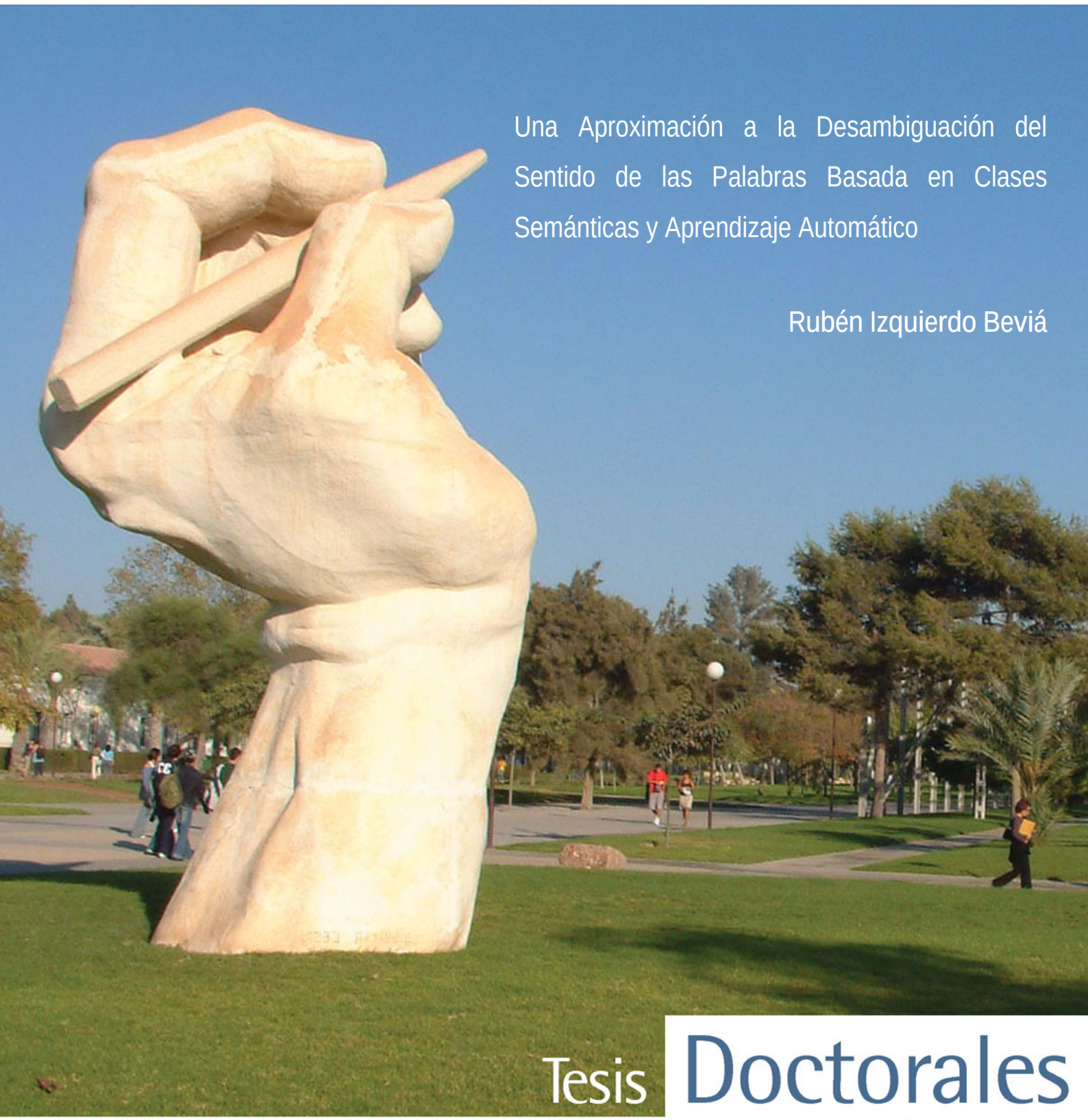
Universitat d'Alacant
Universidad de Alicante

Una Aproximación a la Desambiguación del Sentido de las Palabras Basada en Clases Semánticas y Aprendizaje Automático

Rubén Izquierdo Beviá



Tesis Doctorales

www.eltallerdigital.com

UNIVERSIDAD de ALICANTE

Universitat d'Alacant
Universidad de Alicante
Departament de Llenguatges i Sistemes Informàtics
Departamento de Lenguajes y Sistemas Informáticos

Una Aproximación a la Desambiguación del Sentido de las Palabras Basada en Clases Semánticas y Aprendizaje Automático

Rubén Izquierdo Beviá

Directores:

Armando Suárez Cueto
German Rigau i Claramunt

Alicante, 2010

Agradecimientos

A pesar de ser ésta la primera sección, es la última que he dejado para escribir, y es que estas cosas siempre me han costado bastante. La verdad que empecé un poco, o bastante, sin saber muy bien de qué iba esto del Procesamiento del Lenguaje Natural o lo que era un *paper*. De hecho creo que tenía la misma sensación que cuando trato de explicar en qué trabajo a alguien que no sabe nada de este tema. Poco a poco fui entrando en este mundillo, sabiendo qué era *Question Answering*, una anáfora o la Desambiguación del Sentido de las Palabras. Y precisamente es ahora, al finalizar este trabajo y echar la vista atrás, cuando me doy cuenta de dónde encajo yo dentro de este campo de investigación, las cosas en las que he trabajado y lo que he podido aportar. Alguien me comentaba en alguna ocasión, sentados en una terraza frente al Partenón de Atenas, que esto precisamente era lo más valioso que sacaría de una tesis: abstraerse del trabajo diario de investigación, de las pruebas y los experimentos, para pasar a un nivel superior donde se pudiera ver de un solo vistazo, como un todo, el progreso seguido y el trabajo realizado durante todos estos años. Tampoco me quiero poner mucho más filosófico, pero creo que era una buena reflexión.

Como decía, poco a poco fui entrando en todo este "mundillo", y de cada persona con la que me he cruzado he aprendido algo. Por eso no tendría mucho sentido ponerme a nombrar de uno en uno a todos a los que les agradezco algo, porque seguramente a todos tenga algo que agradecer, aunque seguro también algo que criticar, seguramente mucho menos. Así que todos los miembros del grupo de investigación merecen mi agradecimiento, porque seguro que en algún momento en mayor o menor medida, me han sido de gran ayuda. Especialmente fueron de gran ayuda al principio, cuando no tenía

ni idea de dónde estaba ni a qué me dedicaba. Fue extraordinariamente sencillo integrarme en el grupo y sentir que era uno más de ellos. Siempre me trataron como uno más desde el principio, y pronto dejé de preguntarme eso de: "A esa reunión/cena/invitación que me ha llegado por mail, ¿estaré invitado?". Repito que empezando en el despacho de un lado del pasillo y acabando en el del otro lado, a todos debo agradecer algo. Así que a todos ellos: gracias.

Creo que no sería justo si no nombrara especialmente (y además me apetece hacerlo), a Oscar y a Sergio. Con ellos ha sido con los que más contacto he tenido, habiendo llegado ahora mismo, creo, a ser amigos. Primero, hemos compartido trabajo durante más de 3 años empezando con el proyecto europeo QALL–ME, con buenos momentos y otros de más agobio en el trabajo, pero siempre llevándonos bien. También hemos compartido despacho, con lo cual hemos pasado muchas horas al día juntos. Y más aun, hemos tenido muchos momentos de ocio y viajes fuera del ambiente de trabajo. Todo esto une mucho, y además desde el principio conectamos muy bien, con lo que puedo decir que tenemos una buena (bonita quedaría "cursi") amistad. Por supuesto también les agradezco la ayuda que me han prestado en muchas ocasiones, tanto en el trabajo como fuera del trabajo: gracias!.

No puedo dejar de nombrar a mis directores, Armando y German. Ha sido un placer trabajar con ellos, a pesar de que coordinarnos y entendernos los tres desde la lejanía ha sido bastante complicado. Esas reuniones de toda una mañana y/o tarde aprovechando que German venía para dar alguna asignatura de doctorado han sido muy productivas. Cuando acababa la reunión y me ponía a ordenar mis notas, me daba cuenta que había ideas, comentarios y sugerencias para realizar al menos tres tesis. Sinceramente he aprendido gran cantidad de cosas de ambos, y creo que han sido más incluso de lo que se espera de un director de tesis (o de dos). A German, agradecerle que a pesar de las miles de cosas que lleva entre manos y su situación personal, haya sacado siempre tiempo para mi. Y a Armando querría pedirle perdón y agradecerle por aclararme que se escribe *"Aparte de estos experimentos presentamos otros..."* y no *"A parte de estos experimentos..."*. A los dos: gracias.

Aparte de esto, también me gustaría hacer un pequeño e irónico, aunque real inciso, y nombrar a mis compañeros y demás gente relacionada con mi equipo de fútbol sala. En realidad no sé si son ellos los responsables o soy yo. El caso es que de un modo u otro, "gracias" a las dos fracturas de tobillo que he sufrido en los últimos 4 años, he dispuesto de muchísimo tiempo libre para dedicar a este trabajo. Así que, a pesar de todo, gracias también.

Por supuesto se merecen enorme agradecimiento toda mi familia. En ellos era precisamente en los que más pensaba cuando decía al principio que era muy difícil explicar en qué consiste mi trabajo. A pesar de esto, siempre me han apoyado en todo, y al final hasta comprendían que tuviera que quedarme hasta las tantas el día antes de un *deadline* para acabar un artículo. En realidad tengo muchísimo más que agradecerles que esto, pero creo que no es éste el lugar y momento adecuados, y además nunca he sido de exponer públicamente mis sentimientos. Así que simplemente a toda mi familia, tanto a los que están como a los que ya no están: muchísimas gracias.

Y dejo para el final a la persona más importante, a la que más agradezco y la que más ha sufrido en primera persona los efectos colaterales de mi trabajo: Laura. En realidad ya vivió los últimos 4 años de carrera, cuando iba agobiado con las entregas de prácticas, o cuando me ponía a hablarle de temas informáticos que posiblemente no le interesaran ni entendiera, pero parecía que sí. Luego siguió conviviendo con los efectos colaterales del doctorado, de los *papers* y *deadlines* y de los agobios de última hora que muchas veces me ponían de mala leche, aunque a veces conseguir esto tampoco es difícil. Todo este tiempo que le he "robado" espero poder devolverselo poco a poco, y con intereses. Además, la ayuda anímica y el cariño que siempre me ha dado, tanto en buenos como en malos momentos, me ha ayudado a salir adelante en muchas ocasiones, a no agobiarme por cosas que tenían solución (aunque yo no la viera) y en general a tomarme la vida de forma más relajada y menos rígida de lo que tenía costumbre. Como ya he dicho que no soy de publicar mis sentimientos, lo dejaré aquí y el resto se lo diré en privado. Muchísimas gracias por todo "Laurilla".

P.D: si has leído esto y no encajas en ninguno de los grupos anteriores, no es porque me haya olvidado de ti, simplemente que no he podido poner a todo el mundo. De todos modos, gracias a ti también.

Índice general

Índice de tablas

Índice de figuras

Acrónimos

PLN Procesamiento del Lenguaje Natural 1
RAE Real Academia Española 6
WSD *Word Sense Disambiguacion* 1
SVM *Support Vector Machines* 14
kNN *k-Nearest Neighbor* 27
PoS *Part of Speech* 30
HMM *Hidden Markov Models* 30
MUC *Message Understanding Conference* 32
SUMO *Suggested Upper Merged Ontology* 34
SE *SensEval* 39
SE1 *SensEval–1* 39
SE2 *SensEval–2* 34
SE3 *SensEval–3* 36
LSA *Latent Semantic Analysis* 43
SEM1 *SemEval–1* 84
SEM2 *SemEval–2* 46
SVD *Singular Value Decomposition* 46
WND *WordNet Domains* 60
BC *Base Concepts* 63
BLC *Base Level Concepts* 64

1. Introducción

Leyendo únicamente el título de este trabajo podemos intuir los tres puntos principales de los que trata: técnicas de aprendizaje automático, desambiguación del sentido de las palabras y clases semánticas. Cada uno de estos tres elementos es de naturaleza distinta. Si los combinamos, obtendremos la idea principal de nuestro trabajo: desarrollar un sistema de desambiguación semántica dentro del marco del Procesamiento del Lenguaje Natural, haciendo uso de técnicas de aprendizaje automático y trabajando a un nivel de abstracción de clase semántica, superior al que ofrecen tradicionalmente los sentidos lexicográficos asociados a las palabras.

El Procesamiento del Lenguaje Natural (PLN) es una disciplina dentro de la inteligencia artificial que estudia y analiza el modo de procesar lenguaje natural, tal y como un humano lo genera, mediante mecanismos computacionales automáticos. El principal problema con el que se enfrentan todas las técnicas de PLN es la ambigüedad del lenguaje natural. Un tipo de ambigüedad muy conocida y extendida es la ambigüedad léxica o semántica, derivada de la polisemia de las palabras. Con esto queremos decir que una misma palabra pueden tomar varios significados, y habrá que ser capaces de decidir cuál es el correcto en cada caso para poder conocer el significado de un texto en donde aparezcan palabras polisémicas. Precisamente esta ambigüedad es la que trata de resolver la desambiguación del sentido de las palabras, o *Word Sense Disambiguacion* (WSD) en inglés. WSD es una tarea intermedia en el campo del PLN, donde se estudian y desarrollan técnicas para asignar el sentido correcto a las palabras polisémicas en un texto, en función del contexto en el que aparezca cada una de dichas palabras.

Como hemos dicho anteriormente, un punto importante son los sentidos de una palabra. Tradicionalmente es el modo de representar los diferentes significados de una palabra. Por ejemplo, en los diccionarios clásicos, cada entrada corresponde con una palabra, y cada posible acepción o significado de dicha palabra se representa con un sentido. En diccionarios electrónicos también se suelen representar los significados de las palabras mediante sentidos. Esto ha determinado en gran parte el modo en que se ha abordado computacionalmente la tarea de WSD. En la mayoría de las aproximaciones, se afronta dicha tarea como un problema de clasificación, en el que hay que elegir para cada palabra polisémica su sentido apropiado, teniendo en cuenta los posibles sentidos que se hayan considerado para esa palabra en un cierto recurso o diccionario electrónico.

En nuestro caso hemos desarrollado también nuestro sistema de WSD siguiendo una técnica de clasificación. La diferencia con los tradicionales sistemas que eligen el sentido apropiado para una cierta palabra, nuestros clasificadores decidirán entre las posibles clases semánticas. El uso de clases semánticas en lugar de sentidos propone una serie de ventajas que veremos más detalladamente a lo largo de este trabajo. La filosofía y funcionamiento de nuestros clasificadores quedarán más claras viendo un ejemplo. Haciendo uso de ellos podríamos procesar un texto de entrada y asignarle a cada palabra su etiqueta semántica apropiada.

> *Chess*_ACTIVITY *is a board*_ARTIFACT *game*_ACTIVITY *played between two players*_PERSON. *It is played on a chessboard*_ARTIFACT, *which is a square-checkered board*_ARTIFACT *with 64 squares*_POLYGON *arranged in an eight-by-eight grid*_ARTIFACT. *At the start*_HAPPENING, *each player*_PERSON *controls sixteen pieces*_OBJECT: *one king*_EQUIPMENT, *one queen*_EQUIPMENT, *two rooks*_EQUIPMENT, *two knights*_EQUIPMENT, *two bishops*_EQUIPMENT, *and eight pawns*_EQUIPMENT.

*The object*_CONTENT *of the game*_ACTIVITY *is to checkmate*_BEAT *the opponent*_PERSON*'s king*_EQUIPMENT*, whereby the king*_EQUIPMENT *is under immediate attack*_ACTIVITY *and there is no way*_ACT *to remove or defend it from attack*_ACTIVITY *on the next move*_ACT*.*

Como podemos ver, asignaremos conceptos generales, o clases semánticas, a las palabras, en lugar de los tradicionales sentidos representados en diccionarios. Este modo de representación mediante sentidos ha sido muy criticada, debido a que, a menudo, el grado de especificidad de dichos sentidos es demasiado alto, y la diferencia entre sentidos distintos para una misma palabra es mínimo. Esto hace que sea muy difícil obtener un modelo computacional que sea capaz de discriminar entre estos sentidos tan detallados. Esto puede ser aliviado en cierto modo con el uso de clases semánticas.

Volviendo a las clases semánticas, en primer lugar, y a simple vista, la información de dichas clases semánticas es más informativa que la de sentidos, y, además, posiblemente también sea más útil. Por otro lado, se mantiene una coherencia semántica entre las etiquetas asignadas a las diferentes palabras, de modo que es posible inducir la temática de la que trata el texto a partir de las clases semánticas. También sería posible utilizar las clases semánticas de las palabras monosémicas de un texto para ayudarnos a decidir cuáles son las clases más coherentes para el resto de palabras polisémicas. El uso de clases semánticas en nuestro marco de trabajo proporcionará también otra serie de ventajas que, como hemos dicho, iremos detallando a lo largo de este trabajo.

En general describiremos:

- La aproximación general de nuestro sistema.
- La arquitectura del sistema.
- El conjunto de clases semánticas con diferentes niveles de abstracción.
- El método de aprendizaje automático: algoritmo, atributos, corpus, etc.
- Los experimentos realizados para desarrollar y ajustar el sistema.
- Las distintas evaluaciones del sistema de etiquetado semántico.

Veremos primero el marco general en el que se enmarca este trabajo: el Procesamiento del Lenguaje Natural.

1.1 El Procesamiento del Lenguaje Natural

El Procesamiento del Lenguaje Natural (PLN) es una rama de la inteligencia artificial que trata de desarrollar mecanismos y recursos computacionalmente efectivos que faciliten la interrelación hombre–máquina y permitan una comunicación mucho más fluida y menos rígida que la que ofrecen los lenguajes formales. Dicho con otras palabras: se trata de dotar a un ordenador de los mecanismos, recursos y técnicas necesarias para que sea capaz de entender y generar lenguaje natural del mismo modo que lo hacemos nosotros los humanos. Podríamos considerar que ésta es la meta del PLN, pero ni mucho menos está cerca de alcanzarse. Haciendo un paralelismo con el modo en que nosotros analizamos (casi inconscientemente) el lenguaje, se establecen diferentes niveles de comprensión y análisis para afrontar el procesamiento de un texto mediante un ordenador: nivel fonológico, léxico, morfológico, sintáctico, semántico, pragmático. . . En cada uno de estos niveles se trata el texto de una forma distinta, tratando de extraer un tipo de información determinada y por tanto desarrollando y requiriendo distintos recursos y técnicas.

También podemos clasificar las técnicas de PLN dependiendo de la funcionalidad que desempeñan. Algunas aplicaciones son conocidas como **aplicaciones finales**. Estas aplicaciones combinan métodos y mecanismos de PLN de más bajo nivel y pretenden realizar una tarea determinada de forma que el resultado que generan tenga una utilidad inmediata y directamente observable para el usuario. Entre estas aplicaciones podemos destacar algunas como son:

- *Recuperación de información*: permite obtener un subconjunto de documentos relevantes para el usuario a partir de un conjunto mucho mayor.
- *Extracción de información*: extrae información relevante de un documento que contiene texto no estructurado (lenguaje natural).
- *Búsqueda de respuestas*: devuelve respuestas a preguntas realizadas por el usuario en lenguaje natural.

- *Traducción automática*: traduce un texto en lenguaje natural de un idioma a otro.

Como ya hemos comentado, aparte de estas aplicaciones que generan una información directamente útil para el usuario y que hemos llamado aplicaciones finales, existe otra serie de aplicaciones cuyo objetivo es realizar una conjunto de **tareas intermedias** para obtener algún tipo de información, útil para otras aplicaciones, sobre el texto de entrada que reciben. Muchas de estas tareas funcionan en cadena: la salida de una de ellas es la entrada para otra, es decir, hay tareas que necesitan la información que ofrecen otras para realizar su procesamiento. La utilidad real de estas tareas intermedias está en su uso dentro de una aplicación final. Las aplicaciones finales utilizan muchas de estas tareas intermedias para enriquecer con distintos tipos de información el texto en lenguaje natural, y a partir de toda esta información obtener su resultado final. Algunas de estas tareas intermedias son:

- *Tokenización*: divide el texto en componentes simples como palabras, números, signos de puntuación...
- *Análisis morfológico*: extrae la categoría gramatical, rasgos sobre género, número y persona de las palabras de un texto
- *Análisis sintáctico*: obtiene sintagmas y relaciones estructurales (sintácticas) entre ellos
- *Reconocimiento de entidades*: reconoce conjuntos de palabras que hacen referencia a una entidad como por ejemplo: el nombre de una persona, empresa o localidad, una fecha, una cantidad...
- *Desambiguación del sentido de las palabras*: trata de asignar a una palabra su interpretación correcta dentro del contexto en que aparece.

En general aquellas que generan una información de alto nivel a partir de una entrada, requieren de otras que procesan dicha entrada previamente y obtienen una información de más bajo nivel. Como ya hemos comentado, nuestro trabajo se centra en esta última tarea: **Desambiguación del sentido de las palabras**.

1.2 La Ambigüedad en el Lenguaje

Si revisamos la lista anterior de tareas intermedias de PLN, el principal problema que tratan de solucionar es la ambigüedad del lenguaje a diferentes niveles lingüísticos. De hecho éste es uno de los principales problemas con el que el PLN se enfrenta. Si el lenguaje natural fuera no ambiguo y cada frase significara únicamente una cosa (tal y cómo sucede por ejemplo con los lenguajes formales, matemáticos o de programación), la mayoría de tareas intermedias de PLN no tendrían razón de ser.

Es obvio que, en los diferentes niveles de análisis del lenguaje natural, existen diferentes tipos de ambigüedad, que tratarán de ser resueltas por las subtareas intermedias adecuadas. Así por ejemplo, un tokenizador tendrá que decidir si un punto ortográfico es realmente el fin de una oración o por contrario forma parte de una abreviatura. Un reconocedor de entidades deberá seleccionar un conjunto de palabras contiguas como pertenecientes a la misma entidad, o un analizador sintáctico tendrá que resolver diferentes tipos de ambigüedades, como la conocida ambigüedad por ligamiento del sintagma preposicional[1]. La ambigüedad semántica, derivada de la polisemia de las palabras, la trata la desambiguación del sentido de las palabras.

La ambigüedad semántica está muy presente en el lenguaje a diario, tanto en actos de comunicación orales como en escritos. De hecho, muchos chistes y juegos de palabras hacen uso de este fenómeno para generar el humor. Tomemos como ejemplo la frase:

> *"El capitán dice que ya pasó revista, pero a mi no me ha llegado ninguna."*

Como vemos, en esta frase se juega con el sentido de la palabra *"revista"*. Si consultamos dicha palabra en un diccionario, como por ejemplo el diccionario de la Real Academia Española (RAE), obtenemos el siguiente resultado para la palabra en cuestión:

[1] Este tipo de ambigüedad se da por ejemplo en la frase: "Juan vio al ladrón con los prismáticos". Incluso para un humano que no disponga de más contexto, no queda claro si era Juan quien usó los prismáticos para ver al ladrón, o por el contrario, era el ladrón quien portaba los prismáticos.

revista

1. f. Segunda vista, o examen hecho con cuidado y diligencia.

2. f. Inspección que un jefe hace de las personas o cosas sometidas a su autoridad o a su cuidado.

3. f. Examen que se hace y publica de producciones literarias, representaciones teatrales, funciones, etc.

4. f. Formación de las tropas para que un general o jefe las inspeccione, conozca el estado de su instrucción, etc.

5. f. Publicación periódica por cuadernos, con escritos sobre varias materias, o sobre una sola especialmente.

6. f. Espectáculo teatral de carácter frívolo, en el que alternan números dialogados y musicales.

7. f. Espectáculo teatral consistente en una serie de cuadros sueltos, por lo común tomados de la actualidad.

8. f. Der. Nuevo juicio criminal ante segundo jurado cuando el tribunal de derecho aprecia error evidente o deficiencia grave no subsanada en el veredicto del primero.

9. f. Der. Antiguamente, segunda vista en los pleitos, en otra sala del mismo tribunal.

Según la clasificación que nos ofrece la RAE, está claro que la *"revista"* a la que se refiere el capitán es aquella representada en la acepción número 2 ó 4 del diccionario, mientras que la otra aparición de la palabra hace referencia a la acepción número 5. Precisamente la polisemia de la palabra *"revista"* es lo que origina la ambigüedad. Recordemos que el fenómeno de la **polisemia** se da cuando una palabra puede tener más de un significado, y adquiere su semántica concreta en función del contexto en el que aparece. A menudo se confunde este fenómeno con otro similar, pero de origen totalmente diferente: la **homonimia**. Dos palabras son homónimas cuando se escriben del mismo modo (homógrafas) o suenan de la misma manera (homófonas) pero tienen sentidos totalmente diferentes. La diferencia con la polisemia está en que, en el caso de la homonimia, estamos hablando

realmente de dos palabras totalmente diferentes, que provienen de orígenes distintos y poseen significados distintos. En el caso de la polisemia, es una única palabra, con un único origen, la que posee varios significados. Con los siguientes ejemplos quedará más clara la diferencia:

Homonimia: dos palabras iguales, y dos orígenes y significados distintos

- *Homógrafos:* vino (verbo venir, del latín *venit*) – vino (bebida, del latín *vinum*)
- *Homófonos:* tubo (cañería, del latín *tubus*) – tuvo (verbo tener, del latín *tenuit*, que evoluciona como *habuit* desde *habere* dando lugar a *hubo*)

Polisemia: una palabra, varios significados

- Clave: "La *clave* del problema" (solución) – "La *clave* de la caja fuerte" (combinación)
- Sierra: "Lo corté con la sierra" (herramienta) – "Subimos a la sierra más alta" (montaña)

Este último fenómeno, la polisemia, es el que trata la Desambiguación del Sentido de las Palabras, el cual genera la ambigüedad semántica, que trata de ser resuelta por dicha tarea.

1.3 La Desambiguación del Sentido de las Palabras

Tal y como hemos descrito, la Desambiguación del Sentido de las palabras (o WSD en inglés) trata de asignar o decidir cuál es el significado concreto de una palabra polisémica en un determinado contexto (Weaver, 1957). Se la conoce también por desambiguación léxica o etiquetado semántico. A pesar de ser intermedia y no obtener ningún resultado tangible directamente para el usuario final, esta tarea posee mucha importancia. La relevancia y popularidad de WSD ha sido ampliamente reconocida en el campo de la Lingüística Computacional: en abril de 2010, hay alrededor de 3.500 artículos en la recopilación

digital *ACM Anthology*[2] que contienen los términos *"word sense disambiguation"*, mientras que, por ejemplo, solo alrededor de 1.600 hablan de la resolución de la anáfora.

Aunque normalmente cualquier persona parece que es capaz de entender el significado de cada palabra en un texto sin mucha dificultad[3], automáticamente mediante un ordenador es una tarea muy difícil. Existen diversos problemas que hacen esta tarea tan compleja. En primer lugar, un humano dispone de un aprendizaje obtenido a lo largo de los años, posee un conocimiento muy extenso del mundo real (objetos, relaciones y propiedades), y posiblemente dispone de un contexto amplio sobre el tema del que está hablando o leyendo. Todo esto facilita en gran medida la tarea de desambiguación semántica para un humano, pero es muy difícil de incorporar a un programa.

Desde un punto de vista computacional, WSD se describe como una tarea AI–completa por analogía con los problemas NP–completos en la teoría de complejidad (Mallery, 1988). Esto quiere decir que resolver completamente la tarea de WSD sería equivalente a resolver todos los problemas en el campo de la Inteligencia Artificial, como por ejemplo el test de Turing, o dotar a los computadores de la misma capacidad de comprensión que los humanos. Existen muchos problemas que contribuyen a dificultar dicha tarea para un sistema automático. Cabe destacar que los sistemas que mejores resultados han obtenido tradicionalmente son aquellos basados en *aprendizaje automático*, y más concretamente en *aprendizaje automático supervisado* (Agirre & Edmonds, 2006, capítulo 7). Aunque entraremos en este tema más extensamente a lo largo de esta memoria, podemos adelantar que este tipo de sistemas hacen uso de un algoritmo de aprendizaje automático, y de un conjunto de ejemplos anotados semánticamente para entrenar el sistema. Precisamente el uso de determinados algo-

[2] `http://www.acm.org`, consultado en julio del 2010.

[3] Hablamos de trivial refiriéndonos al proceso que le supone a un humano asignarle el sentido correcto a una palabra. Sin embargo resolver la tarea en general no es sencilla, ni tan sólo para humanos especialistas. Dos anotadores pueden asignarle sentidos distintos a una misma palabra. Por ejemplo, la tasa de acuerdo entre anotadores en el corpus de la tarea internacional de etiquetado de sentidos de las palabras SensEval–3 fue 67,3 %, lo que indica que la tarea no es sencilla de resolver (Snyder & Palmer, 2004).

ritmos de aprendizaje automático determina el punto de vista desde el que se aborda la tarea de WSD: una tarea de clasificación.

Las aproximaciones a WSD basadas en aprendizaje automático supervisado se enfocan como un problema de clasificación. Para una ocurrencia de una palabra, se tienen en cuenta los sentidos posibles de dicha palabra, y se clasifica la ocurrencia entre estos posibles sentidos, asignándole aquel que sea más adecuado. Sin embargo, esta aproximación tampoco está exenta de problemas.

En primer lugar e independientemente de la aproximación tomada, supervisada o no supervisada, existe un enorme debate alrededor de cuál es la granularidad óptima de sentidos. Esta idea de granularidad indica cómo de sutiles o detallados son los sentidos de una palabra, si hay diferencias mínimas o interdependencias entre sentidos distintos (granularidad fina), o si los sentidos representan a conceptos totalmente diferentes (granularidad gruesa). Los sentidos posibles para un conjunto de palabras podemos encontrarlos en diferentes recursos, por ejemplo, un diccionario.

A pesar de ser criticado por su excesiva granularidad, el diccionario de referencia en WSD es WordNet (Fellbaum, 1998). WordNet es una red semántica que define conceptos, sentidos de palabras y relaciones entre ellos (Fellbaum, 1998). Es el más extendido para el inglés, por su amplia cobertura, por el número de recursos anotados semánticamente según dicho diccionario, y por ser el recurso más usado en competiciones de evaluación internacionales.

La crítica a su excesiva granularidad está relacionada por una parte con que, en general, se definen demasiados sentidos para una palabra en concreto. Por otra parte más bien se refiere a que la diferencia entre ciertos sentidos de una misma palabra es mínima, y estas mínimas diferencias son muy difíciles de captar por un sistema de aprendizaje automático. Volviendo al ejemplo de la palabra *"revista"* que comentábamos anteriormente, según la RAE, los sentidos número *6* y *7* hacen referencia a dos tipos similares de espectáculos teatrales. Estas dos acepciones de *"revista"* son muy similares, y es muy posible que las frases en que aparecen ejemplos de uso de *"revista"* con ambos sentido sean muy similares. Incluso se puede dar el caso de que manualmente un anotador haya asignado el sentido *6* cuando sería más correcto el sentido *7*, sin estar cometiendo un error.

Todo esto conduce a disponer de ejemplos muy similares de diferentes sentidos de una misma palabra, lo cual no beneficia en absoluto a los algoritmos de clasificación y dificulta su aprendizaje.

En segundo lugar, y más importante, es el llamado *cuello de botella en la adquisición de conocimiento*. Como hemos dicho, los mejores resultados para la tarea de WSD se han obtenido usando aproximaciones basadas en aprendizaje automático supervisado. Este tipo de aproximaciones se centran principalmente en fuentes de conocimiento anotados como corpus, colecciones de documentos, redes semánticas, diccionarios, etc. El problema es que la creación de este tipo de recursos de forma manual es muy compleja y costosa, como se argumenta en (Ng, 1997), y se necesitarían recursos nuevos para cada nuevo dominio, lenguaje o escenario que se quisiera contemplar. Esto da lugar a una escasez que impide que se puedan obtener sistemas de etiquetado automático con un alto rendimiento, de forma que se pudiera crear nuevos recursos anotados haciendo uso de estos sistemas. Se trata de un círculo vicioso que obstaculiza seriamente a las aproximaciones basadas en aprendizaje automático supervisado.

Algunos trabajos se han centrado recientemente en el estudio y desarrollo de sistemas de WSD en dominios específicos. Tradicionalmente los sistemas de WSD se han basado en aproximaciones de aprendizaje automático supervisado, entrenados sobre corpus de dominios generales o abiertos. El problema es cuando estos modelos se utilizan sobre corpus de dominios concretos, en los que las distribuciones de sentidos de palabras suelen cambiar respecto a dominios generales. Ciertas palabras aparecen frecuentemente asociados con algunos sentidos en textos de dominio general, y con otros totalmente diferentes en textos de dominio concreto. Por tanto, sistemas entrenados sobre datos de temática general, no presentan buen rendimiento sobre datos de dominios específicos. Recientemente se ha presentado un trabajo de tesis que trata ampliamente sobre el impacto del cambio de dominio en sistemas de WSD, y algunas aproximaciones para tratar de solucionar los problemas que se presentan (de Lacalle, 2009).

Finalmente, otro punto en debate para los sistemas de WSD, es su utilidad e integración en aplicaciones de lenguaje natural de alto nivel. Probablemente, la información semántica que proporciona un

sistema de WSD pueda ser de gran utilidad integrada en otro sistema de PLN, pero todavía hay muchas dudas sobre cómo hacer uso de esta información y qué aplicaciones son las más adecuadas para integrar aprovecharse de la información semántica. Además las tasas de acierto de los sistemas actuales son demasiado bajas como para garantizar la calidad de la información que proporciona, como se ha visto en las últimas competiciones internacionales de SensEval y SemEval. Algunas de las aplicaciones donde se integran sistemas de WSD se presentan en la sección siguiente.

1.4 Aplicaciones de *Word Sense Disambiguation*

El increíble aumento de textos digitales e información en general que ha supuesto la explosión de Internet y las tecnologías de la comunicación en la moderna sociedad de la información, han puesto de relieve la necesidad del uso de técnicas de PLN. La mayoría de técnicas automáticas que se utilizan para procesar y gestionar grandes cantidades de información textual, se basan únicamente en análisis superficiales (morfológico, como mucho sintáctico) y en métodos estadísticos. Estas técnicas presentan varios problemas. Por ejemplo, estos sistemas no son capaces de identificar fragmentos de información relevante para un usuario, cuando éstos están expresados con unas palabras diferentes a las que utilizó dicho usuario, aunque en ambos casos el significado sea el mismo. Los sistemas de WSD pueden ayudar a salvar este obstáculo, asignando el sentido correcto a las diferentes palabras, e identificando de este modo posibles sinónimos. Dentro del nuevo marco de la Web Semántica, este tipo de sistemas de WSD encaja perfectamente, como medio para asignar significado estructurado a páginas web de texto plano en principio no estructurado. Otras aplicaciones de PLN en las que se ha integrado un sistema de WSD son:

Traducción automática: conocer el sentido correcto de las palabras es esencial para traducirlas correctamente. De este modo *sierra* (herramienta) sería traducida al inglés como *saw*, mientras que *sierra* (montaña) se traduciría como *mountain* o *mountain*

range, o *revista* podría ser *magazine* (publicación) o *review* (inspección).

Recuperación de Información: es importante recuperar documentos relevantes para una consulta en concreto, aunque no posea exactamente las mismas palabras, sino que contenga palabras sinónimas o relacionadas semánticamente. Además es crítico conocer la semántica de la consulta que se quiere procesar. Por ejemplo, si un usuario lanza una búsqueda mediante la consulta "intereses del banco Central", seguramente esperaría recuperar documentos hablando sobre el beneficio que obtendremos por ingresar nuestro dinero en la entidad financiera Central. Sin embargo, no desearía recuperar documentos sobre otros tipos de *bancos* (asiento, conjunto de peces, masa de arena, etc.).

Clasificación de Documentos: para determinar la temática de un texto o fragmento de texto es imprescindible conocer exactamente que quiere decir cada una de las palabras contenidas en el texto, y por tanto es necesario conocer su significado.

Búsquedas de Respuestas: obviamente comprender el significado correcto de una pregunta es crucial para poder obtener una respuesta correcta. Conocer el sentido de una palabra permite obtener sinónimos de ella, que pueden ser cruciales para obtener la respuesta. Por ejemplo, ante la pregunta "¿Quién concibió el primer telescopio?", la respuesta sólo la obtendríamos a partir del fragmento de texto "Hans Lippershey inventó el primer telescopio...", si fuéramos capaces de determinar que en este caso *concebir* e *inventar* son sinónimos.

En (Agirre & Edmonds, 2006, capítulo 11) podemos ver una descripción más extensa y detallada del uso de algoritmos de WSD en aplicaciones de PLN. Por tanto, parece que disponer del significado de las palabras puede ser algo básico para cualquier tarea.

Tradicionalmente no se han obtenido grandes mejoras con el hecho de integrar la información de sentidos de palabras a diferentes aplicaciones de PLN. Los principales problemas son dos. Por una parte, no se sabe la forma óptima de integrar dicha información de sentidos en aplicaciones de alto nivel. Por otra, los mejores sistemas actuales obtienen un acierto alrededor del 60 % para sentidos finos, el cual

no es suficiente como para poder disponer de esta información con garantías de que no se están introduciendo errores y ruido en lugar de información útil.

1.5 Contribuciones de este trabajo

Teniendo en cuenta el escenario actual de los sistemas de desambiguación semántica, las aproximaciones que se han seguido generalmente, los problemas principales con que se enfrentan y los límites que parecen no poder superar, planteamos una aproximación alternativa.

Diseñamos una sistema de desambiguación semántica basado en aprendizaje automático supervisado que hace uso de una arquitectura basada en clases semánticas. Esta arquitectura supone un cambio de punto de vista respecto a los sistemas tradicionales que intentan discriminar entre los sentidos de una palabra. En nuestro caso, creamos un clasificador para cada clase semántica, en lugar de para cada palabra en concreto. Esto tiene varias ventajas que iremos describiendo a lo largo de nuestro trabajo. El algoritmo de aprendizaje que hemos empleado ha sido las Máquinas de Vector Soporte (o *Support Vector Machines* (SVM) en inglés). Hemos elegido este algoritmo debido a los buenos resultados que ha demostrado en distintas evaluaciones internacionales (J. & Shawe-Taylor, 2000), así como por su buena eficiencia temporal para poder realizar gran cantidad de experimentos. Además se ha demostrado que funciona muy bien en espacios de dimensionalidad muy grandes, con gran número de atributos y muchos de ellos irrelevantes, por lo cual también se adapta muy bien a nuestra aproximación basada en clases (Joachims, 1999).

El motivo de centrar nuestro trabajo en las clases semánticas ha sido por las ventajas que suponen dichas clases, y que pueden ser explotadas mediante la aproximación que tomamos. Estas ventajas, a grandes rasgos, son las siguientes:

- Reducción de polisemia: la agrupación de sentidos derivada del uso de clases semánticas produce una reducción de polisemia que hará obtener mejores resultados.

- Aumento de ejemplos de entrenamiento: se alivia el cuello de botella en la adquisición de conocimiento. Con la nueva arquitectura basada en clases, para cada clasificador disponemos de todos los ejemplos de aquellas palabras que pertenecen a dicha clase semántica, con lo que la cantidad de ejemplos de aprendizaje aumenta.
- Nivel de abstracción posiblemente más adecuado: la información que pueden proporcionar las clases semánticas, seguramente es de mayor calidad y utilidad en aplicaciones de más alto nivel que la información de sentidos, en general demasiado específica y poco informativa.
- Mayor robustez e independencia de los clasificadores respecto al dominio. Mediante las clases semánticas se aprenden modelos que se adaptan y funcionan mejor sobre textos y ejemplos de dominios específicos y distintos a los de los ejemplos de entrenamiento.

En cuanto a las clases semánticas, podemos decir que son conceptos, como por ejemplo BUILDING, VEHICLE o FOOD, que engloban a un conjunto de palabras que poseen coherencia léxica y comparten una serie de propiedades. Una gran parte de la terminología humana puede ser agrupada bajo dichas clases semánticas, ya que su significado hace referencia a dominios específicos y tienden a aparecer en discursos semánticamente similares. Por tanto, se puede considerar que las clases semánticas representan tanto a textos o contextos, como a sentidos de palabras en concreto. Este punto es muy importante, ya que establece la conexión entre clases semánticas y la tarea de WSD. En (Yarowsky, 1992) además se establece dos principios acerca de las clases semánticas:

1. Palabras iguales que aparecen en contextos diferentes suelen pertenecer a clases semánticas distintas.
2. Sentidos diferentes de una misma palabra suelen pertenecer a clases semánticas distintas.

Estos principios son una modificación de la teoría descrita en (Gale *et al.*, 1992b), donde se propone que palabras iguales dentro de un mismo discurso tienden a poseer normalmente el mismo sentido.

Existen diversos repositorios de clases semánticas que agrupan los sentidos de las palabras de forma distinta, dando lugar a conjuntos

con diferentes niveles de abstracción, cada uno con sus ventajas e inconvenientes. A lo largo de este trabajo analizamos varios conjuntos, para elegir aquel más adecuado a la tarea de WSD.

Sin embargo, más importante para nosotros era desarrollar un método de obtención automática clase semánticas. Hasta el momento, todos los estudios que se habían hecho en esta dirección usaban conjuntos de clases semánticas estáticas, predefinidas de antemano y con un nivel de abstracción fijo. Mediante nuestro método pretendemos parametrizar el algoritmo, de modo que podamos generar diferentes conjuntos de clases, con el grado de abstracción necesario para los objetivos concretos que persigamos.

Otro de los principales objetivos de nuestro trabajo es comprobar que estas clases son un recurso válido para la tarea de WSD. Alrededor de este objetivo principal se han desarrollado multitud de experimentos adicionales, para analizar diferentes grados de abstracción, diversos tipos de atributos o diferentes arquitecturas de nuestro sistema, entre otros.

Por tanto, esta memoria está enfocada a demostrar la validez y ventajas del uso de clases semánticas en la desambiguación léxica. Alrededor de este objetivo principal hemos desarrollado un gran número de experimentos para estudiar una arquitectura orientada a clases semánticas, la adaptación de atributos para representar los ejemplos de aprendizaje y la definición de otros nuevos, y la posibilidad de utilizar niveles de abstracción variables, entre otros. En resumen, las contribuciones de nuestra trabajo son:

1. Arquitectura de WSD centrada en clases semánticas.
2. Análisis de diferentes niveles de abstracción a través de distintos conjuntos de clases semánticas.
3. Desarrollo de un método automático para la obtención de Conceptos Base semánticos desde WordNet.
4. Estudio de atributos en el marco de la desambiguación basada en clases semánticas.
5. Uso de clases semánticas en la construcción de atributos para el proceso de aprendizaje.
6. Evaluación del sistema con diferentes configuraciones y participación en campañas internacionales.

7. Integración de clases semánticas en un sistema de Recuperación de Información.
8. Estudio de la robustez de nuestros clasificadores basados en clases semánticas cuando se entrena y evalúa el sistema en dominios diferentes.
9. Desarrollo de una serie de recursos que pueden ser descargados y usados libremente

Como hemos dicho en el último punto, hemos desarrollado un software que implementa nuestro sistema de desambiguación basado en clases semánticas. El paquete lo hemos puesto a disposición de toda la comunidad científica, y puede ser descargado y usado libremente desde `http://www.dlsi.ua.es/~ruben`. También se puede probar el sistema de forma *online* sin necesidad de instalar nada. El software está implementado en Python[4] y hace uso del paquete de herramientas de PLN NLTK[5]. También en esa misma web (`http://www.dlsi.ua.es/~ruben`) se puede descargar el software que implementa el método de extracción automática de conceptos desde WordNet, así como los distintos conjuntos ya extraídos desde distintas versiones de WordNet.

1.6 Estructura de la memoria de tesis

En resumen, este trabajo la hemos estructurado del siguiente modo.

- **Capítulo I**: esta introducción
- **Capítulo II**: en este capítulo repasamos los sistemas más destacados que se encuadran dentro de la desambiguación del sentido de las palabras. Mostramos sistemas pertenecientes a diferentes aproximaciones, centrándonos en aquellos basados en aprendizaje automático supervisado, y en concreto en aquellos que trabajan con clases semánticas.

[4] `http://www.python.org`, consultado en julio de 2010.
[5] `http://www.nltk.org`, consultado en julio de 2010.

- **Capítulo III**: mostramos los recursos que utilizamos en nuestro trabajo, tales como corpus anotados semánticamente y clases semánticas. También describimos el método de aprendizaje automático que hemos empleado en nuestro sistema, Máquinas de Soporte Vectorial.
- **Capítulo IV**: este capítulo describe el método de obtención automática de los Conceptos Base de WordNet, así como los diferentes conjuntos de clases semánticas que se generan con dicho algoritmo. Además presentamos una evaluación de la calidad y robustez de dichos conjuntos de clases semánticas. En primer lugar, en la tarea de desambiguación semántica, haciendo uso de una heurística simple basada en seleccionar la clase más frecuente, sin disponer de contexto ni algoritmo de aprendizaje. En segundo lugar, una evaluación indirecta en otra tarea de lenguaje natural como es la Recuperación de Información. Además describimos el uso que se le ha dado a los Conceptos Base en el proyecto europeo Kyoto[6]. Este proyecto propone una plataforma común para compartir la información entre diferentes culturas con distinta lengua, organizando la información en diversos niveles. Los Conceptos Base se utilizan en uno de los niveles intermedios como punto de conexión entre conceptos generales y específicos.
- **Capítulo V**: aquí se muestra el sistema de WSD desarrollado, junto con los experimentos realizados inicialmente. Definimos la arquitectura basada en clases semánticas, realizando una comparación con la tradicional basada en sentidos. Llevamos a cabo también un profundo análisis de atributos para seleccionar aquellos más idóneos para el aprendizaje. También mostramos nuestra participación en un foro de evaluación internacional (SemEval–1) con una versión muy básica e inicial de nuestro sistema, así como nuestra participación en SemEval–2, donde hemos hecho uso de nuestro sistema básico enriquecido con ejemplos monosémicos extraídos desde textos de un dominio específico. En este punto es donde evaluamos la robustez del sistema frente al cambio de dominio. Finalmente describimos una aplicación de nuestras clases semánticas en la tarea

[6] `http://www.kyoto-project.eu`

de detección y extracción de conceptos clave, una tarea similar a la de detección de tópicos.

- **Capítulo VI**: este capítulo se dedica a una evaluación a fondo del rendimiento del sistema a diferentes niveles de abstracción, haciendo uso de los diferentes conjuntos de clases semánticas descritos. En primer lugar, definimos dos conjuntos de atributos que emplearemos en estos experimentos. También analizamos el impacto de los atributos semánticos en el rendimiento del sistema de desambiguación. Realizamos una serie de pruebas para estudiar el comportamiento del sistema de etiquetado semántico a medida que incrementamos la cantidad de ejemplos de entrenamiento. Por último, se dedica una sección a comparar este sistema con otro sistemas participantes en competiciones internacionales.
- **Capítulo VII**: finalmente mostramos las conclusiones que hemos obtenido de nuestros experimentos y algunas posibles líneas de investigación por donde nos gustaría continuar nuestro trabajo. También describimos la producción científica derivada de este trabajo.
- **Anexos**: incluimos dos anexos. El primero contiene una relación detallada de los resultados obtenidos por nuestro sistema, obtenidos a lo largo del proceso de desarrollo del mismo combinando diferentes tipos de atributos, tamaños de contexto y configuraciones. El segundo anexo corresponde con un conjunto de pruebas que hemos realizado para comprobar que nuestra aproximación era la más adecuada, y que no obtuvieron buenos resultados, y, por tanto nos han proporcionado información sobre cual es la dirección correcta a seguir en el desarrollo de un sistema de este tipo.

2. Estado de la Cuestión

En este capítulo analizaremos el estado actual de los sistemas de resolución de la ambigüedad semántica de las palabras de un texto, así como la evolución desde sus inicios. En primer lugar incluiremos una sección para describir las medidas que se utilizan generalmente para evaluar los sistemas de WSD. Posteriormente, haremos un breve repaso de las principales líneas de investigación que se han adoptado en las tareas de *Word Sense Disambiguacion* (WSD). Nos centraremos en los sistemas que siguen más de cerca las líneas de investigación que nosotros hemos tomado. Existen diversas publicaciones y resúmenes donde se presentan de forma más detallada las aproximaciones a WSD y los sistemas más relevantes (Agirre & Edmonds, 2006; Navigli, 2009).

En general, las diferentes aproximaciones a la tarea de WSD se suelen clasificar en función de los recursos y métodos usados para implementar el sistema. Aquellos sistemas que utilizan principalmente diccionarios, lexicones, tesauros y bases de conocimiento, sin utilizar información estadística obtenida de corpus, son conocidos como *métodos basados en conocimiento* (Agirre & Edmonds, 2006, Capítulo 5). Aquellos otros métodos que utilizan la información contenida en corpus, para extraer evidencias estadísticas o para realizar aprendizaje automático son conocidos como *métodos supervisados* o *no supervisados*, en función del tipo de anotación del corpus que utilicen. Los no supervisados utilizan corpus no anotados semánticamente, y básicamente se centran en agrupar sentidos de palabras similares, crear *clusters* de palabras con significados similares, pero no en discriminar dichos sentidos. Los métodos supervisados hacen uso de corpus con información semántica anotada para aprender y entrenar modelos que posteriormente permitan clasificar nuevas ocurrencias de

palabras ambiguas. Un tipo de sistemas supervisados son conocidos como *semisupervisados*. La diferencia entre los métodos supervisados y los semisupervisados radica en la cantidad de información de la que parten, los semisupervisados suelen utilizar una pequeña cantidad de información inicial, para obtener un primer sistema, con el cual anotar nueva información y refinar sucesivamente dicho sistema. Los supervisados parten de un conjunto de información de un tamaño considerable, que se utiliza para entrenar y desarrollar el sistema final (Agirre & Edmonds, 2006, Capítulo 7).

2.1 Medidas de evaluación

Vamos a comentar las medidas que se utilizan para evaluar muchos sistemas de PLN y en concreto de WSD, y que también usaremos nosotros para evaluar nuestros experimentos. Sea N el número de contextos o ejemplos que queremos clasificar, A el número de contextos clasificados correctamente y E el número errores de clasificación. Puede suceder que haya contextos que no clasifiquemos porque no tengamos suficiente información y que por tanto no pertenecerían ni al conjunto de aciertos ni al de errores. Esto se puede se refleja en la fórmula 2.1.

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

La precisión (P) nos da una idea del número de contextos clasificados correctamente respecto al total de contextos clasificados. La fórmula muestra el modo de obtener la precisión.

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

Otra medida utilizada es la cobertura (C). La cobertura mide el número de aciertos respecto del número total de contextos que deberíamos haber clasificado. Por tanto la fórmula es:

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

La cobertura absoluta (CA) establece la relación entre contextos clasificados y contextos totales que se deberían haber clasificado. La forma de obtenerla es mediante la fórmula 2.4.

$$CT = \frac{A+E}{N} \tag{2.4}$$

Finalmente una medida que combina precisión y cobertura es la llamada $F_{\alpha=1}$, que por comodidad denotaremos F1. La fórmula 2.5 detalla el modo de obtenerla.

$$F1 = \frac{2 \times P \times C}{P+C} \tag{2.5}$$

Tengamos en cuenta que en el caso de que se clasifiquen (acertadamente o no) todos los contextos, se da que $N = A+E$ y por tanto la precisión, cobertura y F1 tienen el mismo valor ($P = C = F1$, en este caso se suele hablar de "acierto", puesto que también coincide con la 'tasa de acierto') y la cobertura total sería igual a 1 ($CT = 1$). En los experimentos descritos utilizaremos normalmente el valor $F1$ ya que combina precisión y cobertura en un único valor.

2.2 *Word Sense Disambiguation*

La tarea de WSD es una de las más antiguas dentro del Procesamiento del Lenguaje Natural (PLN). Ya a finales de los años 40, se consideraba la tarea como un bloque fundamental dentro de la Traducción Automática (Weaver, 1957). En aquel entonces ya se tenían en cuenta los principales puntos en los que se suelen centrar la mayoría de sistemas de WSD actuales: el contexto de las palabras, información estadística sobre palabras y sentidos de palabras, fuentes de conocimiento externas, etc. Pronto quedó claro que WSD iba a ser una tarea muy difícil de abordar, y de hecho fue el máximo obstáculo con el que se encontró la Traducción Automática en los años 60 (Bar-Hillel, 1960). Por ejemplo Bar-Hillel comentó que ningún sistema existente o imaginable sería capaz de hacer a una computadora determinar que la palabra *"pen"* es usada con el sentido de *"corral"* en la frase siguiente:

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

Durante los años 70 las aproximaciones a WSD se centraron en técnicas y procedimientos tomados del campo de la Inteligencia Artificial, aunque los resultados que se alcanzaron no fueron demasiado buenos debido a la poca cantidad de bases de conocimiento disponibles para el desarrollo de este tipo de técnicas. Wilks desarrollo uno de los primeros sistemas explícitamente adaptado a WSD (Wilks, 1975). El sistema usaba un conjunto de restricciones de selección y una semántica basada en marcos (en inglés *frames*) para encontrar un conjunto de significados para las palabras de una frase que fuera consistente.

Fue durante los años 80, cuando los sistemas de WSD avanzaron considerablemente debido al aumento de recursos léxicos de gran tamaño, que facilitaron el desarrollo de sistemas de extracción automática de conocimiento (Wilks *et al.*, 1990), y el uso de este conocimiento en sistemas de WSD en lugar de la información manual utilizada hasta este momento. En este sentido, (Lesk, 1986) desarrolló un sistema de WSD general basado en el conocimiento contenido en el Diccionario Oxford Avanzado[1]. Este sistema utilizaba las posibles definiciones de dos palabras ambiguas para decidir los dos mejores sentidos de ambas palabras. El sistema seleccionaba aquellos sentidos que obtuvieran un solapamiento máximo de sus definiciones en el diccionario.

Durante los años 90, se desarrollaron extensamente las aproximaciones basadas en técnicas estadísticas, se establecieron diferentes marcos de evaluación[2] donde poder comparar los diferentes sistemas de WSD, y WordNet se puso a disposición de los investigadores de PLN (Fellbaum, 1998). Este recurso impulsó la investigación en WSD debido a que proporcionó un base de conocimiento de un tamaño considerable y que era accesible por un computador, estableciendo un repositorio de sentidos que solucionaba momentáneamente el debate sobre las diferentes definiciones de sentidos que las palabras en inglés podían tener.

En los últimos años, las aproximaciones supervisadas han tenido una gran difusión y muy buenos resultados, como veremos posteriormente. Por otra parte, recientemente las aproximaciones basadas en

[1] `http://www.oed.com`, consultado en julio de 2010

[2] `http://www.senseval.org`, consultado en julio de 2010

conocimiento, y en particular los métodos basados en grafos de conceptos de gran tamaño y alta conectividad han acaparado la atención de los investigadores del área. Existe una gran cantidad de sistemas de WSD que siguen ese tipo de técnicas (Navigli & Velardi, 2005; Agirre & Soroa, 2009; Hughes & Ramage, 2007; Cuadros & Rigau, 2008). En la última edición de la competición internacional SemEval (celebrada en julio de 2010 en Uppsala, Suecia), este tipo de aproximaciones han obtenido muy buenos resultados.

2.3 Aproximaciones supervisadas a *Word Sense Disambiguation*

Debido a que la aproximación presentada en este trabajo se engloba dentro de la familia de técnicas supervisadas, haremos especial énfasis en este tipo de sistemas. En los últimos 15 años el campo del Aprendizaje Automático ha experimentado un auge espectacular, y muchas de las técnicas y algoritmos se han aplicado también en el PLN, y en concreto en las aproximaciones supervisadas a la tarea de WSD (Agirre & Edmonds, 2006, Capítulo 7). En general, dentro del PLN se han aplicado en tareas de clasificación, es decir, en aquellas que consisten en seleccionar una alternativa entre varias posibles. Por ejemplo, la selección de la palabra adecuada en el reconocimiento del habla, decidir la etiqueta correcta en el etiquetado morfológico o el referente adecuado en la resolución de la anáfora. En todos estos casos, los algoritmos y técnicas desarrollados en el ámbito del Aprendizaje Automático son de una aplicación inmediata. Más recientemente, también se han usado aproximaciones basadas en Aprendizaje Automático en otras tareas de PLN que no se limitan a problemas de clasificación. Por ejemplo, el etiquetado secuencial o la asignación de estructuras jerárquicas.

Volviendo a WSD, las aproximaciones que más éxito han tenido son las supervisadas basadas en corpus, a pesar del problema del cuello de botella en la adquisición de conocimiento que ya comentamos. Podemos clasificar las aproximaciones en función de la base teórica usada para generar los modelos de clasificación.

2.3.1 Métodos probabilísticos

Estos métodos intentan estimar las probabilidades y ajustar los parámetros de distribuciones de probabilidad, que relacionan contextos con categorías o sentidos de palabras. Siguiendo esta aproximación, los contextos son representados por los atributos que se extraen de ellos. Así, los nuevos ejemplos de palabras ambiguas son clasificadas seleccionando el sentido cuyos atributos observados en el contexto coinciden con una mayor probabilidad conjunta.

El algoritmo más usado en este tipo de métodos es el *Naïve Bayes*, que utiliza la regla de inversión de Bayes y asume la independencia condicional de los atributos con las clases o sentidos de las palabras. Ha sido aplicado en numerosas investigaciones en WSD con buenos resultados (Gale *et al.*, 1992b; Leacock *et al.*, 1993; Escudero *et al.*, 2000b; Pedersen, 2000).

También el principio de Máxima Entropía ha sido utilizado para combinar evidencias estadísticas de diferentes fuentes de información. En este caso no se asume ningún conocimiento *a priori* sobre los datos, y se ha demostrado que es un mecanismo muy robusto. En concreto en WSD ha sido aplicado con buenos resultados en (Suárez & Palomar, 2002), y también en otras tareas de PLN como en análisis sintáctico (Ratnaparkhi, 1999).

2.3.2 Métodos basados en similitud de ejemplos

Este conjunto de métodos se basa en la similitud entre ejemplos. En su versión más simple, el entrenamiento de estos sistemas se basa en almacenar ejemplos junto con su sentido correcto, obtenidos de corpus anotados. La clasificación de un nuevo ejemplo consiste en obtener el ejemplo más similar y devolver el sentido almacenado correspondiente a ese ejemplo. Esto tiene dos implicaciones: cómo representar un ejemplo y cómo comparar dos ejemplos. El modo de representación es mediante atributos extraídos del contexto de las palabras, del mismo modo que en la mayoría de las aproximaciones a WSD. En cuanto a la comparación de ejemplos, se debe definir una medida de similitud que devuelva el grado de semejanza de dos ejemplos, normalmente en función de sus atributos. Por ejemplo, en el

Modelo Espacio Vectorial, se representan los ejemplos mediante vectores de atributos o características, y se obtiene la similitud de dos ejemplos calculando el coseno del ángulo que forman ambos vectores. En (Leacock *et al.*, 1993) se demuestra que las aproximaciones basadas en el Modelo Espacio Vectorial superan a métodos estadísticos basados en *Naïve Bayes*.

El método más usado dentro de esta familia de técnicas es el *k-Nearest Neighbor* (kNN). Este método consiste en almacenar en memoria todos los ejemplos de entrenamiento, junto con sus atributos y su sentido correcto. Para clasificar un nuevo ejemplo se obtienen los *k* vecinos más próximos en el espacio de representación, los más similares, y se combinan los *k* sentidos de dichos ejemplos para asignar el sentido apropiado al nuevo ejemplo. Este método es muy sensible a la medida de similitud utilizada para comparar los ejemplos, y también al tipo de atributos usados para representar los ejemplos y a los pesos que se le asignan a cada uno de ellos. Este técnica ha sido aplicado a WSD por (Ng & Lee, 1996; Stevenson & Wilks, 2001; Escudero *et al.*, 2000a; Agirre *et al.*, 2006), entre otros.

2.3.3 Métodos basados en reglas

Entre estos métodos encontramos las listas y los árboles de decisión. Se basan en un conjunto de reglas que se aprenden automáticamente a partir del corpus de entrenamiento y se utilizan posteriormente para clasificar nuevos ejemplos. Las lista de decisión son un conjunto de reglas ordenadas en función de una probabilidad, cada una de las cuales establece una relación entre una condición (normalmente un valor de atributo), y una clase o sentido de palabra. En la fase de entrenamiento del sistema, se ajustan los pesos de cada regla, y se ordenan de tal forma que las reglas más generales obtienen menor peso que las normas más específicas, las cuales más claramente van a identificar a un sentido de una palabra. En la fase de clasificación de un nuevo ejemplo, se comprueban las reglas una por una por orden de su peso, y se asigna el sentido de la primera regla que se cumple. Por ejemplo, (Yarowsky, 1995) utilizó listas de decisión en WSD. En su aproximación, cada regla se correspondía con un valor para un atributo concreto, asignaba un sentido para una palabra, y

su peso se calculaba usando una medida de similitud logarítmica que representaba la probabilidad de que una palabra perteneciera a un sentido concreto en función de algún valor de atributo.

Los métodos basados en árboles de decisión utilizan una aproximación similar, pero la forma de representar las reglas y condiciones es mediante una estructura de árbol jerárquica. En ella cada nodo comprueba una condición y decide por qué subárbol continuar el proceso de comprobación de condiciones. Aunque este tipo de técnicas se ha aplicado ampliamente en Inteligencia Artificial a problemas de clasificación, no hay muchos trabajos de WSD en los que se hayan utilizado. Por ejemplo, (Mooney, 1996) usó el algoritmo C4.5, basado en árboles de decisión, y concluyó que este método no se encontraba entre los más apropiados para la tarea de WSD. Sin embargo en (Pedersen, 2001) se describe un trabajo usando árboles de decisión a partir de bigramas que obtuvo muy buenos resultados comparados con los participantes de en SensEval–1[3].

2.3.4 Métodos basados en clasificadores lineales

Esta familia de técnicas se ha adaptado a WSD debido a los buenos resultados obtenidos en la tarea de clasificación de textos (Joachims, 1998). En general, un clasificador lineal binario es un hiperplano que separa los ejemplos en el espacio de características que se esté utilizando, y genera dos grupos. Si se utilizan *n* atributos para representar los ejemplos de aprendizaje, cada uno de estos ejemplos se codificará con un vector de *n* elementos, y se representará en un espacio n–dimensional. Por tanto, la fase de entrenamiento intenta encontrar los parámetros del hiperplano que separe a los ejemplos de entrenamiento en el espacio de características.

Aunque en principio estos clasificadores son lineales, su expresividad se puede aumentar mediante el uso de funciones *kernel*. Estas funciones permiten establecer una correspondencia entre los ejemplos de entrenamiento y cambiar la topología del espacio de características. De este modo, se pueden transformar problemas que no sean

[3] Tanto las competiciones SensEval como las tareas propuestas en tales competiciones serán descritas más detalladamente en la sección 2.5 y en el capítulo 3

linealmente separables en otros que sí lo sean, de forma que se puedan usar métodos de clasificación lineales para obtener el modelo de clasificación.

Entre estos tipos de métodos se encuentran las Máquinas de Soporte Vectorial (*Support Vector Machines* (SVM) en inglés) (Vapnik, 1995). SVM intenta encontrar el hiperplano que maximiza el margen de separación entre los ejemplos positivos y los negativos. Los sistemas de WSD basados en esta técnica han estado entre los que han obtenido los mejores resultados. Podemos resaltar entre otros a los sistemas que obtuvieron muy buenos resultados en SenseEval–3 (Strapparava *et al.*, 2004a; Lee *et al.*, 2004a; Agirre & Martínez, 2004; Escudero *et al.*, 2004).

2.4 Aproximaciones basadas en clases

Dentro de las aproximaciones tradicionales a WSD vamos a centrarnos especialmente en sistemas que hacen uso de clases semánticas, debido a que nuestra aproximación se basa en el uso de dichas clases semánticas para construir clasificadores. Estos clasificadores deciden la clase semántica apropiada para una palabra en lugar de su sentido. En realidad, el uso de clases semánticas en WSD surgió casi al mismo tiempo que la propia tarea, pero no se le dio la importancia suficiente, ni se realizó un estudio empírico completo usando diferentes grados de abstracción.

(Yarowsky, 1992) describe un método basado en corpus y no supervisado que asigna el sentido correcto haciendo uso del conjunto de categorías descritas en el tesauro Roget (Roget, 1852). Utiliza este conjunto de categorías como una aproximación a clases conceptuales, como modo de diferenciación para los sentidos de las palabras. Propone una aproximación estadística basada en corpus partiendo de tres hipótesis:

- Distintas clases conceptuales de palabras, como por ejemplo ANIMAL o MAQUINARIA, tienden a aparecer en contextos diferentes y diferenciables.
- Los sentidos distintos de una palabra tienden a pertenecer a clases conceptuales diferentes.

- Si se consiguen construir clasificadores para un conjunto de clases conceptuales, automáticamente se habrá conseguido clasificadores para todas las palabras que son miembros de esas clases.

En primer lugar, utiliza la enciclopedia Grolier[4] para extraer contextos de 100 palabras alrededor de los miembros de cada categoría del tesauro Roget. De esta forma se obtienen contextos representativos para cada categoría. En segundo lugar, se calcula el peso de cada palabra asociada a cada categoría, para ello se tienen en cuenta la frecuencia de aparición de la palabra asociada a la categoría determinada, la frecuencia en los contextos extraídos para la categoría, y la frecuencia total de la palabra en todos los contextos. Las palabras que obtienen un mayor peso son las más representativas para dicha categoría. Cada categoría, posee una lista de palabras asociadas, cada una con un peso determinado. Finalmente, para clasificar una nueva palabra, se obtiene su contexto, y se le asigna aquella categoría que maximiza la suma de pesos asociados a las palabras del contexto obtenido. La evaluación se realiza sobre un conjunto de ejemplos para 12 palabras polisémicas, obteniendo tasas de acierto que varían entre el 72 % y el 99 %, con una media aproximada de 92 %. A pesar de estos resultados, el método tenía ciertas limitaciones: no era capaz de caracterizar correctamente clases que no estuvieran fuertemente asociadas a ciertas palabras, no era capaz de realizar distinciones finas entre sentidos que pertenecen a una misma categoría, o se aplicaba solo a nombres, entre otras.

En (Segond *et al.*, 1997) se presenta un sistema de etiquetado semántico haciendo uso de técnicas tradicionales de etiquetado morfosintáctico (*Part of Speech* (PoS) *tagging* en inglés). La aproximación seguida en este trabajo se basa en un aprendizaje automático supervisado. En concreto toman la idea de que ciertas secuencias de etiquetas, ya sea morfosintácticas o semánticas, son más probables que otras. Algoritmos que se adaptan a este tipo de hipótesis son los basados en Modelos Ocultos de Markov (*Hidden Markov Models* (HMM) en inglés), y por tanto el algoritmo que utilizan en este trabajo es de este tipo. Seleccionan un trozo del corpus *Brown* (Kuĉera & Francis, 1967), el cual fue etiquetado semánticamente

[4] `http://go.grolier.com`

por el equipo que desarrolló WordNet, y lo dividen en dos conjuntos disjuntos, para entrenamiento y test. El conjunto de clases con el que está etiquetado este fragmento de corpus son los ficheros lexicográficos de WordNet[5]. Dicho de otro modo, para cada palabra se dispone del fichero lexicográfico de WordNet como clase semántica, y por tanto generan clasificadores que utilizan esas clases semánticas como conjunto de etiquetas para anotar nuevas ocurrencias de palabras. Realizan tres pruebas. La primera, a modo de sistema base, asigna a cada palabra su clase más frecuente en el corpus de entrenamiento (atendiendo a su categoría gramatical), y obtiene un acierto del 81 %. La segunda prueba utiliza los clasificadores creados por el algoritmo de aprendizaje sin hacer diferenciación en cuanto al tipo de categoría gramatical, es decir, para una palabra en concreto, se consideran sus clases posibles sin diferenciar si es nombre, verbo, adjetivo o adverbio. En este caso la tasa de acierto llega hasta un 86 %. Finalmente, se realiza la misma prueba, usando los clasificadores pero restringiendo las clases posibles para una palabra en función de su categoría gramatical, con lo cual la polisemia media por palabra se reduce considerablemente. En este último caso se alcanza un acierto del 89 %.

En (Peh & Ng, 1997) se propone una aproximación a la desambiguación semántica, basada en clases y en WordNet. Para ello diseñan una ontología de clases de un dominio específico, y la alinean manualmente con synsets de WordNet. El sistema que diseñan se compone principalmente de dos módulos: un módulo de desambiguación basado en sentidos, y un módulo de selección de la clase semántica apropiada. Para el módulo de desambiguación seleccionan dos algoritmos basados en conocimiento, los cuales usan WordNet como fuente de conocimiento: un algoritmo de solapamiento de información desarrollado a partir del algoritmo descrito en (Resnik, 1995) y un algoritmo basado en la técnica de densidad conceptual presentada en (Agirre & Rigau, 1996). Mediante estos algoritmos seleccionan el sentido o synset de WordNet apropiado para una palabra. En el

[5] Aunque serán descritos más adelante, los ficheros lexicográficos son un conjunto de 45 categorías, que representan los usos más frecuentes y generales de las palabras. Estas clases están creadas considerando tanto la categoría léxica como diferentes agrupaciones lógicas

módulo de selección de la clase, se busca el concepto de la ontología específica más similar al synset seleccionado por el algoritmo de desambiguación. Para ello hacen uso de dos medidas de similitud: distancia conceptual y probabilidad de enlace. El corpus de test se corresponde con 1.023 nombres extraídos del corpus de la competición MUC–4[6] de 18 textos del dominio terrorismo. Todas las palabras son etiquetadas manualmente con su sentido y clase semántica correcta. En cuanto a la desambiguación a nivel de sentidos, obtienen un resultado de 63,30 %. Cuando se realiza el mapeo a clases y evalúan a nivel de clases, obtienen un resultado de 80,16 %. Concluyen que su aproximación no supervisada obtiene unos resultados comparables con otras aproximaciones supervisadas, muy similares a los obtenidos mediante anotadores humanos y además es adaptable al dominio.

Otra aproximación al etiquetado mediante clases semánticas en el marco de WSD se presenta en (Ciaramita & Johnson, 2003). En concreto se aborda la tarea del etiquetado de palabras desconocidas, es decir, que no aparecen en el corpus disponible para el entrenamiento. Este es un problema que afecta típicamente a las aproximaciones supervisadas a WSD. La aproximación, a pesar de ser supervisada, no hace uso de corpus anotados semánticamente puesto que únicamente utiliza las **palabras monosémicas** extraídas del corpus *Bllip*[7] (Charniak *et al.*, 2000). Además utilizan la información contenida en WordNet, las definiciones y las frases de ejemplo, para aumentar los ejemplos de entrenamiento. Con esta información, extraen un conjunto de atributos (etiquetas morfológicas, palabras en el contexto, bigramas y trigramas, información sintáctica, etc.), y hacen uso de un algoritmo de etiquetado multiclase basado en un perceptrón. En cuanto al conjunto de etiquetas semánticas para realizar el aprendizaje y la anotación, utilizan los ficheros lexicográficos de WordNet, a los cuales llaman por primera vez *SuperSenses*[8]. El marco de evaluación del sistema lo establecen haciendo uso de WordNet. Entrenan el

[6] Las competiciones *Message Understanding Conference* (MUC) son una serie de conferencias que se vienen realizando desde 1990, como foro de evaluación y análisis de los sistemas de extracción de información más actuales. `http://www-nlpir.nist.gov/related_projects/muc`

[7] `http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2000T43`

[8] En el capítulo 3 describiremos este recurso.

sistema con los nombres contenidos en la versión 1.6 de WordNet, y realizan la evaluación sobre los nuevos nombres que se incorporan en la versión 1.71. Realizan diferentes experimentos, tomando diferentes fragmentos del corpus *Bllip* y teniendo en cuenta la información contenida en WordNet o prescindiendo de ella. Además utilizan dos corpus de evaluación, uno formado por palabras del corpus que aparecen en WordNet 1.7.1 pero no en WordNet 1.6, y otro creado a partir de un conjunto de palabras de WordNet 1.6 que se eliminan manualmente del corpus de aprendizaje. Los mejores resultados los obtienen utilizando el 55 % del corpus *Bllip* y añadiendo la información de WordNet. Obtienen un resultado de 36,9 % a nivel de *token* y 52,9 % considerando clases semánticas, sobre el corpus compilado sobre WordNet 1.7.1 Para el corpus generado a partir de WordNet 1.6, alcanzan un resultado de 52,3 % para los *tokens* y 53,4 % para las clases semánticas. Se observa, por tanto, una sustancial mejora de los resultados usando la información contenida en WordNet. Esto pone de manifiesto la importancia para WSD de utilizar información libre de ruido, así como la relevancia de utilizar información de palabras polisémicas desambiguadas (tal y como aparecen en WordNet), lo cual pone de relieve de nuevo el problema del cuello de botella en la adquisición de conocimiento.

(Gliozzo *et al.*, 2004) hace un estudio del uso de información semántica en la desambiguación léxica. En concreto describen el recurso semántico *WordNet Domains*[9], el modo en que los dominios semánticos pueden usarse en el Procesamiento del Lenguaje Natural en general, y en la desambiguación en concreto. Definen una técnica para modelar los dominios de forma computacional. En concreto describen cómo generar vectores que representen los dominios más relacionadas con cierto objeto, ya sea un concepto (o synset o sentido de una palabra), una palabra o un fragmento de texto. Posteriormente, mediante medidas de similitud entre vectores, serán capaces de determinar la similitud entre objetos de los tipos indicados en términos de su información semántica. Estudian también la aplicación de esta información de dominios a WSD, y las ventajas que pueden aportar dichos dominios tanto en aproximaciones tradicionales basadas

[9] En el siguiente capítulo se describe este recurso.

en algoritmos supervisados (mediante su uso como atributos para el sistema), como su uso como nivel de abstracción para generar clasificadores. En primer lugar describen una aproximación no supervisada a la desambiguación, que a grandes rasgos asigna a una palabra ambigua aquel sentido cuyo vector de dominio maximiza la similitud respecto al vector de dominio correspondiente al contexto. En segundo lugar, seleccionan un sistema de desambiguación basada en listas de decisión, en el que se usa la información semántica de dominios como atributos para el aprendizaje. Evaluando los sistemas sobre el corpus de la tarea *Lexical Sample* de *SensEval–2* (SE2)[10] obtienen que, a pesar de que el sistema supervisado alcanza mejores resultados que el no supervisado, haciendo uso de información semántica en el sistema no supervisado se reduce la cantidad necesaria de ejemplos de entrenamiento casi en un 33 %. Por otra parte, evaluando a nivel de clases semánticas, los resultados son mucho mejores (80 % frente a 56 %de acierto). En el experimento sobre el corpus de la tarea *All Words* de SE2, obtienen resultados competitivos mediante el sistema no supervisado, lo que pone de manifiesto la robustez de la aproximación y el interés del uso de información semántica.

En (Villarejo *et al.*, 2005) se presenta una aproximación supervisada basada en clases a la desambiguación semántica de las palabras. No crean clasificadores centrados en palabras, sino clasificadores basados en clases semánticas, los cuales deciden para cada palabra su clase semántica en lugar de su sentido apropiado. Para realizar el entrenamiento utilizan el corpus SemCor[11], dos conjuntos diferentes de clases semánticas (los ficheros lexicográficos de WordNet y las clases de la ontología *Suggested Upper Merged Ontology* (SUMO)[12]) y hacen uso de dos algoritmos de aprendizaje: SVM y AdaBoost. Para realizar la evaluación definen dos conjuntos de atributos, uno reducido que contiene información de la palabra objetivo y su clase más frecuente, y otro más amplio que incluye únicamente información del contexto de la palabra objetivo. La evaluación se realiza por medio de validación cruzada sobre el mismo corpus SemCor y, en general, los mejores resultados son alcanzados utilizando SVM, y el conjun-

[10] Este corpus también será descrito en el capítulo
[11] Este corpus será explicado en el capítulo 3
[12] Estos recursos se describen en el capítulo 3

to reducido de atributos: un 82,5 % de acierto para los *SuperSenses* de WordNet, y un 71,9 % para SUMO. Concluyen que la aproximación basada en clases a WSD propone una serie de mejoras (evitar la dispersión de datos, reducir la polisemia y aumentar el número de ejemplos de aprendizaje por clase), aunque esta tarea aun debe ser analizada en profundidad y estudiar los atributos más adecuados para ella ya que, si cabe, es todavía mas compleja que la aproximación clásica por sentidos a WSD, tal y como revelan los resultados base de ambas aproximaciones.

(Kohomban & Lee, 2005) presentan otra aproximación supervisada al aprendizaje de clasificadores automáticos basados en clases semánticas. De nuevo hacen uso de SemCor como corpus de aprendizaje y desarrollo, y de las clases semánticas de WordNet, o *SuperSenses*. Utilizan tres conjuntos de atributos (palabras en el contexto, etiquetas morfológicas en el contexto y relaciones sintácticas) para crear tres clasificadores diferentes que posteriormente combinan junto con el clasificador correspondiente al sentido más frecuente, mediante un proceso de votación. Para crear cada clasificador hacen uso de un algoritmo kNN, y definen una medida de similitud derivada de usadas tradicionalmente, tanto para asignar pesos a los ejemplos de entrenamiento, como para obtener la similitud entre ejemplos de entrenamiento y nuevos ejemplos. La novedad de este trabajo estriba en que estudian hasta dónde podrían ayudar las clases semánticas a la desambiguación por sentidos, es decir, analizan el paso de la información de clases semánticas a sentidos. Para ello definen una heurística muy sencilla que etiqueta una palabra con el sentido más frecuente según WordNet que encaje con la clase semántica que se le haya asignado[13].

En primer lugar evalúan hasta dónde llegaría esta heurística utilizando la información correcta de sentidos sobre los corpus de SensEval–2 y SensEval–3. Para ello obtienen la clase correcta de cada palabra a partir de su sentido, y posteriormente, a partir de esta clase, se vuelve a asignar el sentido mediante la heurística descrita.

[13] Por ejemplo: la palabra ha sido clasificada como perteneciente a la clase semántica `economia`, y según WordNet, el primer sentido de la palabra pertenece al dominio `historia`, y el segundo sentido a `economia`. Por tanto el sentido asignado a la palabra sería el segundo.

Obtienen que este método obtendría valores superiores a los resultados base e incluso superiores a los mejores sistemas participantes en tales competiciones. El siguiente paso es evaluar su sistema de aprendizaje automático desde dos perspectiva: la de clase semántica y la de sentido (haciendo uso de la heurística comentada para transformar clases semánticas en sentidos). Obtienen un resultado de 77,7 % sobre SE2 y 80,6 % sobre *SensEval–3* (SE3) a nivel de clase semántica, superando los resultados base (75,6 % y 78,3 %, respectivamente). En cuanto a la evaluación considerando sentidos, consiguen el segundo mejor resultado sobre SE2 con un 66,4 %, por detrás del sistema *SMUaw* que alcanza 69,0 %. En el caso de SE3, alcanzan el mejor resultado comparado con el resto de participantes de la tarea *All Words*, un 66,1 %, por delante del segundo sistema, *GAMBL–AW–S*, que obtiene un 65,2 %.

El trabajo anterior es continuado 2 años después (Kohomban & Lee, 2007). En este caso, argumentan que usar los *SuperSenses* a modo de clases semánticas para WSD no es adecuado. Estas clases no funcionan bien agrupando palabras y ejemplos para luego ser usadas en el marco de un sistema de aprendizaje automático. A menudo agrupan palabras totalmente diferentes, con unos patrones de uso muy distintos, mientras que en otras ocasiones palabras muy relacionadas semánticamente aparecen bajo *SuperSenses* distintos. Esto conduce a que no exista una coherencia en cuanto a los atributos y patrones del conjunto de palabras que se engloban bajo un mismo *SuperSense*, lo cual no es nada beneficioso para un sistema de aprendizaje automático. Por tanto, proponen utilizar técnicas de *clustering* para obtener agrupaciones de palabras relacionadas semánticamente, y usar estas agrupaciones a modo de clases semánticas para generar los clasificadores. El modo de agrupar los sentidos de las palabras es en función de los atributos. A partir del corpus de aprendizaje SemCor, se agrupan sentidos de palabras que poseen atributos y patrones de uso similares. De este modo se generan grupos con gran coherencia, y salvan el problema que se tenía en este aspecto con los *SuperSenses*. Utilizan tres conjuntos de atributos: palabras en el contexto local, etiquetas morfológicas y relaciones gramaticales. Solo las dos primeras las usan para crear agrupaciones de sentidos. Una vez generados dichos *clusters*, que funcionarán a modo de clases semánticas,

aplican la misma aproximación basada en un algoritmo de similitud de ejemplos que emplean en (Kohomban & Lee, 2005). También generan otros conjuntos de agrupaciones de sentidos, basados en la estructura jerárquica de WordNet. En cuanto a los resultados, los que obtienen son a nivel de sentido, transformando la clase semántica que asignan sus clasificadores en el primer sentido de la palabra que encaja en dicha clase. Logran unos buenos resultados, mejores en los casos en que utilizan los *clusters* generados en base a los atributos que en aquellos obtenidos a partir de WordNet. En todos los casos superan al *baseline* y al sistema anterior, que hacía uso de los *SuperSenses*. Mediante una votación de sus clasificadores utilizando diferentes conjuntos de atributos, consiguen un 68,70 % y un 66,40 % sobre el corpus SE2, usando los *clusters* basados en atributos y los basados en WordNet respectivamente, mientras que usando *SuperSenses* alcanzan un 67,40 %. Sobre SE3 obtienen un 67,70 % con los *clusters* basados en atributos, 65,90 % para los desarrollados según WordNet, y 66,10 % utilizando *SuperSenses*. Comentan que sus resultados superan a todos los participantes en SE2 y SE3, a excepción de un único sistema.

También existen aproximaciones no supervisadas, como por ejemplo la descrita en (Curran, 2005). En este trabajo utilizan una técnica que no requiere corpus etiquetado para el aprendizaje, ni poseer ejemplos anotados de una palabra concreta para poder clasificarla posteriormente. Para esto, utilizan una medida de similitud semántica que compara dos palabras basándose en el contexto que las rodea. La medida de similitud tiene en cuenta las relaciones gramaticales que se extraen del contexto de una palabra, haciendo uso de diversas herramientas de PLN. Disponen de varios corpus, procesados con dichas herramientas, para generar un corpus global de 2 billones de palabras. Para clasificar una nueva palabra, obtienen una serie de sinónimos desde el corpus generado, por medio de la función de similitud definida. Para cada uno de esos sinónimos ordenados según su similitud, extraen las posibles categorías según las clases *SuperSenses*, y realizan una combinación para devolver aquella categoría a la que más probablemente pertenezca la palabra. Por otro lado también desarrollan manualmente un conjunto de reglas a partir de los sufijos de las palabras, para clasificar aquellas palabras que no aparecen en

el corpus, o que no aparecen con una frecuencia mínima para extraer sinónimos de forma fiable. La evaluación la realizan sobre conjuntos de palabras extraídas de WordNet 1.6, y sobre otros conjuntos de palabras nuevas incluidas en la versión 1.7.1 de WordNet. La tasa de acierto sobre WordNet 1.6 es de 68 %, mientras que sobre la versión 1.7.1 es de 63 %.

En (Ciaramita & Altun, 2006) proponen tratar la tarea de WSD como un problema de etiquetado secuencial. Eligen los *SuperSenses* como conjunto de clase semánticas para generar los clasificadores. Debido a que este conjunto también dispone de etiquetas específicas para las clases más usuales detectadas en reconocimiento de entidades (persona, localización, organización, tiempo...), el etiquetador semántico que desarrollan también hace la función de detector de entidades. Utilizan una aproximación supervisada usando SemCor como corpus de aprendizaje y un algoritmo de etiquetado secuencial basado en HMM e implementado mediante un perceptrón. Definen un conjunto de atributos para realizar la representación de los ejemplos de entrenamiento, entre los cuales se incluyen palabras, el sentido más frecuente de dichas palabras, etiquetas morfológicas, la etiqueta anterior asignada por el etiquetador y un atributo similar a una expresión regular que representa la forma de la palabra (mayúsculas, minúsculas, letras, dígitos...). Realizan dos tipos de experimentos, uno usando el mismo corpus SemCor como conjunto de test mediante técnicas de validación cruzada, y otro empleando el corpus de SenseEval–3 para evaluación. La tasa de acierto que obtienen es de 77,18 % sobre SemCor, y de 70,54 % sobre SE3. A pesar de los buenos resultados, que superan los resultados base según la heurística de la clase más frecuente, la mejora obtenida respecto al uso de sentidos no es tan sustancial como la reducción de polisemia resultante con el uso de clases semánticas, por lo que concluyen que el problema de la excesiva granularidad es solo uno de los problemas que dificulta la tarea de WSD.

2.5 Los sistemas de SensEval

Las competiciones internacionales *SensEval* (SE)[14], surgieron por la necesidad de disponer de un marco común de evaluación de los sistemas de WSD. Antes de la aparición de esta serie de competiciones, no existía un marco común para evaluar de forma comparable y repetible un sistema de WSD: repositorio de sentidos, fuentes de conocimiento, corpus utilizados, y lo más importante, corpus para la evaluación. Sin dicho marco común, era muy difícil realizar comparaciones entre sistemas. Además, generan una importante cantidad de recursos valiosos para desarrollar y evaluar sistemas de WSD y suponen un foro para todos los investigadores en este campo donde hacer públicos sus resultados, poner en común sus sistemas y discutir con otros investigadores de la materia. A lo largo de los últimos doce años se han celebrado un total de cinco ediciones de estas competiciones.

2.5.1 SensEval–1

La primera edición, *SensEval–1* (SE1)[15], tuvo lugar en 1998, en Sussex (Kilgarriff & Rosenzweig, 2001). Debido a que fue el punto de inicio, hubo diversos debates para establecer las bases concretas del marco de evaluación, sobre repositorios de sentidos, posibles tareas a definir, conjuntos de evaluación, etc. En principio se definió una única tarea, la llamada muestra léxica o, en inglés, *Lexical Sample* en inglés, en la que se trataba de anotar con su sentido apropiado diferentes ocurrencias de una misma palabra en diversos contextos. Participaron un total de 25 sistemas, pertenecientes a 23 grupos de investigación diferentes.

Las lenguas en las que se podía participar fueron inglés, italiano y francés. En concreto, para el inglés se dispuso un conjunto de evaluación con 35 palabras con un total de 3.500 ocurrencias de dichas palabras en diferentes contextos. El repositorio de sentidos que se empleó en esta competición fue el HECTOR (Atkins, 1992). El mejor sistema obtuvo un resultado de acierto cercano a 74 %, y se estableció un *baseline* basado en el sentido más frecuente que alcanzó un

[14] `http://www.senseval.org`, consultado en julio de 2010.

[15] `http://www.itri.brighton.ac.uk/events/senseval/ARCHIVE/index.html`, consultado en julio de 2010.

57 % de acierto. La gran mayoría de sistemas que obtuvieron los mejores resultados fueron aquellos basados en aprendizaje automático supervisado, que hicieron uso de recursos anotados manualmente.

El sistema que mejor resultado obtuvo fue el sistema *JHU* (Yarowsky, 1999). Dicho sistema implementaba un algoritmo supervisado basado en listas de decisión jerárquicas. Se introdujo una serie de condiciones para la ramificación, reduciendo al mismo tiempo la excesiva fragmentación que los datos sufren en los árboles de decisión. Utilizó diferentes conjuntos de atributos tales como colocaciones, atributos morfológicos, sintácticos y contextuales, extraídos automáticamente de los datos de entrenamiento. Además les asignaron pesos a los atributos en función de la naturaleza del atributo y del tipo de dicho atributo.

El sistema que obtuvo el segundo mejor resultado fue el de (Hawkins & Nettleton, 2000). En este caso se trató de una aproximación que combinaba diferentes técnicas: una técnica estocástica basada en la frecuencia de los sentidos en el texto, un conjunto de reglas extraídas de las asociaciones de palabras en el corpus de entrenamiento, y otra que trataba de obtener la similitud entre conceptos.

2.5.2 SensEval–2

El segundo de estos foros de evaluación fue llamado SE2[16], y tuvo lugar en Toulouse, Francia, en 2001. Se definieron tres tareas para un total de 12 lenguas. En primer lugar, la misma tarea que se había definido en SE1, una muestra léxica o *Lexical Sample* en que se trataba de anotar semánticamente diferentes ocurrencias de un conjunto palabras. En segundo lugar, una tarea en la que se solicitaba anotar todas las palabras con contenido semántico que aparecían en un texto plano (en inglés se llamó *All Words* a esta tarea). Y en tercer lugar, una tarea similar a la tarea *Lexical Sample*, pero en la cual los sentidos se definieron a través de su traducción a otra lengua, en concreto al japonés. En esta ocasión se empleó el repositorio de sentidos propuesto por WordNet 1.7 para anotar los corpus de evaluación.

La mayoría de los 34 grupos participantes lo hicieron en las dos primeras tareas. Los resultados de los mejores sistemas los podemos

[16] `http://193.133.140.102/senseval2/`, consultado en julio de 2010.

ver en la tabla 2.1. Estos resultados fueron peores que en la anterior edición de SE debido a que el repositorio de sentidos utilizado fue de mucha mayor granularidad, lo que contribuyó a hacer más difíciles las tareas. De nuevo los mejores resultados se obtuvieron haciendo uso de aprendizaje automático supervisado. Además hubo una serie de técnicas que demostraron también producir buenos resultados, entre las que estaban: votación de sistemas heterogéneos, uso de atributos complejos, selección de atributos y uso de material de entrenamiento adicional.

Lexical Sample		***All Words***	
Sistema	**Precisión**	**Sistema**	**Precisión**
JHU	64,2	SMU-aw	69,0
SMU-ls	63,8	CNTS-Antwerp	63,6
KUNLP	62,9	Sinequa–LIA	61,8
Standford–CS224N	61,7	UNED–AW-U2	57,5
TALP	59,4	UNED–AW-U	55,6
baseline	*47,6*	*baseline*	*57,0*

Tabla 2.1. Mejores resultados en SensEval–2

El sistema *JHU* (Yarowsky *et al.*, 2001) obtuvo los mejores resultados en la tarea *Lexical Sample*, con un 64,2 % de precisión. Emplearon una votación (probaron varias configuraciones mediante validación cruzada) de diferentes clasificadores que hacían uso de aprendizaje automático supervisado y conjuntos de atributos complejos, entre los cuales podíamos encontrar relaciones sintácticas o expresiones regulares construidas en base a las etiquetas morfológicas alrededor de la palabra objetivo. Para la implementación de los diferentes clasificadores utilizaron diversos algoritmos: similitud de ejemplos, modelos Bayesianos y lista de decisión.

(Mihalcea & Moldovan, 2001b) presentaron los sistemas *SMU–ls* y *SMU–aw*, en las tareas *Lexical Sample* y *all-words*, respectivamente. En el caso de *Lexical Sample*, hicieron uso de un algoritmo de aprendizaje basado en ejemplos, mientras para la tarea *all-words* usaron aprendizaje de patrones. Para *Lexical Sample* se disponía de corpus para el aprendizaje, con lo que emplearon un algoritmo basado en ejemplos, en el que incluyeron una selección de atributos específico por palabra, de tal modo que se seleccionaban aquellos atributos

mejores para cada clasificador. En el caso de la tarea *All Words*, se aplicó el mismo algoritmo cuando disponían de suficiente cantidad de ejemplos de entrenamiento para la palabra. En otro caso se aplicó el aprendizaje de patrones. Estos patrones los obtuvieron desde el corpus SemCor (Miller *et al.*, 1993), WordNet (Fellbaum, 1998) y otro corpus que generaron automáticamente, el GenCor. Los patrones se produjeron a partir del contexto local de las palabras, y de cada *token* utilizaron su forma base, su etiqueta morfológica, su sentido y su hiperónimo. En el caso de que tampoco pudieran asignar el sentido mediante esta técnica, asignaban el sentido de alguna ocurrencia de la misma palabra, cercana en el texto y ya desambiguada de la misma palabra cercana en el texto, o el primer sentido de WordNet en último caso.

2.5.3 SensEval–3

Esta tercera edición, *SensEval–3* (SE3)[17], tuvo lugar en 2004 en Barcelona (Mihalcea & Edmonds, 2004). En este caso se propusieron hasta 14 tareas incluyendo las tradicionales *Lexical Sample* y *All Words*, como por ejemplo desambiguación de glosas, etiquetado de roles semánticos, o adquisición de marcos de subcategorización entre otras. Nos centraremos en las dos primeras, ya que son en las que mejor encaja nuestro trabajo de tesis. En total participaron 55 grupos de investigación con 160 sistemas diferentes.

Los cinco mejores resultados para la tarea *Lexical Sample* y *All Words*, junto con los resultados para el *baseline* basado en el sentido más frecuente, los podemos ver en la tabla 2.2.

Lexical Sample		***All Words***	
Sistema	**Acierto**	**Sistema**	**Precisión**
htsa3	72,9	GAMBL–AW	65,1
IRST–Kernels	72,6	SenseLearner	65,1
nusels	72,40	Koc University	64,8
htsa4	72,40	R2D2	62,6
BCU comb	72,3	Meaning–aw	72,5
baseline	*55,2*	*baseline*	*60,9*

Tabla 2.2. Mejores resultados en SensEval–3

[17] `http://www.senseval.org/senseval3`, consultado en julio de 2010.

En el caso de la tarea *Lexical Sample* los resultados de los catorce mejores participantes oscilaron entre 72,9 % y 70,9 % en su tasa de acierto, lo cual sugirió que en cierto modo se establecía una barrera muy difícil de superar. Además surgió el debate de la utilidad de esta tarea, puesto que conocer el sentido de una palabra en una frase, o en un párrafo, quizá no fuera muy informativo para una aplicación de alto nivel de PLN. En concreto, la tarea consistió en anotar con su sentido adecuado un total de 3.944 instancias de evaluación (se usó WordNet 1.7.1 para nombres y adjetivos, y WordSmith[18] para verbos).

El sistema *htsa3* (Grozea, 2004) obtuvo el mejor resultado con una técnica basado en *kernels* y regularización según Thikonov (Tikhonov, 1943). Hacía uso de un *kernel* lineal, y de un conjunto de atributos que incluía colocaciones y lemas en torno a la palabra objetivo. Dicha aproximación empleaba un conjunto de pesos ajustados en función de la distribución de sentidos en el corpus, y además mediante la técnica de regularización se evitó favorecer a aquellos sentidos más frecuentes.

El segundo mejor sistema fue el *IRST-Kernels* (Strapparava *et al.*, 2004b), que alcanzó un 72,6 % de precisión. Este sistema implementó de nuevo una aproximación basada en una función *kernel* que combinaba diferentes fuentes de información. En concreto emplearon dos funciones *kernel*, una sintagmática y otra paradigmática. La sintagmática modelaba la similitud de dos contextos en función de las secuencias de palabras comunes que poseyeran. Para este recuento de secuencias comunes se consideraron colocaciones de palabras y etiquetas morfológicas. La función paradigmática intentaba extraer información del dominio del texto, y hacía uso a su vez de dos *kernels*, uno consistente en una bolsa de palabras alrededor de la palabra objetivo, y otro que implementaba la técnica de Análisis de Semántica Latente (o *Latent Semantic Analysis* (LSA), en inglés).

Por otra parte, en la tarea *All Words* participaron un total de 16 equipos, que anotaron semánticamente 2.037 palabras. El mejor sistema fue el *GAMBL* (Decadt *et al.*, 2004), que obtuvo una precisión de 65,1 %. Usaba un algoritmo de aprendizaje automático basado en

[18] `http://wordsmith.org`

ejemplos, los cuales fueron obtenidos de diferentes fuentes: SemCor, corpus de SE anteriores, WordNet, etc. Los atributos fueron extraídos del contexto local, y de un conjunto de palabras clave que representaban a las palabras objetivo. Además incluyeron una optimización de atributos y un ajuste de parámetros por medio de algoritmos genéticos.

El segundo mejor sistema fue el *SenseLearner* (Mihalcea & Faruque, 2004). Éste usó ejemplos etiquetados manualmente, además de SemCor y WordNet. En una primera fase, usaba un conjunto de modelos aprendidos mediante una técnica de aprendizaje automático basada en ejemplos, utilizando SemCor como corpus de aprendizaje. En una segunda fase, por medio de WordNet y SemCor, extrajeron dependencias entre nombres y verbos, y posteriormente obtuvieron los hiperónimos de dichas palabras, para generalizar el patrón. Posteriormente, empleando un algoritmo de aprendizaje basado en ejemplos, obtuvieron la clasificación usando estos patrones de dependencias como atributos. El sentido final asignado a una palabra estaba basado en los dos valores obtenidos en ambas fases.

2.5.4 SemEval–1

La cuarta edición[19] de esta serie se celebró en 2007 en Praga, y se cambió el nombre de la misma, de *SensEval* a *SemEval*, del inglés *Semantic Evaluations* (E. Agirre & Wicentowski, 2007), dado que se incluyeron numerosas tareas relacionadas con el análisis semántico de textos en general, y no solo con la desambiguación del sentido de las palabras. En total se definieron un total de 18 tareas, entre las que podíamos encontrar las tradicionales *Lexical Sample* y *All Words*, y otras como la desambiguación de preposiciones, la evaluación de WSD en un marco de Recuperación de Información, la inducción y discriminación de sentidos, etc. Un punto importante y novedoso en las tradicionales tareas de WSD fue su cambio de perspectiva en cuanto a la granularidad de sentidos, ya debatido en SensEval–3. Se pensó que el uso de los sentidos demasiado detallados de WordNet no tenía mucha utilidad para otras aplicaciones de más alto nivel dentro del campo del PLN. Además, haciendo uso de este repositorio de

[19] `http://nlp.cs.swarthmore.edu/semeval`, consultado en julio de 2010.

sentidos de alta granularidad, sería muy difícil superar los resultados ya obtenidos. Por tanto, se propuso una agrupación de sentidos, para generar otro repositorio de menor granularidad y utilizarlo en las tareas tradicionales, para anotar los corpus y como conjunto de nuevos sentidos a aprender por los clasificadores. En la tabla 2.3 podemos ver los mejores resultados participantes en ambas tareas, con el uso del los repositorios de sentidos de mayor granularidad.

Lexical Sample (tarea #17)		**All Words** (tarea #7)	
Sistema	**Precisión)**	**Sistema**	**Precisión**
NUS–ML	88,7	NUS–PT	82,5
UBC–ALM	86,9	NUS–ML	81,6
I2R	86,4	LCC–WSD	81,5
USP–IBM2	85,4	GPLSI	79,6
USP–IBM1	85,1	UPV–WSD	78,7
baseline	*78,0*	*baseline*	*78,9*

Tabla 2.3. Mejores resultados en SemEval 2007

La tarea *Lexical Sample* hizo uso del recurso *Ontonotes*[20] (Hovy *et al.*, 2006) como repositorio de sentidos y propuso la desambiguación de 4.851 ocurrencias de 100 palabras distintas (Pradhan *et al.*, 2007). Participaron un total de 13 grupos. Como vemos, los resultados fueron muy superiores a los obtenidos en la misma tarea de SE3. La causa más probable pudo ser la reducción de polisemia obtenida con el uso de sentidos más gruesos que los de WordNet.

El sistema que mejor resultados obtuvo fue el *NUS–ML* (Cai *et al.*, 2007). Este sistema utilizó una técnica de aprendizaje automático supervisado, que combinaba modelos bayesianos estructurados en una jerarquía de tres niveles. Emplearon atributos complejos para realizar el entrenamiento, entre los que se encontraban atributos léxicos, sintácticos y atributos de tópico.

El segundo mejor resultado lo obtuvo el sistema *UBC–ALM* (Agirre & de Lacalle, 2007). Este trabajo combinaba diferentes clasificadores kNN basados en similitud de ejemplos. Cada uno de estos clasificadores disponía de un conjunto de atributos propio: atributos locales, atributos de tópico y atributos latentes, que fueron obtenidos

[20] `http://www.bbn.com/ontonotes`, consultado en julio de 2010.

mediante una técnica de reducción de dimensionalidad de espacios, *Singular Value Decomposition* (SVD) en inglés.

En cuanto a la tarea *All Words*, se propusieron dos de estas tareas, la tradicional que hacía uso de sentidos finos de WordNet (Pradhan *et al.*, 2007), y una nueva que proponía el empleo de sentidos más gruesos (Navigli *et al.*, 2007). Esta última es en la que nos centraremos, debido a que es la más similar a nuestro trabajo, y además fue en la que participamos con nuestro sistema *GPLSI*, obteniendo un buen resultado (79,6 %) y alcanzando la cuarta mejor posición (Izquierdo *et al.*, 2007). Esta aproximación será descrita posteriormente en esta memoria de tesis, en el capítulo 5.

En este caso la tarea consistió en clasificar 2.269 ocurrencias de palabras, que habían sido anotadas haciendo uso de los denominados *sense cluster*. Estos fueron creados semiautomáticamente agrupando bajo un mismo *cluster* aquellos sentidos de una misma palabra compatibles y muy similares. De nuevo se obtuvieron mejores resultados que en la misma tarea de la edición anterior (SE3), posiblemente como consecuencia de la reducción de polisemia. De hecho, el *baseline* por sí solo alcanzó un valor de 78,9 %, que, por otro lado, establecía un valor mínimo muy elevado para ser superado por los sistemas automáticos participantes.

Cabe destacar la primera posición obtenida por el sistema *NUS-PT* (Chan *et al.*, 2007). Usaron un arquitectura de aprendizaje automático, basado en SVM, en la que se integraron numerosos tipos de atributos léxicos y sintácticos. Aparte de SemCor, utilizaron también el corpus DSO[21] y corpus paralelos para realizar el proceso de entrenamiento.

2.5.5 SemEval–2

La quinta edición de las competiciones[22] ha tenido lugar en julio de 2010, en Uppsala, junto con la conferencia internacional ACL (*Association for Computational Linguistics*). En *SemEval–2* (SEM2) se han propuesto un total de 18 tareas, de naturaleza muy variada. Encontramos tareas relacionadas con desambiguación semántica y

[21] `http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC97T12`
[22] `http://semeval2.fbk.eu/semeval2.php`, consultado en julio de 2010.

también con otras tareas de PLN, como por ejemplo resolución de coreferencias, extracción de información, etiquetado de roles semánticos, implicación textual, reconocimiento de expresiones temporales o análisis de sentimientos.

La parte que más nos interesa es la de desambiguación. Relacionada con esta temática, han propuesto varias tareas:

- Desambiguación de nombres en inglés mediante corpus paralelos en varias lenguas.
- Inducción no supervisada de sentidos.
- Desambiguación para el japonés
- Identificación de sentidos poco frecuentes para mandarín.
- Desambiguación de adjetivos referidos a sentimientos
- La tradicional tarea *All Words*, en un dominio restringido

La última tarea es la tarea que más se adaptaba a nuestro trabajo, a pesar de haber sufrido algunas modificaciones. En concreto la más importante es que se enfocó al estudio de la dependencia del dominio de los sistemas de WSD. Para ello, los conjuntos para la evaluación de los sistemas pertenecen a dominios específicos, relacionados con el medio ambiente. De este modo se pudo comparar los sistemas que entrenaban con corpus de propósito general con los que utilizaron corpus especializados, cuando el objetivo es texto relativo a diferentes dominios. En el capítulo 5, donde hablaremos de nuestro sistema de WSD basado en clases semánticas, profundizaremos en nuestra participación en dicha competición.

Para finalizar presentamos en la tabla 2.4 una comparativa de las diferentes ediciones de la competición SensEval. Los datos se corresponden en todos los casos con la tarea *All Words*, a excepción de SE1, donde no hubo dicha tarea y los datos se refieren a la tarea *Lexical Sample*. Podemos ver el año y lugar de celebración, el diccionario utilizado como repositorio de sentidos, el número de instancias de test para la evaluación que se propusieron, el número de sistemas que participaron en la tarea, el baseline que se estableció, el valor F1 obtenido por el mejor sistema y la aproximación que seguía este sistema.

	SE1	**SE2**	**SE3**	**SEM1**	**SEM2**
Año	1998	2001	2004	2007	2010
Lugar	Sussex	Toulouse	Barcelona	Praga	Uppsala
Dicc.	HECTOR	WN1.7	WN1.7.1	WN2.1	WN3.0
Num. instancias test	3500	2473	2037	3500	1398
Num. sistemas	25	22	26	15	29
Baseline (F1)	68,9	57,0	60,9	51,4	50,5
Mejor resultado (F1)	78,1	69,0	65,1	59,1	57,0
Aprox. mejor	Lista decisión	Aprendizaje de patrones	Aprendizaje basado en memoria	Máxima Entropía	?

Tabla 2.4. Resumen competiciones SensEval

3. Herramientas y Recursos

En este capítulo describiremos los recursos y herramientas más importantes que usamos en nuestro trabajo. Tanto las herramientas como los recursos que utilizamos son ampliamente conocidos y empleados en tareas y aplicaciones del Procesamiento del Lenguaje Natural (PLN) en general, y en *Word Sense Disambiguacion* (WSD) en particular. En primer lugar haremos una introducción al método de aprendizaje automático del que hemos hecho uso, Máquinas de Soporte Vectorial (Vapnik, 1995).

Posteriormente describiremos WordNet[1] (Fellbaum, 1998) y EuroWordNet[2] (Vossen, 1998). WordNet es el recurso más empleado en tareas de WSD como repositorio de significados. En nuestro trabajo, dispondremos de él como tal y como repositorio de clases semánticas, ya que normalmente obtendremos dicha información a partir de la organización de sus sentidos.

Una aproximación basada en aprendizaje automático supervisado (Agirre & Edmonds, 2006, Capítulo 7) como la nuestra, hace uso de un corpus anotado para realizar el entrenamiento. También se necesitan corpus anotados para realizar la posterior evaluación del sistema aprendido. Por tanto, también describimos los corpus que usamos, tanto para aprendizaje como para desarrollo y evaluación del sistema.

El punto más importante y en torno al cual gira nuestro trabajo es el uso de clases semánticas. Finalmente se detallan los diferentes conjuntos de clases semánticas que hemos estudiado, junto a sus características más importantes. Veremos los diferentes niveles de abstracción que podemos tratar por medio del uso de los diferentes con-

[1] `http://wordnet.princeton.edu`, consultado en julio de 2010.
[2] `http://www.illc.uva.nl/EuroWordNet`, consultado en julio de 2010.

juntos de clases semánticas. Hay que considerar que estos conjuntos de clases están ya predefinidos. Aparte de ellos, también hemos empleado diversos conjuntos que obtenemos nosotros automáticamente desde WordNet. Estas clases serán detalladas con mayor profundidad en el próximo capítulo.

3.1 Máquinas de Soporte Vectorial (*SVM*)

La teoría de SVM fue desarrollada entre los años 70 y 80 por Vapnik y otros (1995), en el campo del aprendizaje estadístico. Es, por tanto, un método de aprendizaje estadístico basado en ejemplos. Este método también representa los ejemplos mediante vectores de atributos. *SVMLight*[3] es la implementación de SVM que hemos utilizado en este trabajo.

Como se dice en (J. Hernández Orallo, 2004), los modelos SVM pertenecen al grupo de los clasificadores lineales debido a que inducen separadores lineales o hiperplanos en espacios de características de alta dimensionalidad, y maximizan el margen de separación de los ejemplos de entrenamiento en dos o más clases. Suponiendo que el conjunto de datos es linealmente separable, la idea que hay detrás de las SVM de margen máximo es seleccionar el hiperplano separador que está a la misma distancia de los ejemplos más cercanos de cada clase, es decir, el hiperplano que maximiza la distancia mínima entre los ejemplos del conjunto de datos y el hiperplano (es la posición más neutra posible con respecto a las clases representadas por el conjunto de datos sin estar sesgada, por ejemplo, hacia la clase más numerosa). En la figura 3.1 podemos ver el esquema de funcionamiento de una máquina SVM[4]. Los puntos blancos representan ejemplos de una clase, mientras que los negros representan los ejemplos de otra clase. Entre ambos conjuntos se obtiene el hiperplano que los separa, maximizando el margen de separación. Además aparecen resaltados sobre las líneas punteadas aquellos ejemplos que se consideran como vectores de soporte, que son los más cercanos al hiperplano calculado.

[3] `http://svmlight.joachims.org/`, consultado en julio de 2010.

[4] La figura ha sido obtenida de *Wikimedia Commons*, `http://commons.wikimedia.org`

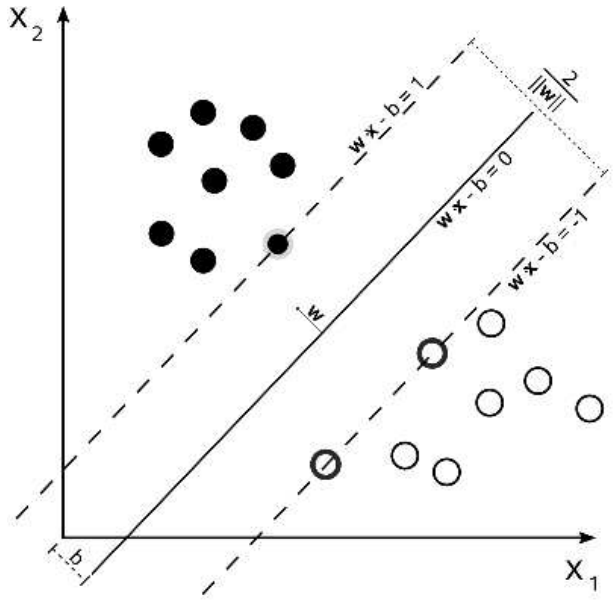


Figura 3.1. Esquema de funcionamiento de SVM

Desde el punto de vista algorítmico, el aprendizaje de SVM representa un problema de optimización con restricciones que se puede resolver con técnicas de programación cuadrática. Ese tipo de técnicas garantiza que siempre habrá una solución única.

En principio los clasificadores generados por SVM son binarios y deberemos realizar ciertos ajustes para afrontar problemas de clasificación donde las posibles clases son más de dos. Una posible aproximación es "binarizar" el problema y construir un clasificador binario para cada clase a la que puedan pertenecer los ejemplos. Una vez que tengamos la salida de todos estos clasificadores deberemos combinarla para obtener el resultado final mediante alguna técnica como "uno contra uno" (*one vs one*), "uno contra todos"(*one vs all*), "todos contra todos", liga, eliminatoria, etc. También existen implementaciones multiclase, que permiten hacer la clasificación en varias clases en lugar de únicamente en dos.

Otra particularidad de los sistemas basados en SVM es que, en su forma básica, aprenden funciones lineales. El aprendizaje de separadores no lineales con SVM se puede conseguir mediante una

transformación no lineal del espacio de atributos de entrada en un espacio de características de dimensionalidad mucho mayor y donde sí sea posible separar los ejemplos. Para esto se introducen las funciones núcleo (en inglés *kernel*), que calculan el producto escalar de dos vectores en el nuevo espacio de características sin necesidad de obtener las transformaciones de los ejemplos. Mediante el uso de las funciones núcleo se pueden aprender clasificadores polinómicos, redes RBF (*Radial Basic Form*), redes neuronales de tres capas,etc. Por el gran número de funciones que son capaces de aprender las SVM, se les suele llamar **clasificadores universales**. En algunos casos los ejemplos no son separables ni tan siquiera en el espacio original de características, o no deseamos obtener un clasificador que se ajuste perfectamente a los datos de entrenamiento (porque sabemos que contiene errores, por ejemplo). En estas ocasiones se permiten ejemplos de entrenamiento mal clasificados con el objetivo de obtener mayor generalización, intentado evitar situaciones de sobreentrenamiento. Son las SVM de margen blando (*soft margin*).

Otra importante característica es que el aprendizaje mediante SVM puede ser independiente de la dimensionalidad del espacio de características. El aprendizaje se basa en el margen de separación de los ejemplos en dos clases, no en el número de atributos que se usen para codificar cada ejemplo. Esto hace que los modelos SVM sean adecuados para problemas de PLN, donde normalmente el número de atributos es muy elevado. Algunas tareas en las que se han utilizado las SVM en el campo del PLN son: recuperación de información (Drucker *et al.*, 2002), clasificación de textos (Joachims, 1998), extracción de información y reconocimiento de entidades (Takeuchi & Collier, 2002), extracción de sintagmas (*chunking* en inglés) (Kudo & Matsumoto, 1995), y, por supuesto, WSD (Lee *et al.*, 2004b), etc.

Un tipo interesante de Máquinas de Soporte Vectorial son las SVM transductivas (Gammerman *et al.*, 1998). El caso general de SVM del que hemos hablado se refiere a las SVM inductivas. En este caso, se genera un único modelo con todos los datos de entrenamiento. Este modelo se usará posteriormente para clasificar nuevos ejemplos desconocidos. En el caso de las SVM transductivas no se genera un modelo general con toda la información de entrenamiento. En esta aproximación, cuando se quiere clasificar un nuevo ejemplo se

selecciona un subconjunto de ejemplos de entrenamiento que sean similares al nuevo ejemplo. Con este subconjunto se genera un modelo específico que será empleado para clasificar el nuevo ejemplo. Este tipo de SVM es muy interesante en aquellos casos en que se dispone de poca información anotada para el entrenamiento, y además la dimensionalidad y dispersión de los ejemplos de entrenamiento es muy elevada. Esto hace que ese tipo de SVM sea interesante para la tarea de WSD.

3.2 WordNet y EuroWordNet

WordNet[5] es una red semántica en inglés, basada en principios psicolingüísticos, creada y mantenida por la Universidad de Princeton (Fellbaum, 1998). Representa conceptos por medio de conjuntos de sinónimos a los que se llama *synsets*. Todas las palabras incluidas en un mismo synset poseen la misma categoría morfológica, y los synsets están enlazados entre sí mediante relaciones semántico–conceptuales y léxicas, conformando así una red semántica. WordNet, en su última versión, la 3.0, contiene alrededor de 155.000 palabras organizadas en 117.000 synsets. Podemos ver un ejemplo del synset para el concepto *automobile* a continuación[6]:

{car.n#1, auto.n#1, automobile.n#1, machine.n#4,
motorcar.n#1}

En cada uno de los synsets, aparte de la información de sentidos de palabras, podemos encontrar también:

- Una glosa, que es una definición textual del concepto que representa el synset, y muchas veces varios ejemplos de uso de las palabras contenidas en el synset
- Un conjunto de relaciones léxicas, que conectan sentidos concretos de palabras. Podemos resaltar:
  - *Antonimia*: representa conceptos opuestos. Por ejemplo *bueno* es antónimo de *"malo"*.

[5] `http://wordnet.princeton.edu`

[6] Debe recordarse que *"car.n#1"* indica el sentido número 1 del nombre *car*

  - *Pertenencia*: *"dental"* pertenece a *"diente"*.
  - *Nominalización*: el nombre *"servicio"* nominaliza el verbo *"servir"*.
- Un conjunto de relaciones semánticas entre synsets, entre las que destacamos:
  - *Hiperonimia*: también conocida como relación *"es–un"* o *"tipo–de"*. Por ejemplo *"vehículo"* es un hiperónimo de *coche*. Es la principal relación semántica que modela la red de WordNet, y se establece entre nombres y verbos únicamente.
  - *Hiponimia o Troponimia*: es la relación contraria a la hiperonimia, para nombres y verbos respectivamente.
  - *Meronimia*: o relación de tipo *"parte–de"*: *"pulpa"* es un merónimo de *"fruta"*. Solo se establece para nombres.
  - *Holonimia*: es la relación contraria a meronimia.
  - *Implicación*: un verbo Y es implicado por un verbo X, si haciendo X debes hacer también Y. Por ejemplo, *"roncar"* implica *"dormir"*.

En la figura 3.2 podemos ver un extracto de WordNet, con varios synsets y varias relaciones entre ellos.

WordNet es también un repositorio de sentidos de palabras, es decir, establece para cada palabra un conjunto de sentidos posibles. Debido a su amplia difusión y al gran número de trabajos de WSD que lo han utilizado, las principales evaluaciones internacionales adoptaron WordNet como repositorio "oficial" de sentidos, estableciendo así esta red semántica como estándar para el campo de WSD. En la tabla 3.1 podemos ver las estadísticas para las diversas versiones de WordNet que hemos utilizado en nuestro trabajo. Describimos el número total de palabras, el número de palabras polisémicas, la cantidad de synsets y sentidos, y finalmente la polisemia media únicamente para las palabras polisémicas.

Siguiendo la filosofía de desarrollo de WordNet, el proyecto EuroWordNet (Vossen, 1998) propuso el alineamiento de diferentes WordNet, que se habían desarrollado en distintas lenguas, mediante un módulo interlingua que conectara los synsets de las diferentes lenguas con los synsets de WordNet en inglés, generándose una base de datos multilingüe (holandés, italiano, español, alemán, francés, checo

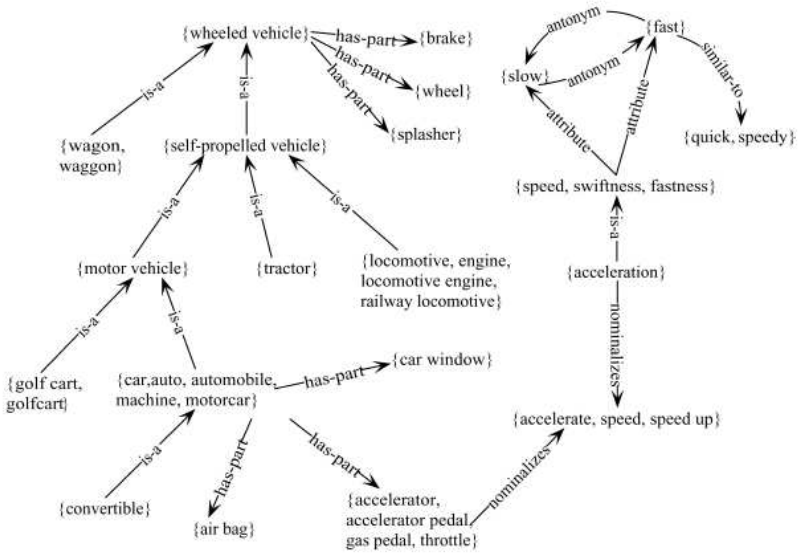


Figura 3.2. Ejemplo extraído de WordNet

Versión	Palabras	Polisémicas	Synsets	Sentidos	Polisemia
WN 1.6	121.962	23.255	99.642	173.941	2,91
WN 1.7	144.684	24.735	109.377	192.460	2,93
WN 1.7.1	146.350	25.944	111.223	195.817	2,86
WN 2.0	152.059	26.275	115.424	203.145	2,94
WN 2.1	155.327	27.006	117.597	207.016	2,89
WN 3.0	155.287	26.896	120.982	206.941	2,89

Tabla 3.1. Estadísticas de las diferentes versiones de WordNet

y estonio). Cada uno de los WordNets representa un sistema de lexicalización para cada una de las lenguas, y está estructurado del mismo modo que el WordNet inglés, como conjuntos de sinónimos relacionados semánticamente. El alineamiento entre los WordNets con el módulo central multilingüe permite el acceso a una ontología de alto nivel con 63 conceptos, que se comparten por todos los WordNets, a pesar de que localmente cada WordNet posea las características concretas que describen la lengua en cuestión. Esta ontología proporciona un marco semántico común a todas las lenguas, definiendo un recurso multilingüe que facilitaría el desarrollo de tareas de WSD en idiomas diferentes al inglés. En la figura 3.3 vemos un esquema de la estructura semántica definida en EuroWordNet.

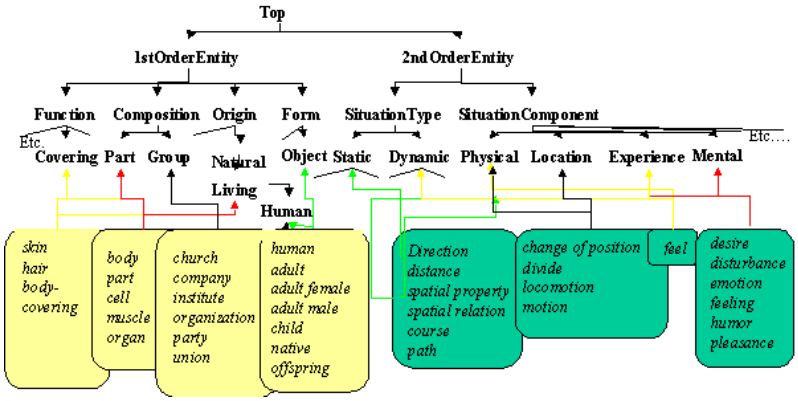


Figura 3.3. Ontología definida en EuroWordNet

Aunque el proyecto EuroWordNet se completó en el verano de 1999, dejando el diseño de la base de datos, las relaciones, la ontología común y el módulo interlingua finalizado, la arquitectura del sistema permite que cualquier otro WordNet que se desarrolle siguiendo las guías del proyecto y se alinee adecuadamente con el módulo interlingua pueda ser añadido a la base de datos multilingüe.

3.3 Corpus

Desafortunadamente, los conjuntos de ejemplos anotados semánticamente en general, y particularmente con sentidos (ya sean de WordNet o de otro diccionario) son escasos, y la mayoría excesivamente pequeños. Esta escasez es aún mayor si nuestro idioma objetivo es alguno distinto del inglés. En esta sección describiremos los corpus utilizados en este trabajo de investigación. Hemos utilizado el corpus SemCor como principal fuente para extraer información anotada semánticamente para entrenar nuestros clasificadores, y varios corpus de evaluación de las competiciones SensEval. Pasamos a ver más en detalle cada uno de ellos.

3.3.1 SemCor

El corpus SemCor (Miller *et al.*, 1993) fue creado en la Universidad de Princeton a partir del corpus en inglés *Brown*. SemCor posee unas 700.000 palabras etiquetadas con su categoría léxica y, de todas estas, unas 200.000 están etiquetadas también con su lema y su sentido (según WordNet 1.6). En concreto se dispone de 35.770 ocurrencias anotadas de adjetivos, 88.398 de nombres y 90.080 de verbos.

El corpus SemCor está compuesto por 352 textos. En 186 de ellos, todas las palabras con contenido semántico (nombres, verbos, adjetivos y adverbios) están anotadas con su categoría léxica, lema y sentido, mientras que en los 166 restantes solo los verbos están anotados con esta información. Los textos para componer el corpus fueron seleccionados a partir de 15 fuentes de información de distinto género (prensa, religión, humor, etc.). De esta forma sabemos el tema general del que trata cada uno de los textos contenidos en el corpus. En cuanto al proceso de anotación, SemCor ha sido anotado de forma paralela por varios anotadores.

3.3.2 Corpus de la tarea *"English all words"* en SensEval–2

Para la tarea *All Words* en inglés (Palmer *et al.*, 2001), en la competición SE2[7], no se proporcionó corpus de entrenamiento, pero sí corpus de evaluación. El corpus consiste en fragmentos de texto extraídos de tres artículos del *Wall Street Journal*[8] pertenecientes a diferentes dominios del corpus *Penn TreeBank II*[9]. En total se había que anotar 2.473 instancias de test, con los sentidos correspondientes de WordNet 1.7.

Ted Pedersen propone una serie de herramientas[10] para transformar de formato los corpus de SE1 y SE2, de modo que las herramientas que trabajen sobre ambos sean compatibles. Además también desarrolla una serie de etiquetadores morfológicos y sintácticos que aceptan texto en el mismo formato que el usado en SE2.

[7] `http://www.sle.sharp.co.uk/senseval2`
[8] `http://europe.wsj.com/home-page`
[9] http://www.cis.upenn.edu/ treebank
[10] `http://www.d.umn.edu/~tpederse/data.html`, consultado en julio de 2010

3.3.3 Corpus de la tarea *"English all words"* en SensEval–3

En este caso, el corpus de evaluación SE3 desarrollado para esta tarea consistió en unas 5.000 palabras extraídas de dos artículos del diario *Wall Street Journal*, y de otro texto del corpus *Brown*. Los tres pertenecen a dominios diferentes, artículos destacados, noticias y ciencia ficción, y fueron seleccionados del corpus *Penn Treebank II*.

Todos los nombres, verbos y adjetivos fueron doblemente anotados haciendo uso de la versión 1.7.1 de WordNet, y finalmente se revisaron por una tercera persona, para aumentar la consistencia de la anotación. También anotaron construcciones multipalabra, en aquellas situaciones en que existía una entrada específica para dicha palabra en WordNet. Además se permitió asignarle a una palabra varios sentidos, aunque se recomendó evitar esta múltiple anotación siempre que fuera posible. También se incluyó en la anotación una etiqueta de sentido especial, para aquellos casos en que ninguna acepción de WordNet representara correctamente el significado de la palabra en el contexto en que aparecía (se usó en estos casos el código "U").

En total se anotaron 2.212 palabras y, sin tener en cuenta algunas construcciones multipalabra, el número total de palabras propuestas para evaluar los sistemas fue 2.037. Debido a los casos en que se disponía de varios sentidos por palabra, el número medio de sentidos anotados por palabra fue 1,03.

3.3.4 Corpus de SemEval–1

En esta competición volvieron a proponer la tarea de *All Words* con granularidad fina (Pradhan *et al.*, 2007). Seleccionaron tres artículos de diferentes secciones del *Wall Street Journal*: uno sobre mendicidad, uno sobre corrupción y otro sobre globos aerostáticos. El texto seleccionado en total contenía 3.500 palabras para ser etiquetadas usando los sentidos de WordNet 2.1. Cada palabra fue etiquetada con su sentido por dos estudiantes de lingüística, y en casos de discrepancia, un experto se encargó de asignar el sentido correcto. La tasa de acuerdo entre ambos anotadores fue de 72 % para verbos y 86 % para nombres.

3.4 Clases semánticas

Existen diferentes conjuntos de clases semánticas, creadas manualmente o semiautomáticamente, de diferentes niveles de abstracción, y centradas en dominios diferentes. En nuestros trabajo hemos usado los ficheros lexicográficos de WordNet, las clases de la ontología SUMO, un conjunto de clases que se conoce como *WordNet Domains* y los conceptos base. Explicamos brevemente a continuación cada uno de estos conjuntos de clases. Los conceptos base serán descritos profundamente en el capítulo 4.

3.4.1 Ficheros lexicográficos de WordNet o *SuperSenses*

Este conjunto de clases ha recibido varios nombres: *lexicographer files of WordNet (Fellbaum, 1998)*, *WordNet Semantic Tags* (Segond *et al.*, 1997), *Semantic Classes of WordNet Semantic Fields of WordNet* (Villarejo *et al.*, 2005) o *SuperSenses* (Ciaramita & Johnson, 2003). En adelante nos vamos a referir a ellas como *SuperSenses*.

Como hemos comentado, WordNet está compuesto por una serie de synsets conectados y relacionados entre sí por diversas relaciones semánticas. Cada uno de estos synsets representa un concepto y contiene un conjunto de palabras que hacen referencia a ese concepto, y por tanto son sinónimos. Estos synsets están clasificados en 45 grupos en función tanto de categorías léxicas (adjetivos, nombres, verbos y adverbios), como de agrupamientos semánticos (persona, fenómeno, sentimiento, lugar, etc.). Existen 26 categorías para nombres, 15 para verbos, 3 para adjetivos y 1 para adverbios. En la tabla 3.2 podemos ver las 45 categorías. Esta organización se ha hecho de este modo para que cada lexicógrafo especialista en una cierta área, se encargue de crear y editar el conjunto de palabras y significados bajo una misma clase semántica. Posteriormente se genera la base de datos de WordNet a partir de los ficheros creados por los lexicógrafos.

Estas categorías semánticas pueden verse como clases semánticas más generales que sus sentidos. Así, varios sentidos de WordNet pueden corresponder con la misma clase semántica. Además se tiene en cuenta los usos más frecuentes y generales de las palabras. Por ejemplo, la palabra *"blood"*, se etiqueta como *"noun.attribute"* (atributos de personas y objetos), como *"noun.body"* (partes del cuerpo)

Adjetivos (3)	Nombres (26)	Verbos (15)	Adverbios (1)
adj.all	noun.Tops	verb.body	adv.all
adj.pert	noun.act	verb.change	
adj.ppl	noun.animal	verb.cognition	
	noun.artifact	verb.communication	
	noun.attribute	verb.competition	
	noun.body	verb.consumption	
	noun.cognition	verb.contact	
	noun.communication	verb.creation	
	noun.event	verb.emotion	
	noun.feeling	verb.motion	
	noun.food	verb.perception	
	noun.group	verb.possesion	
	noun.location	verb.social	
	noun.motive	verb.stative	
	noun.object	verb.weather	
	noun.person		
	noun.phenomenon		
	noun.plant		
	noun.possesion		
	noun.process		
	noun.quantity		
	noun.relation		
	noun.shape		
	noun.state		
	noun.substance		
	noun.time		

Tabla 3.2. Categorías lexicográficas de WordNet

y como *"noun.group"* (grupos de personas y objetos). Sin embargo, *"blood"* no se etiqueta como *"noun.substance"* (sustancias) o como *"noun.food"* (comida), aunque puede poseer estas interpretaciones en ciertos contextos.

Un detalle interesante es que esta categorización está disponible para cualquier lengua que esté contenida en el recurso EuroWordNet, ya que todas los WordNets de distintas lenguas están enlazados con el WordNet en Inglés.

3.4.2 WordNet Domains

WordNet Domains (WND)[11] (Magnini & Cavaglià, 2000) es una extensión de WordNet en donde cada synset se ha anotado con una o varias etiquetas de dominio. Estas etiquetas se estructuran en una

[11] `http://wndomains.fbk.eu`, consultado en julio de 2010

taxonomía de acuerdo a un sistema de clasificación de libros, el *Dewey Decimal Classification System* (este sistema fue seleccionado debido a que asegura una buena cobertura, es fácilmente accesible y es usado muy comúnmente para clasificar material escrito). En total hay unas 160 clases diferentes.

La información de dominios es complementaria a la información contenida en WordNet. En primer lugar, un dominio puede contener synsets de distinta categoría léxica: por ejemplo el dominio MEDICINE agrupa sentidos de nombres, como *"doctor.n#1"* y *"hospital.n#1"*, y de verbos, como *"operate.v#7"*. En segundo lugar, un dominio puede agrupar synsets de distintas subjerarquías de WordNet (de hiperonimia, derivadas de los *SuperSenses*, etc.). Por ejemplo, SPORT contiene sentidos como *"athlete.n#1"*, derivado de *"noun.person"*, o *"game_equipment.n#1"* derivado de *"noun.artifact"*, *"sport.n#1"* de *"noun.act"* y *"playing_field.n#2"* de *"noun.location"*.

Además, WND agrupa sentidos de diferentes palabras de manera que se reduce el grado de detalle y polisemia. Podemos ver este fenómeno en el cuadro 3.3. La palabra *"bank"* tiene diez sentidos distintos en WordNet 1.6, pero tres de ellos (el #1,el #3 y el #6) pueden ser agrupados bajo el dominio ECONOMY, mientras que los sentidos #2 y #7 pertenecen a los dominios GEOGRAPHY y GEOLOGY respectivamente. Obviamente, agrupar sentidos para obtener un grado de detalle más bajo y reducir la polisemia es un tema de gran interés en WSD.

En cuanto al proceso de creación y anotación de los synsets, se realizó de forma semiautomática: un pequeño número de synsets de alto nivel se anotaron manualmente con su dominio correcto y posteriormente, se utilizaron las relaciones semánticas de WordNet para expandir esa anotación al resto de synsets.

3.4.3 La Ontología SUMO

Suggested Upper Merged Ontology (SUMO)[12] (Niles & Pease, 2001), auspiciado por el grupo *IEEE Standard Upper Ontology Working Group* pretende ser una ontología estándar de alto nivel que promueva la interoperabilidad de datos, la búsqueda y recuperación

[12] `http://www.ontologyportal.org`, consultado en julio de 2010.

Dominios	Sent.	Glosa
ECONOMY	#1	depository financial institution, bank, bank concern, banking company
	#3	bank (a supply or stock held in reserve...)
	#6	savings bank, coin bank, money box, bank (a container...)
GEOGRAPHY GEOLOGY	#2	bank (sloping land...)
	#7	bank (a long ridge or pile...)
ARCHITECTURE ECONOMY	#4	bank, bank building
FACTOTUM	#5	bank (an arrangement of similar objects)
ECONOMY PLAY	#8	bank (the funds held by a gambling house)
ARCHITECTURE	#9	bank, cant, camber (a slope in the turn of a road)
TRANSPORT	#10	bank (a flight maneuver)

Tabla 3.3. Sentidos de WordNet y WordNet Domains para la palabra *bank* (nombre)

de información, la inferencia automática y el procesamiento del lenguaje natural entre otras muchas áreas.

Una ontología es en cierto modo similar a un diccionario, pero con mayor formalismo y una estructura que haga posible procesar su contenido de forma automática mediante un ordenador. Además, una ontología contiene tanto conceptos como relaciones y axiomas que modelan un campo de interés determinado, mediante un lenguaje formal de consulta, creación y mantenimiento. Se dice que una ontología es de alto nivel cuando define conceptos genéricos o abstractos que cubren la terminología de un conjunto de áreas y dominios suficientemente grande. Este tipo de ontologías no incluye conceptos de dominios específicos, pero establece la forma en que pueden ser construidas otras ontologías de dominios específicos y como pueden enlazarse con los conceptos genéricos de la ontología de alto nivel.

SUMO provee definiciones para términos de propósito general y, además, ofrece varías ontologías de dominios específicos (como comunicación, países y regiones, economía y finanzas,etc.) relacionadas con los términos de la ontología principal de alto nivel. SUMO también ofrece mapeos creados de forma automática para conectar sus conceptos con los synsets y conceptos de WordNet 1.6. Con ello, podemos establecer para cada synset de WordNet su categoría en SUMO asociada. Por ejemplo, *"bank.n#1"* posee la categoría CORPORATION en SUMO, *"bank.n#1"* se corresponde con LAND AREA

o *"bank.n#4"* con BUILDING. En su última versión contiene 20.000 términos, 60.000 axiomas y 687 clases diferentes. En la figura 3.4 podemos ver las clases de más alto nivel de SUMO. Por encima de *Physical* y *Abstract* estaría el nodo raíz de la ontología: *Entity*.

```
Entity
    Physical
        Object
            SelfConnectedObject
                ContinuosObject
                CorpuscularObject
            Collection
        Process
    Abstract
        SetClass
            Relation
        Proposition
        Quantity
            Number
            PhysicalQuantity
        Attribute
```

Figura 3.4. Nivel superior de SUMO

3.4.4 Conceptos Base

La noción de Conceptos Base (*Base Concepts* (BC) en inglés) surgió en el proyecto EuroWordNet como medio de asegurar el máximo solapamiento y compatibilidad entre los WordNets de las diferentes lenguas que se desarrollarían en EuroWordNet. Los BC son por tanto aquellos conceptos que juegan un papel muy importante en varios WordNets. Las dos características más importantes que definen a estos conceptos son:

1. Poseer una posición elevada en la jerarquía semántica.
2. Poseer muchas relaciones con otros conceptos.

Los BC, por tanto, son aquellos conceptos que proporcionan las relaciones básicas en un WordNet y definen los patrones de lexicalización dominantes en cada lengua. Este tipo de conceptos no se

debe confundir con los Conceptos de Nivel Básico (*Base Level Concepts* (BLC) en inglés), que fueron definidos en (Rosch, 1977), y surgen del compromiso entre dos principios de caracterización:

1. Representar tantos conceptos como sea posible.
2. Representar tantas características como sea posible.

Como resultado de este compromiso, los BLC tienden a aparecer en zonas intermedias de las jerarquías semánticas, como nivel de abstracción medio entre conceptos muy generales y conceptos muy concretos.

Volviendo a los BC, podemos diferenciar entre tres tipos en función de las lenguas (o WordNets) en los que dichos conceptos actúan como Concepto Base:

1. BC Comunes (CBC): son aquellos que actúan como Conceptos Base en al menos dos lenguas.
2. BC Locales (LBC): son aquellos que actúan como Conceptos Base en una sola lengua.
3. BC Globales (GBC): son aquellos que actúan como Conceptos Base en todas las lenguas.

En EuroWordNet inicialmente se seleccionó un conjunto de 1.024 conceptos como CBC, aquellos que actúan como Conceptos Base en al menos dos lenguas independientes. Solo los WordNets en inglés, holandés, y español fueron usados para la selección de este conjunto de CBC. Teniendo en cuenta los conceptos importantes que aparecían en al menos 3 WordNets, se construyó una lista de 164 conceptos, que se llamó *Core Base Concepts*. Esta lista se revisó, eliminando algunos synsets, y juntando otros que estaban muy próximos (por ejemplo, *act* y *action*), y se compiló una lista de 71 conceptos conocidos como *Base Types*, que son aquellos conceptos más fundamentales y generales. A partir de la lista de 1.024 CBC se seleccionaron los 63 conceptos que actúan como ontología de nivel superior en EuroWordNet, y que hemos descrito en la sección 3.2.

En el proyecto BalkaNet[13] (Sofia *et al.*, 2002), se siguió una aproximación similar a la de EuroWordNet, pero considerando otro conjunto de lenguas: griego, rumano, serbio, turco y búlgaro. En este caso el

[13] `http://www.ceid.upatras.gr/Balkanet`, consultado en julio de 2010.

conjunto de CBC seleccionados fue de 4.689 conceptos (3.210 para nombres, 1.442 para verbos y 37 para adjetivos). En el proyecto MEANING[14] (Rigau *et al.*, 2003), el número de BC seleccionados de WordNet 1.6 fue 1.535 (793 para nombres y 742 para verbos). En este caso las lenguas representadas fueron español, inglés, italiano, euskera y catalán.

[14] `http://www.lsi.upc.edu/~nlp/meaning/documentation/index.html`, consultado en julio 2010.

4. Un Recurso Semántico: los "*Basic Level Concepts*"

El punto central de nuestro trabajo, como ya hemos comentado anteriormente, son las clases semánticas. Además, el principal objetivo es el uso de estas clases para generar los clasificadores de un sistema de WSD, de modo que dispongamos de un nivel de abstracción superior al que proporcionan los sentidos demasiado detallados de las palabras.

Por tanto, podemos considerar dos aspectos, el uso y estudio de diferentes conjuntos de clases semánticas, y su aplicación en un sistema de WSD como nuevo nivel de abstracción para la generación de los clasificadores. En este capítulo nos centraremos en el primero, en el estudio de conjuntos de clases semánticas, y más concretamente, en un método que hemos desarrollado para obtener conjuntos de clases semánticas aprovechando la estructura de WordNet. También realizaremos un análisis de dichos conjuntos, así como su adecuación a la tarea de WSD.

Muchas investigaciones se han dirigido a obtener agrupaciones de sentidos para solucionar el problema la excesiva granularidad de los sentidos de WordNet (Hearst & Schütze, 1993; Peters *et al.*, 1998; Mihalcea & Moldovan, 2001a; Agirre & LopezDeLaCalle, 2003; Navigli, 2006; Snow *et al.*, 2007). Mediante diferentes técnicas se realizan agrupaciones de sentidos. En todos los casos se han agrupado sentidos de la misma palabra, con lo que se reduce la polisemia y se incrementan por tanto los resultados de un sistema de desambiguación basado en dichas agrupaciones. De cualquier modo, este tipo de aproximaciones solo intenta obtener ventaja de la reducción de polisemia, y no del aumento del número de ejemplos de entrenamiento para entrenar cada clasificador. Por otra parte, existen muchas otras aproximaciones que se han centrado en hacer uso de conjuntos de

clases semánticas, o agrupaciones de sentidos más bien, para su directa aplicación en un sistema de desambiguación semántica basada en clases (Segond *et al.*, 1997; Ciaramita & Johnson, 2003; Curran, 2005; Ciaramita & Altun, 2006).

Sin embargo, existen pocas aproximaciones que hayan obtenido automáticamente dicho conjunto de clases semánticas, con diferentes niveles de abstracción, desde los más generales a los más concretos. La sección 4.1 se centra en la generación de diferentes conjuntos de clases semánticas, cada uno con un cierto nivel de granularidad, y en el análisis de las características más importantes de cada conjunto. Se intuye que la aplicación de WSD a diferentes tareas finales de PLN será más eficaz si se proporciona el grado de abstracción que cada una necesite, de ahí el interés de nuestra propuesta.

Además, diseñamos una serie de experimentos para evaluar la calidad de dichas clases semánticas. En este caso no pretendemos desarrollar un sistema completo de WSD, solo hacer una primera aproximación de cómo funcionaría un sistema de este tipo haciendo uso de dichas clases semánticas. Para ello, seguimos una heurística muy sencilla para realizar una evaluación sobre SE3, que consiste en asignar la clase más frecuente en SemCor para cada palabra. En la sección 4.2 podemos ver este punto desarrollado extensamente.

También estudiamos la calidad de los conjuntos de clases obtenidas, mediante de su aplicación en un sistema de alto nivel de PLN. En la sección 4.3 mostramos cómo integramos nuestros conjuntos semánticos en una aplicación de recuperación de información, así como la participación de dicho sistema en una competición internacional de recuperación de información.

Finalmente, detallaremos el uso que se le ha dado a nuestros conjuntos de clases semánticas en el proyecto europeo Kyoto[1] El objetivo de este proyecto es permitir que diferentes comunidades de personas puedan representar sus términos y palabras junto con su significado en una plataforma de tipo Wiki. Esta plataforma servirá de puente común entre culturas, permitiendo que dos personas de dos culturas con distinta lengua puedan compartir conocimiento, pero que

[1] `http://www.kyoto-project.eu`, consultado en julio de 2010.

también una computadora sea capaz de acceder a la información contenida en dicha plataforma y procesarla.

4.1 Generación de los *Basic Level Concepts*

En esta sección explicamos el desarrollo de un método para obtener un conjunto de clases semánticas, llamadas BLC, a partir de la jerarquía de WordNet. Explicaremos el algoritmo que hemos desarrollado, su funcionamiento y la parametrización que podemos ajustar. Posteriormente, presentaremos un estudio y comparación de los diferentes conjuntos que obtenemos mediante la modificación de los parámetros del algoritmo. Tanto el *software* que implementa nuestro método de selección, como los distintos conjuntos de BLC extraídos desde WordNet pueden ser usados y descargados libremente desde `http://www.dlsi.ua.es/~ruben`.

A modo de recordatorio citaremos de nuevo los criterios que hemos seguido para diseñar nuestro algoritmo de selección de BLC, que ya fueron descritos en (Rosch, 1977). En concreto los BLC son conceptos que surgen de un compromiso entre varios criterios:

- Que representen un número adecuado de conceptos (que sean generales)
- Que representen un gran número de características
- Que tengan una frecuencia elevada

4.1.1 Método de Selección Automática de BLC

Como ya hemos comentado, hemos desarrollado un método para obtener un conjunto BLC desde WordNet. El método puede ser aplicado a cualquier versión de WordNet, tanto en su parte nominal como verbal[2]. La idea principal del método es que, synsets con muchas relaciones son synsets importantes, y por tanto son candidatos a ser conceptos relevantes. Básicamente, el algoritmo selecciona el

[2] El algoritmo hace uso únicamente de los ficheros de texto plano de WordNet. Además, sólo es posible aplicarlo a nombres y verbos debido a que son las únicas dos categorías para las que se dispone de estructura jerárquica en WordNet ya que el algoritmo se basa en estas estructuras.

BLC apropiado para un synset en concreto considerando el número relativo de relaciones de los hiperónimos de dicho synset. Tanto el algoritmo de selección automática como los distintos conjuntos de BLC que generamos pueden ser descargados de forma libre desde `http://www.dlsi.ua.es/~ruben`.

En concreto, se obtienen conjuntos de BLC diferentes en función del tipo de relaciones consideradas:

1. *All*: todos los tipos de relaciones codificadas en WordNet.
2. *Hypo*: solo las relaciones de hiponimia contenidas en WordNet.

La selección de estos criterios no es casual. En el primer caso, es la aproximación más obvia, podemos disponer de todas las relaciones codificadas en WordNet, para extraer la mayor información posible en cuanto a la importancia de los synsets. En el caso *Hypo*, elegimos las relaciones de hiperonimia/hiponimia debido a que son las que definen la jerarquía de WordNet, en la que se centra el algoritmo para extraer BLC.

El proceso sigue un recorrido ascendente a través de la cadena de hiperonimia definida mediante las relaciones correspondientes en WordNet. Para cada synset de WordNet, se selecciona como su BLC apropiado el primer máximo local de acuerdo al número relativo de relaciones. Para synsets que tienen varios hiperónimos, se selecciona el camino que tenga el máximo local, con el mayor número de relaciones. El proceso termina con un conjunto de conceptos candidatos iniciales, con el conjunto de synsets seleccionados como BLC para algún otro synset. Entre éstos, hay algunos que no representan a un número suficiente de conceptos o synsets. Para evitar estos "falsos" BLC, realizamos un proceso de filtrado final, en el que se eliminan aquellos que no representan a un mínimo número de conceptos determinado. Este número mínimo de conceptos exigidos para ser un BLC válido, es un parámetro de nuestro algoritmo al que llamaremos umbral λ. Mediante la combinación de diferentes valores para el umbral λ y el tipo de relaciones consideradas (todas o solo de hipo/hiperonimia) obtenemos diferentes conjuntos de BLC. Los synsets que después del proceso de filtrado queden sin BLC asignados (debido a que su BLC sea eliminado), son procesados de nuevo para asignarles otro BLC válido en su cadena de hiperonimia. Por ejem-

plo, para el sentido "church.n#1" su BLC es "faith.n#3". Si éste no hubiera podido ser seleccionado como BLC por no superar el umbral filtro, el siguiente BLC asignado hubiera sido "organization.n#2".

En la figura 4.1 podemos ver un ejemplo del funcionamiento del algoritmo. Hay que tener en cuenta que no es un ejemplo real extraído de WordNet, únicamente trata de simular el funcionamiento del algoritmo sobre un grupo de synsets, enlazados mediante relaciones de hiperonimia. Cada uno de los nodos representa un synset de WordNet, y cada una de las aristas entre ellos representa las relaciones de hiperonimia. Así, el nodo A es hiperónimo de D, y el C es hipónimo de A. En el ejemplo vemos el recorrido que seguiría el algoritmo para asignar un BLC al synset J. Comienza el proceso en synset J, y se asciende hasta su hiperónimo, el synset F, cuyo número de relaciones de hiponimia es 2. El proceso continúa ascendiendo hasta el synset D, cuyo número de relaciones es 3. Es importante notar que aunque el synset F tiene 2 hiperónimos, B y D, el proceso elige aquel que tiene un mayor número de relaciones: el D. Finalmente se asciende hasta el synset A, y debido a que el número de relaciones del nuevo nodo desciende (A tiene 2 y D tiene 3), el proceso se detiene y se selecciona D como BLC para el synset J. El conjunto global de BLC se compone de todos los synsets seleccionados como BLC.

En la tabla 4.1 podemos ver un ejemplo real de los posibles BLC para el nombre *church* en WordNet 1.6. Se muestra la cadena de hiperonimia para cada synset junto al numero de relaciones codificadas en WordNet para el synset (estaríamos haciendo uso del conjunto de relaciones *All*). El máximo local a lo largo de la cadena de hiperonimia para cada synset aparece resaltado. Como ya comentamos, diferentes criterios hubieran producido un conjunto diferente de BLC.

Además del criterio de seleccionar conceptos con muchas relaciones, podemos tener en cuenta también la frecuencia como posible indicador de clases que representan un gran número de características, por tanto importantes. Definimos otra medida para la selección de BLC: la frecuencia relativa de los synsets[3] en la cadena de hiperonimia. Ahora el algoritmo hace uso de esta nueva medida para encontrar el máximo local según dicha medida en la cadena de hi-

[3] Calculada sobre algún corpus anotado a nivel de sentidos, como SemCor.

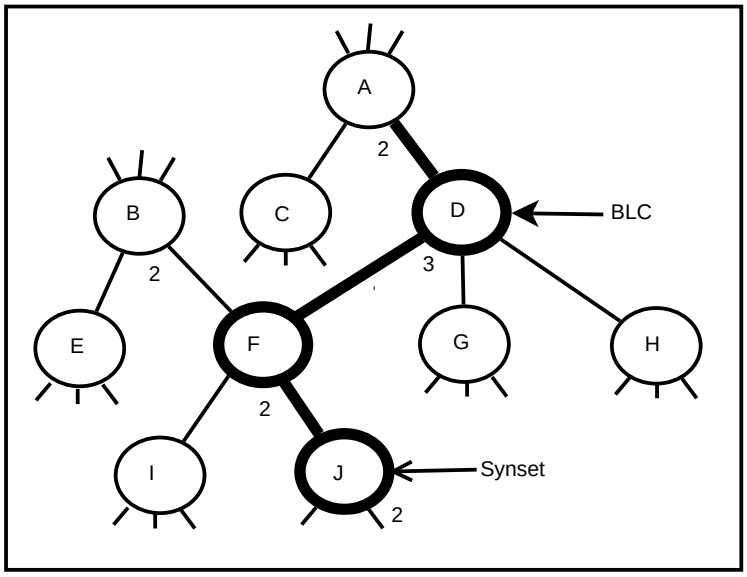


Figura 4.1. Ejemplo del algoritmo

#rel.	**synset**
18	group_1,grouping_1
19	social_group_1
37	organisation_2,organization_1
10	establishment_2,institution_1
12	faith_3,religion_2
5	Christianity_2,**church_1**,Christian_church_1

#rel.	**synset**
14	entity_1,something_1
29	object_1,physical_object_1
39	artifact_1,artefact_1
63	construction_3,structure_1
79	building_1,edifice_1
11	place_of_worship_1, ...
19	**church_2**,church_building_1

#rel.	**synset**
20	act_2,human_action_1,human_activity_1
69	activity_1
5	ceremony_3
11	religious_ceremony_1,religious_ritual_1
7	service_3,religious_service_1,divine_service_1
1	**church_3**,church_service_1

Tabla 4.1. Posibles BLC para el nombre *Church* en WordNet 1.6

peronimia de un synset y seleccionar así su BLC. Generamos otros posibles conjuntos de BLC dependiendo de la fuente sobre donde se calcula la frecuencia relativa de los synsets[4]:

1. *FreqSC*: se obtiene la frecuencia sobre SemCor
2. *FreqWN*: se utiliza la frecuencia para los synsets contenida en WordNet.

La frecuencia para un synset se obtiene sumando la frecuencia de las palabras (con su sentido correcto) contenidas en él. De hecho, las frecuencias representadas en WordNet para las palabras se calcularon sobre SemCor y una parte del corpus Brown anotada a nivel de sentidos. Por tanto, la frecuencia de una palabra según WordNet y la calculada manualmente sobre SemCor no son necesariamente iguales.

4.1.2 Estudio y Comparación de BLC

Aplicando el algoritmo descrito en la sección anterior sobre distintas versiones de WordNet, decidiendo si tener en cuenta el número de relaciones (*All* o *Hypo*) o la frecuencia de los synsets (*FreqSC* o *FreqWN*), así como ajustando el umbral λ[5], se consiguen diversos conjuntos de BLC. Hemos seleccionado varios valores para dicho umbral, variando desde 0 (sin realizar ningún filtrado por número de conceptos englobados), hasta 50, debido a que con este nivel de filtrado se obtienen conjuntos de BLC más abstractos. En la tabla 4.2 presentamos el número total de synsets considerados como conceptos para cada conjunto, además de la profundidad media de cada conjunto en la jerarquía de WordNet[6]. Podemos ver las estadísticas para los diferentes conjuntos que se obtienen en función del umbral λ que limita el mínimo número de conceptos, y según el tipo de criterio utilizado por el algoritmo para obtener el conjunto de máximos locales: número de relaciones (todas, *All* o las de hiponimia, *Hypo*), o la

[4] De hecho, para WordNet 3.0 podríamos usar también los sentidos anotados de las glosas.

[5] Como hemos comentado, λ representa el mínimo número exigido de conceptos englobados por un BLC

[6] Para calcular la profundidad de un synset, contamos el camino más corto en la jerarquía de hiperonimia de WordNet, desde el synset ascendiendo hasta un synset raíz, que no tenga más hiperónimos.

frecuencia de los synsets (calculada sobre SemCor, *freqSC* o obtenida de WordNet, *freqWN*).

Umbral	Tipo	BLC		Profundidad	
		Nombres	Verbos	Nombres	Verbos
0	All	3.094	1.256	7,09	3,32
	Hypo	2.490	1.041	7,09	3,31
	freqSC	34.865	3.070	7,44	3,41
	freqWN	34.183	2.615	7,44	3,30
10	All	971	719	6,20	1,39
	Hypo	993	718	6,23	1,36
	freqSC	690	731	5,74	1,38
	freqWN	691	738	5,77	1,40
20	relAll	558	673	5,81	1,25
	relHypo	558	672	5,80	1,21
	freqSC	339	659	5,43	1,22
	freqWN	340	667	5,47	1,23
50	All	253	633	5,21	1,13
	Hypo	248	633	5,21	1,10
	freqSC	94	630	4,35	1,12
	freqWN	99	631	4,41	1,12

Tabla 4.2. Conjuntos de BLC obtenidos automáticamente sobre WordNet 1.6

Tal y como cabía esperar, el aumento en el umbral λ tiene un efecto inmediato tanto en el número de conceptos seleccionados, como en su profundidad media. Ambos valores descienden, lo que implica un aumento en la abstracción de nuestros conjuntos de BLC. Dicho de otro modo, con umbrales mayores se seleccionan conceptos más generales y que además, aparecen en posiciones más elevadas dentro de la jerarquía de WordNet. Por ejemplo, utilizando todas las relaciones en la parte nominal de WordNet, el número total de BLC varía desde 3.904 (sin umbral) hasta 253 (con umbral 50). Sin embargo, aunque el número total de BLC para los nombres disminuye, la profundidad media de los synsets seleccionados solo varía desde 7,09 a 5,21 usando ambos tipos de relaciones (todas y solo hiponimia). Este hecho demuestra la robustez de nuestro método, ya que aunque disminuimos fuertemente el número de conceptos totales, mantenemos un nivel de abstracción intermedio (BLC en niveles medios de la jerarquía de hiperonimia de WordNet).

Por otra parte, atendiendo a la parte verbal de WordNet se obtienen resultados diferentes. En este caso, debido a que las jerarquías de

hiperonimia son mucho más cortas para verbos que para nombres, la profundidad media de los synsets seleccionados varía de 3,32 a 1,13 utilizando todas las relaciones, y de 3,31 a 1,10 usando solo las de hiponimia.

En general, cuando utilizamos el criterio basado en frecuencia de los synsets, se puede observar un comportamiento similar al obtenido considerando el número de relaciones. Sin embargo, siguiendo el criterio de frecuencia de synsets, el efecto ocasionado por la modificación del umbral es más acusado, en especial para los nombres. De nuevo, aunque el número de conceptos para nombres desciende considerablemente aumentado el umbral, la profundidad media solo desciende desde 7,44 a 4,35 y 4,41. Del mismo modo, la parte verbal obtiene resultados diferentes. También es interesante destacar que, el número de BLC en ambos casos (relaciones o frecuencia), está en torno a 600 utilizando umbrales intermedios, lo que está muy próximo al número de synsets de más alto nivel para verbos[7].

A modo de comparación, mostramos también las estadísticas para los conjuntos de BC seleccionados en los proyectos BALKANET y MEANING. Estos valores podemos verlos en la tabla 4.3.

	Categoría	**Núm. BLC**	**Profundidad**
BALKANET	Nombre	3.210	5,08
	Verbo	1.442	2,45
MEANING	Nombre	793	4,93
	Verbo	742	1,36

Tabla 4.3. Estadísticas para los BLC seleccionados en MEANING y BALKANET

En el caso de BALKANET el número de conceptos seleccionados es elevado, del mismo orden que nuestros BLC cuando no realizamos filtrado, pero la profundidad media es menor que en el mismo caso para nuestros BLC (con umbral $\lambda = 0$). En el caso de MEANING, ambos valores son coherentes con nuestros datos, tanto el número de BLC como su profundidad media, seguramente derivado de la similitud de las aproximaciones de selección de conceptos en ambos casos.

[7] En inglés, *top beginners*, son aquellos synsets más altos en la cadena de hiperonimia, que no poseen hiperónimos. Están por tanto en el nivel más alto de la jerarquía

4.2 Evaluación Inicial de los BLC

Como hemos mencionado, en esta sección obtenemos una primera aproximación de la calidad de los conjuntos de BLC obtenidos, mediante su uso en una tarea de desambiguación semántica. No se trata de obtener un sistema de desambiguación, no utilizaremos el contexto de las palabras, ni ningún método de aprendizaje. Tan solo aplicamos una heurística muy sencilla, basada en asignar la clase más frecuente, obtenida a partir de un recurso semántico, a las palabras del corpus de evaluación de la tarea *All Words* de *SensEval–3* (SE3). Con esta aproximación, nos haremos una idea sobre el rendimiento que podemos esperar de dichas clases semánticas en un sistema de desambiguación de las palabras.

4.2.1 Agrupando sentidos a modo de clases semánticas

Para estudiar hasta dónde pueden ser de utilidad las agrupaciones de sentidos generadas mediante las clases semánticas para la desambiguación del sentido de las palabras, presentamos una comparativa de su aplicación en un marco de evaluación controlado. En concreto probamos el comportamiento de las diferentes agrupaciones de sentidos (BC de BALKANET, BC de MEANING, BLC obtenidos automáticamente y SuperSenses), además de los tradicionales sentidos en la tarea *All Words* de SE3. Obviamente, diferentes agrupaciones proporcionarán representaciones con distinto nivel de abstracción del contenido de WordNet.

La forma más básica de probar la adecuación de dichas clases semánticas a un sistema de desambiguación, es utilizar el sentido o clase más frecuente. De hecho, este *baseline* se tiene en cuenta en todas las competiciones y evaluaciones de sistemas de desambiguación, a modo de resultado mínimo que cualquier sistema de desambiguación competitivo debe superar (Gale *et al.*, 1992a). En este estudio empleamos esta técnica, simple pero robusta, para probar el comportamiento de las diferentes clases semánticas. Usamos SemCor para calcular los sentidos o clases más frecuentes para una determinada palabra. En concreto solo utilizamos las partes *brown1* y *brown2* de SemCor para calcular estas frecuencias, ya que la parte *brown–v*

solo tiene etiquetados los verbos. Utilizamos las medidas estándar de precisión, cobertura (en inglés *recall*) y valor F1[8] para evaluar el funcionamiento de cada clasificador.

Para los sentidos de WordNet, los BC de MEANING, nuestros BLC automáticamente generados y los ficheros Lexicográficos de WordNet, utilizamos WordNet en su versión 1.6, que será la usada como repositorio de referencia en nuestros experimentos[9]. Los BC de BALKANET están alineados con synsets de WordNet 2.0, por tanto empleamos un mapeo facilitado por (Daudé *et al.*, 2003) para alinear dichos BC con synsets de WordNet 1.6. También, para usar el corpus de SE3 necesitamos transformar los sentidos utilizados en dicho corpus (pertenecientes a WordNet 1.7.1) a sentidos de WordNet 1.6, para mantener la homogeneidad de versiones en los experimentos.

En la tabla 4.4 presentamos el grado de polisemia medio para nombres y verbos de las palabras contenidas en el corpus SE3, de acuerdo a diferentes conjuntos de clases semánticas. La columna Sentidos representa la polisemia utilizando sentidos tradicionales, la columna BLC–A representa los BLC automáticamente obtenidos utilizando un umbral de 20[10] y considerando todos los tipos de relaciones (*All*), BLC–S representa a los BLC obtenidos utilizando un umbral de 20 y considerando la frecuencia sobre SemCor (*FreqSC*) y *SuperSenses* representa a los Ficheros Lexicográficos de WordNet.

	Sentidos	**BLC-A**	**BLC-S**	**SuperSenses**
Nombres	4,93	4,07	4,00	3,06
Verbos	11,00	8,64	8,72	4,08
Nom. & Ver.	7,66	6,13	6,13	3,52

Tabla 4.4. Grado de polisemia sobre el corpus SensEval–3

Lógicamente, cuando aumentamos el nivel de abstracción, la polisemia decrece. Particularmente para verbos: de 11,0 a sólo 4,08.

[8] Recordemos que el valor F1 resulta de obtener la media harmónica de precisión y cobertura con el valor β=1. $F1 = 2 * P * C/(P + C)$.

[9] Utilizamos esta versión ya que los sentidos anotados inicialmente en SemCor corresponden con la versión 1.6 de WordNet

[10] Hemos elegido este umbral ya que proporciona unas clases con nivel de generalidad medio.

En el caso de los BLC, se mantiene un grado de polisemia bastante elevado tanto para nombres como para verbos.

En las tablas 4.5 y 4.6 podemos ver el funcionamiento de nuestro sistema para las palabras polisémicas del corpus de la tarea *All words* de SE3, para diferentes conjuntos de clases semánticas, y utilizando el corpus SemCor como recurso para calcular la clase semántica más frecuente para una palabra en concreto. En otras palabras, para cada palabra polisémica de SE3, obtenemos la clase más frecuente para dicha palabra sobre el corpus SemCor, según el conjunto de clases semánticas que estemos considerando, y asignamos dicha clase como resultado de la desambiguación. Los mejores resultados aparecen resaltados en negrita.

	All		***Hypo***	
Clase	**Nombres**	**Verbos**	**Nombres**	**Verbos**
Sentidos	63,69	49,78	63,69	49,78
Balkanet	65,15	50,84	65,15	50,84
Meaning	65,28	53,11	65,28	53,11
BLC–0	66,36	54,30	65,76	54,30
BLC–10	66,31	54,45	65,86	54,45
BLC–20	**67,64**	54,60	**67,28**	54,60
BLC–30	67,03	54,60	66,72	54,60
BLC–40	66,61	55,54	66,77	**55,54**
BLC–50	67,19	**55,69**	67,19	**55,54**
SuperSenses	**73,05**	**76,41**	**73,05**	**76,41**

Tabla 4.5. Valores F1 del *baseline* para las palabras polisémicas en el corpus de SE3, teniendo en cuenta el número de relaciones

	Semcor		**WordNet**	
Clase	**Nombres**	**Verbos**	**Nombres**	**Verbos**
Sentidos	63,69	49,78	63,69	49,78
Balkanet	65,15	50,84	65,15	50,84
Meaning	65,28	53,11	65,28	53,11
BLC–0	64,45	52,27	64,95	51,75
BLC–10	64,98	53,21	65,59	53,29
BLC–20	65,73	53,97	66,30	53,44
BLC–30	66,46	54,15	66,67	53,61
BLC–40	68,46	**54,63**	**69,16**	54,22
BLC–50	**68,84**	**54,63**	69,11	**54,63**
SuperSenses	**73,05**	**76,41**	**73,05**	**76,41**

Tabla 4.6. Valores F1 del *baseline* para las palabras polisémicas en el corpus de SE3 teniendo en cuenta la frecuencia de las palabras

Como era de esperar según los datos sobre la polisemia de la tabla 4.4, los mejores resultados se obtienen con *SuperSenses*, con un valor F1 muy alto tanto para nombres como para verbos. Comparando los resultados obtenidos con los Conceptos Base seleccionados en los proyectos MEANING y BALKANET, los primeros superan a los segundos, tanto para nombres como para verbos. Si tenemos en cuenta que el conjunto de BC de BALKANET es más extenso que el de MEANING, nuestros resultados indican que los BC de MEANING proporcionan un mejor nivel de abstracción. En este punto es importante resaltar que, a pesar de que se obtengan mejores resultados con SuperSenses que con nuestros BLC, lo que buscamos no es obtener los mejores resultados, si no disponer de un conjunto de clases con una abstracción intermedia lo suficientemente rica como para obtener buenos resultados sin perder poder de representación.

Considerando los conjuntos de BLC creados mediante el criterio de número de relaciones, todos los conjuntos generados poseen unos resultados mejores que los conseguidos por los BC de MEANING o BALKANET. Los mejores se obtienen utilizando un umbral de 20 (BLC–20). Este resultado es muy interesante, debido a que este conjunto obtiene mejor resultado que los restantes BLC, manteniendo más información de los synsets originales y además, menos genérica. Por ejemplo, BLC–20, utilizando todos los tipos de relaciones (558 clases), obtiene un valor de F1 igual a 67,64, mientras que *SuperSenses*, utilizando un conjunto mucho menor (26 clases) obtiene 73,05. Ambos resultados no están desmesuradamente lejanos, a pesar de estar hablando de dos niveles de abstracción totalmente distintos.

También podemos observar que, en general, usando el criterio basado en relaciones de hiponimia se obtienen resultados ligeramente inferiores que utilizando todo tipo de relaciones. Posiblemente este hecho indique que se caracterizan mejor los BLC al tener en cuenta todas las relaciones, lo cual es lógico ya que se utiliza mayor cantidad de información. Los resultados en general sugieren que niveles de abstracción intermedios, como los proporcionados por los BLC automáticamente creados, podrían ser apropiados para el aprendizaje y construcción de clasificadores basados en clases semánticas para la tarea de desambiguación del sentido de las palabras.

Atendiendo a los resultados obtenidos con los BLC según elcriterio de frecuencia de palabras, mostrados en la tabla 4.6, podemos ver que en este caso no todos los conjuntos de BLC superan los resultados obtenidos con los BC de MEANING o BALKANET. Los mejores resultados se obtienen utilizando umbrales más altos. Sin embargo, ahora la parte verbal de los BLC obtiene resultados ligeramente peores que en el caso de usar el criterio del número de relaciones. También podemos observar que, en general, utilizar las frecuencias calculadas sobre SemCor obtiene resultados peores que utilizar las frecuencias contenidas en WordNet.

En general, los resultados refuerzan nuestras observaciones e hipótesis iniciales. Nuestro método para la selección automática de conceptos ha demostrado ser un método robusto y extrae conceptos con un nivel de abstracción intermedia, que son apropiados para su uso en el aprendizaje de clasificadores basados en clases semánticas.

4.2.2 Análisis de resultados

Podemos evaluar nuestros resultados, aunque de forma indirecta, comparando con los resultados obtenidos por los participantes en la tarea *All Words* de SE3. En esta competición, el mejor sistema obtuvo un acierto del 65,1 % , mientras que la heurística basada en el primer sentido de WordNet alcanzó 62,4 %[11]. Además hay que mencionar que en esta competición, sólo 4 sistemas de los 26 participantes superaron los resultados de la heurística basada en el sentido más frecuente, o primer sentido según WordNet. Como hemos comentado anteriormente, este resultado base es muy competitivo en tareas de desambiguación, y muy difícil de superar (Gale *et al.*, 1992a).

En la tabla 4.7 se muestran los resultados para nombres y verbos, tanto monosémicos como polisémicos, de las diferentes agrupaciones de sentidos obtenidas usando los criterios de todos los tipos de relaciones (*All*) y frecuencias sobre WordNet (*FreqWN*). De nuevo la clasificación consiste en asignar a cada palabra del corpus de evaluación SE3 su clase o sentido más frecuente sobre el corpus de entre-

[11] Este resultado puede variar en función del tratamiento de las multipalabras y las palabras unidas mediante guiones

namiento de SemCor. Los mejores resultados aparecen resaltados en negrita.

	Relaciones (*All*)			**Frecuencia *FreqWN***		
Clase	**Nombres**	**Verbos**	**N&V**	**Nombres**	**Verbos**	**N&V**
Sentidos	71,79	52,89	63,24	71,79	52,89	63,24
Balkanet	73,06	53,82	64,37	73,06	53,82	64,37
Meaning	73,40	56,40	65,71	73,40	56,40	65,71
BLC–0	74,80	58,32	67,35	72,99	55,33	65,01
BLC–10	74,99	58,46	67,52	74,60	57,08	66,69
BLC–20	76,12	58,60	68,20	75,62	57,22	67,31
BLC–30	75,99	58,60	68,14	76,10	57,63	67,76
BLC–40	75,76	59,70	68,51	**78,03**	58,18	69,07
BLC–50	**76,22**	**59,83**	**68,82**	**78,03**	**58,87**	**69,38**
SuperSense	**81,87**	**79,23**	**80,68**	**81,87**	**79,23**	**80,68**

Tabla 4.7. Medida F1 para nombres y verbos de todas las palabras utilizando todas las relaciones y las frecuencias calculadas sobre WordNet para generar los BLC

Obviamente se obtienen resultados más altos cuando se incluyen las palabras monosémicas, puesto que estas son siempre clasificadas correctamente, pero es la evaluación estándar de SE, y por eso la incluimos. Cabe resaltar que este sencillo sistema que proponemos obtiene un acierto de 63,24 %, muy similar al mejor resultado obtenido en SensEval–3, y que *SuperSenses* obtiene un resultado muy alto, con un valor de F1 igual a 80,68 %. En cuanto a nuestros conjuntos BLC, los mejores resultados son obtenidos usando un umbral $\lambda = 50$, además en todos los casos se superan los resultados obtenidos con los BC de BALKANET y MEANING. Sin embargo, para nombres y utilizando todas las relaciones con umbral $\lambda = 20$ (BLC–20 con 558 clases), se obtiene un acierto ligeramente menor que utilizando umbral $\lambda = 50$ (BLC–50 con 253 clases).

Esto demuestra de nuevo que el conjunto BLC–20 ofrece un nivel de abstracción intermedio sin la penalización de empeorar excesivamente los resultados. Cuando se usa la frecuencia de las palabras para generar los conceptos, los resultados obtenidos utilizando dichos conceptos son elevados, pero no superan en todos los casos a los obtenidos utilizando los conceptos de MEANING o BALKANET, aunque a partir de BLC–20 si sobrepasan los resultados de los BC de dichos proyectos.

Sorprendentemente, los clasificadores construidos con esta técnica tan simple son capaces de obtener niveles muy altos de acierto. Para nombres, utilizando BLC–20 (considerando todas las relaciones, con 558 clases semánticas) el sistema alcanza un acierto de 76,12 %, mientras que usando BLC–50 (seleccionados mediante frecuencias de WordNet, 132 clases semánticas), el sistema alcanza un 78,03 %. Finalmente, utilizando *SuperSenses* para verbos (15 clases semánticas) el sistema obtiene un acierto del 79,23 %.

Como hemos comentado en la sección 2.4 (Aproximaciones basadas en clases), nuestros datos pueden compararse con los de (Ciaramita & Altun, 2006) puesto que usan una aproximación similar con los mismos conjuntos de datos de entrenamiento y test. Su sistema implementa un etiquetador secuencial, basado en cadenas ocultas de Markov y un perceptrón. Utilizan SuperSenses como repositorio de clases semánticas, SemCor como corpus de entrenamiento y el corpus de SE3 para evaluación. Consiguen un valor de F1 igual a 70,74, obteniendo una sensible mejora con respecto a su *baseline*, que consigue un 64,09. En estas mismas condiciones nuestro sistema alcanza un valor F1 de 80,69. En su caso, el *baseline* implementa la heurística de asignar el SuperSense correspondiente al synset más frecuente para la palabra, de acuerdo a la ordenación de frecuencias de WordNet. Posiblemente el origen de las discrepancias entre su *baseline* y los nuestros, puesto que ambos métodos utilizan la misma heurística, se debe a dos razones fundamentalmente: en primer lugar, ellos utilizan un esquema de etiquetado secuencial, donde marcan el inicio y fin de cada entidad, y en segundo lugar, debido al uso de la parte brown–v de SemCor para el calculo de las frecuencias de sentidos, parte que nosotros no utilizamos.

4.2.3 Conclusiones

La tarea de desambiguación supervisada parece haber alcanzado su límite superior utilizando las aproximaciones tradicionales basadas en palabras. Algunas de las limitaciones de los trabajos existentes podrían estar relacionadas con el uso de WordNet, recurso mayoritariamente utilizado en todos los sistemas de desambiguación. Aparentemente, WordNet presenta un repositorio de sentidos con una

granularidad muy detallada, con diferencias demasiado pequeñas entre sentidos de una misma palabra como para poder ser capturadas por un sistema automáticamente.

También se debe lidiar con el problema de la falta de suficiente información etiquetada a nivel de sentidos para poder afrontar tareas de aprendizaje automático con garantías. Cambiar el conjunto de clases semánticas puede ser una solución a este problema, enriqueciendo el número de ejemplos disponibles para cada clasificador. De hecho, nuestro sistema utilizando una técnica simple basada en asignar la clase más frecuente, obtiene una tasa de acierto muy elevada. Por ejemplo, usando BLC–20 para nombres y SuperSense para verbos, la tasa de acierto sobre SE3 sería de un 75 %.

4.3 Evaluación de los BLC aplicados: "*Robust–WSD*", CLEF–09

Además de la evaluación directa que hemos mostrado en la sección anterior, a continuación vamos a presentar otra aplicación donde el uso de BLC se ha mostrado de utilidad. En esta sección mostramos su aplicación en el marco de una tarea de Recuperación de Información, propuesta dentro de la conferencia internacional CLEF[12] (*Cross Lingual Evaluation Forum*). Esta conferencia promueve la investigación y desarrollo de técnicas de acceso y recuperación de información multilingüe, estableciendo una plataforma común de evaluación para sistemas de este tipo que trabajen en lenguas Europeas, tanto en contextos monolingües como multilingües. Además se generan una serie de recursos y corpus para las diferentes tareas, que son de gran utilidad para toda la comunidad científica. La conferencia se celebra cada año y se proponen una serie de tareas específicas entre las que en 2009 se encontraba la tarea de recuperación de información *Robust–WSD*.

Esta tarea tenía como objetivo estudiar la contribución de la información semántica de sentidos de las palabras sobre la tarea de recuperación de información. Los organizadores de la tarea proporcionaron un conjunto de consultas y el corpus de documentos anotados

[12] `http://clef-campaign.org`

previamente con sus sentidos adecuados, por medio de dos sistemas automáticos de WSD, con muy buen funcionamiento entre los sistemas de desambiguación actuales, el sistema *UBC–ALM* descrito en (Agirre & de Lacalle, 2007), y el sistema *NUS–PT* presentado en (Chan *et al.*, 2007), ambos participantes en *SemEval–1* (SEM1). Por tanto, la tarea no consistió en desarrollar un sistema de WSD que funcionase con buenos resultados, sino en estudiar el modo de aprovechar e integrar la información semántica de sentidos en un sistema de recuperación de información.

4.3.1 Integración de clases semánticas

En este caso nuestra aproximación (Fernández *et al.*, 2009) se centró en hacer uso de nuevo de las clases semánticas y explotar sus ventajas: reducción de polisemia, agrupación coherente de sentidos y nivel de abstracción y representación más apropiado que el nivel de sentidos. La idea fue representar tanto las preguntas como los documentos por medio de las clases semánticas de las palabras contenidas en ellos, y definir una medida de similitud semántica que nos proporcionara la semejanza entre dos objetos (en nuestro caso pregunta y documento) en términos de sus clases semánticas. En cierto modo, se trataba de extraer los tópicos o temas principales más representativos de preguntas y documentos. Esto no podría hacerse tratando con sentidos, ya que un documento y una pregunta pueden hablar del mismo tema utilizando palabras totalmente diferentes, mientras que en el mismo caso, las clases semánticas de las palabras de documento y pregunta serían muy similares. Esta medida de similitud entre pregunta y documento se empleó para reordenar la lista final de documentos que el sistema de recuperación generaba: obtuvimos la similitud semántica entre la pregunta y cada uno de los documentos recuperados inicialmente, y se reordenó la lista final en función de dicha similitud.

Para estos experimentos utilizamos como conjuntos de clases semánticas, WND y BLC–20. El motivo fue que ambos conjuntos poseen diferente nivel de abstracción, y distinto modo de generación, por lo que podrían proporcionar resultados diferentes e interesantes. El modo de representar un documento o una pregunta, un objeto

de texto en general, fue mediante las clases semánticas de las palabras contenidas en dicho objeto. Tal y como hemos comentado anteriormente, tanto las preguntas como los documentos proporcionados para la tarea estaban anotados con la información de sentidos. Cada palabra poseía un *ranking* ordenado por probabilidad de todos sus sentidos (usando WordNet). Este *ranking* se obtuvo de forma automática mediante los dos sistemas de desambiguación que hemos comentado.

Nuestro primer paso fue establecer una correspondencia enter cada uno de estos sentidos y su correspondiente clase semántica, manteniendo el mismo valor de probabilidad que tenía el sentido concreto. A partir de la información de clases semánticas para un objeto (pregunta o documento), creamos un vector semántico. Dicho vector contenía tantos elementos como número de clases semánticas tenía el conjunto que utilizáramos en ese caso (165 clases para nombres en WND y 558 clases para nombres en BLC–20). Cada uno de los elementos del vector representaba el valor de asociación del objeto en cuestión con la clase semántica correspondiente a dicho elemento. Ya que las anotaciones semánticas de cada una de las palabras fueron realizadas automáticamente y no podíamos estar seguros de que fueran totalmente correctas, decidimos utilizar todas las clases semánticas de cada palabra, y no quedarnos únicamente con la que obtenía mayor resultado. Cada palabra contribuyó en la generación del vector semántico con todas sus clases semánticas, en cada caso con el valor de probabilidad correspondiente. El proceso se puede resumir en:


```
Entrada: un objeto textual O
  Para cada palabra P contenida en O
    Para cada clase C de P
      vector[C] := vector[C] + probabilidad(C,P)
    FinPara
  FinPara
Salida: Vector de clases
```

Una vez finalizado el proceso anterior, disponíamos de un vector semántico que representaba los tópicos o temáticas del objeto tratado. La forma de obtener la similitud semántica entre dos objetos a partir de sus vector semánticos (ya fuera entre pregunta y documento, o entre documento y documento, o entre pregunta y pregunta) fue utilizando la medida del coseno del ángulo formado entre los dos vectores. Mediante esta medida obtuvimos un valor de asociación semántica entre la pregunta y cada uno de los documentos que el sistema recuperaba inicialmente. Otro valor del que disponíamos era el que obtenía directamente el sistema de recuperación de información entre la pregunta y un conjunto de documentos (con esta información se generaba el primer *ranking* de documentos). Diseñamos diferentes fórmulas que combinaban ambos valores, el del sistema de recuperación de información y la similitud semántica, para obtener un valor final de asociación pregunta–documento, y así poder generar la lista final de documentos para la consulta tratada.

En general el proceso es el siguiente. La consulta del usuario es procesada para obtener un conjunto de términos sin palabras de parada (*stopwords*) ni símbolos especiales. Utilizando el motor *Lucene*[13] se obtiene una primera lista de documentos relevantes. A partir de esta lista, utilizando métodos tradicionales de expansión de la pregunta y las relaciones de Wordnet, se expande la consulta inicial del usuario. Cada término de la nueva consulta posee un peso específico en función de su peso original y la expansión realizada. Se utiliza de nuevo *Lucene* para obtener un conjunto de documentos relevantes, los cuales son reordenados del modo que hemos descrito para generar la lista final de documentos. Para más detalles sobre las fórmulas utilizadas y la configuración del sistema se puede consultar (Fernández *et al.*, 2009).

4.3.2 Resultados

Se realizaron diferentes experimentos, utilizando distintas configuraciones del sistema de búsqueda Lucene[14], varias estrategias de expansión de los términos de la pregunta y diversas fórmulas pa-

[13] `http://lucene.apache.org`, consultado en julio de 2010.
[14] `http://lucene.apache.org`

ra combinar el valor devuelto por el sistema Lucene y el valor de similitud semántica y reordenar la lista final de documentos. Se utilizaron las medidas clásicas de evaluación para sistemas de recuperación de información: *MAP* (media aritmética de la precisión sobre cada pregunta), *GMAP* (media geométrica de la precisión para cada pregunta), *R–Prec* (Precisión tras haber recuperado un número de documentos igual al número de documentos relevantes), *P@5* y *P@10* (precisión tras haber recuperado 5 y 10 documentos). A continuación mostramos tres resultados de nuestro sistema: un resultado que actuó como sistema base, el cual consistía en utilizar el sistema de recuperación de información basado en el modelo *BM25* (Robertson & Jones, 1976) y la técnica de expansión de la pregunta *Bo1*, y dos resultados más añadiendo a este sistema base el módulo de reordenación de documentos utilizando WND y utilizando BLC–20 como repositorio de clases semánticas para obtener la similitud. En la tabla 4.8 podemos ver los resultados que comentamos.

	MAP	**GMAP**	**R–Prec**	**P@5**	**P@10**
BM25 + Bo1 (baseline)	0,3737	0,1294	0,3585	**0,4475**	0,3825
BM25 + Bo1 + WND	0,3752	0,1298	**0,3638**	0,4462	**0,3862**
BM25 + Bo1 + BLC–20	**0,3776**	**0,1317**	0,3609	0,4437	0,3806

Tabla 4.8. Resultados del sistema con reordenación semántica y sin ella

Como podemos ver, utilizando la reordenación basada en clases semánticas mejoramos ligeramente los resultados con la incorporación de la reordenación según similitud semántica. Además, en general se obtuvieron mejores resultados en el caso de BLC–20 que usando WND, lo que pone de relieve de nuevo la robusta y adecuada agrupación de sentidos que hace el conjunto BLC–20, así como el buen nivel de abstracción que ofrece dicho conjunto. Con WND mejoramos la medida *MAP* del baseline en un 0,4 % y un 0,31 % en *GMAP*. Con BLC–20 se obtuvo una mejora respecto al baseline de 0,64 % según la medida *MAP* y un 1,77 % según *GMAP*. Aunque la mejora fue muy ligera, nos indicaba que íbamos en una buena dirección, y más teniendo en cuenta lo simple de nuestra aproximación. Posiblemente si dispusiéramos de la información correcta de cual es el sentido de cada palabra, la representación semántica mediante el

vector de clases sería más precisa y el resultado final podría verse incrementado. Posiblemente algunos documentos relevantes quedaron demasiado alejados de las primeras posiciones del *ranking* como para que el módulo de reordenación semántica lo volviera a situar en las primeras posiciones.

4.4 Nuestros BLC en el proyecto Kyoto

Kyoto es un proyecto relacionado con Semántica y Contenido Inteligente, y con Librerías Digitales, perteneciente al 7° Programa Marco de la Unión Europea. El proyecto comenzó en marzo de 2008, y tiene una duración de 3 años. Además, participan 9 grupos de investigación pertenecientes a 7 países distintos, y dos compañías más. El objetivo del proyecto es establecer una plataforma que permita a diferentes comunidades ser capaces de compartir términos, palabras y conceptos, conocimiento al fin y al cabo, sin depender de las distintas lenguas o culturas particulares de cada comunidad. Además todo este conocimiento, no solo se almacenará y representará de un modo que sea útil para los humanos, sino que también un computador será capaz de entenderlo y realizar inferencias sobre él. El punto de conexión con nuestro trabajo es precisamente que han hecho uso de las mismas clases semánticas BLC que nosotros usamos, para diseñar parte de la ontología de interconexión del conocimiento.

El proyecto se ha centrado en el dominio medioambiental, y en temas de actualidad relacionados, como el cambio climático o el problema del calentamiento global. Para manejar toda la información lingüística han dispuesto una arquitectura de 3 niveles. En el nivel más bajo se encuentra la capa perteneciente al vocabulario, que incluye términos y palabras relacionadas con el dominio. En la segunda capa se integra un conjunto de WordNets, generales para las diferentes lenguas y específicos para el ámbito que tratan. Por encima se encuentra la tercera capa, la ontología, que contiene conceptos generales y subontologías específicas. Además se disponen de los mappings y relaciones necesarias para conectar los términos y conceptos de las diferentes capas.

En cuanto a la ontología general de Kyoto, se ha dividido también en tres niveles. El nivel más alto se basa en la ontología DOLCE y OntoWordNet, y contiene conceptos muy generales. El segundo nivel consiste precisamente en los conceptos BLC. Se trata de conceptos de este tipo, obtenidos sobre diferentes WordNets en distintas lenguas. El nivel más bajo de la ontología contiene conceptos y relaciones específicas del dominio ambiental.

De este modo, los BLC se usan a modo de conexión entre los WordNets locales para cada idioma, y los WordNets específicos, hacia las clases generales de la ontología, proporcionando una cobertura completa de los conceptos específicos, asegurando de este modo que para todos los términos y conceptos específicos de los niveles bajos existe un concepto de la ontología de mayor nivel con el que se puedan conectar. Hay que tener en cuenta, que todos los synsets de WordNet están relacionados con su BLC, y también estos BLC están enlazados con las clases generales de la ontología Kyoto. Por tanto, es sencillo pasar del conocimiento contenido en términos y palabras de bajo nivel, fuertemente ligadas al dominio específico, a conocimiento de alto nivel, independiente del dominio y de la lengua, a través de diferentes capas de abstracción, una de ellas creada haciendo uso de los conceptos BLC. Por ejemplo, podemos ver en la siguiente figura el uso del BLC *animal.n#1* en la capa intermedia, para relacionar el término *Animalia* con el concepto general *organism*. Además, otros términos del dominio como *Anura*, *Amphibia* o *Chordata* son enlazados con la clase *organism* a través del mismo BLC *animal.n#1* y siguiendo las relaciones entre synsets de WordNet.

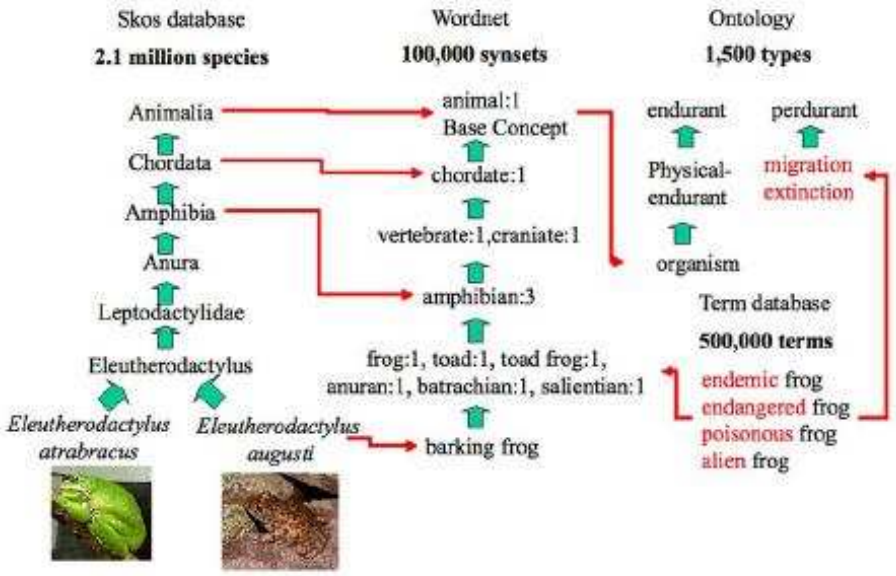


Figura 4.2. Ejemplo de uso de los BLC en el proyecto Kyoto

5. Un Sistema de WSD Basado en Clases Semánticas

En el capítulo anterior mostramos ya algunos resultados de un sistema de etiquetado semántico que usaba clases semánticas (nuestros BLC) en lugar de sentidos de las palabras. Sin embargo, ese sistema utilizaba una heurística muy sencilla basada en seleccionar la clase más frecuente para cada palabra, y sólo lo usamos para obtener una primera aproximación de hasta dónde podrían llegar en tareas de WSD dichas clases semánticas y evaluar la calidad de los conjuntos creados.

Ahora en este capítulo describiremos el proceso de desarrollo de un sistema de etiquetado semántico basado en aprendizaje automático y orientado a clases semánticas en lugar de los tradicionales sentidos de las palabras. Explicaremos tanto el proceso de desarrollo como los experimentos que hemos ido realizando para refinar y ajustar nuestro sistema de etiquetado basado en clases semánticas.

Dedicaremos la sección 5.1 a detallar específicamente el diseño de la arquitectura basada en clases semánticas y compararla con otras basadas en sentidos de palabras. También haremos énfasis en la forma en que se distribuyen los ejemplos en las diferentes aproximaciones, y qué ventajas supone en nuestro caso el uso de clases semánticas.

Otro punto determinante de todo sistema de aprendizaje automático basado en ejemplos es la forma de representar estos ejemplos mediante atributos. Explicamos también, en la sección 5.2, las diferentes pruebas que hemos realizado con diferentes tipos de atributos para seleccionar aquel conjunto que mejor se adapta a nuestra aproximación basada en clases. En estos experimentos partimos de un conjunto de atributos tradicionalmente utilizados en tareas de WSD, y realizamos diferentes pruebas para ver cuáles y de qué manera se adaptan mejor a nuestros clasificadores semánticos. Para compro-

bar la calidad de estos atributos llevamos a cabo en la sección 5.3 unos experimentos con pruebas de validación cruzada sobre el corpus SemCor.

En la sección 5.4 presentamos la participación en la competición internacional *SemEval–1* (SEM1) con una versión muy inicial de nuestro sistema, en la que se combina una aproximación basada en sentidos con una basada en clases semánticas. La sección 5.5 muestra la participación en la competición internacional *SemEval–2* (SEM2). En este caso utilizamos la configuración estándar de nuestro sistema de WSD. El aporte es el uso que hacemos de nuestras clases semánticas BLC para obtener ejemplos monosémicos de entrenamiento desde un conjunto de textos no etiquetados de un dominio específico. Esta era la novedad de la tarea en esta ocasión. Los corpus de evaluación pertenecían a un dominio concreto, al dominio medioambiental. De este modo se quería analizar como un sistema entrenado sobre un corpus de propósito general se comportaba sobre un contexto específico, y cómo se podía ajustar este sistema para adaptarlo al nuevo dominio.

Finalmente mostramos una aplicación de nuestro sistema de WSD basado en clases semánticas. Se trata de una tarea no muy difundida en el campo de PLN: la detección y extracción de palabras clave de un texto. Estas palabras clave se pueden utilizar a modo de resumen muy breve sobre el texto, como detector del tópico del texto, o como palabras clave para indexar el texto en un sistema de recuperación de información. En concreto utilizaremos nuestras clases semánticas BLC–20 a modo de palabras clave. Estas clases BLC–20 serán asignadas a las palabras de un texto mediante nuestro sistema de WSD.

5.1 Arquitectura General del Sistema

A la hora de desarrollar nuestro sistema, hemos pensado en la mejor arquitectura para explotar las ventajas que, *a priori*, le suponíamos al uso de clases semánticas en el marco de un sistema de desambiguación semántica. Por recordar, como hemos comentado en varias ocasiones, las tres ventajas más importantes que suponen el uso de clase semánticas son:

- Aumento de ejemplos de entrenamiento por clasificador
- Reducción de polisemia
- Nivel de abstracción probablemente más apropiado para su uso en aplicaciones de PLN

Por tanto, para explotar estas ventajas, la aproximación y arquitectura del sistema no puede ser la misma que la usada tradicionalmente en los sistemas de desambiguación basados en el sentido de las palabras. Además, la elección de uno u otro método de aprendizaje determinará la arquitectura del sistema, debido a que definirá la forma de codificar los ejemplos, y por tanto, la forma de tratarlos y de organizarlos para construir los clasificadores.

En nuestro caso, el paquete software que seleccionamos para realizar el aprendizaje es *SVMLight*[1]. Este *software* implementa un motor SVM con clasificación binaria, es decir, un clasificador concreto asociado a una clase C, sería capaz de discriminar si un ejemplo desconocido pertenece a esa clase C o no. Para entrenar ese clasificador se usan ejemplos positivos y negativos de la clase C . Este es otro punto importante de implementación, decidir qué ejemplos considerar como negativos para una clase C y cuáles no. Por otra parte, la forma de codificar los ejemplos para *SVMLight* es por medio de números enteros: cada atributo, aunque sea una cadena de caracteres, debe ser codificado mediante un valor entero. Estos requerimientos hacen que el primer paso para entrenar un clasificador sea obtener un diccionario o conjunto completo de atributos a partir de los ejemplos utilizados para dicho clasificador, para luego poder asignarle a cada atributo un valor entero distinto. Cada uno de estos atributos posee un peso asociado que permite ajustar la relevancia que le asignamos, y que por defecto es siempre 1.

Tradicionalmente, los sistemas de WSD se han **centrado en palabras**, es decir, se genera un clasificador para cada palabra o sentido de la palabra. El clasificador creado para una palabra W, se encarga de asignarle el sentido más apropiado a un nueva ocurrencia de esa palabra W, en un cierto contexto. Los ejemplos que se utilizan para construir el clasificador asociado a la palabra W son todas las ocurrencias que aparezcan en el corpus de entrenamiento de W con sus

[1] `http://svmlight.joachims.org`, consultado en julio de 2010

diferentes sentidos. El hecho de usar un paquete basado en clasificación binaria hace que cada clasificador, CL_w, para cada palabra W, se componga a su vez de varios subclasificadores $CL_{w_{s_i}}$, uno por cada sentido s_i de la palabra w. La salida del clasificador CL_w, por tanto, se construye a partir de las salidas individuales de los clasificadores $CL_{w_{s_i}}$. Cada uno de estos clasificadores $CL_{w_{s_i}}$ nos proporciona una estimación de que el sentido correcto para el ejemplo de w sea el i. Finalmente se debe emplear alguna estrategia para combinar las salidas individuales y generar una salida global. Una estrategia sencilla de combinación es quedarse con el sentido con un valor de evidencia mayor según SVM, como podemos ver en la fórmula 5.1.

$$CL_w = \arg\max_i CL_{w_{s_i}} \tag{5.1}$$

En estas aproximaciones basadas en sentidos, para un clasificador concreto $CL_{w_{s_i}}$ asociado al sentido s_i de la palabra w, se utiliza como ejemplos positivos todos aquellos del corpus de entrenamiento pertenecientes a la palabra w anotados con el sentido s_i. Como ejemplos negativos se usarían el resto de ocurrencias de la palabra w anotadas con otro sentido diferente al s_i. En la figura 5.1 podemos ver un esquema de la distribución de ejemplos en la aproximación basada en sentidos. Suponiendo que la palabra *casa* posee 3 sentidos, se generarían tres clasificadores, uno para cada sentido. También podemos observar cómo se distribuyen los ejemplos de entrenamiento (en la parte inferior) en ejemplos positivos y negativos para cada uno de los clasificadores individuales. Finalmente los clasificadores individuales se combinan para dar lugar al clasificador general para la palabra *casa*.

Nuestra aproximación **centrada en clases semánticas**, supone cambiar el foco de atención del sentido de la palabra a la clase semántica. En lugar de generar clasificadores para cada sentido de cada palabra, se genera un clasificador para cada clase semántica del conjunto que estemos utilizando. Para etiquetar una nueva palabra, se obtendrían las posibles clases semánticas de dicha palabra, C_i, y posteriormente se obtendría la salida de cada clasificador asociado a cada clase C_i. De nuevo la misma aproximación sencilla sería elegir la clase asociada al clasificador que más certeza de pertenencia indicara.

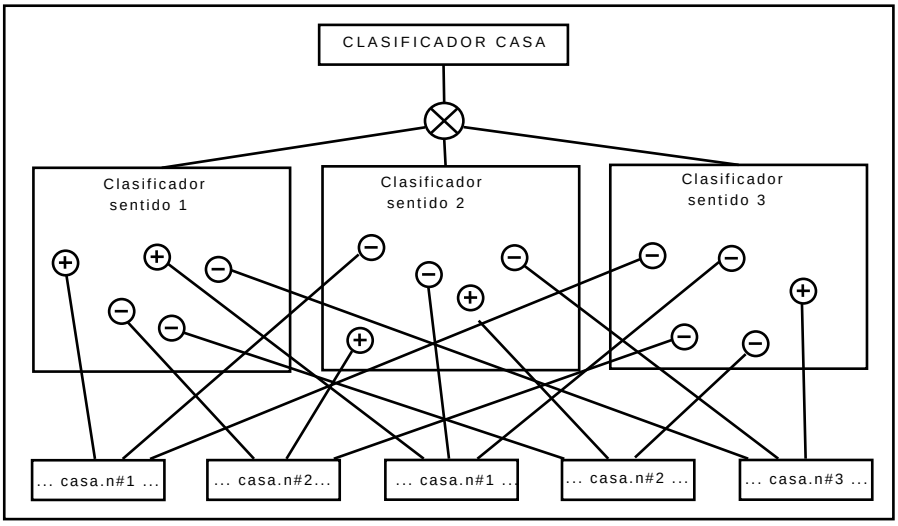


Figura 5.1. Distribución de ejemplos por sentidos

Como hemos comentado anteriormente, una de las ventajas del uso de clases semánticas es la *reducción de polisemia*, lo que hace más sencilla la tarea de desambiguación. Esta reducción de polisemia se produce directamente por la agrupación de sentidos que supone el uso de dichas clases semánticas. Por otra parte, el *incremento de la cantidad de ejemplos de entrenamiento* lo explotamos a través del modo en que se distribuyen los ejemplos para cada uno de los clasificadores. En la aproximación basada en clases semánticas, cada clasificador puede usar como ejemplos positivos a todos aquellos etiquetados con la clase correspondiente al clasificador. Hay que tener en cuenta que, para un mismo clasificador se pueden utilizar ejemplos de palabras totalmente diferentes, siempre que estén englobadas dentro de la misma clase semántica. Como ejemplos negativos para un clasificador, utilizamos el resto de ejemplos de palabras anotadas con una clase semántica diferente. Podemos ver el incremento en la cantidad de ejemplos positivos de entrenamiento por clasificador en la tabla 5.1.

En cuanto a la forma en que se distribuyen los ejemplos en la aproximación basada en clases, podemos ver un esquema en la figura 5.2. Así pues, el clasificador para la clase EDIFICIO utilizará a

Clasificador	Ejemplos	Num. ejemplos
church.n#2 *(Aprox. sentidos)*	church.n#2	58
building, edifice *(Aprox. clases)*	church.n#2	58
	building.n#1	48
	hotel.n#1	39
	hospital.n#1	20
	barn.n#1	17
	.....	.....
		TOTAL= 371 ejemplos

Tabla 5.1. Ejemplos y su frecuencia en el corpus SemCor, para la aproximación basada en sentidos y en clases

todas las palabras que estén anotadas con la clase semántica EDIFICIO, como por ejemplo una ocurrencia de la palabra *casa* y otra de la palabra *hospital*. Como negativos utiliza al resto de ejemplos de palabras asociadas a otras clases semánticas distintas.

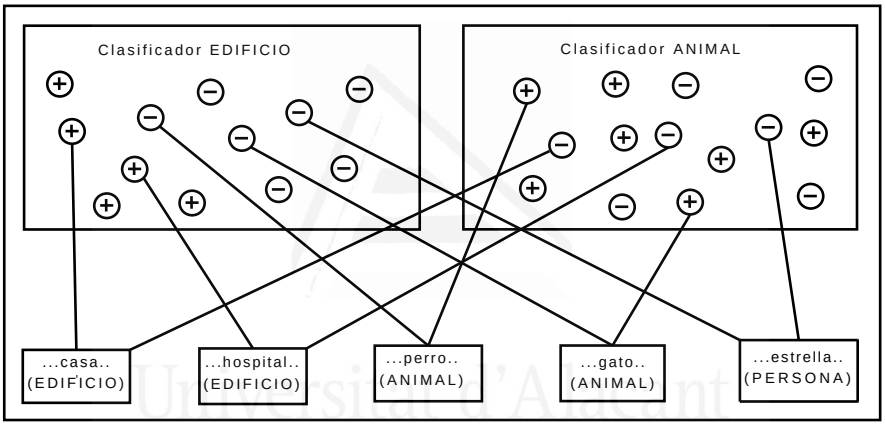


Figura 5.2. Distribución de ejemplos por clases

Hemos visto la forma de distribuir los ejemplos positivos y negativos. De este modo, cada clase posee dos conjuntos de ejemplos: el de ejemplos positivos y el de ejemplos negativos. De cada uno de estos conjuntos se extrae una serie de atributos, que posteriormente se utilizará para codificar los ejemplos de cada grupo (los ejemplos po-

sitivos se codifican mediante los atributos extraídos de ellos mismos, y lo mismo en el caso de los negativos).

Finalmente, el procedimiento para etiquetar una nueva palabra en la aproximación basada en clases consiste en los siguientes pasos:

1. Se obtienen las posibles clases semánticas para la palabra.
2. Para cada clase semántica, se obtiene el valor de pertenencia de la palabra a dicha clase, haciendo uso de los clasificadores apropiados.
3. Se asigna como etiqueta aquella clase semántica que haya obtenido mayor evidencia de pertenencia.

5.2 Estudio de Atributos y Características

Una parte muy importante de un sistema de aprendizaje automático basado en ejemplos es la forma de mostrar dichos ejemplos al módulo de aprendizaje, es decir, su representación. Esto se realiza tradicionalmente mediante atributos. Se define una serie de tipos de atributo, como por ejemplo las palabras o los lemas del contexto, y se determina un tamaño de contexto alrededor de la palabra objetivo que queremos representar. Posteriormente para representar esta palabra objetivo, se extraen atributos concretos para estos tipos definidos, dentro del contexto determinado, y se representa directamente a este ejemplo, normalmente mediante un vector de características. Esto quedará más claro mediante un ejemplo. Tomemos la frase siguiente correspondiente al artículo *wsj_0123.mrg* de Wall Street Journal, en la que cada palabra aparece junto a su etiqueta morfológica asociada.

$$They_{PRP}\ raised_{VBD}\ the_{DT}\ electrical_{JJ}\ current{-}carrying_{JJ}\ \textbf{capacity}_{NN}\ of_{IN}\ new_{JJ}\ superconductors_{NN}\ crystals_{NN}\ by_{IN}\ a_{DT}\ factor_{NN}.$$

Por simplicidad, suponemos un contexto de 3 palabras alrededor de la palabra objetivo, que en nuestro caso será *"capacity"*. Como tipos de atributos vamos a usar, las palabras en el contexto (*W* para codificación), y bigramas, trigramas que contengan a la palabra objetivo, tanto de palabras (*WB* y *WT*) como de etiquetas morfológicas (*PB* y *PT*). En nuestra aproximación utilizamos la posición relativa

del atributo respecto a la palabra objetivo así como el tipo de atributo para codificar cada atributo de forma única. Por tanto en nuestro sencillo ejemplo obtendríamos este vector de características:

(the-W-3, electrical-W-2, current-carrying-W-1, capacity-W0, of-W1, new-W2, superconductors-W3, current-carrying-capacity-WB-1, capacity-of-WB1, JJ-NN-PB-1, NN-IN-PB1, current-carrying-capacity-of-WT-1, JJ-NN-IN-WT-1)

El primer paso para desarrollar nuestro sistema de desambiguación, tras la definición de la arquitectura del sistema, es analizar diferentes tipos de atributos para ver cuáles se adaptan mejor a las necesidades de un sistema de desambiguación semántica. En esta sección describiremos el proceso que hemos seguido en el análisis y experimentación con diferentes conjuntos de atributos, pero sin mostrar los valores concretos para cada experimento, debido a que realizamos alrededor de 100 experimentos diferentes, y la gran cantidad de datos tendería a confundir más que a aclarar el proceso. Estos valores los podemos ver en el anexo A.

Empezamos utilizando únicamente dos conjuntos de clases semánticas para los primeros experimentos: BLC–20 y BLC–50. Esto es debido a que ambos conjuntos presentan un grado de abstracción bastante distinto, y nos pueden proporcionar puntos de vista diferentes sobre el uso de unos u otros tipos de atributos. En principio, el corpus que hemos utilizado para la evaluación en todos estos casos ha sido el correspondiente a la tarea *All Words* de *SensEval–3* (SE3).

Para representar los ejemplos de entrenamiento y extraer la información necesaria para construir un modelo que sea capaz de clasificarlos, definimos un conjunto de características a partir de trabajos previos de WSD y adaptado a la naturaleza de la nueva aproximación basada en clases. Hemos partido de un conjunto de características tradicionalmente usadas y descritas en (Yarowsky, 1994). Generalmente son fragmentos de información que aparecen en el contexto de la palabra objetivo que queremos representar. Pueden ser organizadas del siguiente modo:

Atributos locales. Bigramas y trigramas que contienen a la palabra objetivo y construidos con categorías morfológicas, lemas y

formas de palabras. Tratan de capturar palabras relacionadas frecuentemente con la palabra objetivo.

Atributos de tópico. Son usualmente lemas o formas de palabras que aparecen en una cierta ventana alrededor de la palabra objetivo, tratan de capturar de algún modo la semántica relacionada con la palabra en cuestión.

Los primeros experimentos los realizamos utilizando como atributos únicamente las palabras y los lemas de dichas palabras en una ventana de 50 palabras alrededor de la palabra objetivo. Esta configuración era superada si incluíamos también las etiquetas morfológicas de los 5 *tokens* alrededor de la palabra objetivo y los bigramas de lemas y clases de palabras monosémicas en el contexto de las 50 palabras. Combinando estos atributos lanzamos varias pruebas sin obtener mejora considerable. En cambio, sí observamos que se obtenía una mejora al utilizar una ventana de 5 palabras en lugar de las 50 iniciales. En este punto todavía no utilizábamos ningún tipo de filtrado de atributos, por tanto el contexto de 50 palabras introducía demasiado ruido a través de un número demasiado elevado de atributos (de 71.000 atributos en un contexto de 50 a 5.000 en un contexto de 5 palabras). Con esta configuración de ventana 5, realizamos una serie de experimentos que demostraron que el uso de las clases de las palabras monosémicas aportaban un incremento considerable en nuestro sistema.

Teniendo en cuenta de nuevo la configuración con ventana 5, realizamos diferentes combinaciones de tipos de atributos y valores para el parámetro de regularización c del módulo SVM[2]. Estas pruebas determinaron que el mejor valor para este parámetro era 0,01 que, por otra parte, ya se había utilizado con éxito en otros trabajos aplicando SVM a desambiguación.

El siguiente paso fue probar con una diferente aproximación para codificar los atributos: la *bolsa de palabras*. En esta aproximación no se diferencia entre posiciones para los atributos alrededor de la

[2] Este parámetro proporciona una idea sobre la cantidad de ejemplos mal clasificados que vamos a permitir en el proceso de entrenamiento, o el error máximo permitido, con el fin de obtener unos clasificadores más precisos, con mayor capacidad de generalización que eviten ser excesivamente dependientes de los datos de entrenamiento

palabra objetivo, es indiferente, por ejemplo, si una palabra concreta aparece en el token siguiente a la palabra objetivo o diez tokens más adelante. Estas pruebas obtuvieron buenos resultados, incluso sólo usando los lemas como tipo de atributos. Precisamente éste pudo ser el motivo de los buenos resultados: utilizar este tipo de representación basado en bolsa de palabras decrementa en gran medida el número de atributos que se extraen, y aquellos que se seleccionan son realmente representativos.

Volviendo a la codificación normal de los atributos en la ventana de tamaño 5 (sin bolsa de palabras), añadimos un atributo más, los bigramas de palabras, al mismo tiempo que modificamos la definición de los bigramas de lemas y de etiquetas morfológicas. En concreto, las etiquetas morfológicas correspondieron a la concatenación de las 3 o 5 etiquetas de los 3 o 5 tokens siguientes o anteriores. En caso de que uno de estos tokens perteneciera a una oración diferente, la etiqueta considerada era *NULL*[3]. Si uno de los tokens era un signo de puntuación, la etiqueta de dicho token era *PUNC*. En caso de bigramas, se tuvieron en cuenta todos los tipos de tokens para construir dichos bigramas, y se descartaron aquellos en los que algún elemento de los que componían el bigrama perteneciera a una oración diferente a la de la palabra objetivo. También en las nuevas pruebas con estos cambios empezamos a introducir un filtro muy sencillo para los atributos, basado únicamente en la frecuencia global del atributo. Los nuevos experimentos mantuvieron los buenos resultados, utilizando una cantidad mucho menor de atributos, y reduciendo por tanto el tiempo de aprendizaje.

En este punto centramos nuestra investigación en el análisis de un filtro más preciso y eficiente, que tenía en cuenta la relación entre la frecuencia de un atributo para una clase y la frecuencia total del atributo. De este modo se trataba de seleccionar atributos que eran muy frecuentes para una cierta clase pero no lo eran para el resto, lo que indicaría una clara asociación entre el atributo y dicha clase. Este filtro será detallado más extensamente en el siguiente capítulo. También incluimos en estos experimentos, como nuevos atributos, a

[3] En las pruebas hasta este momento, no se tuvo en cuenta esta característica.

los trigramas, de palabras y de lemas. Estos nuevos atributos también mejoraron el funcionamiento del sistema.

Todos los atributos que empleamos hasta el momento estaban demasiado centrados en palabras, y no tanto en clases semánticas, aunque algunos de ellos los hubiéramos modificado ligeramente. El siguiente paso que intentamos fue generalizar los atributos para adaptarlos a clases semánticas. En concreto, los bigramas y trigramas que contuvieran la palabra objetivo, se modificaron sustituyendo dicha palabra objetivo (o su lema) por una cadena comodín, por ejemplo "X". De este modo dos bigramas diferentes como podrían ser *"de_la_casa"* y *"de_la_residencia"* podían generalizarse en un único atributo *"de_la_X"* para la clase semántica EDIFICIO.

Esta modificación se podía llevar a cabo directamente sobre los atributos originales, y usar únicamente los nuevos modificados para el aprendizaje, o incluir ambos atributos, originales y modificados, a la lista final de atributos. Esta última opción fue la que mejor resultados obtuvo y la que adoptamos por tanto. Con todo ello (uso de trigramas, y refinamiento y generalización de n–gramas) obtuvimos mejoras considerables. Esto se puede entender desde el punto de vista de que ambos tipos de atributos tratan de capturar un tipo de información diferente. Mientras que los atributos sin generalizar intentan extraer características muy específicas y concretas de una clase semántica, los atributos generalizados intentan obtener patrones generales asociados frecuentemente con una clase semántica.

Con el conjunto de atributos que hasta el momento nos produjeron mejores resultados realizamos diferentes pruebas ahora variando el tamaño del contexto, con diferentes valores para el umbral de filtrado y con diversos conjuntos de clases semánticas. No se observaron mejoras respecto a utilizar tamaños de ventana diferentes a 5, o valores para el umbral distintos a 0,25. Por tanto, continuamos con estos ajustes. Se confirmó también que esta configuración y conjunto de atributos obtenía buenos resultados para otros tipos de clases semánticas, como SuperSenses o WordNet Domains.

El último tipo de atributo que añadimos con buenos resultados fue la clase más frecuente para la palabra objetivo, según un conjunto de clases semánticas (utilizamos en principio BLC–20 y BLC–50). Incluyendo este atributo se incrementaba en varios puntos el rendi-

miento del sistema. Probamos diferentes configuraciones para este atributo: extraer también la clase más frecuente para las palabras del contexto y añadirle al atributo también el lema de la palabra objetivo. Ninguna de estas modificaciones provocó una mejora de nuestros clasificadores. Finalmente probamos otros tipos de atributos, en concreto incluimos la etiqueta semántica anterior, por si el etiquetado semántico siguiera alguna coherencia secuencial y palabras próximas tuvieran clases semánticas similares. Este atributo no aportó mejoras. También añadimos un atributo que representara la forma de la palabra, si estaba construido con mayúsculas, minúsculas, dígitos, etc. Tampoco este atributo aportó ninguna mejora.

El conjunto de atributos con el que mejores resultados obtuvimos es el que llamamos *atributos Base* y lo describimos detalladamente en el capítulo siguiente, en la sección 6.1.1.

5.3 Pruebas de Validación Cruzada sobre SemCor

También utilizamos un conjunto de atributos que ha sido empleado en otro sistema de desambiguación en el seno del grupo de investigación IXA[4]. El nombre que le hemos dado a este conjunto de atributos ha sido *atributos IXA*. La descripción completa de estos atributos la podemos ver en el capítulo siguiente, en la sección 6.1.2. Debido a que no hemos realizado una experimentación sobre este conjunto tan profunda como en el caso de los atributos *BASE*, para evaluar su calidad realizamos dos experimentos iniciales sobre el propio SemCor, por medio de validación cruzada. De este modo eliminamos la pérdida de rendimiento que puede suponer el cambio de dominio entre entrenamiento y evaluación, es decir, entrenar con un corpus y evaluar con otro diferente. Las pruebas con validación cruzada (en inglés *k–fold cross validation*) sobre un mismo corpus consisten en dividir el corpus en K conjuntos disjuntos, y realizar K evaluaciones diferentes con el mismo sistema, cada una utilizando $K-1$ particiones para entrenar y la restante para evaluar (obviamente en cada ejecución se seleccionan $K-1$ porciones diferentes). El

[4] `http://ixa.si.ehu.es`, consultado en julio de 2010

resultado final del sistema se obtiene promediando los resultados individuales de cada ejecución. En nuestros experimentos hemos dividido al corpus SemCor en 10 partes ($K = 10$).

Para realizar estos experimentos elegimos dos subconjuntos de atributos dentro del conjunto IXA: el conjunto de atributos local, y el que incluye todos los atributos (locales, contextuales y semánticos según BLC–20). En la tabla 5.2 mostramos los valores para la medida F1 (la cobertura es 100 %) para las palabras polisémicas y para todas las palabras (monosémicas y polisémicas) de cada una de los 10 ejecuciones, y del resultado promedio. El conjunto de clases semánticas elegidas para construir los clasificadores y realizar la evaluación ha sido BLC–20.

Num. Ejecución	**Atrib. Local**		**Todos atrib.**	
	Polisémicas	**Todas**	**Polisémicas**	**Todas**
1	64,58	72,47	65,04	72,83
2	61,90	71,35	61,87	71,32
3	59,44	70,27	58,81	69,81
4	69,76	77,02	69,93	77,14
5	72,62	78,62	72,88	78,82
6	65,40	74,39	65,61	74,55
7	65,64	74,35	65,46	74,21
8	63,76	71,80	64,11	72,07
9	66,20	73,04	66,32	73,14
10	70,28	76,16	70,63	76,44
PROMEDIO	**65,69**	**73,75**	**65,80**	**73,83**

Tabla 5.2. Validación cruzada sobre SemCor

Los resultados son altos en ambos casos, comparados con otros sistemas, como los participantes en SensEval, y ligeramente superiores utilizando el conjunto más rico de atributos. Podemos intuir que nuestra aproximación basada en clases obtendrá buenos resultados, es una aproximación robusta y con buen rendimiento. Además, los atributos seleccionados en el conjunto IXA, también parecen representar de forma adecuada a los ejemplos de entrenamiento.

5.4 Nuestro sistema en SemEval–1

En la evaluación internacional SemEval–1[5], que tuvo lugar en junio de 2007, se propusieron diferentes tareas para evaluación y comparación de sistemas de desambiguación del sentido de las palabras. Una de las tareas se centraba en estudiar si realmente la excesiva granularidad de los sentidos que ofrece WordNet representa un problema para los sistemas actuales de WSD. Para ello se propuso una tarea de desambiguación en la que el repositorio de sentidos que se utilizaba era de una granularidad mayor que la que ofrece WordNet tradicionalmente.

5.4.1 La tarea "*Coarse–grained English all words*"

La tarea se llamó *coarse–grained English all words task*[6], debido al uso de sentidos de granularidad gruesa por una parte, y a su similitud con las tareas de desambiguación *All Words* propuestas en competiciones SensEval anteriores. Se anotaron alrededor de 6.000 palabras de tres textos diferentes, análogos a los utilizados previamente en las competiciones SE.

El repositorio de granularidad gruesa se obtuvo agrupando sentidos de WordNet para una misma palabra, con lo que directamente se redujo la polisemia. Este repositorio se generó mediante un *mapping* entre las definiciones de WordNet y las del diccionario electrónico Oxford en inglés[7]. El procedimiento de generación del nuevo repositorio de sentidos fue semiautomático: se generó primeramente una agrupación automática mediante el algoritmo SSI (*Structural Semantic Interconnections* (Navigli & Velardi, 2005), que luego sería revisada manualmente por dos anotadores. Finalmente, un tercer revisor resolvería los casos en que hubiera discrepancias. Además del sentido *"grueso"* para cada palabra, se disponía de su lema y su etiqueta morfológica. La evaluación se realizó mediante la medida F1, y se consideró como correcto asignar a una palabra cualquiera de los sen-

[5] `http://nlp.cs.swarthmore.edu/semeval`, consultado en julio de 2010

[6] `http://nlp.cs.swarthmore.edu/semeval/tasks/task07/summary.shtml`, consultado en julio de 2010

[7] `http://www.oed.com`, consultado en julio de 2010

tidos que estuvieran contenidos dentro del sentido grueso considerado correcto.

5.4.2 Aproximación basada en clases

Aprovechando la similitud de dicha tarea con nuestra aproximación basada en clases a WSD, decidimos participar con nuestro sistema en dicha tarea. La aproximación fue muy simple, ya que nos encontrábamos en los primeros pasos de desarrollo del sistema. El método que seguimos fue usar la misma arquitectura que la utilizada con nuestras clases semánticas, pero en este caso con los sentidos gruesos en lugar de las clases semánticas. De este modo no obtendríamos solo ventaja de la reducción de polisemia, sino también del aumento de ejemplos de aprendizaje para cada clasificador. De nuevo podíamos utilizar ejemplos de todos los sentidos detallados agrupados bajo un mismo sentido grueso para entrenar el clasificador correspondiente.

Además, incluimos nuestras clases semánticas BLC–20 como atributos para el proceso de entrenamiento de los clasificadores de sentidos gruesos. Para disponer de esta información tuvimos que generar primero clasificadores a este nivel de abstracción.

Por tanto, la aproximación constó de dos fases: una primera que se apoyaba en una técnica basada en clases semánticas, haciendo uso de nuestro conjunto de clases BLC–20, y una segunda fase que se centraba en sentidos gruesos para construir los clasificadores. Con la primera fase pretendíamos generar los clasificadores necesarios para obtener las etiquetas BLC–20 asociadas a las palabras del corpus de test, y con la segunda fase obtener directamente el sentido grueso apropiado para cada palabra de test.

Ambas fases de generación de clasificadores hicieron uso de SVM, y de un conjunto de atributos simple y similar en ambos casos. Este conjunto no es ninguno de los que hemos descrito anteriormente (BASE o IXA), y consistía en un conjunto inicial o preliminar. Se consideró un contexto de tres palabras alrededor de la palabra objetivo para extraer dichos atributos (las propias palabras, sus lemas y sus etiquetas morfológicas). Incluimos también el sentido más frecuente (o primer sentido según WordNet) para la palabra objetivo. Este atributo depende del tipo de clase seleccionada para los clasificado-

res, es decir, para los clasificadores basados en BLC–20, se empleó la clase BLC–20 más frecuente para la palabra objetivo, y para los basados en sentidos "gruesos", usamos el atributo correspondiente al sentido grueso más frecuente para dicha palabra. El motivo de incluir este tipo de atributos fue asegurar que nuestro sistema asegurara alcanzar al menos al *baseline*. Además, utilizamos la clase BLC–20 asociado a la palabra objetivo automáticamente por nuestros clasificadores semánticos, como atributos para el entrenamiento de los clasificadores centrados en sentidos gruesos.

En la figura 5.4.2 podemos ver la arquitectura general del sistema de forma más clara. La fase 1 corresponde con la parte que usa nuestras clases semánticas BLC–20. Haciendo uso de SemCor como corpus de aprendizaje, y del conjunto de atributos que definimos, mediante SVM generamos el conjunto de clasificadores semánticos según BLC–20. Estos clasificadores fueron utilizados para etiquetar las instancias del corpus de test de SemEval con sus clases BLC–20 correspondientes.

La fase 2 es la orientada a sentidos gruesos. De nuevo haciendo uso de SemCor como repositorio de ejemplos de entrenamiento, y con el conjunto de atributos extendido añadiendo las etiquetas BLC–20, se generó un conjunto de clasificadores para estos sentidos gruesos. Finalmente, mediante los modelos aprendidos se etiquetaron las instancias de test del corpus de SemEval.

5.4.3 Resultados de nuestro sistema

Participaron 12 equipos en total que enviaron los resultados de 14 sistemas, más 2 sistemas que presentaron los organizadores de la tarea. Se calculó un *baseline* siguiendo la heurística de seleccionar el sentido grueso más frecuente sobre SemCor, y se obtuvo un resultado de 78,89 de medida F1. Se calculó también el resultado de un sistema que seleccionara aleatoriamente el sentido para cada palabra, alcanzando un valor de F1 igual a 52,43.

En la tabla 5.3 podemos ver el valor F1 de los sistemas participantes. Hay que resaltar que los resultados que mostramos no son los enviados realmente por todos los equipos, ya que hubo 5 participantes que no utilizaron la heurística del sentido más frecuente en

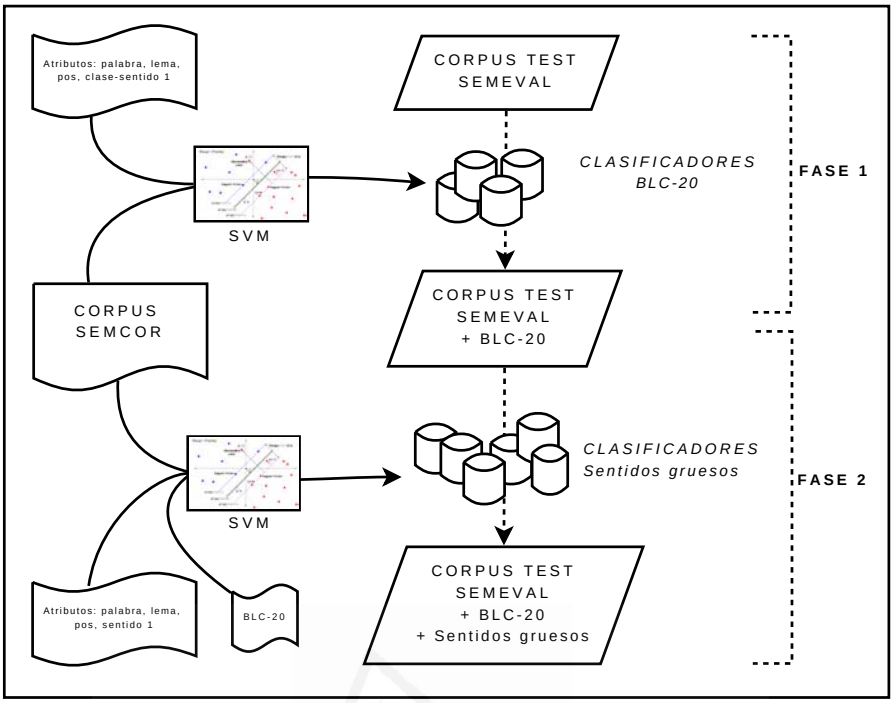


Figura 5.3. Arquitectura del sistema en SemEval 2007

caso de que su sistema no fuera capaz de obtener una respuesta. Por tanto, para realizar una comparación justa entre todos los sistemas, los organizadores añadieron esta heurística a aquellos que no la implementaban, únicamente en los casos que no devolvieron respuesta. Estos resultados son los que se muestran en la tabla 5.3.

Nuestro sistema, que aparece resaltado en negrita, ocupó la quinta posición. Es importante resaltar que superamos el *baseline*, a pesar de lo elevado que es dicho resultado (en gran parte derivado de la fuerte disminución de polisemia asociada a la agrupación de sentidos finos en sentidos más gruesos). Por otra parte, el buen resultado obtenido por nuestro sistema puso de relieve la robustez y buen rendimiento de nuestra arquitectura con 2 niveles diferentes de abstracción, ya que ni usamos atributos lingüísticos muy complejos, ni grandes recursos léxicos o semánticos. Por tanto, gran parte del buen funcionamiento de nuestra aproximación residió en la arquitectura basada en clases

Sistema	F1
UoR–SII	83,21
NUS–PT	82,50
NUS–ML	81,58
LCC–WSD	81,45
GPLSI	**79,55**
Baseline	*78,89*
UPV–WSD	78,63
SUSSX–FR	77,04
TKB–UO	70,21
PU–CBD	69,72
RACAI–SynWSD	65,71
SUSSX–C–WD	64,52
SUSSX–CR	64,35
USYD	58,79
UoFL	54,61

Tabla 5.3. Resultados de los sistemas en SemEval–1

semánticas en lugar de palabras, y en la información que aportaron los BLC–20 como atributos para el aprendizaje.

El primer sistema de la tabla (UoR-SII) corresponde con el sistema de los organizadores de la tarea. Los otros sistemas que quedaron por delante del nuestro utilizaban en todos casos mayor número de corpus para realizar el aprendizaje. En primer lugar NUS-PT hizo uso de corpus paralelos y el corpus DSO. En el caso de NUS-ML se utilizaron corpus no anotados. Finalmente LCC-WSD integraron la información de los recursos *Open Mind Word Expert* y *Extended Wordnet* en su sistema.

5.5 Nuestro sistema en SemEval–2

En julio de 2010 ha tenido lugar evaluación internacional SemEval–2[8]. El propósito de nuevo es evaluar sistemas de análisis y procesamiento semántico. Se han propuesto un total de 19 tareas, entre las cuales podemos encontrar 7 relacionadas con WSD. En concreto nuestra participación se ha centrado en la tarea número #17, que se corresponde con la tradicional tarea *all–words* aplicada sobre un dominio específico.

[8] `http://semeval2.fbk.eu/semeval2.php`

5.5.1 La tarea "*all–words WSD on a Specific Domain*"

La tarea se ha llamado *All-Words WSD on a Specific Domain* (Agirre *et al.*, 2010), y consiste en anotar con su sentido correcto de WordNet 3.0 los nombres y verbos de un corpus de test. La novedad de la tarea es que el corpus de test pertenece a un dominio específico, el dominio medioambiental. En concreto los textos seleccionados para anotar corresponden a textos del Centro experto para la biodiversidad y desarrollo sostenible (ECNC[9]) y el Fondo Mundial para la Naturaleza (WWF[10]). La tarea se ha desarrollado en el marco del proyecto europeo Kyoto[11], el cual trata de establecer una plataforma para compartir el conocimiento a diferentes niveles, hasta el nivel semántico, entre comunidades distintas, con diferentes lenguas. Además este proyecto también trabaja en el dominio medioambiental.

Es ampliamente conocido que sistemas entrenados sobre dominios generales sufren una grave penalización en sus resultados cuando se utilizan en dominios especializados. Por tanto éste es el reto que plantea esta tarea, estudiar cómo desarrollar un sistema para un dominio en concreto, o analizar la manera de adaptar un sistema de dominio general a un dominio más concreto y reducido.

5.5.2 Nuestra aproximación

Debido a la cercanía entre las fechas de envío de resultados para la tarea y los últimos retoques de este trabajo, no hemos dispuesto del tiempo que nos hubiera gustado para preparar nuestra participación. Por tanto, hemos hecho uso de nuestro sistema basado en clases semánticas, para lo cual hemos seleccionado aquellas obtenidas sobre WordNet 3.0 (debido a que es éste el repositorio seleccionado para la tarea), haciendo uso de todo tipo de relaciones en el algoritmo de selección, y un umbral mínimo de 20 conceptos (son las llamadas BLC–20). En la tabla 5.4 podemos ver las tres clases BLC–20 más frecuentes en SemCor.

Como conjunto de atributos hemos seleccionado una ventana de 5 tokens alrededor de la palabra objetivo para extraer: palabras, lemas,

[9] `http://www.ecnc.org`, consultado en julio de 2010
[10] `http://www.panda.org`, consultado en julio de 2010
[11] `http://www.kyoto-project.eu`, consultado en julio de 2010

PoS	Num.	BLC	Glosa
Nombres	4,792	person.n.01	a human being
	1,935	activity.n.01	any specific behavior
	1,846	act.n.02	something that people do or cause to happen
Verbos	1,541	change.v.01	cause to change; make different; cause a transformation
	1,085	change.v.02	undergo a change; become different in essence; losing one's or its original nature
	519	move.v.02	cause to move or shift into a new position or place, both in a concrete and in an abstract sense

Tabla 5.4. Clases BLC–20 más frecuentes en SemCor

bigramas y trigramas de palabras y lemas, trigramas de etiquetas morfológicas, y la clase más frecuente para la palabra objetivo.

Nuestro sistema en principio está orientado a un dominio general, debido a que usamos SemCor como fuente de información desde donde extraer los ejemplos de entrenamiento. Para adaptar nuestro sistema al dominio medioambiental hemos pensado en utilizar ejemplos de entrenamiento pertenecientes a este dominio para entrenar nuestro sistema. La organización ha facilitado un conjunto de textos de *background* pertenecientes a la nueva temática. El problema es que estos documentos son texto plano, y no está etiquetado semánticamente, lo cual es necesario para nuestra aproximación supervisada. La estrategia seguida en este caso ha sido hacer uso de las clases BLC–20 para extraer los ejemplos monosémicos desde dichos textos de *background*, y utilizarlos como datos de aprendizaje. En la tabla 5.5 podemos ver el número de ejemplos de entrenamiento de que disponemos, en SemCor y monosémicos en los documentos de *background*

	Nombres	Verbos	Nombres+Verbos
SemCor	87,978	48,267	136,245
Background	193,536	10,821	204,357
Total	*281,514*	*59,088*	*340,602*

Tabla 5.5. Número de ejemplos de entrenamiento

El número de ejemplos de la tabla anterior corresponde al total de ejemplos, sin tener en cuenta a la palabra objetivo a la que per-

tenecen. Centrándonos en dichas palabras objetivo, mostramos en la tabla 5.6 las palabras monosémicas más frecuentes en los ejemplos extraídos desde los documentos de *background* pertenecientes al dominio medioambiente.

	Nombres		**Verbos**	
	Lema	**Num. ejemplos**	**Lema**	**Num. ejemplos**
1	biodiversity	7,476	monitor	788
2	habitat	7,206	achieve	784
3	specie	7,067	target	484
4	climate	3,539	select	345
5	european	2,818	enable	334
6	ecosystem	2,669	seem	287
7	river	2,420	pine	281
8	grassland	2,303	evaluate	246
9	datum	2,276	explore	200
10	directive	2,197	believe	172

Tabla 5.6. Las 10 palabras más frecuentes monosémicas según BLC–20 en documentos de *background*

Por tanto, planteamos tres experimentos diferentes partiendo de la misma configuración, pero usando en cada caso un conjunto de datos de entrenamiento distinto. En concreto los experimentos son:

- modelBLC20_SC: solo utilizamos los ejemplos de entrenamiento extraídos de SemCor
- modelBLC20_BG: solo se hace uso de los ejemplos monosémicos contenidos en los datos de *background*
- modelBLC20_SCBG: usamos tanto los ejemplos extraídos de SemCor como los monosémicos obtenidos de los documentos de *background*

Con el primer experimento, *modelBLC20_SC* trataremos de ver cómo se comporta un sistema entrenado sobre un dominio general cuando se aplica a un dominio específico. Con el segundo experimento analizaremos la calidad de los ejemplos monosémicos obtenidos automáticamente desde los documentos de *background*. Finalmente con el último experimento comprobaremos si nuestro sistema de dominio general se adapta bien al dominio específico añadiendo ejemplos de entrenamiento de este último dominio.

5.5.3 Resultados

Nuestra participación se ha centrado en el idioma inglés, en el cual ha habido un total de 29 experimentos de distintos grupos de investigación. En la tabla 5.7 podemos ver la precisión y la cobertura para los distintos participantes, además del resultado de la heurística basada en el sentido más frecuente (*MFS*), y el resultado de la heurística basada en seleccionar un sentido aleatoriamente (*Random*).

	Precision	**Cobertura**
1	0,570	0,555
2	0,554	0,540
3	0,534	0,528
4	0,522	0,516
(modelBLC20_SCBG) 5	**0,513**	**0,513**
MFS	*0,505*	*0,505*
(modelBLC20_SC) 6	**0,505**	**0,505**
7	0,512	0,495
8	0,506	0,493
9	0,504	0,491
10	0,481	0,481
11	0,492	0,479
12	0,461	0,460
13	0,447	0,441
14	0,436	0,435
15	0,440	0,434
16	0,496	0,433
17	0,498	0,432
18	0,433	0,431
19	0,426	0,425
20	0,424	0,422
21	0,437	0,392
22	0,384	0,384
(modelBLC20_BG) 23	**0,380**	**0,380**
24	0,381	0,356
25	0,351	0,350
26	0,370	0,345
27	0,328	0,322
28	0,321	0,315
29	0,312	0,303
Random	*0,230*	*0,230*

Tabla 5.7. Resultados en SemEval–2

Como podemos ver, en general los resultados no han sido muy elevados en ningún caso. Incluso el *baseline* basado en el sentido más frecuente, que siempre obtiene un rendimiento alto, solo ha conse-

guido un discreto 0,505. De cualquier modo, nuestro mejor sistema obtiene una meritoria quinta posición. Además, como esperábamos, nuestro mejor experimento es aquel que utiliza SemCor y los ejemplos monosémicos extraídos del *background* para realizar el entrenamiento. Esto nos hace pensar que realmente el sistema aprovecha estos ejemplos monosémicos para especializarse, aunque ligeramente, en el dominio específico. Además, también vemos que nuestro sistema entrenado sobre SemCor, lo cual sería un sistema de dominio general, consigue unos buenos resultados, igualando al *baseline* del sentido más frecuente y alcanzando la séptima mejor posición. Por otra parte, vemos que usando únicamente los ejemplos monosémicos extraídos del *background* no es suficiente. Posiblemente se necesiten también ejemplos polisémicos para que los modelos realmente aprendan a clasificar de forma correcta.

5.6 Detección de Palabras Clave mediante Clases Semánticas

Otro modo de evaluar la calidad de nuestras clases semánticas y del sistema de desambiguación que etiqueta un texto haciendo uso de estas clases semánticas, ha sido mediante su aplicación a otra tarea de PLN, la detección de palabras clave.

La detección de palabras clave (*keyphrases* en inglés) es una tarea intermedia de PLN que consiste en obtener las palabras más representativas y relevantes de un texto (Frank *et al.*, 1999). Se trata de obtener un conjunto de palabras a partir de las cuales sea posible deducir la semántica de un documento y el dominio al que pertenece dicho texto. En realidad esta tarea se puede enfocar desde dos puntos de vista, y recibe en cada caso un nombre.

En primer lugar se puede considerar un conjunto predefinido de etiquetas o clases. En este caso la tarea consiste en asignar a un texto un conjunto de estas etiquetas. Se trata por tanto de un problema de clasificación, y está muy relacionado con otra tarea de PLN, la detección de tópico. Esta tarea suele llamarse asignación o detección de palabras clave.

En segundo lugar, otra modalidad de la tarea consiste en extraer palabras o sintagmas contenidos en el texto, sin disponer de una lista predefinida. En este caso se habla de extracción de palabras clave.

De uno u otro modo, las *keyphrases* proporcionan una descripción de alto nivel de un documento, y pueden ser usadas en muchas aplicaciones: sirven para calcular la similitud entre dos documentos; también es posible utilizarlas en otras aplicaciones de PLN como producción automática de resúmenes, recuperación de información, implicación textual, incluso detección de autoría.

La analogía de esta tarea con nuestro sistema basado en clases semánticas es obvio, y se podría usar nuestro conjunto de clases semánticas asignadas automáticamente como palabras clave. Mediante nuestros clasificadores semánticos asignaremos a cada palabra su clase semántica apropiada y, posteriormente, a partir de estas clases semánticas obtendremos aquellas más relevantes o representativas que compondrán el conjunto de *keyphrases*.

5.6.1 Evaluación

Nuestra intención es únicamente ver cómo pueden funcionar nuestras clases semánticas a modo de *keyphrases* y en ningún caso desarrollar un sistema completo de detección de palabras clave. Por tanto adaptamos la evaluación a nuestro marco de trabajo. Nuestra idea ha sido emplear los textos que componen los corpus de evaluación de SensEval y SemEval, ya que para cada texto suponemos cierta coherencia semántica. La aproximación consiste en etiquetar las palabras de cada uno de estos textos con nuestros clasificadores semánticos, obtener un *ranking* ordenado por frecuencia de las clases semánticas de todas las palabras, y comparar dicho *ranking* con el de clases semánticas correctas para las palabras del texto. Para comparar ambos *rankings* utilizaremos un test estadístico, el test de *Spearman* (Spearman, 1904). Este test devuelve un estadístico (ρ) que indica la correlación entre dos variables aleatorias continuas, que representan datos ordenados, y en nuestro caso los *rankings* de clases semánticas. El valor ρ puede oscilar entre -1 y $+1$, indicando desde correlaciones totalmente negativas (*rankings* exactamente inversos uno respecto al otro) hasta correlaciones totalmente positivas (*rankings* idénticos).

Por tanto que el *ranking* de clases según nuestro clasificador sea muy similar al de clases correctas quiere decir que se están detectando las clases semánticas más importantes y frecuentes del texto, las cuales pueden actuar a modo de *keyphrases*.

Como corpus para evaluar nuestro sistema a modo de detector de *keyphrases* mediante las clases semánticas, hemos elegido los corpus de SE2, SE3 y SEM1. Como ya hemos comentado la comparación de *rankings* la hemos hecho individualmente para cada fichero de cada corpus de evaluación. En la tabla 5.8 podemos ver una descripción de los ficheros que componen cada uno de los corpus.

Corpus	**Fichero**	**Origen**	**Descripción**
SV2	d00	Fichero 0089 del WSJ	Novela de misterio "The Nine Tailors" de D.L.Sayers
	d01	Fichero 0465 del WSJ	Artículo médico sobre genes y cancer
	d02	Fichero 1286 del WSJ	Artículo de opinión sobre educación, de I.Kristol
SV3	d000	Fichero *cl23* del Brown	Novela de misterio "Try my sample murders" de J.W.Rose
	d001	Fichero 1695 del WSJ	Artículo editorial
	d002	Fichero 1778 del WSJ	Noticia sobre el terremoto de 1989 en San Francisco
SEM1	d00	Fichero 0105 del WSJ	Sobre mendicidad y gente sin techo
	d01	Fichero 0186 del WSJ	Sobre un libro de corrupción
	d02	Fichero 0239 del WSJ	Trata sobre globos aerostáticos

Tabla 5.8. Descripción de los corpus de SensEval y SemEval

Aunque el conjunto completo de experimentos será descrito en el siguiente capítulo, pensamos que es más apropiado incluir esta sección en este capítulo, ya que se corresponde con una aplicación de nuestro sistema en una tarea de PLN, y no con una evaluación propia del sistema de WSD como tal. Para realizar esta evaluación hemos elegido el experimento que utiliza el conjunto de atributos básico y un conjunto de atributos semánticos utilizando WordNet Domains como tipo de clase semántica. Este es uno de los experimentos que mejores resultados obtienen. La evaluación la hemos hecho utilizando tres tipos de clases semánticas para generar los clasificadores, BLC–

20, BLC–50 y SuperSenses. En cada caso se ha usado el mismo tipo de atributos. Por tanto veremos los resultados para cada una de estas clases semánticas.

Resultados para BLC–20

. En la tabla 5.12 presentamos la medida F1 de los resultados para la clase BLC–20 de este experimento (Exp-WND), y también los del *baseline* según la clase más frecuente (CMF). Estos resultados son únicamente para nombres, y sobre los tres corpus que hemos descrito.

Corpus	**CMF**		**Exp-WND**	
	Poli.	Todas	Poli.	Todas
SV2	65,92	75,71	68,97	77,88
SV3	67,98	76,29	65,25	74,27
SEM1	74,10	75,74	71,69	73,45

Tabla 5.9. Medida F1 del experimento utilizado para la evaluación para BLC–20

Haciendo uso del clasificador que corresponde con el experimento anterior etiquetamos semánticamente los tres ficheros de los tres corpus, y obtenemos los correspondientes *rankings* tal y cómo hemos descrito, el de las clases semánticas asignadas por nuestro clasificador (CL), y el de clases correctas (OK). También obtenemos el *ranking* de clases asignadas según la heurística de la clase más frecuente (MF). Aplicando el test de Spearman a los diferentes rankings obtenemos las siguientes correlaciones:

- CL-OK: comparamos nuestro clasificador con las soluciones
- CL-MF: comparamos nuestro clasificador con la heurística de la clase más frecuente
- MF-OK: comparamos el *baseline* de la clase más frecuente con las clases correctas.

En la tabla 5.10 podemos ver el valor del test Spearman sobre los distintos corpus de evaluación.

Como vemos, en la gran mayoría de los casos el valor está muy próximo a 1, lo que indica que los rankings son muy similares. Cabe recordar que los *rankings* corresponden a las clases semánticas asignadas a las palabras y ordenadas por frecuencia.

Corpus	Fichero	CL-OK	CL-MF	MF-OK
SV2	d00	0,9974	0,9961	0,9906
	d01	0,9906	0,9901	0,9697
	d02	0,9899	0,9983	0,9905
SV3	d000	0,9981	0,9995	0,9978
	d001	0,9885	0,9962	0,9897
	d002	0,9977	0,9998	0,9981
SEM1	d00	0,9515	0,9893	0,9103
	d01	0,9514	0,9995	0,9514
	d02	0,9721	0,9444	0,9690

Tabla 5.10. Valores de correlación Spearman para BLC–20

Una posible mejora de este sistema consistiría en considerar únicamente aquellas palabras que representen al dominio del texto, y que no sean generales o de dominio abierto. Para obtener esta importancia nos basamos en una fórmula ampliamente utilizada en trabajos de recuperación de información: TF-IDF (Jones, 1972). En nuestro caso hemos adaptado la fórmula a nuestras necesidades, de forma que el valor TF representa la frecuencia de un término en un texto concreto (por ejemplo el texto d00 del corpus SV2). El valor IDF representa el logaritmo del inverso del número de documentos en los que aparece el término en una colección, en nuestro caso, el corpus BNC[12]. De este modo obtienen un mayor valor, para un texto concreto, aquellos términos con mayor frecuencia en el texto, y que aparecen en menor número de documentos en un corpus de dominio general. Son por tanto, palabras específicas del dominio del texto. Nuestra aproximación por tanto ha sido calcular el valor TF-IDF modificado para los términos de cada texto de evaluación, ordenar dichas palabras según el valor obtenido, y no considerar aquellas palabras que queden dentro del 1 % con menor puntuación. Solo el restante 99 % se utiliza para calcular los distintos *rankings* de clases semánticas. Con este refinamiento se obtienen los valores para el test de Spearman contenidos en la tabla 5.11.

Como vemos, los resultados son similares al proceso sin filtrado, y en muchos casos ligeramente inferiores. Esto puede significar que el filtro propuesto no es muy eficiente, o bien que estamos descartando palabras que son muy generales, pero que contienen información

[12] El *British National Corpus* ha sido desarrollado por la universidad de Oxford, y es gestionado por el consorcio BNC (`http://www.natcorp.ox.ac.uk`)

Corpus	Fichero	CL-OK	CL-MF	MF-OK
SV2	d00	0,9948	0,9894	0,9773
	d01	0,9761	0,9661	0,9930
	d02	0,9835	0,9979	0,9828
SV3	d000	0,9969	0,9987	0,9962
	d001	0,9786	0,9966	0,9637
	d002	0,9966	0,9991	0,9971
SEM1	d00	0,9253	0,9940	0,8593
	d01	0,9040	1	0,9040
	d02	0,9560	0,9995	0,9679

Tabla 5.11. Valores de correlación Spearman para BLC–20. Filtrado por TF-IDF

semántica determinante sobre el texto. También podemos ver que los *rankings* de clases según nuestro sistema y según la clase más frecuente ahora son más similares.

Resultados para BLC–50. La misma experimentación y comparación la realizamos teniendo en cuenta las clases semánticas BLC–50. Los resultados de los clasificadores sobre los corpus de evaluación se muestran en la tabla (los atributos utilizados son los mismos que para BLC–20, atributos básicos y semánticos según WordNet Domains). En la tabla mostramos el valor F1 obtenido en la tarea de etiquetado semántico de el clasificador que utilizaremos en estos experimentos.

Corpus	CMF		Exp-WND	
	Poli.	Todas	Poli.	Todas
SV2	67,20	76,75	70,39	78,92
SV3	68,01	76,74	65,38	74,83
SEM1	72,61	75,71	69,43	72,88

Tabla 5.12. Medida F1 del experimento utilizando BLC–50

Realizamos el mismo proceso que en el caso anterior con BLC–20 para la obtención de los *rankings* de clases semánticas, en esta ocasión según BLC–50, y obtenemos los valores del test de Spearman, sin realizar ningún filtrado de palabras. La tabla 5.13 muestra estos valores.

Y en la tabla 5.14 se muestran los resultados del valor de correlación de Spearman aplicando el filtrado de palabras según TF-IDF, del mismo modo que anteriormente.

Corpus	Fichero	CL-OK	CL-MF	MF-OK
SV2	d00	0,9939	0,9884	0,9776
	d01	0,9779	0,9856	0,9248
	d02	0,9842	0,9977	0,9831
SV3	d000	0,9945	0,9983	0,9938
	d001	0,9834	0,9937	0,9787
	d002	0,9965	0,9990	0,9966
SEM1	d00	0,9350	0,9748	0,8985
	d01	0,9781	0,9990	0,9781
	d02	0,9901	0,9252	0,9930

Tabla 5.13. Valores de correlación Spearman para BLC–50.

Corpus	Fichero	CL-OK	CL-MF	MF-OK
SV2	d00	0,9927	0,9813	0,9618
	d01	0,9624	0,9557	0,9874
	d02	0,9608	0,9975	0,9707
SV3	d000	0,9959	0,9985	0,9949
	d001	0,9776	0,9952	0,9599
	d002	0,9952	0,9979	0,9956
SEM1	d00	0,9179	0,9917	0,8593
	d01	0,9539	1	0,9539
	d02	0,9866	0,9989	0,9921

Tabla 5.14. Valores de correlación Spearman para BLC–50. Filtrado por TF-IDF

En este caso los resultados en este caso utilizando BLC–50 son ligeramente mejores que para BLC–20, como era de esperar. Además hay algunos casos en los que el valor de Spearman alcanza un 1, aplicando filtrado, lo que quiere decir que los *rankings* obtenidos son totalmente idénticos.

Resultados con SuperSense. De nuevo repetimos el mismo proceso con las clases semánticas SuperSenses. El clasificador basado en SuperSenses que utiliza el conjunto de atributos básicos y atributos semánticos según WordNet Domains obtiene el rendimiento que mostramos en la tabla 5.15.

Corpus	CMF		Exp-WND	
SV2	70,48	80,41	73,59	83,47
SV3	72,59	81,50	70,10	79,82
SEM1	72,46	78,53	65,94	73,45

Tabla 5.15. Resultados del experimento utilizado para la evaluación para SuperSense

En las tablas 5.16 y 5.17 presentamos los valores de correlación según Spearman entre los distintos pares de *rankings* de clases semánticas SuperSenses, sin filtrado y con filtrado según TF-IDF respectivamente.

Corpus	Fichero	CL-OK	CL-MF	MF-OK
SV2	d00	0,7994	0,7325	0,6150
	d01	0,7069	0,6151	-0,2917
	d02	0,7875	0,9754	0,7911
SV3	d000	0,8029	0,9201	0,8431
	d001	-0,4923	0,9701	-0,2352
	d002	0,8820	0,9802	0,8323
SEM1	d00	0,4524	0,6426	0,5
	d01	0,9182	0,5412	0,9318
	d02	0,5955	-0,1448	0,9126

Tabla 5.16. Valores de correlación Spearman para SuperSense

Corpus	Fichero	CL-OK	CL-MF	MF-OK
SV2	d00	0,8464	0,6091	0,3275
	d01	0,7309	0,6029	-0,3221
	d02	0,5182	0,9231	0,4818
SV3	d000	0,8765	0,9456	0,9235
	d001	0,5099	0,9868	0,5099
	d002	0,9657	0,9816	0,9397
SEM1	d00	0,4405	0,9818	0,5
	d01	0,9030	0,9955	0,9212
	d02	0,3333	0,7394	0,9273

Tabla 5.17. Valores de correlación Spearman para SuperSense. Filtrado por TF-IDF

Ahora los resultados son notablemente menores, a pesar de que la clase semántica SuperSense posee un mayor nivel de abstracción y una polisemia menor, con lo que era de esperar que los resultados fueran altos. Esto puede indicarnos que realmente nuestras clases BLC funcionan bien a modo de *keyphrases*, debido a que las palabras de un texto mantienen una coherencia semántica según dichas clases, mientras que en el caso de SuperSenses parece que no se mantiene esta coherencia. Por tanto los buenos resultados de un sistema de este tipo, no solo dependen del nivel de abstracción de las clases seleccionadas (podríamos pensar: con más abstracción, menos poli-

semia y mejores resultados), sino también de la calidad y coherencia que mantengan entre sí el conjunto de clases semánticas, y del modo en que las palabras sean agrupadas bajo los distintos conceptos semánticos.

6. Evaluación Empírica Usando Diferentes Niveles de Abstracción

En el capítulo anterior nos centramos en explicar el desarrollo del sistema general de desambiguación semántica, haciendo énfasis en la arquitectura del mismo y en un estudio de los atributos más apropiados para el entrenamiento, así como en unas primeras evaluaciones que nos indicaran el rendimiento de nuestro sistema, aunque posteriormente también hemos incluido nuestra participación en SEM1 y SEM2, por mantener la coherencia.

En este capítulo, nos centraremos en evaluar el funcionamiento del sistema desarrollado usando distintos niveles de abstracción. Estos niveles de abstracción nos los proporcionarán las diversas clases semánticas de las que disponemos para crear los clasificadores. Por tanto, diseñaremos diferentes experimentos que utilicen los distintos conjuntos de clases semánticas para generar los clasificadores.

En primer lugar, en la sección 6.2 se muestran los resultados de nuestros clasificadores usando distintos grados de abstracción y utilizando diferentes conjuntos de atributos. Esta evaluación se realiza sobre los corpus SE2, SE3 y SEM1. En estos experimentos también incluimos el estudio de las curvas de aprendizaje para ver cómo evolucionan nuestros clasificadores a medida que se incrementa la cantidad de ejemplos de entrenamiento. Así comprobaremos si realmente en la aproximación de clases semánticas se reduce la cantidad de datos de entrenamiento necesitado respecto a las técnicas tradicionales que usan sentidos. También se presentan unos experimentos en los que se combinan diferentes clasificadores con distinto grado de abstracción, para generar un único clasificador a nivel de sentidos de las palabras. En este caso tratamos de ver si nuestras clases semánticas son capaces de mantener el poder discriminatorio cuando se transforman en sentidos detallados.

La evaluación que hemos descrito en el párrafo anterior está hecha de forma aislada, no presenta ninguna comparación con otros sistemas de WSD. Por ello realizamos la evaluación que se presenta en la sección 6.3. Se trata de una serie de experimentos en los que contrastamos nuestro sistema frente a los sistemas participantes en las tareas *All Words* en SE2 y SE3. Debido a que los sistemas de SE trabajan con sentidos detallados, y nuestro sistema emplea clases semánticas, debemos realizar dicha comparación de dos modos distintos. Para llevar a cabo esta comparación de una forma justa, se evalúa al mismo nivel de granularidad para todos los sistemas, los participantes y el nuestro: en primer lugar a nivel de sentido de WordNet, y, en segundo lugar, a nivel de clase semántica. Hay que tener en cuenta que nuestro sistema no participó en dichas competiciones, SE2 y SE3 y, por tanto, la evaluación ha sido realizada *a posteriori*, haciendo uso de los corpus que proporcionaron y disponiendo de las salidas concretas de cada sistema participante y no únicamente su resultado global.

6.1 Conjuntos de Atributos

A partir de las pruebas realizadas en el capítulo anterior, y descritas en la sección 5.2, definimos un conjunto de atributos que nos proporciona buenos resultados. A este conjunto lo hemos denominado BASE. Nuestra intención no era encontrar aquellos que obtuvieran los mejores resultados, sino seleccionar un conjunto de atributos simples para evaluar cómo se comporta nuestro sistema basado en clases semánticas y poder realizar un número elevado de pruebas en un tiempo razonable desde el punto de vista de la eficiencia computacional.

Por otra parte hacemos uso de otro conjunto, al cual denominamos IXA por el nombre del grupo de investigación que definió este conjunto de atributos (Agirre *et al.*, 2006). Aunque son similares a los atributos BASE, tienen ligeras diferencias y están agrupados de otro modo, por lo que pensamos que es interesante probar el funcionamiento de estos atributos también bajo nuestro marco de desarrollo basado en clases semánticas, ya que posiblemente nos proporcionen un punto de vista distinto del rendimiento de nuestro sistema.

6.1.1 Atributos BASE

En este conjunto de atributos podemos diferenciar los siguientes: **atributos básicos**:

Formas de palabras y lemas en una ventana de 10 palabras alrededor de la palabra objetivo

Etiquetas morfológicas: la concatenación de tres/cinco precedentes/siguientes etiquetas morfológicas

Bigramas y trigramas formados por lemas y formas de palabras y obtenidos dentro de una ventana de 5 palabras. Se usan todos los *tokens* sin tener en cuenta su categoría morfológica para construir estos n-gramas. La palabra objetivo es reemplazada por una *X* en estos atributos para no hacer a los clasificadores semánticos dependientes de la palabra concreta en cuestión.

Además definimos un conjunto de **atributos semánticos** para explotar los diferentes recursos semánticos que usamos, y enriquecer el conjunto de atributos básicos:

Clase semántica más frecuente de la palabra objetivo, calculada sobre el corpus SemCor

Clase semántica de palabras monosémicas contenidas en un contexto de 5 palabras alrededor de la palabra objetivo.

Diferentes conjuntos de clases semánticas son considerados para construir estas características semánticas. En particular, dos conjuntos diferentes de BLC (BLC–20 y BLC–50[1]), SuperSenses, WordNet Domains (WND) y las clases de la ontología SUMO.

6.1.2 Atributos IXA

Podemos agrupar los atributos IXA de la siguiente manera:

Atributos locales: (*L*) entre los que consideramos

[1] Seleccionamos estos dos conjuntos debido a que representan diferentes niveles de abstracción. Debemos recordar que los valores 20 y 50 indican el umbral establecido como mínimo numero de conceptos que un BLC candidato debe representar para ser aceptado finalmente como BLC. Se eligen los BLC construidos considerando todo tipo de relaciones (*All*). Podemos consultar los distintos tipos de BLC creados y el método de selección en la sección 4.1.

- Unigramas: para la palabra objetivo
- Bigramas: que contienen a la palabra objetivo, formados con palabras, lemas y etiquetas morfológicas
- Trigramas: que contienen a la palabra objetivo, formados con palabras, lemas y etiquetas morfológicas
- Token anterior y posterior: lema y palabra del primer token con contenido semántico (nombre, verbo, adjetivo o adverbio) anterior/posterior a la palabra objetivo

Bolsa de palabras: conjuntos de palabras en el contexto de la palabra objetivo. Diferenciamos entre dos tipos de contextos

- Ventana (*BOW–win*): el contexto es una ventana de 4 tokens alrededor de la palabra objetivo
- Párrafo (*BOW–par*): el contexto es todo el párrafo en el que aparece la palabra objetivo

Atributos semánticos: (*S*) los mismos que utilizamos en el caso de los atributos BASE.

- Clase más frecuente para la palabra objetivo
- Clase de las palabras monosémicas que aparecen en la misma oración que la palabra objetivo

Como vemos, hemos extendido este conjunto con los atributos semánticos, concretamente utilizamos el conjunto BLC–20 para construirlos. Seleccionamos este conjunto de clases porque demostró un buen funcionamiento en los experimentos con los atributos BASE.

Debemos destacar algunas peculiaridades de este conjunto de atributos. Por ejemplo, en el caso de bigramas y trigramas se realiza una generalización de dichos atributos, y se obtienen otros nuevos en los que se sustituye la palabra objetivo por una cadena comodín idéntica en todos los casos. De esta forma se elimina la dependencia del atributo respecto a la palabra objetivo. En este mismo caso de generalización de atributos, cuando tratamos con bigramas o trigramas construidos con etiquetas morfológicas sólo se tienen en cuenta el primer carácter de dicha etiqueta, que es el que indica la categoría gramatical (nombre, verbo...).

6.1.3 Filtrado de atributos

Para aumentar las capacidades de generalización de los clasificadores y seleccionar aquellos atributos más informativos, diseñamos y empleamos un proceso de filtrado de atributos. En nuestra aproximación basada en clases, cada clasificador posee un conjunto de atributos propio, extraído de los ejemplos anotados con la clase a la que representa el clasificador, es decir, de los ejemplos positivos. Por tanto, una vez obtenidos todas las características para cada clasificador desde el corpus de entrenamiento, se filtra cada uno de estos conjuntos para seleccionar únicamente aquellos más importantes e informativos.

El filtrado para una clase en concreto trata de seleccionar únicamente aquellos atributos relevantes para esa clase y no para el resto. Establecemos una medida para calcular la relevancia de un atributo f para una clase concreta c[2]. El proceso de filtrado se puede ver detallado a continuación (recordemos que partimos de un conjunto de atributos para cada clase extraídos del corpus de entrenamiento):

```
Para cada clase c
  Para cada atributo f de la clase c
    f_clase = frecuencia de f para clase c
    f_total = frecuencia de f para todas las clases
    Si f_clase / f_total < UMBRAL Entonces
      ELIMINAR f de la lista de atributos de c
    Fin Si
  Fin Para
Fin Para
```


De este modo aseguramos que para cada clase seleccionamos los atributos más frecuentemente relacionados con dicha clase, y que además no son muy frecuentes para todas las clases en general. Ajustamos el umbral empíricamente a 0,25, a partir de unas pruebas iniciales realizadas sobre el corpus SE3.

[2] Nótese la diferencia entre el valor de un atributo concreto y un tipo de atributo. Por ejemplo un tipo puede ser la *forma de la palabra* y un atributo concreto sería *houses*.

6.2 Evaluación de clases semánticas y atributos

Para evaluar nuestra arquitectura basada en clases semánticas y analizar el rendimiento del sistema en diferentes niveles de abstracción, hemos diseñado un gran conjunto de experimentos. Cada experimento queda definido por dos conjuntos:

1. El de clases semánticas seleccionadas para construir los clasificadores, que determinarán el nivel de abstracción
2. El de atributos empleados para realizar el entrenamiento

Como clases semánticas para construir los clasificadores hemos utilizado BLC–20, BLC–50, WND, SUMO y SuperSenses, además de los tradicionales sentidos[3]. Por otra parte, la selección del conjunto de atributos será crucial para el éxito en la evaluación del sistema y es, precisamente, la tarea más difícil de todas: ¿cuál es el conjunto de atributos ideal para el aprendizaje? La mayor parte de la experimentación que se mostrará de ahora en adelante intenta responder a esta cuestión.

Es obvio que los resultados de la tarea de desambiguación semántica variarán en función del corpus de evaluación y del nivel de abstracción o granularidad. Para obtener una aproximación de la dificultad de dicha tarea, obtenemos la polisemia media para cada corpus y conjunto de clases semánticas. En la tabla 6.1 podemos ver esta información.

Test	Cat.	Sentidos	BLC–20	BLC–50	WND	SUMO	SS
SensEval–2	N	4,02	3,45	3,34	2,66	3,33	2,73
	V	9,82	7,11	6,94	2,69	5,94	4,06
SensEval–3	N	4,93	4,08	3,92	3,05	3,94	3,06
	V	10,95	8,64	8,46	2,49	7,60	4,08
SemEval–1	N	4,85	4,19	3,95	3,01	3,94	2,93
	V	10,41	7,72	7,49	2,65	6,65	4,23

Tabla 6.1. Polisemia media sobre SensEval–2 y SensEval–3

Como podemos ver, la polisemia disminuye cuando el grado de abstracción aumenta. Para aquellas clases semánticas que presentan

[3] El clasificador para sentidos detallados sigue la aproximación tradicional, en la que se crea un clasificador para cada palabra. Por tanto, en este caso no explotamos las ventajas de las clases semánticas

granularidad mayor, es decir, un nivel de abstracción más concreto (sentidos y BLC–20) la polisemia es más elevada. Al contrario en el otro extremo, en aquellas clases más abstractas (SuperSenses), la polisemia es más baja. Este fenómeno es el esperado. Por otra parte, la evolución de la polisemia en función de la clase semántica es la misma en los tres corpus; presenta por tanto un comportamiento coherente.

6.2.1 Sistema base o *baseline*

Con el sistema base, o *baseline*, tratamos de establecer una cota inferior de los resultados. Lo deseable es que nuestro sistema supere este umbral mínimo como garantía de buen funcionamiento. Definimos una heurística muy sencilla como *baseline*. Se trata de asignar a una palabra ambigua su clase semántica más frecuente sobre un cierto recurso semántico anotado, en nuestro caso SemCor. Dicho de otro modo, consiste en obtener la clase con la cual aparece más frecuentemente anotada la palabra en cuestión. En el caso de que haya un empate en la frecuencia de varias clases para una misma palabra, se recurre a calcular la frecuencia general de dichas clases en el corpus, sin ceñirnos únicamente a aquellas ocurrencias de la palabra, y elegir aquella clase que mayor valor obtenga.

Un caso particular es aquel en que SemCor no dispone de ocurrencias de la palabra ambigua. En este caso no podemos calcular las frecuencias de anotación de la palabra con sus posibles clases, y optamos por asignarle la clase más frecuente globalmente (asociada a cualquier palabra) de entre sus posibles clases

Este sistema base es ampliamente usado en tareas de WSD a modo de evaluación (Gale *et al.*, 1992a), y, como hemos comentado, establece un resultado mínimo que cualquiera sistema competitivo debería superar. De cualquier modo, los baseline en WSD suelen obtener resultados muy altos y son difíciles de superar, con lo cual todavía suponen un límite a partir del cual se puede considerar que se está obteniendo resultados satisfactorios.

6.2.2 Resultados BASE

En primer lugar mostramos los resultados utilizando el conjunto de atributos BASE definidos en la sección 6.1.1. En este caso cada experimento queda definido por la clase semántica seleccionada para generar los clasificadores, y por la clase semántica usada para construir los atributos semánticos (BLC–20, BLC–50, WND, SUMO y SuperSenses). Por tanto, entrenamos nuestro sistema utilizando diferentes combinaciones de atributos y clases semánticas, sobre el corpus SemCor, y lo evaluamos sobre los corpus de SE2, SE3 y SEM1, para nombres y verbos[4].

En las tablas 6.2 y 6.3 presentamos la medida F1 para nombres y para verbos, respectivamente. Aquellos resultados que muestran una diferencia estadísticamente significativa[5] respecto a los resultados de la heurística basada en la clase más frecuente se han señalado en negrita.

La columna *Clase* indica el repositorio de clases semánticas usadas para construir los clasificadores, la granularidad de los mismos. La columna *Atributos Semánticos* representa el tipo de clase semántica usado para construir los atributos semánticos. Es importante tener en cuenta esta diferencia: por ejemplo, el experimento que tiene como clase BLC–20 y como atributos semánticos BLC–50, utiliza el conjunto de clases BCL–20 para crear los clasificadores (se creará un clasificador para cada clase del conjunto BLC–20), y como atributos usa los atributos básicos más los atributos semánticos construidos mediante el repositorio BLC–50.

También incluimos los resultados del *baseline* a modo de referencia, así como los resultados del sistema utilizando únicamente los atributos básicos, para ver el efecto introducido por los atributos semánticos (*atrBasicos*).

Como hemos comentado anteriormente, los resultados del *baseline* siguiendo la heurística de la clase más frecuente son bastante altos, en particular para nombres, con un valor de F1 rondando el 70 % u 80 %. Los *baselines* para nombres muestran un comportamiento similar tanto sobre SE2 como sobre SE3, mientras que son ligeramente

[4] Para una descripción más detallada de dichos corpus, consultar la sección 3.3
[5] Usamos el test de McNemar.

superiores en el caso de SEM1, debido probablemente a la naturaleza homogénea del corpus.

En el caso de **verbos** son mucho más bajos los resultados sobre SE2 que sobre SE3[6]. Los *baseline* para verbos en SEM1 son muy similares a los de SE2. En este último corpus los resultados de dicho *baseline* oscilaron entre 44 % y 68 %, mientras que en SE3 lo hacen entre 52 % y 79 %, y en SEM1 entre un 46 % y un 67 %. Se produce una excepción en el caso de WND para verbos, y es que en este caso la polisemia es muy baja, con lo cual los resultados aumentan en gran medida. Observamos, como es lógico, una correlación entre las polisemias y los resultados usando diferentes conjuntos de clases semánticas: cuando usamos conjuntos más abstractos (con una polisemia menor) se obtienen resultados más altos. Finalmente, teniendo en cuenta los resultados de los *baseline* parece que la tarea de desambiguación es más compleja sobre el corpus SE2 que sobre SE3 y SEM1. Salvando este punto, los resultados del sistema base son siempre muy elevados, con lo que el hecho de que nuestro sistema supere a éstos en casi la totalidad de los casos es todavía más relevante.

Analizando el caso para **nombres**, se observa un comportamiento muy diferente de nuestro sistema sobre los tres corpus. Mientras que en el caso de SE3 ninguna de las ejecuciones de nuestro sistema obtiene una mejora estadísticamente significativa sobre el *baseline*, en el caso de SE2 sí se observa dicha mejora, sobre todo en aquellos experimentos que incluyen atributos semánticos. Sobre SEM1 nuestro sistema no es capaz de superar los resultados base en la mayoría de los casos, exceptuando algunas ejecuciones utilizando SuperSense como conjunto de clases semánticas. Los *baseline* sobre SEM1 son, de nuevo, especialmente elevados, a pesar de que la polisemia no es menor que en el caso de los otros dos corpus. En particular observamos que las clases semánticas que aportan mejores resultados como atributos semánticos son WND y BLC–20, mientras que BLC–50 y SuperSenses no aportan tal mejora.

En el caso de verbos, se obtienen mejoras significativas sobre los tres corpus de evaluación utilizando diferentes combinaciones de atri-

[6] Es importante recordar que en SE2 se utilizó WordSmith como repositorio de sentidos en el caso de verbos.

butos semánticos. Cabe resaltar el caso de WND como atributos semánticos, que consiguen muy buenos resultados.

En general, los resultados obtenidos seleccionando como nivel de abstracción BLC–20 no son muy diferentes de los obtenidos con BLC–50, en muy pocos casos la diferencia es mayor de 2 puntos. Un caso excepcional es SEM1, dónde se obtienen mejores resultados para nombres según BLC–20 que según BLC–50. Por otra parte, si observamos el número de clases contenido en cada conjunto (lo que nos daría una idea del nivel de abstracción de dicho conjunto), vemos que BLC–20 contiene 558 clases para nombres, BLC–50 253 clases, y SuperSenses solo 24 clases. A pesar de este hecho, utilizando los conjuntos de BLC se obtienen tasas de acierto elevadas, manteniendo un nivel de abstracción más bajo y por tanto un nivel expresivo más alto que en el caso de SuperSenses, por ejemplo. De hecho, utilizando SuperSenses obtenemos un etiquetador que funciona con un acierto entorno al 80 %, haciendo uso de un conjunto de 24 etiquetas. Sin embargo, el etiquetador basado en BLC–20 utiliza un conjunto mucho más amplio y rico de etiquetas, 558 clases, y alcanza un resultado bastante próximo al etiquetador de SuperSenses: 75 % de acierto. Esto pone de relieve la calidad de los conjuntos de clases BLC obtenidos automáticamente, y su adecuación para ser usados en el sistema de desambiguación semántica que proponemos.

Cabe resaltar el caso de WND, que obtiene muy buenos resultados tanto para nombres como para verbos, en gran parte derivado de su acusada reducción de polisemia. También en el caso de sentidos se consiguen clasificadores competitivos.

6.2.3 Resultados IXA

En esta sección mostramos los resultados del sistema entrenado utilizando diferentes combinaciones de las agrupaciones de atributos definidas en la sección 6.1.2, y llamados atributos IXA. Además de los resultados de los clasificadores para los diferentes conjuntos de atributos, incluimos también los resultados del sistema base con dos diferentes configuraciones: seleccionando para cada palabra ambigua la clase más frecuente sobre SemCor (*baseSC*), y la clase más frecuente sobre WordNet (*baseWN*, coincide con la clase correspon-

Clase	Atr.Sem.	SE2		SE3		SEM1	
		Poli.	**Todas**	**Poli.**	**Todas**	**Poli.**	**Todas**
Sentidos	baseline	59,66	70,02	64,45	72,30	70,41	71,59
	atrBasicos	61,13	71,20	65,45	73,15	68,05	69,32
	BLC–20	61,93	71,79	65,45	73,15	68,05	69,32
	BLC–50	61,79	71,69	65,30	73,04	68,05	69,32
	SuperSenses	61,00	71,10	64,86	72,70	68,05	69,32
	WND	61,13	71,20	65,45	73,15	68,05	69,32
	SUMO	61,66	71,59	65,45	73,15	68,05	69,32
BLC–20	baseline	65,92	75,71	67,98	76,29	74,10	75,71
	atrBasicos	65,65	75,52	64,64	73,82	71,69	73,45
	BLC–20	**68,70**	**77,69**	68,29	76,52	73,49	75,14
	BLC–50	**68,83**	**77,79**	67,22	75,73	68,67	70,62
	SuperSenses	65,12	75,14	64,64	73,82	68,67	70,62
	WND	**68,97**	**77,88**	65,25	74,24	71,69	73,45
	SUMO	68,57	77,60	64,49	73,71	67,47	69,49
BLC–50	baseline	67,20	76,65	68,01	76,74	72,61	75,71
	atrBasicos	64,28	74,57	66,77	75,84	70,06	73,45
	BLC–20	**69,72**	**78,45**	68,16	76,85	71,97	75,14
	BLC–50	67,20	76,65	68,01	76,74	69,43	72,88
	SuperSenses	65,60	75,52	65,07	74,61	66,24	70,06
	WND	**70,39**	**78,92**	65,38	74,83	69,43	72,88
	SUMO	**71,31**	**79,58**	66,31	75,51	63,69	67,80
WND	baseline	78,97	86,11	76,74	83,8	80,69	84,18
	atrBasicos	70,96	80,81	67,85	77,64	61,38	68,36
	BLC–20	72,53	81,85	72,37	80,79	64,83	71,19
	BLC–50	73,25	82,33	71,41	80,11	64,14	70,79
	SuperSenses	74,39	83,08	68,82	78,31	61,38	68,36
	WND	78,83	86,01	76,58	83,71	66,21	72,32
	SUMO	75,11	83,55	73,02	81,24	64,14	70,62
SUMO	baseline	66,40	76,09	71,96	79,55	71,62	76,27
	atrBasicos	68,53	77,60	68,10	76,74	66,22	71,75
	BLC–20	65,60	75,52	68,10	76,74	66,89	72,32
	BLC–50	65,60	75,52	68,72	77,19	66,22	71,75
	SuperSenses	68,39	77,50	68,41	76,97	64,19	70,06
	WND	**68,92**	**77,88**	69,03	77,42	66,22	71,75
	SUMO	68,92	77,88	70,88	78,76	67,57	67,11
SuperSenses	baseline	70,48	80,41	72,59	81,50	72,46	78,53
	atrBasicos	69,77	79,94	69,60	79,48	69,57	76,27
	BLC–20	71,47	81,07	72,43	81,39	75,36	80,79
	BLC–50	70,20	80,22	72,92	81,73	72,46	78,53
	SuperSenses	70,34	80,32	65,12	76,46	73,91	79,66
	WND	**73,59**	**82,47**	70,10	79,82	65,94	73,45
	SUMO	70,62	80,51	71,93	81,05	70,29	76,84

Tabla 6.2. Resultados para nombres, atributos BASE

Clase	Atr.Sem.	SE2		SE3		SEM1	
		Poli.	Todas	Poli.	Todas	Poli.	Todas
Sentidos	baseline	41,20	44,75	49,78	52,88	42,51	46,42
	atrBasicos	42,01	45,53	**54,19**	**57,02**	**45,75**	**49,43**
	BLC–20	41,59	45,14	**53,74**	**56,61**	**46,56**	**50,19**
	BLC–50	42,01	45,53	**53,6**	**56,47**	**46,56**	**50,19**
	SuperSenses	41,80	45,34	**53,89**	**56,75**	**46,15**	**49,81**
	WND	42,01	45,53	**53,89**	**56,75**	**46,56**	**50,19**
	SUMO	42,22	45,73	**54,19**	**57,02**	**46,56**	**50,19**
BLC–20	baseline	50,21	55,13	54,87	58,82	47,08	52,79
	atrBasicos	52,36	57,06	**57,27**	**61,10**	**51,25**	**56,51**
	BLC–20	52,15	56,87	56,07	59,92	**52,08**	**57,25**
	BLC–50	51,07	55,90	**56,82**	**60,60**	**51,67**	**56,88**
	SuperSenses	51,50	56,29	**57,57**	**61,29**	48,75	54,28
	WND	**54,08**	**58,61**	57,12	60,88	49,58	55,02
	SUMO	52,36	57,06	**57,42**	**61,15**	47,08	52,79
BLC–50	baseline	49,78	54,93	55,96	60,06	47,92	53,53
	atrBasicos	**53,23**	**58,03**	58,07	61,97	**52,50**	**57,62**
	BLC–20	**52,59**	**57,45**	57,32	61,29	**52,92**	**57,99**
	BLC–50	51,72	56,67	57,01	61,01	**52,92**	**57,99**
	SuperSenses	52,59	57,45	57,92	61,83	50,42	55,76
	WND	**55,17**	**59,77**	**58,52**	**62,38**	**51,67**	**56,88**
	SUMO	52,16	57,06	57,92	61,83	47,92	53,53
WND	baseline	84,80	90,33	84,96	92,20	85,12	90,71
	atrBasicos	84,50	90,14	78,63	88,92	88,69	92,57
	BLC–20	84,50	90,14	81,53	90,42	88,69	92,57
	BLC–50	84,50	90,14	81,00	90,15	88,69	92,57
	SuperSenses	83,89	89,75	78,36	88,78	88,69	92,57
	WND	85,11	90,52	84,96	92,20	**89,88**	**93,31**
	SUMO	85,11	90,52	80,47	89,88	88,69	92,57
SUMO	baseline	54,24	60,35	59,69	64,71	53,81	59,48
	atrBasicos	56,25	62,09	61,41	66,21	54,24	59,48
	BLC–20	55,13	61,12	61,25	66,07	52,54	57,99
	BLC–50	56,25	62,09	61,72	66,48	52,12	57,62
	SuperSenses	53,79	59,96	59,69	64,71	50,42	56,13
	WND	55,58	61,51	61,56	66,35	52,97	58,36
	SUMO	54,69	60,74	60,00	64,98	55,51	60,59
SuperSenses	baseline	62,79	68,47	76,24	79,07	59,73	66,91
	atrBasicos	**66,89**	**71,95**	75,47	78,39	60,63	67,66
	BLC–20	63,70	69,25	74,69	77,70	57,01	64,31
	BLC–50	63,70	69,25	74,69	77,70	57,01	64,31
	SuperSenses	63,70	69,25	74,84	77,84	60,18	66,91
	WND	**66,67**	**71,76**	77,02	79,75	58,37	65,43
	SUMO	64,84	70,21	74,69	77,70	59,73	66,54

Tabla 6.3. Resultados para verbos, atributos BASE

diente al sentido 1 de la palabra). En la tabla 6.4 podemos ver los resultados del sistema en el caso de nombres. En negrita aparecen los resultados que muestran una diferencia estadísticamente significativa según el test de McNemar. Por simplificar, en aquellos en que se han utilizado en contexto de ventana alrededor de la palabra objetivo (BOW–w) y también el contexto de párrafo (BOW–p), se presenta en la tabla como BOW–wp. Hay que tener en cuenta que aunque en cierto modo un contexto incluye al otro, el modo de codificar los atributos de ambos ha sido distinto, para reflejar la cercanía a la palabra objetivo de cada atributo.

En el caso de SE2, en casi todos los casos el *baseline* es superado, a excepción del caso de WND. Atendiendo a los resultados según los diferentes conjuntos de atributos, observamos que utilizando únicamente el conjunto de atributos *Local* se alcanzan buenos resultados. En este caso los atributos semánticos no aportan ninguna mejora en general, ni tampoco los atributos extraídos del contexto. Los resultados son coherentes con los diferentes niveles de granularidad de los clasificadores: a mayor nivel de abstracción y menor granularidad, mejores resultados.

Analizando los resultados sobre el corpus SE3, los *baseline* no son superados en todos los casos, en concreto en los casos de mayor abstracción nuestros clasificadores no mejoran dichos resultados base. El rendimiento mediante el conjunto de atributos *Local* en este caso es superado generalmente, en mayor manera por los atributos semánticos, que ahora sí que proporcionan información relevante para nuestros clasificadores.

Por último, para el corpus SEM1, los *baseline* sólo son superados en el caso de BLC–50. Posiblemente la reducción de polisemia y de granularidad derivada del uso de clases semánticas y de la agrupación de sentidos en el corpus, provoca que los resultados base sean muy altos y difíciles de superar por los clasificadores. En cuanto a los diferentes grupos de atributos, los atributos locales proporcionan un rendimiento menor que el conjunto de atributos semánticos. Ninguno de los dos grupos de atributos extraídos del contexto confieren mejoras a los resultados.

En la tabla 6.5 podemos ver los resultados para los verbos. De nuevo en negrita se muestran los que presentan diferencia signifi-

Clase	Atr.Sem.	SE2		SE3		SEM1	
		Poli.	Todas	Poli.	Todas	Poli.	Todas
Sentidos	baseSC	59,66	70,02	64,45	72,30	68,64	69,69
	baseWN	60,05	70,31	62,14	70,50	70,41	71,59
	L	61,50	71,39	63,44	71,51	67,46	68,75
	L S	61,37	71,29	63,58	71,62	68,05	69,32
	L BOW–w	61,50	71,39	63,44	71,51	67,46	68,75
	L BOW–w S	61,37	71,29	63,58	71,62	68,05	69,32
	L BOW–wp	61,50	71,39	63,44	71,51	67,46	68,75
	L BOW–wp S	61,37	71,29	63,58	71,62	68,05	69,32
BLC–20	baseSC	65,92	75,71	67,98	76,29	73,49	75,14
	baseWN	64,72	74,86	65,10	74,16	74,10	75,71
	L	**69,89**	**78,54**	66,31	75,06	71,69	73,45
	L S	66,05	75,80	68,44	76,63	73,49	75,14
	L BOW–w	**70,16**	**78,73**	65,86	74,72	71,69	73,45
	L BOW–w S	66,05	75,80	68,44	76,63	73,49	75,14
	L BOW–wp	**70,16**	**78,73**	65,86	74,72	71,69	73,45
	L BOW–wp S	66,05	75,80	68,44	76,63	73,49	75,14
BLC–50	baseSC	67,20	76,65	68,01	76,74	70,06	73,45
	baseWN	66,00	75,80	64,91	74,49	72,61	75,71
	L	**69,06**	**77,98**	64,14	73,93	68,15	71,75
	L S	67,20	76,65	68,01	76,74	71,97	75,14
	L BOW–w	**69,06**	**77,98**	64,14	73,93	68,15	71,75
	L BOW–w S	67,20	76,65	68,01	76,74	71,97	75,14
	L BOW–wp	**69,06**	**77,98**	64,14	73,93	68,15	71,75
	L BOW–wp S	67,20	76,65	68,01	76,74	71,97	75,14
WND	baseSC	78,97	86,11	76,74	83,80	80,69	84,18
	baseWN	79,40	86,39	73,74	81,46	80,69	84,18
	L	54,36	69,85	47,33	63,37	61,38	68,36
	L S	57,22	71,74	51,37	66,18	66,90	72,88
	L BOW–w	55,65	70,70	46,20	62,58	60,69	67,80
	L BOW–w S	57,65	72,02	49,11	64,61	66,21	72,32
	L BOW–wp	57,22	71,74	47,33	63,37	46,21	55,93
	L BOW–wp S	62,52	75,24	49,27	64,72	46,21	55,93
SUMO	baseSC	66,40	76,09	71,96	79,55	68,92	74,01
	baseWN	61,09	72,31	68,57	77,08	71,62	76,27
	L	69,19	78,07	68,57	77,08	68,92	74,01
	L S	65,34	75,33	68,41	76,97	68,92	74,01
	L BOW–w	68,53	77,60	68,57	77,08	68,92	74,01
	L BOW–w S	65,21	75,24	68,41	76,97	68,92	74,01
	L BOW–wp	68,53	77,60	68,57	77,08	68,92	74,01
	L BOW–wp S	68,92	77,88	69,18	77,53	68,24	73,45
SuperSenses	baseSC	70,48	80,41	72,59	81,50	73,91	79,66
	baseWN	69,07	79,48	68,11	78,48	72,46	78,53
	L	71,89	81,35	69,93	79,71	72,46	78,53
	L S	72,03	81,44	68,94	79,04	72,46	78,53
	L BOW–w	71,75	81,26	69,77	79,60	73,19	79,10
	L BOW–w S	72,18	81,54	69,93	79,71	72,46	78,53
	L BOW–wp	71,33	80,97	70,60	80,16	73,19	79,10
	L BOW–wp S	71,61	81,16	69,77	79,60	72,46	78,53

Tabla 6.4. Resultados para nombres, atributos IXA

cativa según el test de McNemar. En el caso de SE2 los *baselines* se sobrepasan, excepto para WND y SuperSenses. La agrupación de atributos *Local* obtienen resultados bastante altos en comparación con el resto de agrupaciones de atributos. Aunque en este caso los atributos semánticos no parecen aportar ninguna mejora respecto al uso de los atributos locales, los atributos extraídos del contexto sí que proporcionan un ligero incremento en el acierto de los clasificadores.

Para SE3 sucede lo mismo: los *baseline* se sobrepasan a excepción del caso de WND y SuperSenses. Los atributos locales también son superados, en algunos casos mediante la inclusión de los atributos semánticos, y en la mayoría de los casos mediante el uso de los atributos extraídos del contexto.

Finalmente para SEM1, los *baselines* únicamente son vencidos en los 3 casos de menor granularidad (mayor grado de detalle, menor nivel de abstracción) de los clasificadores: sentidos, BLC–20 y BLC–50. Los atributos locales se superan en la gran mayoría de los casos. La información semántica mejora en algunos ocasiones el rendimiento de los clasificadores (BLC–20, BLC–50 y WND). Además, en general los atributos extraídos del contexto aportan una mejora en el funcionamiento de nuestros clasificadores en este caso.

Podemos considerar que, tanto para nombres como para verbos, solo con la información extraída del contexto local de la palabra objetivo se consiguen resultados que difícilmente superamos incluyendo la información semántica o el contexto más amplio en torno a la palabra objetivo. Sí es cierto que, en casos concretos, ambos tipos de información, semántica y contextual, mejoran notablemente el rendimiento del sistema, incluso en algunos casos de forma muy acusada. Quizá el contexto debiera representarse de otra forma, posiblemente seleccionando una ventana más amplia, ya que en el caso de trabajar con clases semánticas, la información representativa se encuentra en ventanas bastante amplias en torno a la palabra objetivo, llegando incluso a tener que buscar en una ventana de 50 palabras para obtener atributos que representen correctamente a la palabra objetivo, como se indica en (Yarowsky, 1992). El uso de ventanas contextuales mayores aumentaría enormemente el número de atributos con el que trabajamos, y posiblemente necesitáramos un método de filtrado y selección de atributos más potente. Tanto este problema como el del

Clase	Atr.Sem.	SE2		SE3		SEM1	
		Poli.	**Todas**	**Poli.**	**Todas**	**Poli.**	**Todas**
Sentidos	baseSC	41,20	44,75	49,78	52,88	44,94	48,68
	baseWN	40,79	44,36	49,78	52,88	42,51	46,42
	L	43,06	46,50	49,49	52,60	**48,18**	**51,70**
	L S	42,03	45,53	50,51	53,56	47,77	51,32
	L BOW–w	43,06	46,50	49,49	52,60	48,18	51,70
	L BOW–w S	42,03	45,53	50,51	53,56	47,77	51,32
	L BOW–wp	43,06	46,50	49,49	52,60	**48,18**	**51,70**
	L BOW–wp S	42,03	45,53	50,51	53,56	47,77	51,32
BLC–20	baseSC	50,21	55,13	54,87	58,82	51,67	56,88
	baseWN	49,36	54,35	54,27	58,28	47,08	52,79
	L	51,72	56,48	**58,47**	**62,11**	50,00	55,39
	L S	50,43	55,32	55,47	59,38	53,33	58,36
	L BOW–w	51,93	56,67	**58,77**	**62,38**	50,00	55,39
	L BOW–w S	50,43	55,32	55,47	59,38	53,33	58,36
	L BOW–wp	51,93	56,67	**58,77**	**62,38**	50,00	55,39
	L BOW–wp S	50,43	55,32	55,47	59,37	53,33	58,36
BLC–50	baseSC	49,78	54,93	55,96	60,06	52,50	57,62
	baseWN	49,57	54,74	54,60	58,82	47,92	53,53
	L	51,72	56,67	59,28	63,06	50,83	56,13
	L S	50,43	55,51	56,56	60,60	54,17	59,11
	L BOW–w	51,94	56,87	**59,28**	**63,06**	50,83	56,13
	L BOW–w S	50,43	55,51	56,56	60,60	54,17	59,11
	L BOW–wp	51,94	56,87	**59,28**	**63,06**	50,83	56,13
	L BOW–wp S	50,43	55,51	56,56	60,60	54,17	59,11
WND	baseSC	84,80	90,33	84,96	92,20	88,10	92,57
	baseWN	83,89	89,75	82,59	90,97	85,12	90,71
	L	74,77	83,95	70,98	84,95	80,36	87,73
	L S	81,46	88,20	77,84	88,51	87,50	92,19
	L BOW–w	69,91	80,85	73,09	86,05	80,36	87,73
	L BOW–w S	79,33	86,85	76,52	87,82	83,33	89,59
	L BOW–wp	80,85	87,81	76,25	87,69	83,33	89,59
	L BOW–wp S	80,85	87,81	76,25	87,69	83,33	89,59
SUMO	baseSC	54,24	60,35	59,69	64,71	55,51	60,97
	baseWN	55,36	61,32	58,91	64,02	53,81	59,48
	L	56,03	61,90	**63,91**	**68,40**	52,54	58,36
	L S	52,68	58,99	58,28	63,47	51,27	57,25
	L BOW–w	55,36	61,32	63,59	68,13	52,97	58,74
	L BOW–w S	52,46	58,80	58,44	63,61	51,27	57,25
	L BOW–wp	55,36	61,32	63,59	68,13	52,97	58,74
	L BOW–wp S	55,36	61,32	63,44	67,99	52,97	58,74
SuperSenses	baseSC	62,79	68,47	76,24	79,07	60,18	67,29
	baseWN	63,47	69,05	75,31	78,25	59,73	66,91
	L	55,48	62,28	68,94	72,64	49,77	58,74
	L S	55,25	62,09	69,10	72,78	49,32	58,36
	L BOW–w	55,71	62,48	68,94	72,64	49,77	58,74
	L BOW–w S	55,25	62,09	69,10	72,78	49,32	58,36
	L BOW–wp	55,71	62,48	68,94	72,64	49,77	58,74
	L BOW–wp S	55,25	62,09	69,10	72,78	49,32	58,36

Tabla 6.5. Resultados para verbos, atributos IXA

uso de contextos mayores se nos presentan como líneas de trabajo futuro que serían interesantes desarrollar.

6.2.4 Curvas de aprendizaje con atributos BASE

Diseñamos un conjuntos de experimentos para evaluar el comportamiento del sistema de desambiguación basado en clases cuando gradualmente se incrementa la cantidad de ejemplos de entrenamiento. Estos experimentos se llevan a cabo para nombres y verbos sobre SE2 y SE3, y ya que la tendencia es muy similar, solo mostramos los resultados para el caso de nombres, en el que el efecto es más acusado.

Para este experimento, el corpus de entrenamiento, SemCor, se divide en fragmentos del 5 % del total de archivos que lo componen. No se fragmenta ningún archivo del corpus, sino que se seleccionan archivos completos aleatoriamente[7] para ir aumentando el corpus de aprendizaje paulatinamente y generar así porciones de entrenamiento correspondientes al 5 %, 10 %, 15 %... del total del corpus[8]. El procedimiento consiste en utilizar la misma configuración de nuestro sistema, para entrenarlo sobre las diferentes porciones del corpus SemCor, y evaluarlo sobre la totalidad de los corpus SE2 y SE3. También comparamos nuestro sistema respecto al *baseline* calculado sobre el mismo fragmento de entrenamiento.

En las figuras 6.1 y 6.2 se presentan las curvas de aprendizaje del sistema evaluado sobre SE2 y SE3. El sistema que utilizamos usa BLC–20 como nivel de abstracción (conjunto de etiquetas que asigna), y conjunto de atributos básicos ampliados con los atributos semánticos construidos según WND, dentro de lo que llamamos atributos BASE en la sección 6.1.1.

Sorprendentemente, sobre SE2 solo se obtiene una mejora del 2 % cuando se aumenta el corpus de entrenamiento desde el 25 % del total hasta el 100 % del corpus SemCor. Sobre SE3, solo se incrementa un 6 % con el uso del total del corpus frente al uso de sólo el 50 %

[7] De esta forma nos aseguramos de que mantenemos la distribución de sentidos en los diferentes subconjuntos de entrenamiento que se generan.

[8] Cada porción que generamos contiene la porción anterior más un 5 % de nuevos archivos del corpus. Por ejemplo, la porción del 25 % contiene los archivos de la porción del 20 %.

del mismo. Estos nos da la idea de que la mayoría del conocimiento requerido por nuestro sistema de WSD basado en clases parece estar contenido en un pequeño fragmento del corpus de entrenamiento. Dicho de otro modo, la cantidad de ejemplos de entrenamiento requeridos por un sistema de desambiguación basado en clases para obtener resultados competitivos se reduce drásticamente.

El mismo análisis lo realizamos seleccionando BLC–50 como clases semánticas para construir los clasificadores. Podemos ver la curva de aprendizaje sobre SE2 en la figura 6.3 y sobre SE3 en 6.4. El comportamiento del sistema frente al incremento de la cantidad de ejemplos de aprendizaje es muy similar al observado para BLC–20. Obviamente se consiguen tasas de acierto mayores, derivados del menor grado de polisemia que ofrece el conjunto BLC–50.

Las figuras 6.5 y 6.6 muestran de nuevo las curvas de aprendizaje sobre SE2 y SE3, con la misma configuración del sistema y utilizando SuperSenses como conjunto de clases para construir los clasificadores.

Observamos la misma tendencia: sobre SE2 solo se mejora un 2 % cuando se incrementaba el corpus de entrenamiento del 25 % al 100 % del total de SemCor. En el caso de SE3 sólo aumentamos un 3 % pasando de entrenar con el 30 % del corpus a entrenar con la totalidad del SemCor.

En general, en todos los casos, usando BLC–20, BLC–50 o SuperSenses como clases semánticas para el etiquetado, el comportamiento del sistema es muy similar al *baseline* basado en la clase más frecuente. Esto es especialmente interesante si tenemos en cuenta el modo en que está definido este *baseline*, de modo que se asigna la clase más frecuente sobre el corpus para las palabras que no aparecen en los ejemplos de entrenamiento, con lo cual nunca queda ninguna palabra sin etiquetar. Sin esta definición, muchas palabras del conjunto de test quedarían sin etiquetar por dicho *baseline*, sobre todo utilizando fragmentos pequeños de entrenamiento, y con ello la cobertura (y por tanto el valor F1) del *baseline* descendería bruscamente. Otro de los problemas que pueden afectar en gran medida a nuestros clasificadores cuando los entrenamos sobre fragmentos pequeños de corpus, es el reducido número de ocurrencias de que disponen los corpus de evaluación de SE2 y SE3. Con este reducido número de ejemplos de

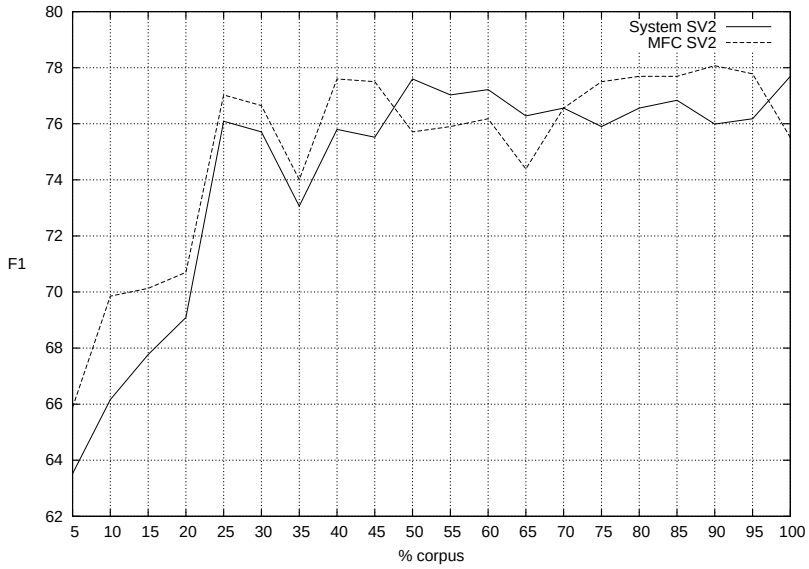


Figura 6.1. Curva de aprendizaje para BLC–20 sobre SE2

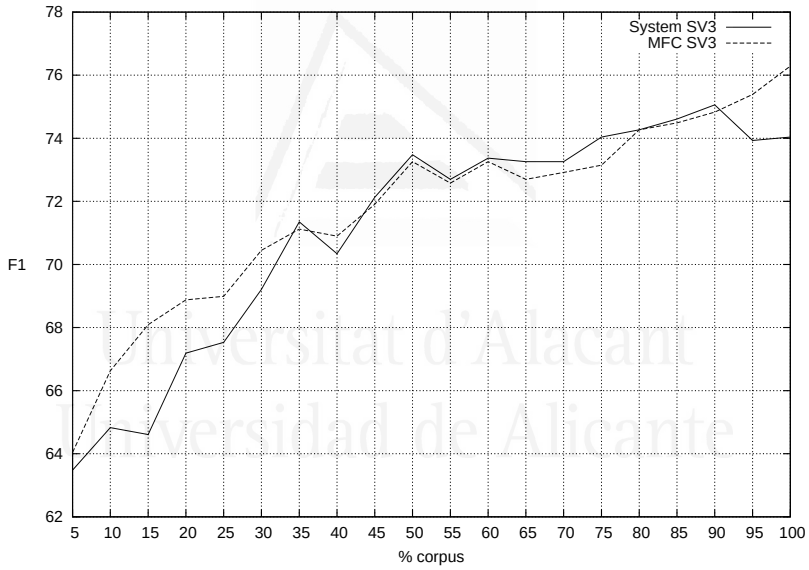


Figura 6.2. Curva de aprendizaje para BLC–20 sobre SE3

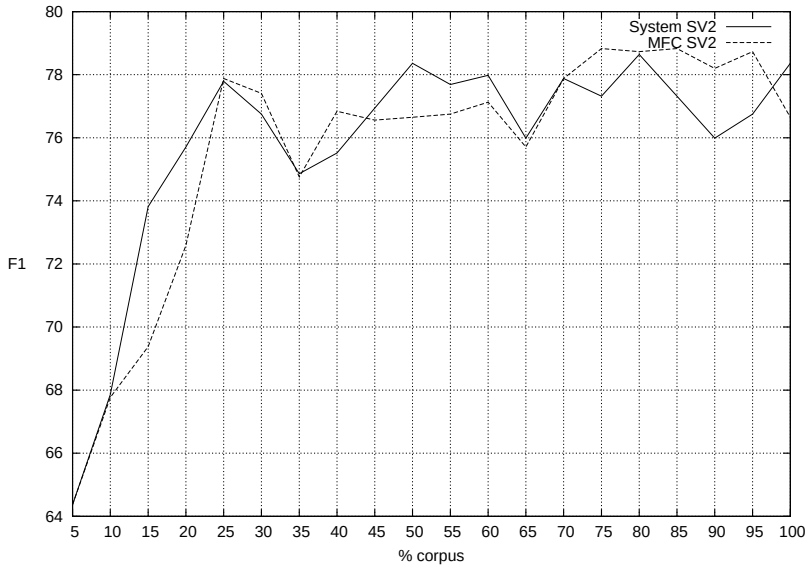


Figura 6.3. Curva de aprendizaje para BLC–50 sobre SE2

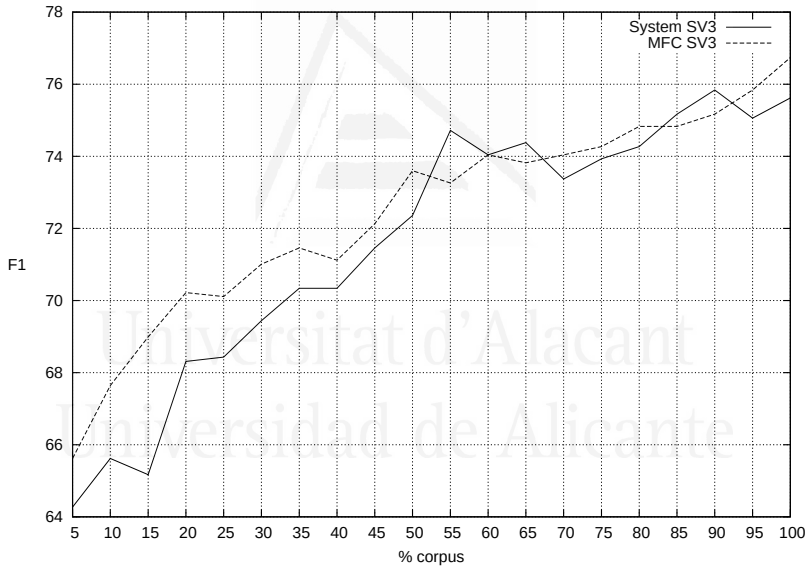


Figura 6.4. Curva de aprendizaje para BLC–50 sobre SE3

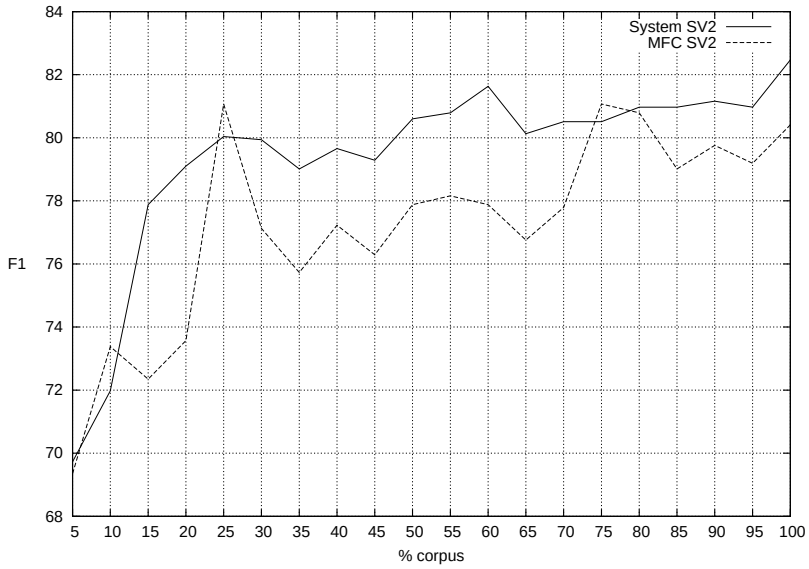


Figura 6.5. Curva de aprendizaje para SuperSenses sobre SE2

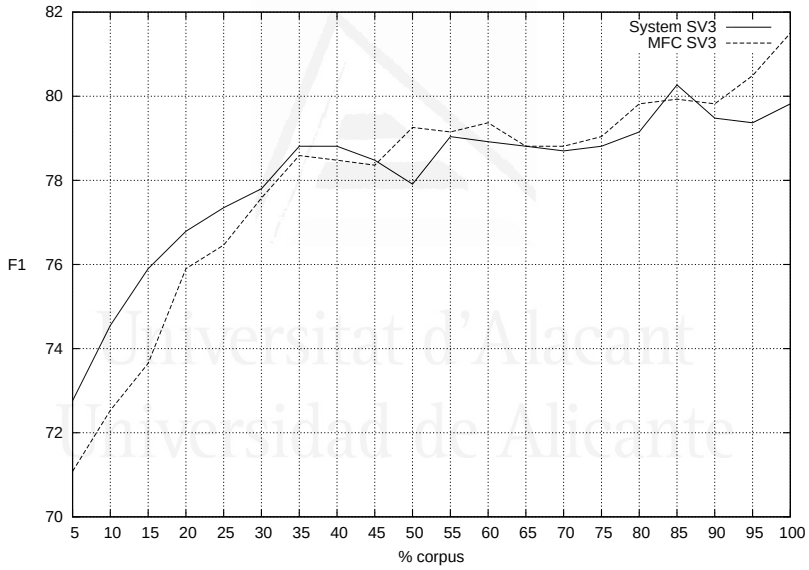


Figura 6.6. Curva de aprendizaje para SuperSenses sobre SE3

evaluación, unos pocos fallos o aciertos pueden hacer variar varios puntos la precisión final del sistema.

6.2.5 Combinación de clasificadores

Como hemos visto, disponemos de diferentes clasificadores, con distinto nivel de abstracción y granularidad, en función de la clase semántica que elijamos para entrenar los clasificadores. Por otra parte, también veremos en la sección 6.3 que podemos obtener clasificadores competitivos a nivel de sentido a partir de nuestros clasificadores semánticos, y siguiendo una heurística muy sencilla[9].

En este experimento nos planteamos la posibilidad de combinar los clasificadores que trabajan en distintos niveles de abstracción, para generar un único clasificador para sentidos de palabras, que aproveche las ventajas que proporcionan los diferentes niveles de abstracción de los clasificadores individuales combinados.

Cada palabra de los corpus de evaluación (SE2 y SE3) puede ser etiquetada con nuestros diferentes clasificadores semánticos: BLC–20, BLC–50, WND, SuperSenses y SUMO (además de su sentido fino). Cada uno de estos clasificadores semánticos utiliza el conjunto de atributos básicos definido en la sección 6.1.1 y los atributos semánticos construidos según WND.

Una vez etiquetada la palabra con las diferentes clases semánticas, realizamos el mapeo de cada una de estas clases al sentido de WordNet correspondiente (primer sentido de la palabra según el *ranking* de WordNet que pertenezca a la clase). Este sentido recibe un voto, y se repite el proceso con el resto de clases, de modo que cada clase vote por un sentido concreto. Al finalizar, aquel sentido que haya obtenido más votos es el asignado. Podemos obtener este clasificador de sentidos a partir de los conjuntos de clases semánticos que deseemos, combinándolos del modo que hemos descrito.

Realizamos varios experimentos, combinando diferentes clasificadores en cada caso. En aquellas situaciones donde únicamente tenemos en cuenta un tipo de clase semántica, no existe combinación,

[9] Podemos etiquetar una palabra con su sentido a partir de su clase semántica seleccionando el primer sentido de la palabra que pertenezca a dicha clase semántica

se trata de un mapeo simple entre clases semánticas y sentidos, del mismo modo que veremos en nuestros experimentos de la sección 6.3. En la tabla 6.6 vemos la medida F1 de los resultados sobre SE2 y SE3, únicamente para nombres, de las diferentes combinaciones de clasificadores semánticos evaluadas a nivel de sentido.

Clasificadores	**SE2**		**SE3**	
	Poli.	**Todas**	**Poli.**	**Todas**
Sentido	61,13	71,20	65,45	73,15
BLC–20	64,45	73,50	63,94	72,30
BLC–50	64,62	73,80	63,94	72,30
WND	60,32	69,50	63,78	71,90
SuperSense	63,14	73,40	64,24	72,60
SUMO	63,75	73,10	63,84	72,10
BLC–20 + BLC–50	64,78	73,83	60,84	69,48
BLC–20 + BLC50 + WND + SS + SUMO	64,26	73,44	62,14	70,50
BLC–20 + BLC50 + WND + SS + SUMO + Sentido	63,47	72,85	63,29	71,40

Tabla 6.6. Valor F1 para nombres de la combinación de clasificadores semánticos, evaluando a nivel de sentidos de WordNet.

Como vemos, nuestros clasificadores semánticos mantienen el poder discriminatorio cuando se transforman a nivel de sentido, tal y como demuestran los buenos resultados del mapeo de clase semántica a sentido, que superan a los resultados por sentidos en la mayoría de los casos. En cuanto a la combinación de clasificadores, en el caso de SE2 se obtiene mejora con las diferentes combinaciones que hemos probado. Es interesante resaltar el caso de la combinación BLC–20 + BLC–50, que es la que mejor resultados consigue. Estos dos conjuntos de clases representan niveles de abstracción diferentes, y al estar generados con la misma técnica, posiblemente la información que contienen ambos sea complementaria y coherente, lo que produce que dicha combinación obtenga tan buenos resultados.

En el caso de SE3 no observamos ninguna mejora con la combinación de clasificadores, ni siquiera con la correspondencia directa de clases semánticas a sentidos. Posiblemente la distribución tanto de sentidos como de clases semánticas en dicho corpus sea muy diferente a la de SemCor, el corpus de entrenamiento, lo que pueda ser, en parte, la causa de la disparidad de resultados.

6.2.6 Conclusiones

Después de nuestra experimentación con la nueva aproximación basada en clases, y con diferentes conjuntos de clases semánticas, podemos concluir que, nuestros conjuntos de BLC obtenidos automáticamente agrupan los sentidos de las palabras en un nivel de abstracción medio, adecuado para su uso en un sistema de desambiguación basado en clases semánticas. Por otra parte, comprobamos que las clases semánticas facilitan un conjunto de atributos muy ricos, que mejoran los resultados de un sistema WSD basado en clases semánticas. También mostramos la reducción del número de ejemplos de entrenamiento necesarios en la aproximación a WSD basada en clases, lo que puede contribuir en gran manera a la construcción de recursos de calidad anotados automáticamente.

En general, los resultados conseguidos usando BLC–20 no son muy diferentes de los obtenidos por BLC–50. Teniendo en cuenta la diferencia de granularidad o abstracción de ambos conjuntos, podemos considerar que es posible seleccionar un nivel intermedio y rico de abstracción para la tarea de desambiguación semántica, sin sufrir un descenso importante en el rendimiento del sistema. Además, sería posible desarrollar un sistema que utilizara las clases BLC para etiquetar los nombres (BLC–20 con 558 clases y un acierto de 75 %) y el conjunto de clases SuperSenses para etiquetar los verbos (14 clases y un acierto del 75 %).

En cuanto a los atributos, observamos que aquellos atributos tradicionalmente utilizados en aproximaciones basadas en sentidos a WSD, no han funcionado bien con la nueva aproximación basada en clases semánticas. Por lo tanto, hemos enriquecido los atributos con nueva información, con los atributos semánticos que hemos descrito, y además ajustando los atributos tradicionales de los sistemas basados en sentidos. En concreto se trata de considerar contextos más grandes alrededor de la palabra objetivo para extraer información relevante, y generalizar los bigramas y trigramas para que no dependan de la palabra en cuestión, sino de la clase semántica de dicha palabra.

6.3 Comparación con sistemas SensEval

Hasta el momento, hemos evaluado nuestro sistema sobre diferentes corpus, pertenecientes a tareas de SE2, SE3 y SEM1. Sin embargo, la única evaluación que hacemos es la de nuestro sistema aisladamente, con diversas configuraciones: diferentes conjuntos de clases semánticas, diferentes conjuntos de atributos o diferentes ajustes para el motor SVM, entre otros. En esta sección comparamos los resultados de nuestro sistema con los resultados de los participantes en las tareas *English All Words* de SE2 y SE3.

Para realizar esta comparación son necesarios algunos ajustes. En primer lugar, nuestro sistema trabaja con clases semánticas, y los sistemas participantes en SE lo hacen a nivel de sentido. Por tanto es necesario realizar un *mapping* o correspondencia entre diferentes niveles de abstracción para poder realizar una evaluación equitativa. Esta evaluación la realizamos a dos niveles: sentido y clase semántica, y en cada caso realizamos una correspondencia concreta que detallaremos en la sección correspondiente. En segundo lugar, nuestro sistema únicamente trabaja para nombres y para verbos, mientras que los resultados publicados de los sistemas de SE incluyen también los resultados para adjetivos y adverbios. Por tanto, también es necesario lanzar de nuevo los *scripts* proporcionados por los organizadores de SE sobre las salidas detalladas de los participantes, para obtener los resultados aislados de dichos sistemas para nombres y verbos.

La configuración del sistema que utilizamos para realizar esta evaluación es aquella que utilizaba el conjunto de atributos BASE descrito en la sección 6.1.1, y diferentes niveles de abstracción (desde sentidos a SuperSenses) para construir los clasificadores. Pasamos a describir a continuación las dos evaluaciones.

6.3.1 Evaluación a nivel de sentidos

El principal objetivo de esta evaluación es verificar si el nivel de abstracción de nuestros clasificadores basados en clases semánticas mantiene la suficiente capacidad discriminatoria cuando se evalúa sobre sentidos. Por tanto, nuestros sistemas basados en clases semánticas se adaptan a sentidos siguiendo una aproximación muy simple.

En principio, para una palabra en concreto, nuestro sistema le asigna la clase semántica correspondiente. La aproximación consiste en asignarle a dicha palabra el primer sentido de acuerdo al orden que establece WordNet[10], que pertenezca a dicha clase semántica. De este modo transformamos la clase semántica original en un sentido concreto según WordNet.

Los resultados de este primer experimento sobre el corpus de SE2 se muestran en la tabla 6.7. Aparecen los cinco mejores resultados de los participantes en SE2, y el peor resultado. Todos nuestros sistemas poseen el prefijo "SVM–", junto con el sufijo de la clase semántica utilizada para construir los clasificadores semánticos. Por ejemplo *SVM–semBLC20* indica el sistema que utiliza BLC–20 como repositorio de clases semánticas para construir los clasificadores, es decir, es la clase que define la granularidad del clasificador[11]. También se incluyen dos *baselines* resaltados en cursiva: el sentido más frecuente según WordNet, o primer sentido de la palabra (*base–WordNet*), y el sentido más frecuente según SemCor (*base–SemCor*). De hecho, estos dos *baselines* son parecidos pero no idénticos, debido a que la ordenación de sentidos en WordNet está hecha en función de la frecuencia de dichos sentidos calculada sobre SemCor, pero también sobre otros corpus de palabras anotadas con información de sentidos.

Tanto para nombres como para verbos, nuestro sistema supera ambos *baselines*. Este tipo de *baselines* son muy competitivos en tareas de WSD, y es muy difícil superarlos incluso ligeramente, tal y como se describe en (McCarthy *et al.*, 2004). Como era de esperar, el uso de diferentes conjuntos de atributos, derivados de la selección de un tipo de clases semánticas, produce resultados ligeramente distintos. Sin tener en cuenta este tipo de diferencias, nuestro sistema basado en clases ocuparía la tercera posición evaluado a nivel de sentidos frente al resto de participantes en SE2.

El mismo experimento lo realizamos sobre SE3, evaluando frente a los participantes en dicha competición, y mostramos los resultados

[10] Recordemos que el orden de los sentidos en WordNet se corresponde con la frecuencia de uso de dichos sentidos. Es decir, al sentido más usado de una palabra se le asigna el sentido 1 en WordNet, y sucesivamente con el resto.

[11] Para cada clase semántica, se ha elegido el experimento que utiliza el conjunto de atributos BASE, eligiendo WND como clases para generar los atributos semánticos. Los resultados de estos experimentos se pueden ver en la tabla 6.2

Clase → Sentido en SE2			
Nombres		**Verbos**	
Sistema	**F1**	**Sistema**	**F1**
SMUam	77,80	SMUaw	52,70
AVe-Antwerp	74,40	AVe-antwerp	47,90
SVM-BLC–50	73,80	SVM-BLC–50	47,10
SVM-BLC–20	73,50	SVM-BLC–20	46,10
SVM-SuperSense	73,40	SVM-Sentido	45,70
SVM-SUMO	73,10	LIA-Sinequa	44,80
SVM-Sentido	71,20	*base-SemCor*	44,80
base-WordNet	70,10	SVM-WND	44,50
base-SemCor	70,00	SVM-SuperSense	44,40
LIA-Sinequa	70,00	*base-WordNet*	43,80
SVM-WND	69,50	SVM-SUMO	73,40
...	...	...	...
Woody-IIT1	6,00	frglc	2,20

Tabla 6.7. F1 de los resultados a nivel de sentidos sobre SE2

en la tabla 6.8. En este caso, nuestro sistema basado en clases supera claramente a los *baselines*, consiguiendo además el mejor resultado para nombres, y el mejor segundo resultado para verbos. Es interesante destacar que para nombres, el mejor sistema participante de SE3 (*GAMB–AW*) no consigue alcanzar el *baseline* basado en SemCor.

Clase → Sentido en SE3			
Nombres		**Verbos**	
Sistema	**F1**	**Sistema**	**F1**
SVM-Sentido	73,15	GAMBL-AW	59,30
SVM-SuperSense	72,60	SVM-Sentido	57,00
SVM-BLC–50	72,30	UNT-aw	56,40
SVM-BLC–20	72,30	SVM-BLC–50	55,70
base-SemCor	72,30	SVM-BLC–20	55,30
SVM–SUMO	72,10	Meaning-allwords	55,20
SVM-WND	71,90	SVM-SUMO	54,50
GAMBL-AW	70,80	kuaw	54,40
base-WordNet	70,70	R2D2	54,40
kuaw	70,60	SVM-SuperSense	54,10
UNT-aw	69,60	UPV	53,40
Meaning-allwords	69,40	Meaning-allwordsII	53,10
LCCaw	69,30	SVM-WND	52,90
R2D2	68,10	*base-SemCor*	52,90
Meaning-allwordsII	68,10	*base-WordNet*	52,80
...	...	...	...
DLSI-UA-dom	36,40	autoPSNVs	9,90

Tabla 6.8. Resultados a nivel de sentidos sobre SE3

Ambos experimentos muestran que los clasificadores basados en clases semánticas son competitivos cuando se transforman a sentidos y se evalúan a este nivel. Estos clasificadores funcionan por encima de los resultados de los *baselines* basados en asignar el sentido más frecuente calculado sobre SemCor o WordNet, y obtienen una posición muy elevada en la evaluación según sentidos, en comparación con el resto de participantes en SE2 y SE3.

6.3.2 Evaluación a nivel de clase semántica

En esta sección describimos los experimentos que exploran el rendimiento evaluado sobre clases semánticas de los sistemas participantes en SE y que funcionan a nivel de sentido de la palabra. Por tanto la correspondencia necesaria en este caso es para las salidas concretas de los sistemas participantes en dicha competición. Transformamos los sentidos a clases semánticas directamente. Las salidas de nuestros sistemas quedan tal cual, a nivel de clase semántica. En función del tipo de clase semántica que elijamos, podemos realizar una evaluación distinta. En concreto realizamos evaluaciones de este tipo según las clases BLC–20, BLC–50, WND, SUMO y SuperSenses.

La tabla 6.9 presenta los resultados ordenados por el valor F1 de los participantes en SE2. De nuevo, en cursiva aparecen los resultados de los *baselines* basados en la clase más frecuente según WordNet (*base–WordNet*, clase correspondiente al primer sentido de WordNet de cada palabra) o según SemCor (*base–SemCor*, clase más frecuente para cada palabra entre todas las ocurrencias de dicha palabra en SemCor).

En general sobre SE2, la mayoría de nuestros sistemas basados en clases superan a los dos baselines (*base–WordNet* y *base–SemCor*) tanto para nombres como para verbos. El mejor sistema de este tipo es *SVM–semWND*, es decir, aquel que utiliza WND para construir los atributos semánticos. En el caso de nombres, este sistema ocupa la primera posición según SuperSenses, la segunda en caso de SUMO, WND y BLC–20, y la tercera posición en la evaluación para BLC–50. En el caso de verbos, el sistema alcanza la segunda mejor posición para BLC–20, BLC–50 y WND, la tercera para SuperSenses, y la quinta en el caso de SUMO, superando en todos los casos a los

Nombres		Verbos	
Sistema	**F1**	**Sistema**	**F1**
Sentido → BLC–20			
SMUaw	78,72	SMUaw	61,22
SVM-semWND	77,88	SVM-semWND	58,61
SVM-semBLC50	77,79	LIA-Sinequa	57,42
SVM-semBLC20	77,79	AVe-antwerp	57,28
SVM-semSUMO	77,60	SVM-semSUMO	57,06
AVe-antwerp	76,71	SVM-semBLC20	56,87
base-SemCor	*75,71*	SVM-semSS	56,29
SVM-semSS	75,14	SVM-semBLC50	55,90
base-WordNet	*74,29*	*base-SemCor*	*55,13*
LIA-Sinequa	73,39	*base-WordNet*	*54,16*
Sentido → BLC–50			
SVM-semSUMO	79,58	SMUaw	61,61
SMUaw	79,01	SVM-semWND	59,77
SVM-semWND	78,92	LIA-Sinequa	57,81
SVM-semBLC20	78,45	AVe-Antwerp	57,67
AVe-antwerp	77,57	SVM-semSS	57,40
SVM-semBLC50	76,65	SVM-semBLC20	57,45
base-SemCor	*76,65*	SVM-semSUMO	57,06
SVM-semSS	75,52	SVM-semBLC50	56,67
base-WordNet	*75,24*	*base-SemCor*	*54,93*
LIA-Sinequa	74,53	*base-WordNet*	*54,55*
Sentido → WND			
SMUaw	88,80	SMUaw	91,16
base-SemCor	*86,11*	SVM-semWND	90,52
SVM-semWND	86,01	SVM-semSUMO	90,52
AVe-Antwerp	87,30	*base-SemCor*	*90,33*
base-WordNet	*85,82*	SVM-semBLC20	90,14
LIA-Sinequa	84,85	SVM-semBLC50	90,14
SVM-semSUMO	83,85	LIA-Sinequa	89,82
SVM-semSS	83,08	SVM-semSS	89,75
SVM-semBLC50	82,33	*base-WordNet*	*89,75*
SVM-semBLC20	81,85	AVe-Antwerp	89,74
Sentido → SUMO			
SMUaw	79,30	SMUaw	68,22
SVM-semWND	77,88	LIA-Sinequa	64,79
SVM-semSUMO	77,88	AVe-Antwerp	62,56
SVM-semSS	77,50	SVM-semBLC50	62,09
base-SemCor	*76,09*	SVM-semWND	61,51
AVe-Antwerp	75,94	*base-SemCor*	*61,33*
SVM-semBLC20	75,52	SVM-semBLC20	61,12
SVM-semBLC50	75,52	SVM-semSUMO	60,74
LIA-Sinequa	*74,92*	*base-WordNet*	*60,35*
base-WordNet	*71,74*	SVM-semSS	59,96
Sentido → SuperSenses			
SVM-semWND	82,47	SMUaw	73,47
SMUaw	81,21	LIA-Sinequa	72,74
SVM-semBLC20	81,07	SVM-semWND	71,76
AVe-Antwerp	80,75	SVM-semSUMO	70,21
SVM-semSUMO	80,51	AVe-Antwerp	69,31
base-SemCor	*80,41*	SVM-semBLC20	69,25
SVM-semSS	80,32	SVM-semBLC50	69,25
SVM-semBLC50	80,22	SVM-semSS	69,25
LIA-Sinequa	79,58	*base-WordNet*	*69,05*
base-WordNet	*78,16*	*base-SemCor*	*68,47*

Tabla 6.9. Resultados para el mapeo de sentidos a clases semánticas sobre SE2

dos *baselines*. Otras configuraciones de nuestro sistema basado en clases también obtienen buenos resultados puntualmente en algunos niveles de abstracción. Por ejemplo, *SVM–semSUMO* consigue el mejor resultado para nombres evaluando sobre BLC–50.

La tabla 6.10 presenta los resultados ordenados por el valor F1 de los mejores resultados entre los participantes en SE3 y nuestros sistemas basados en clases semánticas, evaluando a diferentes niveles de abstracción. En cursiva aparecen los resultados para los dos *baselines*, el que usa SemCor y el que usa WordNet para obtener la clase más frecuente para cada palabra.

Analizando los resultados sobre SE3, algunos de los sistemas basados en clases semánticas superan a los *baselines*, para nombres y para verbos. Sin embargo, esto no se produce en todos los casos, y concretamente el *baseline* basado en SemCor para nombres *base–SemCor* parece muy difícil de superar. Es interesante el hecho de que ninguno de los sistemas participantes en SE3 sea capaz de alcanzar dicho *baseline* en ninguno de los diferentes niveles de abstracción. Solo algunos de nuestros sistemas basados en clases son capaces de superarlo en algún caso. Por ejemplo, *SVM–semBLC20* obtiene la primera posición para nombres sobre BLC–20 y BLC–50. Para verbos, de nuevo *SVM–semWND* funciona especialmente bien, obteniendo la primera posición sobre WND y la segunda posición para BLC–20, BLC–50 y SUMO.

Remarcamos el hecho de que nuestros sistemas alcancen buenos resultados haciendo uso del mismo conjunto de clases semánticas, tanto para construir los clasificadores, como para obtener los atributos semánticos. Por ejemplos en el caso de BLC–20, el mejor sistema es el *SVM–semBLC20* (utiliza BLC–20 para generar los atributos semánticos), para WND el mejor es *SVM–semWND*, y para SUMO el mejor es el *SVM–semSUMO*.

En líneas generales, vemos que nuestro sistema basado en clases semánticas obtiene mejores resultados que los sistemas basados en sentidos de palabra participantes en SE2 y SE3. Además esta comparación se ha realizado evaluando todos los sistemas a nivel de sentido primero, y a nivel de clase semántica después.

En el caso de la evaluación para sentidos, el hecho de que nuestros sistemas basados en clases semánticas y *mapeados* a sentidos

Nombres		Verbos	
Sistema	**F1**	**Sistema**	**F1**
Sentido → BLC–20			
SVM-semBLC20	76,52	GAMBL-AW	63,56
base-SemCor	*76,29*	SVM-semSS	61,29
SVM-semBLC50	75,73	SVM-semSUMO	61,15
GAMBL-AW	74,77	SVM-semWND	60,88
kuaw	74,69	kuaw	60,66
LCCaw	74,44	SVM-semBLC50	60,60
UNTaw	74,40	SVM-semBLC20	59,92
SVM-semWND	74,24	R2D2	59,79
base-WordNet	*74,16*	UNTaw	59,73
SVM-semSS	73,82	Meaning-allwords	59,37
SVM-semSUMO	73,71	*base-SemCor*	*58,82*
Meaning-allwords	73,11	*base-WordNet*	*58,28*
Sentido → BLC–50			
SVM-semBLC20	76,85	GAMBL-AW	64,38
SVM-semBLC50	76,74	SVM-semWND	62,38
base-SemCor	*76,74*	SVM-semSS	61,83
GAMBL-AW	75,56	SVM-semSUMO	61,83
SVM-semSUMO	75,51	SVM-semBLC20	61,29
kuaw	75,25	kuaw	61,22
SVM-semWND	74,83	SVM-semBLC50	61,01
LCCaw	74,78	R2D2	60,35
UNTaw	74,73	UNTaw	60,27
SVM-semSS	74,61	Meaning-allwords	60,19
base-WordNet	*74,49*	*base-SemCor*	*60,06*
R2D2	73,93	*base-WordNet*	*58,82*
Sentido → WND			
base-SemCor	*83,80*	SVM-semWND	92,20
SVM-semWND	83,71	*base-SemCor*	*92,20*
UNTaw	83,62	UNTaw	91,37
kuaw	81,78	GAMBL-AW	91,01
GAMBL-AW	81,53	*base-WordNet*	*90,83*
base-WordNet	*81,46*	R2D2	90,52
SVM-semSUMO	81,24	Meaning-simple	90,50
SVM-semBLC20	80,79	kuaw	90,44
LCCaw	80,64	SVM-semBLC20	90,42
Meaning-allwords	80,50	SVM-semBLC50	90,15
SVM-semBLC50	80,50	SVM-semSUMO	89,88
SVM-semSS	78,31	SVM-semSS	88,78
Sentido → SUMO			
base-SemCor	*79,55*	GAMBL-AW	68,77
SVM-semSUMO	78,76	SVM-semBLC50	66,48
kuaw	78,18	SVM-semWND	66,35
LCCaw	77,54	SVM-semBLC20	66,07
SVM-semWND	77,42	UNTaw	66,03
UNTaw	77,32	kuaw	65,93
SVM-semBLC50	77,19	Meaning-allwords	65,43
GAMBL-AW	77,14	SVM-semSUMO	64,98
SVM-semSS	76,97	upv-eaw2	64,92
base-WordNet	*76,97*	SVM-semSS	64,71
Meaning-allwords	76,75	*base-SemCor*	*64,71*
SVM-semBLC20	76,74	*base-WordNet*	*64,02*
Sentido → SuperSenses			
SVM-semBLC50	81,73	SVM-semWND	79,75
base-SemCor	*81,50*	GAMBL-AW	79,4
SVM-semBLC20	81,39	*base-SemCor*	*79,07*
SVM-semSUMO	81,05	*base-WordNet*	*78,25*
kuaw	79,89	Meaning-allwords	78,14
SVM-semWND	79,82	SVM-semSS	77,84
UNTaw	79,71	Meaning-simple	77,72
GAMBL-AW	79,62	SVM-semBLC20	77,70
upv-eaw2	79,27	SVM-semBLC50	77,70
upv-eaw	78,42	SVM-semSUMO	77,70
base-WordNet	*78,25*	kuaw	77,53
SVM-semSS	76,46	upv-eaw2	77,21

Tabla 6.10. Resultados para el mapeo de sentidos a clases semánticas sobre SE3

obtengan buenos resultados nos indica que estos clasificadores, y las clases semánticas que usan, son capaces de mantener un nivel de discriminación a nivel sentidos detallados, a pesar del incremento de abstracción que supone el uso de dichas clases semánticas. Dicho de otro modo, en el proceso de transformación de sentido a clase y de clase a sentido, no perdemos mucha información.

En el caso de la evaluación para clases semánticas, obtenemos una de las conclusiones que más apoyan nuestra aproximación. Como vemos, con nuestros modelos semánticos superamos en la mayoría de los casos a los mejores participantes en SE según sentidos y transformados a clases semánticas. Por tanto, los buenos resultados que obtienen nuestros clasificadores semánticos basados en clases no son producto únicamente de la reducción de polisemia frente al uso de sentidos detallados, sino que, de hecho, nuestros modelos semánticos aprenden de los sentidos agrupados bajo una misma clase semántica.

7. Conclusiones y Trabajo Futuro

En este trabajo hemos presentado un sistema de desambiguación semántica basado en clases semánticas. Por clase semántica entendemos cualquier agrupación de conceptos con coherencia semántica. Además de clases semánticas reciben el nombre de dominios o tópicos. Ejemplos de clases semánticas pueden ser MEDICINA, DEPORTE, EDIFICIO o ACTO. Hemos hecho uso de este tipo de clases para disponer de un nivel de abstracción superior que el que ofrecen los sentidos de las palabras, y ser capaces de crear clasificadores más generales que se especialicen en estas clases semánticas. Además, mediante las clases semánticas también somos capaces de obtener rasgos de caracterización importantes para asignarle la interpretación correcta a una palabra ambigua.

Nuestra aproximación sigue la tendencia del cambio en el punto de vista consistente en generar clasificadores para cada clase semántica, en lugar de para cada palabra, como se ha hecho tradicionalmente. Hemos demostrado empíricamente que el uso de las clases semánticas supone varias ventajas:

- Un mayor nivel de abstracción que el de sentidos, manteniendo el poder discriminatorio, que puede facilitar su integración en otras tareas, ya que solo deben considerarse un menor número de conceptos.
- Aumento en el número de ejemplos de entrenamiento por clasificador, debido a que cada clasificador puede disponer de ejemplos sentidos de palabras distintos.
- Mejora de los resultados absolutos ya que se reduce la polisemia.
- Una mayor robustez del sistema de desambiguación e independencia respecto al dominio. Como hemos visto en nuestra participación en SemEval–2, donde la evaluación se ha realizado sobre un dominio

específico, con nuestro sistema basado en clases y entrenado sobre el corpus general SemCor alcanzamos el mismo resultado que el *baseline*, el cual ha sido difícil de superar en general.

7.1 Conclusiones generales

Con el uso de clases semánticas, los principales problemas que presentan las aproximaciones tradicionales basadas en sentidos pueden ser tratados de forma satisfactoria. Seleccionamos varios conjuntos de clases semánticas que se han utilizado previamente. En concreto, hemos hecho uso de tres conjuntos de clases semánticas: WordNet Domains, las clases de la ontología SUMO, y los ficheros lexicográficos de WordNet o SuperSenses. Cada uno de ellos contiene un conjunto de clases muy distinto, con una metodología de creación muy diferente y proporciona un nivel de abstracción totalmente distinto. En principio, con el uso de estas clases para el entrenamiento de modelos de WSD, sería difícil extraer conclusiones sobre el uso de diversos niveles de abstracción, precisamente por el hecho de que cada conjunto ha sido creado a su modo, y las diferencias obtenidas podrían residir más en la naturaleza de las clases que en su grado de abstracción. Además, estos lexicones no han sido creados específicamente para ser integrados en un sistema de desambiguación semántica y, por tanto, puede que no sean los más apropiados.

Por estas razones decidimos crear un método de selección de clases semánticas que funciona de forma automática sobre WordNet. Por los criterios que hemos seguido para realizar esta selección, nuestros conjuntos creados pertenecen al tipo BLC (Basic Level Concepts). Mediante varios parámetros somos capaces de obtener conjuntos con distinto nivel de abstracción, pero todos con la misma naturaleza y origen. De este modo, su evaluación en una arquitectura de WSD sí que nos proporcionará conclusiones importantes sobre el uso de uno u otro grado de abstracción.

Nuestro método de selección se basa en recorrer la jerarquía de WordNet, seleccionando aquellos *synsets* más importantes. Su importancia se mide a través del número de relaciones de los *synsets* o de la frecuencia de las palabras contenidas en ellos. Además po-

demos controlar el nivel de abstracción de los conjuntos creados a través de un parámetro de filtrado. En el caso del criterio centrado en el número de relaciones, hemos considerado dos variantes para realizar el cálculo del número de relaciones por synset: emplear todas las relaciones representadas en WordNet, o solo las de hiperonimia. Por otro lado, para computar la frecuencia de un synset a partir de las palabras que contiene, también hemos considerado dos medidas: la frecuencia de palabra en SemCor (y realizar la suma de las frecuencias para obtener la del synset), o la frecuencia del synset contenida en WordNet. Aunque hemos generado diversos conjuntos de BLC, los que posteriormente más hemos empleado a lo largo de este trabajo son los que se generan teniendo en cuenta el número de relaciones totales de cada synset.

Una vez generados los distintos conjuntos BLC, comprobamos su coherencia, robustez y adecuación a la tarea de WSD. Realizamos una primera evaluación de la calidad de los conjuntos de clases semánticas BLC, simulando la tarea de desambiguación semántica sobre el corpus de SensEval–3. Trabajamos con nuestras clases semánticas en lugar de sentidos, y haciendo uso de una heurístico muy sencilla basado en asignar la clase más frecuente sobre el corpus SemCor para cada palabra. En esta primera evaluación, no realizamos ningún otro tipo de aprendizaje automático[1]. Mediante esta aproximación hemos obtenido unos resultados muy altos, a pesar de lo simple de nuestra aproximación, lo cual nos indica la robustez y calidad de nuestros conjuntos de clases semánticas. En general, se obtienen mejores resultados que empleando otros conjuntos de clases generados en otros proyectos, como MEANING o BALKANET. Además, nuestros mejores resultados se obtienen mediante los BLC creados utilizando el criterio del número de relaciones por *synset*, en lugar de la frecuencia de las palabras. A partir de las primeras evaluaciones descritas en el párrafo anterior, hemos podido concluir que, con nuestras clases semánticas (BLC), disponemos de unos conjuntos con un nivel de abstracción intermedio, potencialmente adecuados para ser empleados en el aprendizaje de clasificadores semánticos para la tarea de desambiguación semántica de las palabras.

[1] En realidad aprender cuál es la clase más frecuentemente asociada ya implica cierto aprendizaje.

Por otra parte, también hemos demostrado empíricamente que estas clases semánticas son de utilidad cuando se integran en un sistema de recuperación de información tradicional. Participamos en la tarea de recuperación de información *Robust WSD*, dentro de la competición CLEF–09. En concreto, empleamos nuestras clases semánticas para representar tanto los documentos como las consultas, a través de sus tópicos o dominios. La integración de esta información en el flujo del sistema de recuperación de información la hemos hecho en la fase final. Una vez que se dispone del primer *ranking* de documentos, utilizamos los vectores semánticos de los documentos y de la consulta para reordenar dicha lista y, posteriormente, darle mayor puntuación a aquellos documentos más relacionados semánticamente con la pregunta.

El siguiente paso ha sido definir y desarrollar un sistema de desambiguación semántica basado en aprendizaje automático supervisado y con una arquitectura centrada en clases semánticas. Hemos hecho uso de las clases semánticas BLC generadas con nuestro método, y del resto de conjuntos de clases predefinidos: SUMO, WordNet Domains y SuperSenses. Mediante un motor de Máquinas de Soporte Vectorial (*SVM*) hemos generado un conjunto de clasificadores, uno por clase semántica, haciendo uso de un conjunto de atributos para representar los ejemplos de entrenamiento extraídos del corpus SemCor. Posteriormente, mediante estos clasificadores podemos asignar clases semánticas apropiadas a cada palabra. Para implementar esta aproximación, hemos necesitado una arquitectura distinta de la usada tradicionalmente en los sistemas de WSD basados en sentidos, mediante la cual hemos sido capaces de explorar las ventajas de las clases semánticas: reducción de polisemia, aumento del número de ejemplos de entrenamiento, probablemente mejor nivel de abstracción para la tarea de WSD y mayor robustez e independencia respecto al dominio del corpus de aprendizaje.

En el proceso de desarrollo y ajuste del sistema, investigamos sobre el conjunto adecuado de atributos para el proceso de aprendizaje. La conclusión es que los atributos básicos que se utilizan en WSD no son suficientemente adecuados para la tarea de desambiguación basada en clases semánticas. Entre otras conclusiones, hemos comprobado empíricamente que la ventana contextual de dónde se extraen atri-

butos debe ser mayor en la aproximación por clases semánticas que en la centrada en sentidos. Además algunos atributos deben ser generalizados para aumentar la capacidad de abstracción.

Las primeras evaluaciones de nuestro sistema de desambiguación las hemos realizado con un conjunto de pruebas de validación cruzada sobre el propio SemCor, con una configuración inicial básica. Los resultados que hemos obtenido son altos, lo que nos indica la calidad y robustez de nuestra aproximación.

También hemos participado en la competición internacional SemEval-1 con un sistema inicial, haciendo uso de las clases semánticas para enriquecer el conjunto de atributos. Obtenemos un muy buen resultado, superando el *baseline* y alcanzado la quinta mejor posición (la cuarta sin tener en cuenta el sistema de los organizadores que consiguió el mejor resultado).

En la competición internacional SemEval–2 participamos con nuestro sistema basado en clases semánticas de nuevo, consiguiendo la quinta posición. En este caso hemos hecho uso de nuestras clases semánticas para extraer ejemplos monosémicos desde un conjunto de textos no anotados pertenecientes al dominio concreto en el que se centraba la tarea. Aumentando el conjunto de ejemplos de entrenamiento de SemCor con el de ejemplos monosémicos extraídos desde los textos del dominio, nuestro sistema obtiene la quinta mejor posición. Además con nuestro sistema básico, sin ejemplos monosémicos, alcanzamos el valor del *baseline*. Estos resultados ponen de relieve la robustez e independencia de nuestro sistema respecto al dominio de entrenamiento y evaluación. Además vemos que nuestras clases semánticas nos permiten extraer ejemplos monosémicos desde corpus de dominio específico no anotado. El sistema que añade estos nuevos ejemplos para el aprendizaje, además de los extraídos desde SemCor, obtiene una mejora respecto al que no emplea dichos ejemplos monosémicos. Esto es muy interesante, ya que nos puede proporcionar una técnica muy sencilla para adaptar nuestro sistema a cualquier dominio específico que se desee.

Posteriormente, analizamos el rendimiento del sistema usando diversos niveles de abstracción, a través de numerosos experimentos en los que hemos hecho uso de los diferentes conjuntos de clases semánticas. Además, en estos experimentos también evaluamos el

rendimiento de las clases semánticas como atributos para el aprendizaje. La combinación de los conjuntos de clases semánticas para generar los clasificadores, y construir los atributos semánticos, da lugar a una gran batería de experimentos, los cuales hemos evaluado sobre los corpus de evaluación de la tareas *All Words* de SensEval–2, SensEval–3 y SemEval-1. También separamos los experimentos en dos grupos, dependiendo del conjunto de atributos que utilizan como base: los atributos llamados propiamente BASE, seleccionados empíricamente, y los IXA, utilizados por el grupo de investigación del mismo nombre.

Comparados con el *baseline*, en general los resultados son bastante elevados para todos los experimentos. Debemos tener en cuenta que si estos *baseline* basados en el primer sentido de la palabra son altos, los *baseline* basados en la clase más frecuente son todavía más elevados, en todos los casos y sobre los tres corpus. Aun así, nuestros sistemas superan en la mayoría de ocasiones a estos *baseline*. La tendencia en el rendimiento de nuestro sistema es coherente con el *baseline* en cada caso, obteniendo mejores resultados cuando el *baseline* es más alto, y viceversa. En general, los mejores resultados se obtienen sobre SensEval–3, por encima de los obtenidos en SensEval–2 y SemEval–1, con un rendimiento muy similar en ambos casos. Además, como esperábamos, en general el acierto es superior para nombres que para verbos. Este comportamiento se puede justificar observando la naturaleza de los corpus, y la polisemia que presentan los distintos tipos de palabras en cada uno de ellos.

Atendiendo a las diferencias obtenidas en función del conjunto de atributos utilizado, hemos comprobado que los atributos semánticos suelen mejorar el rendimiento del sistema, sobre todo en el caso de los atributos BASE[2]. Además las clases semánticas que proporcionan mejores resultados son WordNet Domains, y también, aunque en menor grado, BLC–20[3]. También podemos concluir, principalmente a través de los experimentos con los atributos IXA, que la información extraída del contexto de la palabra ambigua es muy importante para asignarle su sentido concreto.

[2] Ver capítulo 6
[3] Ver capítulo 4

Son muy interesantes los resultados que obtenemos con el conjunto de clases BLC–20. Los resultados para nombres según BLC–20 (en torno a 75 de valor F1) no son muy inferiores a los obtenidos con BLC–50 (sobre 77 de F1), ni los de SuperSenses (80 de F1). Si tenemos en cuenta que el conjunto nominal para BLC–20 contiene 558 clases, el de BLC–50 contiene 253, y disponemos de 24 clases únicamente para SuperSenses, podemos decir que, el conjunto de BLC-20, proporciona un nivel de abstracción intermedio adecuado para la desambiguación semántica, manteniendo un nivel discriminatorio elevado, sin una gran pérdida en los resultados.

Por otra parte, comprobamos que se constata una enorme reducción en la cantidad de ejemplos de entrenamiento necesarios para obtener buen rendimiento en nuestra arquitectura basada en clases semánticas. No debemos confundir este punto con el del incremento en la cantidad de entrenamiento. De hecho, para cada clasificador en la aproximación semántica, podemos disponer de un mayor número de ejemplos de entrenamiento que siguiendo la aproximación basada en sentidos. Sin embargo, la cantidad necesaria de ejemplos de entrenamiento para obtener buenos resultados con nuestra aproximación basada en clases se reduce. Para llegar a esta conclusión, diseñamos una serie de experimentos para obtener las curvas de aprendizaje de nuestros clasificadores. En general, haciendo uso de nuestras clases semánticas (BLC–20, BLC–50 y SuperSenses), únicamente con el 25 % del corpus SemCor, se obtiene casi el mismo resultado, unos 2 o 3 puntos menos, que haciendo uso del 100 % del corpus. De este modo confirmamos la hipótesis que habíamos hecho en un principio: la cantidad de ejemplos de entrenamiento se reduce en la aproximación basada en clases semánticas.

También hemos comparado nuestro sistema con aquellos que participaron en SensEval–2 y SensEval–3. Ya que nuestro sistema funciona con clases semánticas, y los participantes en SensEval se basan en sentidos finos, hemos realizado dos tipos de comparaciones: a nivel de sentido y a nivel de clase semántica. A nivel de sentido, las salidas de los sistemas de SensEval se dejan tal cual, sin modificar, mientras que las salidas de nuestros clasificadores semánticos se transforman

a sentidos[4]. Con esta evaluación, nuestro sistema ajustado a nivel de sentido ocuparía la tercera mejor posición, tanto para nombres como para verbos, en SensEval–2, la primera posición para nombres en SensEval–3, y la segunda mejor para verbos en SensEval–3. Estos buenos resultados nos confirman que nuestras clases semánticas mantienen un nivel discriminatorio muy alto. Dicho de otro modo, la capacidad de discriminar entre sentidos finos de una palabra no se pierde, a pesar de transformar de sentido a clase semántica para realizar el entrenamiento y, posteriormente, de clase semántica a sentido para realizar la evaluación.

Para realizar la comparación a nivel de clase semántica, transformamos la salida según sentidos de los sistemas de SensEval a clase semántica. Esta comparación la podemos realizar usando distinto grados de abstracción, en función del tipo de clase semántica que seleccionemos: BLC–20, BLC–50, SuperSenses, SUMO y WordNet Domains. Sobre SensEval–2, nuestro sistema ocupa las primeras y segundas mejores posiciones para la mayoría de clases semánticas. Sobre SensEval–3, nuestro sistema todavía funciona mejor, alcanzando en la mayoría de los casos la mejor posición, y la segunda mejor posición en otros muchos casos. En esta evaluación a nivel de clases, estos buenos resultados nos indican que nuestros clasificadores semánticos realmente están aprendiendo mejor que los mejores sistemas basados en sentidos Además el buen rendimiento no es producto únicamente de la reducción de polisemia.

Finalmente, en el anexo B mostramos una serie de experimentos que hemos ido realizando simultáneamente junto con el desarrollo de este trabajo y el resto de pruebas. Hemos decidido colocar esta experimentación en un anexo debido a que en ellos no se han obtenido buenos resultados. En realidad, nos ha servido más bien para descartar ciertas configuraciones de nuestro sistema, y asegurarnos de que el modo en que hemos ajustado nuestro sistema general es el acertado. De cualquier modo, estos experimentos nos han aportado conclusiones importantes en cuanto al número de ejemplos que

[4] Recordemos que a partir de la clase que asignan nuestros clasificadores semánticos a una palabra, seleccionamos el primer sentido siguiendo el ranking de WordNet que pertenezca a la clase semántica etiquetada.

utilizar como positivos y negativos, acerca del tipo de arquitectura seleccionada y sobre los ajustes del motor SVM.

También hemos desarrollado un software que implementa el sistema de desambiguación basado en clases semánticas. El paquete puede ser descargado y usado de forma libre. Este etiquetador ha sido implementado en Python[5], y hace uso de una librería de herramientas de Procesamiento de Lenguaje Natural, NLTK[6]. El sistema puede ser probado también de forma *online* sin necesidad de realizar ningún tipo de descarga e instalación. El método para seleccionar los BLC desde WordNet, así como los distintos conjuntos ya obtenidos pueden ser también descargados. Ambos recursos están disponibles en `http://www.dlsi.ua.es/~ruben`.

7.2 Trabajo futuro

Hay muchas líneas abiertas en las que se podría continuar nuestro trabajo. En primer lugar sería interesante extender la misma que hemos desarrollado, estudiando atributos más complejos para el aprendizaje como, por ejemplo, atributos sintácticos. Con atributos más complejos seguramente el rendimiento del sistema aumentaría. Por otra parte, sería interesante probar distintos algoritmos de aprendizaje automático. Además, también sería interesante incrementar la cantidad de datos de aprendizaje, añadiendo más recursos semánticos, como pueden ser nuevos corpus anotados distintos de SemCor.

Otro punto a desarrollar sería la integración del método algebraico de factorización de matrices *SVD* (del inglés, *Single Value Decomposition*) en el sistema de desambiguación semántica. Mediante esta técnica, sería posible reducir las dimensiones de los vectores de atributos, seleccionando únicamente aquellos más relevantes. Posiblemente esta técnica podría sustituir a nuestro método de filtrado, para eliminar atributos poco importantes.

Otro tema interesante a desarrollar sería la combinación e integración de clasificadores. Hasta donde sabemos, los sistemas que han combinado clasificadores se han hecho siempre usando el mismo nivel

[5] `http://www.python.org`, consultado en julio de 2010.
[6] `http://www.nltk.org`, consultado en julio de 2010.

de abstracción, normalmente trabajando con sentidos. Dicho de otro modo, diferentes clasificadores basados en sentidos se han combinado para asignar un único sentido concreto a una palabra. En nuestro caso podemos aplicar diferentes clasificadores, y cada uno modela un nivel de abstracción distinto según el conjunto de clases semánticas que usemos. Por tanto, para una palabra disponemos de todas sus etiquetas semánticas. Fácilmente podríamos diseñar distintas estrategias para combinar estas etiquetas semánticas para obtener un sentido concreto. Por ejemplo, emplear la etiqueta BLC–20, WordNet Domains y SuperSense, para decidir el sentido apropiado para la palabra.

Otro aspecto, relacionado con el punto anterior de combinación de clasificadores, consistiría en desarrollar un método para estudiar la coherencia entre clases, detectar incompatibilidades entre etiquetas semánticas, y ayudarnos de las correctas para intentar corregir las erróneas.

También se podría estudiar cómo integrar nuestro sistema en el seno de otras aplicaciones de PLN de nivel superior. Tendríamos que analizar qué aplicación podría beneficiarse en mayor medida de la información semántica, y de qué modo deberíamos integrar dicho tipo de información.

Finalmente, otro frente abierto es el estudio de otros métodos de selección de conceptos a partir de WordNet, mediante los cuales podríamos generar conjuntos de BLC distintos a los que ya tenemos, con diferentes características y propiedades, y que podrían aportarnos nuevos niveles de abstracción. Estos nuevos conjuntos también podrían aportarnos un buen recurso semántico para ser empleado en el marco de nuestro sistema basado en clases semánticas.

7.2.1 Producción científica

Parte del contenido de este trabajo ha sido ya publicado en una serie de artículos en conferencias nacionales e internacionales. A continuación mostramos la descripción de los diferentes artículos, junto con el capítulo, y sección en su caso, donde aparecen explicados en detalle en este trabajo. También incluimos una breve descripción y la referencia completa del artículo.

I. GPLSI: Word coarse-grained Disambiguation aided by Basic Level Concepts. (Capítulo 5, sección 5.4)

En este caso se trata de nuestra participación en la competición internacional SemEval–1. En concreto utilizamos nuestros clasificadores basados en clases semánticas para disponer de las clases semánticas BLC–20 como atributos para el entrenamiento. Se construyeron un conjunto de clasificadores basados en nuestras clases semánticas, y otro conjunto de clasificadores basados en en unas agrupaciones de sentidos que propusieron los organizadores (*SenseClusters*). En ambos casos la arquitectura que adoptamos fue similar a la que empleamos con nuestras clases semánticas. El sistema consistía en dos fases. En la primera fase se hizo uso de los clasificadores BLC–20 para etiquetar los ejemplos de evaluación con su clase semántica BLC-20 correspondiente. Estas etiquetas junto con un conjunto básico de atributos sería empleado en la segunda fase, en la que los clasificadores basados en *SenseClusters* asignarían la etiqueta apropiada a cada palabra. Conseguimos la quinta mejor posición con este sistema.

Izquierdo, Rubén; Suárez, Armando; Rigau German. *"GPLSI: Word coarse-grained Disambiguation aided by Basic Level Concepts"*. Proceedings of the fourth international workshop on semantic evaluations (SemEval 2007). Association for Computational Linguistics, 157-160. Prague, June 2007.

II. Exploring the Automatic Selection of Basic Level Concepts. (Capítulo 4)

Este artículo presenta nuestro método automático de selección de conceptos semánticos BLC a partir de WordNet. Se define y explica el algoritmo, y además se analizan, estudian y presentan las características más importantes de distintos conjuntos de BLC obtenidos mediante este método. Como evaluación inicial se muestran los resultados, sobre el corpus de evaluación de SensEval–3, de un sistema que utiliza dichas clases semánticas y una heurística sencilla. Esta heurística corresponde con seleccionar para cada palabra su clase semántica con la que aparece más frecuentemente anotado sobre el corpus SemCor. Se observa que se consiguen resultados muy competitivos mediante este sistema basado en clases semánticas.

Izquierdo, Rubén; Suárez, Armando; Rigau, German. *"Exploring the Automatic Selection of Basic Level Concepts"*. International Conference Recent Advances in Natural Language Processing : Proceedings / Galia Angelova [et al.] (eds.). Shoumen, Bulgaria : INCOMA, 2007. ISBN 978-954-91743-7-3, pp. 298-302.

III. A Proposal of Automatic Selection of Coarse-grained Semantic Classes for WSD. (Capítulo 4)

Este artículo presenta una extensión del anterior, donde se detalla en más profundidad el método de obtención de conceptos asi como los distintos parámetros para la obtención de distintos conjuntos BLC.

Izquierdo R., Suárez A. and Rigau G. *"A Proposal of Automatic Selection of Coarse-grained Semantic Classes for WSD"*. Proceedings of the 23th Annual Meeting of Sociedad Española para el Procesamiento del Lenguaje Natural, SEPLN07. Sevilla, España. Procesamiento del Lenguaje Natural num. 39 pp. 189-196. ISSN: 1135-5948. 2007.

IV. An empirical Study on Class-based Word Sense Disambiguation. (Capítulo 5 y 6)

En este artículo se presenta la evaluación empírica del sistema de desambiguación basado en clases semánticas y en aprendizaje automático. Se describe la arquitectura del sistema, el algoritmo de aprendizaje, así como los conjuntos de atributos y de clases semánticas. La evaluación se realiza en diferentes niveles de abstracción, en función del conjunto de clases semánticas seleccionado, y teniendo en cuenta distintos tipos de atributos para representar los ejemplos de entrenamiento. Se presenta también una evaluación enfocada a analizar el comportamiento de nuestros clasificadores semánticos cuando se varía la cantidad de información de aprendizaje. Observamos que, en la aproximación basada en clases semánticas, el número de ejemplos de entrenamiento requerido para obtener resultados competitivos se reduce.

Izquierdo, R., Suárez, A., and Rigau, G. *"An Empirical Study on Class-based Word Sense Disambiguation"*. In Proceedings of the 12th Conference of the European Chapter of the Association For Computational Linguistics (Athens, Greece, March 30 - April 03, 2009).

European Chapter Meeting of the ACL. Association for Computational Linguistics, Morristown, NJ, 389-397.

V. Using WordNet Relations and Semantic Classes in Information Retrieval Tasks. (Capítulo 4, sección 4.3)

Presentamos en esta ocasión una aplicación de nuestras clases semánticas. En concreto se trata de una competición de Recuperación de Información, en la que se quería analizar de qué forma integrar la información de los sentidos de las palabras en un sistema de Recuperación de Información para mejorar los resultados. Empleamos nuestras clases semánticas para representar mediante vectores de conceptos tanto las consultas como los documentos. Mediante distintas medidas obteníamos la similitud entre la consulta y el primer *ranking* de documentos obtenido por el sistema de Recuperación de Información. Este valor de similitud fue empleado para reordenar la lista de documentos, y darle mayor importancia a aquellos documentos con mayor similitud semántica con la consulta inicial.

Fernández, J.; Izquierdo, R.; Gómez, J.M; *"Using WordNet Relations and Semantic Classes in Information Retrieval Tasks"*. Working Notes for the CLEF 2009 Workshop, Corfu, Greece.

VI. GPLSI-IXA: Using Semantic Classes to Acquire Monosemous Training Examples from Domain Texts. (Capítulo 5, sección 5.5)

Aquí presentamos nuestra participación en SemEval–2. La tarea en esta ocasión consistía en evaluar sistemas de WSD sobre un dominio específico. De este modo se trataba de analizar cómo desarrollar sistemas de desambiguación para un dominio concreto, o cómo adaptar un sistema de propósito general a un dominio específico. Mediante nuestras clases semánticas obtuvimos un conjunto de ejemplos monosémicos desde corpus no anotados pertenecientes al dominio propuesto. Este conjunto de ejemplos monosémicos, junto con los ejemplos extraídos desde SemCor, lo empleamos para entrenar nuestro sistema basado en clases semánticas. De este modo el sistema de dominio general se adaptó al dominio específico. En este caso obtuvimos la quinta mejor posición con el sistema que hacía uso de

los dos conjuntos de ejemplos (desde SemCor y desde los textos del dominio), mostrando la robustez de nuestro sistema basado en clases semánticas.

Izquierdo, Rubén; Suárez, Armando; Rigau German. *"GPLSI-IXA: Using Semantic Classes to Acquire Monosemous Training Examples from Domain Texts"*. Proceedings of the fifth international workshop on semantic evaluations (SemEval 2010). Association for Computational Linguistics. Uppsala, Sweden. July 2010

Referencias

Agirre, E., & de Lacalle, O. Lopez. 2007. UBC-ALM: Combining k-NN with SVD for WSD. *Pages 342–345 of: Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*. Prague, Czech Republic: Association for Computational Linguistics.

Agirre, E., & Edmonds, P. 2006. *Word Sense Disambiguation: Algorithms and Applications*. Springer.

Agirre, E., & LopezDeLaCalle, O. 2003. Clustering WordNet Word Senses. *In: Proceedings of RANLP'03*.

Agirre, E., & Martínez, D. 2004. The Basque Country University system: English and Basque tasks. *Pages 44–48 of:* Mihalcea, Rada, & Edmonds, Phil (eds), *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*. Barcelona, Spain: Association for Computational Linguistics.

Agirre, E., & Rigau, G. 1996. Word sense disambiguation using Conceptual Density. *Pages 16–22 of: Proceedings of the 16th conference on Computational linguistics*. Morristown, NJ, USA: Association for Computational Linguistics.

Agirre, E., & Soroa, A. 2009. Personalizing PageRank for word sense disambiguation. *Pages 33–41 of: EACL '09: Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*. Morristown, NJ, USA: Association for Computational Linguistics.

Agirre, E., de Lacalle, O. Lopez, Fellbaum, C., Hsieh, S., Tesconi, M., Monachini, M., Vossen, P., & Segers, R. 2010. SemEval-2010 Task 17: All-words Word Sense Disambiguation on a Specific Domain. *In: Proceedings of the 5th International Workshop on*

Semantic Evaluations (SemEval-2010), Association for Computational Linguistics.

Agirre, Eneko, Lacalle, Oier Lopez De, & MartÃnez, David. 2006. Exploring feature set combinations for WSD. *In: In Proc. of the SEPLN.*

Atkins, S. 1992. Tools for computer-aided corpus lexicography: The Hector project. *In:* F. Kiefer, G. Kiss, & Pajzs, J. (eds), *Acta Linguistica Hungarica*, vol. 41.

Bar-Hillel, Y. 1960. The Present Status of Automatic Translation of Languages. *Advances in Computers*, **1**, 91–163.

Brill, E. 1995. Transformation-Based Error-Driven Learning and Natural Language Processing: A Case Study in Part of Speech Tagging. *Computational Linguistics*, **21**(4).

Cai, Jun Fu, Lee, Wee Sun, & Teh, Yee Whye. 2007. NUS-ML:Improving Word Sense Disambiguation Using Topic Features. *Pages 249–252 of: Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*. Prague, Czech Republic: Association for Computational Linguistics.

Chan, Yee Seng, Ng, Hwee Tou, & Zhong, Zhi. 2007. NUS-PT: Exploiting Parallel Texts for Word Sense Disambiguation in the English All-Words Tasks. *Pages 253–256 of: Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*. Prague, Czech Republic: Association for Computational Linguistics.

Charniak, E., Blaheta, D., Ge, N., Hall, K., Hale, J., & Johnson, M. 2000. *BLLIP 1987-89 WSJ Corpus Release 1*. Tech. rept. Linguistic Data Consortium (LDC), Philadelphia.

Ciaramita, M., & Altun, Y. 2006. Broad-Coverage Sense Disambiguation and Information Extraction with a Supersense Sequence Tagger. *Pages 594–602 of: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'06)*. Sydney, Australia: ACL.

Ciaramita, M., & Johnson, M. 2003. Supersense tagging of unknown nouns in WordNet. *Pages 168–175 of: Proceedings of the Conference on Empirical methods in natural language processing (EMNLP'03)*. ACL.

Cuadros, M., & Rigau, G. 2008 (August). KnowNet: using Topic Signatures acquired from the web for building automatically highly dense knowledge bases. *In: Proceedings of 22nd International Conference on Computational Linguistics (COLING'08)*.

Curran, J. 2005. Supersense tagging of unknown nouns using semantic similarity. *Pages 26–33 of: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL'05)*. ACL.

Daudé, J., Padró, Ll., & Rigau, G. 2003. Validation and Tuning of Wordnet Mapping Techniques. *In: Proceedings of the International Conference on Recent Advances on Natural Language Processing (RANLP'03)*.

de Lacalle, O. Lopez. 2009. Domain-Specific Word Sense Disambiguation. *In: Lengoiaia eta Sistema Informatikoak Saila (UPV-EHU). Donostia 2009ko Abenduaren 14ean.*

Decadt, B., Hoste, V., Daelemans, W., & den Bosch, A. Van. 2004. GAMBL, genetic algorithm optimization of memory-based WSD. *Pages 108–112 of:* Mihalcea, Rada, & Edmonds, Phil (eds), *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*. Barcelona, Spain: Association for Computational Linguistics.

Drucker, H., Shahrary, B., & Gibbon, D.C. 2002. Support vector machines: relevance feedback and information retrieval. *Inf. Process. Manage.*, **38**(3), 305–323.

E. Agirre, L. Màrquez, & Wicentowski, R. (eds). 2007. *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval)*. Association for Computational Linguistics, Prague, Czech Republic.

Escudero, G., Marquez, L., & Rigau, G. 2000a. An Empirical Study of the Domain Dependence of Supervised Word Disambiguation Systems. *Pages 172–180 of: 2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*. Hong Kong, China: Association for Computational Linguistics.

Escudero, G., Màrquez, L., & Rigau, G. 2000b. Naive Bayes and Exemplar-Based Approaches to Word Sense Disambiguation Re-

visited. *In: Proceedings of the 14th European Conference on Artificial Intelligence (ECAI)*.

Escudero, G., Màrquez, L., & Rigau, G. 2004. TALP system for the English lexical sample task. *Pages 113–116 of:* Mihalcea, Rada, & Edmonds, Phil (eds), *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*. Barcelona, Spain: Association for Computational Linguistics.

Fellbaum, C. (ed). 1998. *WordNet. An Electronic Lexical Database*. The MIT Press.

Fernández, J., Izquierdo, R., & Gómez, J.M. 2009. Alicante at CLEF 2009 Robust–WSD Task. *In: Working Notes for the CLEF 2009 Workshop, Corfu, Greece*.

Frank, E., Paynter, G., Witten, I., Gutwin, C., & Nevill-Manning, C. 1999. Domain-Specific Keyphrase Extraction. *Pages 668–673 of: IJCAI '99: Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Gale, W., K. Church, K.W. Ward, & Yarowsky, D. 1992a. Estimating upper and lower bounds on the performance of word-sense disambiguation programs. *Pages 249–256 of: Proceedings of the 30th annual meeting on Association for Computational Linguistics*. Morristown, NJ, USA: Association for Computational Linguistics.

Gale, W., Church, K., & Yarowsky, D. 1992b. One sense per discourse. *In: Proceedings of of DARPA speech and Natural Language Workshop*.

Gammerman, A., Vovk, V., & Vapnik, V. 1998. Learning by Transduction. *Pages 148–155 of: In Uncertainty in Artificial Intelligence*. Morgan Kaufmann.

Gliozzo, A., Strapparava, C., & Dagan, I. 2004. Unsupervised and supervised exploitation of semantic domains in lexical disambiguation. *Computer Speech & Language*, **18**(3), 275 – 299. Word Sense Disambiguation.

Grozea, C. 2004. Finding optimal parameter settings for high performance word sense disambiguation. *Pages 125–128 of:* Mihalcea, Rada, & Edmonds, Phil (eds), *Senseval-3: Third Interna-*

tional Workshop on the Evaluation of Systems for the Semantic Analysis of Text. Barcelona, Spain: Association for Computational Linguistics.

Hawkins, P., & Nettleton, D. 2000. Large Scale WSD Using Learning Applied to SENSEVAL. *Computers and the Humanities*, **Volume 34**(1), 135–140.

Hearst, M., & Schütze, H. 1993. Customizing a lexicon to better suit a computational task. *In: Proceedingns of the ACL SIGLEX Workshop on Lexical Acquisition*.

Hovy, E., Marcus, M., Palmer, M., Ramshaw, L., & Weischedel, R. 2006. OntoNotes: the 90 % solution. *Pages 57–60 of: NAACL '06: Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers on XX*. Morristown, NJ, USA: Association for Computational Linguistics.

Hughes, T., & Ramage, D. 2007. Lexical Semantic Relatedness with Random Graph Walks. *Pages 581–589 of: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*. Prague, Czech Republic: Association for Computational Linguistics.

Izquierdo, R., Suárez, A., & Rigau, G. 2007. GPLSI: Word Coarse-grained Disambiguation aided by Basic Level Concepts. *Pages 157–160 of: Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*. Prague, Czech Republic: Association for Computational Linguistics.

J., N. Cristianini, & Shawe-Taylor. 2000. *An introduction to support vector machines : and other kernel-based learning methods*. 1 edn. Cambridge University Press.

J. Hernández Orallo, M.J. Ramírez Quintana y C. Ferri Ramírez. 2004. *Introducción a la Minería de Datos*. Editorial Pearson.

Joachims, T. 1998. Text categorization with support vector machines: learning with many relevant features. *Pages 137–142 of:* Nédellec, Claire, & Rouveirol, Céline (eds), *Proceedings of ECML-98, 10th European Conference on Machine Learning*. Chemnitz, DE: Springer Verlag, Heidelberg, DE.

Joachims, T. 1999. Making large-scale support vector machine learning practical. 169–184.

Jones, K. Spärck. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, **28**, 11–21.

Kilgarriff, A., & Rosenzweig, J. 2001. *English SENSEVAL: Report and Results*.

Kohomban, U., & Lee, W.S. 2005. Learning semantic classes for word sense disambiguation. *Pages 34–41 of: ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. Morristown, NJ, USA: Association for Computational Linguistics.

Kohomban, U., & Lee, W.S. 2007. Optimizing classifier performance in word sense disambiguation by redefining word sense classes. *Pages 1635–1640 of: IJCAI'07: Proceedings of the 20th international joint conference on Artifical intelligence*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Kuĉera, H., & Francis, W. N. 1967. *Computational Analysis of Present-Day American English*. Providence, RI, USA: Brown University Press.

Kudo, T., & Matsumoto, Y. 1995. Chunking with support vector machines. *Pages 1–8 of: NAACL '01: Second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies 2001*. Morristown, NJ, USA: Association for Computational Linguistics.

Leacock, C., Towell, G., & Voorhees, E. 1993. Towards Building Contextual Representations of Word Senses Using Statistical Models. *Pages 97–113 of: Corpus Processing for Lexical Acquisition*. MIT Press.

Lee, Yoong Keok, Ng, Hwee Tou, & Chia, Tee Kiah. 2004a. Supervised Word Sense Disambiguation with Support Vector Machines and multiple knowledge sources. *Pages 137–140 of:* Mihalcea, Rada, & Edmonds, Phil (eds), *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*. Barcelona, Spain: Association for Computational Linguistics.

Lee, Yoong Keok, Ng, Hwee Tou, & Chia, Tee Kiah. 2004b. Supervised Word Sense Disambiguation with Support Vector Machines and multiple knowledge sources. *Pages 137–140 of:* Mihalcea, Rada, & Edmonds, Phil (eds), *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*. Barcelona, Spain: Association for Computational Linguistics.

Lesk, M. 1986. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. *Pages 24–26 of: SIGDOC '86: Proceedings of the 5th annual international conference on Systems documentation*. New York, NY, USA: ACM.

Magnini, B., & Cavaglià, G. 2000. Integrating subject field codes into WordNet. *In: Proceedings of LREC*.

Mallery, John C. 1988. Thinking About Foreign Policy: Finding an Appropriate Role for Artificially Intelligent Computers. *In: Master's thesis, M.I.T. Political Science Department*.

McCarthy, D., Koeling, R., Weeds, J., & Carroll, J. 2004. Finding predominant word senses in untagged text. *In: In 42nd Annual Meeting of the Association for Computational Linguistics, Barcelona, Spain.*

Mihalcea, R., & Edmonds, P. (eds). 2004. *Proceedings of the 3rd International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*. Senseval–3, Barcelona, Spain.

Mihalcea, R., & Faruque, E. 2004. SenseLearner: Minimally supervised Word Sense Disambiguation for all words in open text. *Pages 155–158 of:* Mihalcea, Rada, & Edmonds, Phil (eds), *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*. Barcelona, Spain: Association for Computational Linguistics.

Mihalcea, R., & Moldovan, D. 2001a. Automatic generation of coarse grained wordnet. *In: Proceding of the NAACL workshop on WordNet and Other Lexical Resources: Applications, Extensions and Customizations*.

Mihalcea, R., & Moldovan, D. 2001b. Pattern Learning and Active Feature Selection for Word Sense Disambiguation. *Pages 127–130 of: SensEval–2*.

Miller, G., Leacock, C., Tengi, R., & Bunker, R. 1993. A Semantic Concordance. *In: Proceedings of the ARPA Workshop on Human Language Technology*.

Mooney, R. 1996. Comparative Experiments on Disambiguating Word Senses: An Illustration of the Role of Bias in Machine Learning. *Pages 82–91 of: In Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Navigli, R. 2006. Meaningful clustering of senses helps boost word sense disambiguation performance. *Pages 105–112 of: ACL-44: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*. Morristown, NJ, USA: Association for Computational Linguistics.

Navigli, R. 2009. Word Sense Disambiguation: a Survey. *ACM Computing Surveys*, **41**(2), 1–69.

Navigli, R., & Velardi, P. 2005. Structural Semantic Interconnections: a Knowledge-Based Approach to Word Sense Disambiguation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **27**(7), 1075–1088.

Navigli, R., Litkowski, K., & Hargraves, O. 2007. SemEval-2007 Task 07: Coarse-Grained English All-Words Task. *Pages 30–35 of: Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*. Prague, Czech Republic: Association for Computational Linguistics.

Ng, H. 1997. Getting Serious about Word Sense Disambiguation. *In: Proceedings of the SIGLEX Workshop*.

Ng, Hwee Tou, & Lee, Hian Beng. 1996. Integrating Multiple Knowledge Sources to Disambiguate Word Sense: An Exemplar-Based Approach. *Pages 40–47 of:* Joshi, Arivind, & Palmer, Martha (eds), *Proceedings of the Thirty-Fourth Annual Meeting of the Association for Computational Linguistics*. San Francisco: Morgan Kaufmann Publishers.

Niles, I., & Pease, A. 2001. Towards a Standard Upper Ontology. *Pages 17–19 of: Proceedings of the 2nd International Conference on Formal Ontology in Information Systems (FOIS-2001)*. Chris Welty and Barry Smith, eds.

Palmer, M., Fellbaum, C., Cotton, S., Delfs, L., & Dang, H. Trang. 2001. English tasks: All-words and verb lexical sample. *In: Proceedings of the SENSEVAL-2 Workshop. In conjunction with ACL'2001/EACL'2001*.

Pedersen, T. 2000. A simple approach to building ensembles of Naive Bayesian classifiers for word sense disambiguation. *Pages 63–69 of: Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Pedersen, T. 2001. A decision tree of bigrams is an accurate predictor of word sense. *Pages 1–8 of: NAACL '01: Second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies 2001*. Morristown, NJ, USA: Association for Computational Linguistics.

Peh, Li Shiuan, & Ng, Hwee Tou. 1997. Domain-Specific Semantic Class Disambiguation Using Wordnet. *Pages 56–64 of: Proceedings of the fifth Workshop on Very Large Corpora*.

Peters, W., Peters, I., & Vossen, P. 1998. Automatic Sense Clustering in EuroWordNet. *In: First International Conference on Language Resources and Evaluation (LREC'98)*.

Pradhan, S., Dligach, E. Loperand D., & Palmer, M. 2007. SemEval-2007 task 17: English lexical sample, SRL and all words. *Pages 87–92 of: SemEval '07: Proceedings of the 4th International Workshop on Semantic Evaluations*. Morristown, NJ, USA: Association for Computational Linguistics.

Ratnaparkhi, A. 1999. Learning to Parse Natural Language with Maximum Entropy Models. *Machine Learning*, **34**(1-3), 151–175.

Resnik, P. 1995. Using Information Content to Evaluate Semantic Similarity in a Taxanomy. *Pages 448–453 of: International Joint Conference for Artificial Intelligence (IJCAI-95)*.

Rigau, G., Agirre, E., & Atserias, J. 2003. The MEANING project. *In: Proc. of the XIX Congreso de la Sociedad Española para el Procesamiento del Lenguaje Natural (SEPLN). ISSN 1135-5948. Alcalá de Henares (Madrid)*.

Robertson, S.E., & Jones, K. Sparck. 1976. Relevance weighting of search terms. 143–160.

Roget, P.M. 1852. *Roget's Thesaurus*. Burnt Mill, Harlow, Essex: Longman Group Limited.

Rosch, E. 1977. Human Categorisation. *Studies in Cross-Cultural Psychology*, **I**(1), 1–49.

Segond, F., Schiller, A., Greffenstette, G., & Chanod, J. 1997. An Experiment in Semantic Tagging Using Hidden Markov Model Tagging. *Pages 78–81 of: ACL Workshop on Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications*. New Brunswick, New Jersey: ACL.

Snow, R., S., Prakash, D., Jurafsky, & A., Ng. 2007. Learning to Merge Word Senses. *Pages 1005–1014 of: Proceedings of Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*.

Snyder, B., & Palmer, M. 2004. The English all-words task. *Pages 41–43 of:* Mihalcea, Rada, & Edmonds, Phil (eds), *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*. Barcelona, Spain: Association for Computational Linguistics.

Sofia, S., Kemal, O., Karel, P., Dimitris, C., Dan, C., Dan, T., Svetla, K., George, T., Dominique, D., & Maria, G. 2002. Balkanet: A Multilingual Semantic Network for the Balkan Languages. *In: Proceedings of the* 1^{st} *Global WordNet Association conference*.

Spearman, C. 1904. The Proof and Measurement of Association between Two Things. *The American Journal of Psychology*, **15**(1), 72–101.

Stevenson, M., & Wilks, Y. 2001. The interaction of knowledge sources in word sense disambiguation. *Comput. Linguist.*, **27**(3), 321–349.

Strapparava, C., Gliozzo, A., & Giuliano, C. 2004a. Pattern Abstraction and Term Similarity for Word Sense Disambiguation: IRST at Senseval-3. *In: Proceedings of the Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text (Senseval-3)*.

Strapparava, C., Gliozzo, A., & Giuliano, C. 2004b. Pattern abstraction and term similarity for Word Sense Disambiguation: IRST at Senseval-3. *Pages 229–234 of:* Mihalcea, Rada, & Edmonds,

Phil (eds), *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*. Barcelona, Spain: Association for Computational Linguistics.

Suárez, A., & Palomar, M. 2002. A Maximum Entropy-based Word Sense Disambiguation System. *In: COLING*.

Takeuchi, K., & Collier, N. 2002. *Use of support vector machines in extended named entity recognition*.

Tikhonov, A.N. 1943. On the stability of inverse problems. *Doklady Akademii nauk SSSR*, **39**(5), 195–198.

Vapnik, V. 1995. *The nature of statistical learning theory*. New York, NY, USA: Springer-Verlag New York, Inc.

Villarejo, L., Màrquez, L., & Rigau, G. 2005 (September). Exploring the construction of Semantic Class Classifiers for WSD. *Pages 195–202 of: Proceedings of the 21th Annual Meeting of Sociedad Española para el Procesamiento del Lenguaje Natural SEPLN'05*. ISSN 1136-5948.

Vossen, P. (ed). 1998. *EuroWordNet: A Multilingual Database with Lexical Semantic Networks* . Kluwer Academic Publishers .

Weaver, W. 1957. Translation. *Pages 15–23 of:* Locke, W. N., & Booth, A. D. (eds), *Machine Translation of Languages*. MIT Press.

Wilks, Y. 1975. Preference Semantics. *Pages 329–348 of:* Keenan, E. L. (ed), *Formal Semantics of Natural Language*. Cambridge: Cambridge University Press.

Wilks, Y., Fass, D., J.Mcdonald, Plate, T., & Slator, B. 1990. Providing Machine Tractable Dictionary Tools. *Journal of Machine Translation*, **2**.

Yarowsky, D. 1992 (July). Word-Sense Disambiguation using Statistical Models of Roget's Categories Trained on Large Corpora. *Pages 454–460 of: Proceedings of COLING-92*.

Yarowsky, D. 1994. Decision Lists for Lexical Ambiguity Resolution: Application to Accent Restoration in Spanish and French. *In: Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics (ACL'94)*.

Yarowsky, D. 1995. Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. *Pages 189–196 of: In Proceedings of*

the 33rd Annual Meeting of the Association for Computational Linguistics.

Yarowsky, D. 1999. Hierarchical Decision Lists for Word Sense Disambiguation. *Pages 179–186 of: Computers and the Humanities*.

Yarowsky, D., Cucerzan, S., Florian, R., Schafer, C., & Wicentowski, R. 2001. *The Johns Hopkins senseval2 system descriptions*.

A. Resultados del Sistema. Atributos

En ese apéndice mostramos los resultados más relevantes que obtuvimos con nuestro sistema para las diferentes combinaciones de atributos que fuimos explorando hasta llegar a los atributos BASE descritos en el capítulo 6. Recordemos que el corpus que utilizamos como corpus de desarrollo y evaluación fue el SensEval–3 en este caso. La lista de experimentos que mostramos a continuación coincide con lo descrito en el capítulo 5. Progresivamente fuimos probando distintos atributos, diferentes forma de codificarlos y diversos tamaños para el contexto de la palabra objetivo, y nos quedábamos con aquellos elementos que nos proporcionaron mejores resultados.

La columna *Clase* indica el tipo de clase utilizada para construir los clasificadores. Se debe tener en cuenta que hemos usado *"SS"* en lugar de SuperSenses por abreviar.

En cuanto a los atributos usados en cada experimento, se muestran en la columna *Atributos*, y aunque ya han sido descritos, la lista de abreviaturas que usaremos en la tabla de resultados por simplicidad es:

- *WF*: forma de palabra
- *L*: lema de la palabra
- *P3(5)*: concatenación de las etiquetas *PoS* (morfológicas) de 3(5) *tokens* anteriores y 3(5) posteriores
- *BiL* / *TiL*: bigramas/trigramas de lemas
- *BiW* / *TiW*: bigramas/trigramas de palabras
- *BLC*: clases BLC para palabras monosémicas (será BLC20 o BLC50 o SS, según el tipo de clase que elijamos)
- *BLC50-PO*: clases BLC posibles para la palabra objetivo

- *MF-BLC50*: clase más frecuente para la palabra objetivo (será MF-BLC20, MF-BLC50, MF-WND, MF-SS, según el tipo de clase que elijamos)
- *MF-BLC50+w*: igual que el atributo anterior, pero se le concatena la palabra objetivo también
- *MF-BLC50+l*: igual que el atributo anterior, pero se le concatena el lema de la palabra objetivo también
- *ANT-BLC*: etiqueta BLC de la palabra objetivo inmediatamente anterior
- *WFsh*: expresión regular que representa la forma de la palabra en base a mayúsculas, minúsculas y dígitos

Por otra parte, en la mayoría de nuestros experimentos utilizamos la posición relativa del token respecto a la palabra objetivo para codificar los atributos que extraemos de dicho token. En este caso encontraremos un número entero en la columna *Cont.*. En la aproximación según *Bolsa de Palabras* probamos varios modos de codificar los tokens.

- $B_{-1,0,1}$: se diferencian tres grupos, el grupo de *tokens* anteriores a la palabra objetivo, la palabra objetivo, y el conjunto de *tokens* posteriores a la palabra objetivo. Los atributos extraídos de *tokens* de un mismo grupo siempre poseen la misma posición relativa
- $B_{0,1}$: se diferencia dos grupos, la palabra objetivo, y las palabras en el contexto tanto anterior como posterior
- B_0: solo se considera un grupo de tokens, formado por la palabra objetivo y las palabras en el contexto
- $B_{0,1,2}$: se codifica con un 0 la palabra objetivo, con un 1 los tokens en una ventana de 5 posiciones, y con un 2 los contenidos en una ventana entre 5 y 10
- B_{-3+3}: esto indicaría un rango. En este caso consideramos una ventana de tamaño 15, y se codifica los tokens por grupos. De las posiciones [-15,-11] con un -3, [-10,-6] con -2, y así sucesivamente.

Hay algunos casos en los que sólo hemos utilizado la forma de bolsa de palabras para ciertos tipos de atributos. En este caso encontraremos entre paréntesis los tipos de atributos implicados. Por ejemplo $B_{0,1}(WF, L)$ indicaría que se aplica ese tipo de codificación

$B_{0,1}$ solo a los atributos WF y L. En general, si se especifica algún subíndice referenciando a un tipo de atributo para un determinado contexto, quiere decir que ese contexto se utiliza para ese tipo de atributo.

En cuanto a la columna *Config.*, representa diferentes opciones que hemos utilizado, bien para el motor SVM o como ajustes de atributos. Hay que tener en cuenta que por simplificar la tabla, cuando en esta columna aparece el símbolo "-", quiere decir que se mantiene la misma configuración que en el experimento anterior (si el anterior también es "-" habría que buscar la primera fila que tuviera información de configuración, y todas los experimentos hasta llegar a esa poseerían la misma configuración. A continuación mostramos una leyenda para comprender los símbolos que aparecen en dicha columna *Config*.

- "$-c\ Valor$": representa el valor de regularización *c* usado en el módulo de aprendizaje SVM
- $P_{null,punc}$: en las etiquetas morfológicas, se utiliza NULL cuando el token pertenece a una oración diferente a la palabra objetivo. Para signos de puntuación utilizamos la etiqueta PUNC
- $Big_{noCruza}$: no permitimos bigramas que se extienda en oraciones distintas a la de la palabra objetivo
- $F_{frec} > valor$: filtro de atributos por frecuencia. Su frecuencia debe superar el *valor*
- $F_{uniq} = valor$: filtro de atributos por número de clases a la que pertenece. Se elimina el atributo si no pertenece a *valor* clases
- $F_{ref} > valor$: filtro refinado basado en la relación entre la frecuencia del atributo para la clase, y la frecuencia del atributo en general. La proporción de ambas frecuencias debe superar el *valor*
- $NoF.$: no se usa filtro
- $Gener.$: se generalizan bigramas, sustituyendo la palabra objetivo por una cadena comodín. Se utilizan solo los modificados
- $+Gen.$: como $Gener.$, pero se incluyen bigramas originales y modificados

Finalmente la columna *Num* representa el número medio de atributos por clasificador, y las últimas columnas muestran en valor F1 para las palabras polisémicas y para todas las palabras.

Clase	Atributos	Cont.	Config.	Num	Valor F1	
					Poli.	Todas
BLC50	WF L P3 P5 BiL BLC50	50	-	70.654	60,78	71,57
BLC50	WF L	50	-	40,909	60,62	71,46
BLC50	WF L P3 P5 BiL BLC50	5	-	7.914	63,26	73,37
BLC50	WF L	5	-	3.853	64,50	74,27
BLC50	WF L P3 P5	5	-	4.931	62,95	73,15
BLC50	WF L P3	5	-	4.147	62,64	72,92
BLC50	WF	5	-	2.115	62,79	73,03
BLC50	L	5	-	1.738	64,19	74,04
BLC50	L P3 P5	5	-	2.816	62,95	73,15
BLC50	L BLC50	5	-	1.904	63,57	73,60
BLC50	L	50	-	18.653	60,62	71,46
BLC50	WF L BLC50	50	-	42.843	60,62	71,46
BLC50	L	5	-c 0,1	1.738	64,50	74,27
BLC50	L	5	-c 1	1.738	63,57	73,60
BLC50	L	5	-c 0,001	1.738	63,10	73,26
BLC50	L	5 $B_{-1,0,1}$	-c 0,01	1.310	65,43	74,94
BLC50	L	50 $B_{-1,0,1}$	-	5.688	59,69	70,79
BLC50	L BLC50	5 $B_{-1,0,1}$	-	1.408	65,74	75,17
BLC50	WF L	5 $B_{-1,0,1}$	-	2.978	64,96	74,61
BLC50	WF L P3 P5 BiL BLC50	5 $B_{-1,0,1}$	-	6.809	64,03	73,93
BLC50	L P3 P5 BLC50	5 $B_{-1,0,1}$	-	2.485	64,03	73,93
BLC50	L	5 $B_{0,1}$	-	1.087	65,27	74,83
BLC50	L BLC50	5 $B_{0,1}$	-	1.158	66,20	75,51
BLC50	L BLC50	50 $B_{0,1}$	-	4.193	59,69	70,79
BLC50	L(cont=5) BLC50	50 $B_{0,1}$	-	1.271	64,96	74,61
BLC50	L(cont=5) BLC50	25 $B_{0,1}$	-	1.234	65,12	74,72
BLC50	L BLC50	50 B_0	-	1.118	55,81	67,98
BLC50	L BLC50	10 $B_{0,1,2}$	-	2.340	63,26	73,37
BLC50	WF L P3 P5 BiL BLC50	10 $B_{0,1,2}$	-	11.105	62,95	73,15
BLC50	WF L P3 P5 BiL BLC50	5	$P_{null,punc}$	6.693	65,43	74,94
BLC50	WF L P3 P5 BiL BiW BLC50	5	$Big_{noCruza}$ -	9.982	64,34	74,16
BLC50	WF L P3 P5 BiL BLC50	5	$F_{frec} > 1$	1.656	65,27	74,83
BLC50	WF L P3 P5 BiL BLC50	5	$F_{frec} > 2$	798	65,12	74,72
BLC50	WF L P3 P5 BiL BiW BLC50	5	$F_{frec} > 1$	2.152	64,69	74,61
BLC50	WF L P3 P5 BiL BiW BLC50	5	$F_{frec} > 2$	963	65,12	74,72
BLC50	WF L P3 P5 BiL BiW BLC50	5	$F_{frec} > 5$	312	62,33	72,70
BLC50	WF L P3 P5 BiL BiW BLC50	5	$F_{uniq} = 1$	1.529	45,58	60,56
BLC50	WF L P3 P5 BiL BiW BLC50	5	$F_{uniq} = 5$	5.818	64,65	74,38
BLC50	WF L P3 P5 BiL BiW BLC50	5	$F_{ref} > 0,25$	5.003	66,67	75,84
BLC50	WF L P3 P5 BiL BiW BLC50	5	$F_{ref} > 0,50$	2.527	62,48	72,81

Clase	Atributos	Cont.	Config.	Num	Valor F1	
					Poli.	Todas
BLC50	WF L P3 P5 BiL BiW BLC50	5	$F_{ref} > 0,10$	6.527	64,81	74,49
BLC50	WF L P3 P5 BiL BiW BLC50	5	NoF. Gener.	9.652	63,26	73,37
BLC50	WF L P3 P5 BiL BiW BLC50	5	No atrs. pal. objetivo	9.555	55,66	67,87
BLC50	WF L P3 P5 BiL BiW BLC50	5	$F_{ref} > 0,25$	4.480	65,89	75,28
BLC50	WF L P3 P5 BiL BiW TiL TiW BLC50	5	noFiltro	17.718	63,57	73,60
BLC50	WF L P3 P5 BiL BiW TiL TiW BLC50	5	$F_{ref} > 0,25$	11.281	66,98	76,07
BLC50	WF L P3 P5 BiL BiW TiL TiW BLC50	5	$F_{ref} > 0,28$	10.865	66,36	75,62
BLC50	WF L P3 P5 BiL BiW BLC50	5	+Gen. $F_{ref} > 0,25$	5.179	65,43	74,94
BLC50	WF L P3 P5 BiL BiW TiL TiW BLC50	5	-	13.818	65,58	75,06
BLC50	WF L P3 P5 BiL BiW TiL TiW BLC50	5 y 2 para WF L	-	19.546	64,96	74,61
BLC50	WF L P3 P5 BiL BiW TiL TiW BLC50	10	-	14.342	64,50	74,27
BLC50	WF L P3 P5 BiL BiW TiL TiW BLC50	3	-	7.846	65,74	75,17
BLC50	WF L P3 P5 BiL BiW TiL TiW BLC50	5 $B_{0(WF,L)}$	-	11.124	51,78	65,05
BLC50	WF L P3 P5 BiL BiW TiL TiW BLC50 BLC50-PO	5 $B_{0(WF,L)}$	-	11.126	50,08	63,82
SS	WF L P3 P5 BiL BiW TiL TiW SS	5 $B_{0,1(WF,L)}$	-	69.707	69,77	79,55
SS	WF L P3 P5 BiL BiW TiL TiW SS	5 $B_{0,1(WF,L)}$	NoFiltro	94.237	70,01	79,78
SS	WF L P3 P5 BiL BiW TiL TiW SS	5 $B_{0,1(WF,L)}$	$F_{ref} > 0,5$	48.713	70,76	80,22
SS	WF L P3 P5 BiL BiW TiL TiW SS	5 $B_{0,1(WF,L)}$	$F_{ref} > 0,75$	24.176	61,30	73,82
BLC20	WF L P3 P5 BiL BiW TiL TiW BLC50	5 $B_{0,1(WF,L)}$	$F_{ref} > 0,25$	5.942	65,45	74,49

Clase	Atributos	Cont.	Config.	Num	Valor F1	
					Poli.	Todas
BLC20	WF L P3 P5 BiL BiW TiL TiW BLC50	5 $B_{0,1(WF,L)}$	noFiltro	9.591	64,38	73,71
BLC20	WF L P3 P5 BiL BiW TiL TiW BLC20	5 $B_{0,1(WF,L)}$	noFiltro	9.599	64,08	73,48
BLC20	WF L P3 P5 BiL BiW TiL TiW BLC20	5 $B_{0,1(WF,L)}$	$F_{ref} > 0,25$	5.943	64,54	73,82
BLC50	WF L P3 P5 BiL BiW TiL TiW BLC50	5 $B_{0,1(WF,L)}$ 25_{BLC50}	$F_{ref} > 0,25$	11.284	65,89	75,28
BLC50	WF L P3 P5 BiL BiW TiL TiW BLC50	5 $B_{0,1(WF,L)}$ 25_{BLC50}	noFiltro	17.920	64,19	74,04
BLC50	WF L P3 P5 BiL BiW TiL TiW BLC50 MF-BLC50	5 $B_{0,1(WF,L)}$	$F_{ref} > 0,25$	11.456	67,44	76,40
BLC50	WF L P3 P5 BiL BiW TiL TiW BLC50 MF-BLC50	5 $B_{0,1(WF,L)}$	$F_{ref} > 0,25$ (no MF-BLC50)	11.532	66,82	75,96
BLC50	WF L P3 P5 BiL BiW TiL TiW BLC50 MF-BLC50 MF-BLC50+w	5 $B_{0,1(WF,L)}$	$F_{ref} > 0,25$	11.410	67,44	76,40
BLC50	WF L P3 P5 BiL BiW TiL TiW BLC50 MF-BLC50 MF-BLC50+w ANT-BLC50	5 $B_{0,1(WF,L)}$	-	11.413	67,44	76,40
BLC50	WF L P3 P5 BiL BiW TiL TiW BLC50 MF-BLC50 MF-BLC50+w ANT-BLC50	5 $B_{0,1(WF,L)}$	noFiltro	17.982	66,82	75,96
BLC50	WF L P3 P5 BiL BiW TiL TiW BLC50 MF-BLC50 MF-BLC50+w ANT-BLC50	5 $B_{0,1(WF,L)}$	$F_{ref} > 0,5$	7.836	67,13	76,18
BLC50	WF L P3 P5 BiL BiW TiL TiW BLC50 MF-BLC50 MF-BLC50+w ANT-BLC50	5 $B_{0,1(WF,L)}$	$F_{ref} > 0,1$	13.521	66,98	76,07
BLC50	WF L P3 P5 BiL BiW TiL TiW BLC50 MF-BLC50 MF-BLC50+l ANT-BLC50	5 $B_{0,1(WF,L)}$	$F_{ref} > 0,25$	11.362	66,98	76,07

Clase	Atributos	Cont.	Conf.	Num	Valor F1	
					Poli.	Todas
BLC50	WF L P3 P5 BiL BiW TiL TiW BLC50 MF-BLC50 MF-BLC50+l WFsh ANT-BLC50	5 $B_{0,1(WF,L,WFsh)}$	-	11.367	67,29	76,29
BLC50	WF L P3 P5 BiL BiW TiL TiW BLC50 MF-BLC50 MF-BLC50+l WFsh ANT-BLC50	5	-	11.369	67,29	76,29
BLC50	WF L P3 P5 BiL BiW TiL TiW BLC50 MF-BLC50 MF-BLC50+l WFsh ANT-BLC50	10 $B_{0,1(WF,L,WFsh)}$	-	11.850	66,98	76,07
BLC50	WF L P3 P5 BiL BiW TiL TiW BLC50 MF-BLC50 MF-BLC50+w WFsh ANT-BLC50	10 $B_{0,1(WF,L,WFsh)}$	-	11.901	66,98	76,07
BLC50	WF L P3 P5 BiL BiW TiL TiW BLC50 MF-BLC50 MF-BLC50+w WFsh ANT-BLC50	10	-	14.700	66,98	76,07
BLC50	WF L P3 P5 BiL BiW TiL TiW BLC50 MF-BLC50 MF-BLC50+w ANT-BLC50	10 $B_{-2+2(WF,L)}$	-	12.561	67,60	76,52
BLC50	WF L P3 P5 BiL BiW TiL TiW BLC50 MF-BLC50 MF-BLC50+w ANT-BLC50	20 $B_{-4+4(WF,L)}$	-	14.206	67,44	76,40
BLC50	WF L P3 P5 BiL BiW TiL TiW BLC50 MF-BLC50 MF-BLC50+w ANT-BLC50	15 $B_{-3+3(WF,L)}$	-	13.383	67,60	76,52

B. Peligro, no intentar repetir en casa

Lo primero que hay que tener en cuenta es que no hemos incluido este anexo como un capítulo de nuestra memoria debido a los pobres resultados obtenidos. En general, hemos comprobado que ninguna de las modificaciones que proponemos en este capítulo nos aporta una mejora. En ese capítulo comprobamos más bien lo que no deberíamos hacer en el desarrollo de una arquitectura de este tipo (de ahí el nombre del anexo).

A lo largo de este trabajo hemos visto el desarrollo y evaluación del sistema en varias líneas principales: arquitectura, atributos y niveles de abstracción. En este anexo describiremos una serie de experimentos infructuosos realizados simultáneamente junto con el resto descrito a lo largo de este trabajo. En concreto, en la sección B.1, describimos una serie de experimentos que analizaron el comportamiento y rendimiento del sistema variando el número de ejemplos positivos y negativos, tratando de equilibrar ambos conjuntos.

En la sección B.2 detallamos algunos experimentos que realizamos siguiendo otras arquitecturas de diseño, como por ejemplo la arquitectura multiclase, en lugar de la aproximación binaria que hemos seguido.

El paquete de aprendizaje que hemos utilizado (*SVMLight*), admite que para cada ejemplo, cada uno de los atributos posea un peso determinado. En la sección B.3 se modifican los pesos asociados a dichos atributos, para tratar de darles mayor peso a aquellos que pueden ser más relevantes.

B.1 Variando la cantidad de ejemplos de entrenamiento

Tal y como hemos descrito anteriormente, para cada clasificador se utilizan como ejemplos positivos todos aquellos etiquetados con la clase correspondiente al clasificador, y como ejemplos negativos el resto de ejemplos. Aunque en principio consideramos que sería una ventaja disponer de mayor cantidad de ejemplos para el entrenamiento de cada clasificador, también cabe pensar que posiblemente esto entorpeciera el entrenamiento, debido a que se agruparan ejemplos distintos de palabras diferentes bajo una misma clase semántica, y los atributos y formas de uso que se extraerían podrían ser muy heterogéneos, con lo que podría ser difícil generalizar y realizar un buen entrenamiento. Una forma de reducir esta excesiva dispersión de atributos sería **limitar el número de ejemplos**.

Podemos limitar el número de ejemplos positivos o el de negativos. Seleccionamos aleatoriamente diferentes cantidades de ejemplos para realizar estos experimentos. Únicamente se extraen atributos del conjunto de ejemplos que se seleccione. En la tabla B.1 podemos ver los resultados de los diferentes experimentos variando la proporción de ejemplos de entrenamiento. Se muestra el número de ejemplos positivos, negativos, la media de atributos por clasificador para los ejemplos positivos y los resultados de F1 para todas las palabras (monosémicas y polisémicas) sobre los corpus SE2 y SE3. Elegimos como experimento base a aquel que hace uso de los atributos básicos más los semánticos construidos según WND. El conjunto de clases utilizado para construir los clasificadores es BLC–20. Seleccionamos este experimento debido a que es uno de los que mejores resultados obtiene. La primera línea de la tabla corresponde con los resultados para este experimento.

Como podemos ver, tanto restringir el número de ejemplos positivos como el de negativos no produce ningún incremento en los resultados del sistema. El número de atributos por clasificador se reduce cuando se decrementa el número de ejemplos, como era de esperar. Se producen ligeras variaciones en el número de atributos cuando no se varia el número de ejemplos positivos, debido a que éstos se seleccionan aleatoriamente, y en cada ejecución son conjun-

Num. Positivos	Num. Negativos	Num. Atrib.	Sval–2	Sval–3
Todos	Todos	5.416	77,88	74,27
1.000	1.000	4.190	75,05	71,69
2.000	2.000	4.804	77,69	73,60
Todos	1.000	5.467	77,02	74,04
Todos	5.000	5.404	77,79	74,04
Todos	250	5.345	65,50	60,79
Todos	500	5.789	77,79	73,93

Tabla B.1. Experimentos sobre el número de ejemplos

tos diferentes. En conclusión, tal y como pensábamos en principio, los mejores resultados se obtienen utilizando todos los ejemplos de los que disponemos para realizar el entrenamiento. La gran cantidad de atributos diferentes no presenta ningún problema, debido a que SVM trabaja muy bien en espacios de alta dimensionalidad y con gran cantidad de atributos irrelevantes.

Otro problema en nuestra aproximación basada en clases, derivado del modo en que se distribuyen los ejemplos, podría ser el excesivo desequilibrio entre ejemplos positivos y negativos. Normalmente, el número de ejemplos negativos es mucho mayor al de ejemplos positivos por clasificador, lo que puede influir negativamente en el modelo que genere el módulo de SVM. Por esta razón diseñamos este experimento, en el que se **aumentan los ejemplos de entrenamiento positivos** con nuevos ejemplos extraídos de WordNet. Utilizamos los ejemplos de uso de las palabras, que aparecen contenidos en las glosas de WordNet. Empleamos la versión 3.0 de WordNet, debido a que disponen de los ejemplos etiquetados con sus sentidos correspondientes. Para la anotación morfológica de estos ejemplos hacemos uso del etiquetador Brill Tagger (Brill, 1995).

El proceso sólo se realiza para nombres, por limitar el tiempo de experimentación. Seleccionamos un experimento de los que realizamos anteriormente como referencia, para ejecutarlo con y sin el incremento de ejemplos de entrenamiento, y analizar su comportamiento. En concreto seleccionamos aquel experimento que usa BLC–20 como conjunto de clases semánticas para crear los clasificadores, y como conjunto de atributos a los atributos locales según los atributos IXA, definidos anteriormente en la sección 6.1.2. Realizamos las evaluación sobre los corpus SE2 y SE3.

La forma de añadir los ejemplos de entrenamiento es la siguiente. En primer lugar obtenemos de cada synset nominal de WordNet 3.0 todos los ejemplos que aparecen en él. Marcamos la palabra objetivo en cada uno de éstos, que es una de las palabras contenidas en el synset y representada en el ejemplo. Esta palabra aparece anotada semánticamente, y determina a qué clase semántica se añadirá este nuevo ejemplo para el entrenamiento. El siguiente paso es extraer los diferentes atributos de dicho ejemplo, y finalmente añadir el ejemplo al conjunto de ejemplos de entrenamiento para la clase correspondiente. Es importante tener en cuenta en este punto, que estos nuevos ejemplos se tratan de forma especial. Todos los ejemplos de que disponíamos anteriormente se utilizan como ejemplos positivos para su clase correspondiente y como negativos para el resto. Esto precisamente es lo que provoca el excesivo desequilibrio entre ejemplos positivos y negativos. Por tanto, los nuevos ejemplos sólo se tienen en cuenta como ejemplos positivos, y no se añaden como negativos para evitar continuar incrementando el desequilibrio que comentamos. Tampoco se utilizan estos nuevos ejemplos para extraer más atributos, simplemente se dispone de ellos como si ya pertenecieran al corpus de entrenamiento, y se utilizan como material extra de aprendizaje.

En la tabla B.2 vemos los resultados de este experimento. Mostramos el valor F1 para las palabras polisémicas y para todas, tanto sobre SE2 como sobre SE3. También se muestran el número de clasificadores que se crean, el número medio de atributos por clasificador y el número medio de ejemplos positivos y negativos por cada clasificador.

	local IXA	**local IXA + WN 3.0**
F1 SE2 (poli.-todas)	70,16 - 78,73	70,42 - 78,92
F1 SE3 (poli.-todas)	66,16 - 74,94	65,40 - 74,38
Num. Clasif.	356	356
Atributos / clasif.	1.438	1.438
Ejemplos / clasif. (+ -)	239 - 86.075	249 - 86.075

Tabla B.2. Aumento del número de ejemplos positivos para el entrenamiento

Podemos ver que sobre SE2, el incremento del conjunto de ejemplos positivos de entrenamiento produce una mejora del 0,37 % para

las palabras polisémicas, y un 0,24 % para el total de polisémicas y monosémicas. En el caso de SE3, añadir nuevos ejemplos de entrenamiento conlleva un empeoramiento de los resultados, 1,15 % en el caso de las palabras polisémicas y un 0,75 % en el caso de todas las palabras mono y polisémicas. Por tanto, no existe una tendencia clara, ni una mejora importante con el incremento del número de ejemplos positivos. Por otra parte, es cierto que la cantidad de nuevos ejemplos que añadimos no es muy grande (una media de 10 ejemplos más por clase), y el desequilibrio de ejemplos positivos–negativos aun permanece. La principal razón es que no hay gran cantidad de synsets en WordNet que contengan ejemplos de uso de las palabras contenidas en dichos synsets.

Otro experimento que realizamos es una ligera modificación del anterior. En este caso sí que utilizamos los nuevos ejemplos para extraer más atributos para el aprendizaje. En la tabla B.3 podemos ver los resultados en este caso.

	local IXA	**local IXA + WN 3.0**
F1 SE2 (poli.-todas)	70,16 - 78,73	69,63 - 78,36
F1 SE3 (poli.-todas)	66,16 - 74,94	65,55 - 74,49
Num. Clasif.	356	356
Atributos / clasif.	1.438	1.589
Ejemplos / clase (+ -)	239 - 86.075	268 - 86.075

Tabla B.3. Aumento del número de atributos y ejemplos positivos para el entrenamiento.

Vemos que se obtiene un empeoramiento en ambos casos, para SE2 y SE3. Posiblemente los ejemplos extraídos de WordNet son muy diferentes a los contenidos en el corpus SemCor, y los atributos que obtenemos de ambas fuentes, por tanto, también lo son. De este modo, en lugar de favorecer, empeoran más el entrenamiento de los modelos y clasificadores.

B.2 Otras Arquitecturas

Como venimos describiendo, la arquitectura por la que optamos es una arquitectura basada en clases semánticas. Creamos un clasificador para cada clase, que utiliza como ejemplos positivos a todos

aquellos ejemplos de palabras pertenecientes a la clase semántica en cuestión, y como ejemplos negativos al resto. Concretamente, del conjunto de ejemplos positivos se extrae un conjunto de atributos propio para cada clasificador, que posteriormente se utilizará para codificar y representar dichos ejemplos. Esta aproximación pensamos que es la más adecuada, debido a que cada clase puede disponer de sus atributos más representativos, y se pueden filtrar estos atributos comparándolos con los atributos representativos para otras clases.

Otra aproximación posible sería obtener un **conjunto único de atributos**, común para todas las clases, extraído de todos los ejemplos de aprendizaje. Este conjunto común de atributos se utilizaría posteriormente para codificar tanto los ejemplos positivos como los negativos, que se seguirán distribuyendo del mismo modo que anteriormente. Siguiendo esta aproximación realizamos un nuevo experimento. Para compararlo con alguno de nuestros experimentos utilizamos aquel que usa BLC–20 como conjunto de clases para crear los clasificadores, y el conjunto de atributos local IXA, definido en la sección 6.1.2. En la tabla B.4 mostramos los resultados para las palabras polisémicas y para todas (monosémicas y polisémicas), sobre los corpus SE2 y SE3. De nuevo se muestran el número de clasificadores creados, la media de atributos por clase y el número de ejemplos positivos y negativos por clasificador.

	local IXA	**local IXA Cjto. Único atrib**
F1 SE2 (poli-all)	70,16 - 78,73	65,92 - 75,71
F1 SE3 (poli-all)	66,16 - 74,94	63,73 - 73,15
Num. Clasif.	356	356
Atributos/clasif.	1.438	515.353
Ejemplos/clasif. (+ -)	239 - 86.075	239 - 86.075

Tabla B.4. Resultados usando un único conjunto común de atributos

Como podemos ver, se produce un gran aumento en el número medio de atributos por clasificador (de 1.438 a 515.353). Sin embargo, los resultados empeoran notablemente, lo que pone de relieve que es mejor disponer de un número más reducido de atributos, que sean más específicos para cada clasificador. Una posible forma de reducir esta gran cantidad de atributos es, tal y como hacemos en el res-

to de experimentos descrito en esta memorias, definir un filtro para descartar aquellos atributos que consideremos que no son importantes. La forma en que hacemos este filtrado es teniendo en cuenta la frecuencia de cada atributo para cada clase frente a la frecuencia total de dicho atributo. Este filtro, en este caso, no es posible aplicarlo, debido a que únicamente disponemos de un conjunto global de atributos. Por esta razón, definimos un filtro muy sencillo, basado únicamente en la **frecuencia global de cada atributo**: descartaremos aquellos atributos cuya frecuencia total sea menor que 5. La tabla B.5 presenta las mismas estadísticas que la anterior para este nuevo experimento.

	local IXA	**local IXA Cjto. Único atrib Frec. Min 5**
F1 SE2 (poli.-todas)	70,16 - 78,73	63,00 - 73,63
F1 SE3 (poli.-todas)	66,16 - 74,94	59,64 - 70,11
Num. Clasif.	356	356
Atributos/clasif.	1.438	17.365
Ejemplos/clasif. (+ -)	239 - 86.075	239 - 86.075

Tabla B.5. Resultados usando un único conjunto común de atributos. Filtro por frecuencia mínima 5.

En este caso los resultados con la nueva aproximación todavía son peores que en el caso anterior. Notemos que, con el filtro por frecuencia global, se produce un decremento del número de atributos por clasificador muy importante (hay 497.988 atributos de media menos por clasificador). Como hemos dicho, los resultados son peores que en el caso de no filtrar los atributos. Esto nos indica que el filtro que hemos definido en este caso es demasiado simple, y no sirve para seleccionar atributos relevantes. Quizá la simple frecuencia global de un atributo en un corpus no sea indicador de la importancia de dicho atributo. Por otra parte, como se indica en muchos trabajos, SVM es capaz de trabajar con gran cantidad de atributos irrelevantes, y en el caso de no disponer de una estrategia de filtrado más refinada, es mejor dejar que el motor SVM realice esta selección de atributos por sí solo.

Otra posible arquitectura para la implementación del sistema es hacer uso de una extensión del paquete *SVMLight*, que define cla-

sificadores multiclase: *SVMLight MultiClass*[1]. En esta ocasión no es necesario realizar una "binarización" de los clasificadores, no hay que crear un clasificador para cada clase y posteriormente lanzar cada uno de los clasificadores por separado para decidir qué clase se asocia a una nueva palabra. Ahora se creará un único clasificador multiclase, capaz de decidir entre todas las posibles clases semánticas.

Para entrenar este clasificador, se utiliza un único conjunto de atributos, obtenido de todos los ejemplos de entrenamiento. Cada ejemplo no se trata como positivo o negativo para un cierto clasificador binario, simplemente se considera como un ejemplo más de la clase semántica a la que esté asociado. El hecho de utilizar un único conjunto de atributos para codificar todos los ejemplos, nos lleva de nuevo a no poder aplicar el filtro refinado, y tener que recurrir al filtro básico basado en la frecuencia de atributos. Para realizar estas pruebas tomamos como experimento de referencia el mismo que utilizamos en las pruebas anteriores de esta sección: BLC–20 como conjunto de clases semánticas para entrenar los clasificadores, y el conjunto de atributos locales descritos en la sección 6.1.2.

Tenemos que tener en cuenta que esta implementación multiclase de SVM, internamente, realiza una binarización de la clasificación, ya que la teoría y fundamentos básicos de SVM se desarrollaron para un tipo de clasificación binaria. Por tanto, el tiempo de procesamiento para realizar el entrenamiento del modelo completo es muy superior al que se emplearía para entrenar un clasificador binario individual. Además de esto, algunos parámetros del algoritmo de entrenamiento cambian su forma de influir en el sistema, es decir, no podemos utilizar los mismos valores que en el caso binario, ya que no funcionarían del mismo modo. En concreto, el parámetro C de regularización, en el caso multiclase está escalado respecto al caso binario y debe ser reajustado. Además la gran cantidad de recursos que utiliza el entrenamiento del modelo multiclase, hace que las primeras pruebas que lanzamos sin uso de filtro provocasen un error de falta de memoria en el sistema.

El sistema multiclase genera un único modelo, y cuando trata de clasificar un nuevo ejemplo, genera una salida que consiste en un valor

[1] `http://www.cs.cornell.edu/People/tj/svm_light/svm_multiclass.html`

de pertenencia para cada una de las clases (en este caso de BLC–20 hay que recordar que disponemos de 365 clases). Para evitar problemas de inconsistencia, no nos quedamos con aquella clase que recibe una mayor puntuación en general, sino únicamente con aquella clase mejor puntuada de entre las clases posibles para la palabra. En la tabla B.6 vemos los valores F1 de los diferentes experimentos multiclase con diferentes combinaciones del parámetro c de regularización, y de la frecuencia mínima exigida para cada atributo (los atributos que no superan esta frecuencia son eliminados del conjunto).

Frec. Min.	**Atributos**	**Param.** c	**SE2**		**SE3**	
			Poli.	**Todas**	**Poli.**	**Todas**
Local IXA Caso binario			*70,16*	*78,73*	*66,16*	*74,94*
No filtro	497.375	1.000	Error de memoria			
2	73.683	1.000	60,61	71,93	59,18	69,78
3	35.594	100	47,48	62,57	47,50	61,12
10	6.653	100	49,47	63,99	42,34	57,30
10	6.653	1.000	54,51	67,58	53,72	65,73
10	6.653	10.000	57,56	69,75	55,69	67,19

Tabla B.6. Experimentos con SVM Multiclase

En ningún caso los resultados que se obtienen están próximos a los obtenidos con los clasificadores binarios. Son siempre peores en el caso multiclase. En principio el uso del paquete multiclase no debería suponer ningún problema, ya que la binarización interna que realiza es similar a la que nosotros implementamos. Sin embargo, hay varios problemas que pueden provocar estos malos resultados. En primer lugar, los parámetros que utilizamos son ajustados para los clasificadores binarios, y para el caso multiclase hubiéramos tenido que realizar un estudio en profundidad para ajustarlos adecuadamente. En segundo lugar, y más importante, de nuevo el problema del conjunto único de atributos, que no permite especializar cada uno de los clasificadores, y se obtienen unos modelos menos precisos y con un rendimiento inferior a los obtenidos en el caso binario.

Para finalizar esta sección hablaremos sobre el número de clasificadores que se crean. En principio, tal y como hemos descrito nuestra aproximación, se crea un clasificador por cada clase semántica, por tanto se dispone de tantos clasificadores como clases semánticas ha-

ya en el conjunto semántico elegido para entrenar el sistema (por ejemplo 558 clasificadores para el caso de nombres según BLC–20). Lo que sucede es que, cuando evaluamos nuestro sistema sobre algún corpus (SE2 o SE3), muchos de los clasificadores que se crean no se utilizan nunca, debido a que no todas las clases de un conjunto (por ejemplo, BLC–20) tienen que aparecer representadas en un cierto corpus. Tal y como hemos descrito, el proceso para etiquetar una palabra en concreto es obtener sus posibles clases semánticas, y obtener la pertenencia de la palabra a cada una de sus clases posibles, haciendo uso del clasificador correspondiente. Por tanto, con el único objetivo de **acelerar el proceso y reducir el coste temporal**[2], optamos por construir sólo los clasificadores necesarios para evaluar sobre un corpus en concreto. Es decir, para evaluar sobre un corpus, primero extraemos las palabras que se van a clasificar, y para cada una de estas palabras obtenemos sus posibles clases, las cuales se almacenan en una lista general. Al final del proceso, en esta lista tenemos todos los posibles clasificadores de los que se hará uso para la evaluación y, por tanto, solo se genera dicha lista de clasificadores en lugar de la lista completa para todas las clases según el repositorio de clases seleccionado. De este modo realizamos nuestros experimentos, ganando eficiencia temporal. Por supuesto, hemos hecho uso de esta técnica sólo para el proceso de evaluación y desarrollo del sistema, ya que para la construcción de un etiquetador general, deberemos generar clasificadores para todas las clases, ya que no sabremos de antemano que palabras se van a querer etiquetar.

Sin embargo, debido a la forma en que se filtran los atributos, esta técnica de limitar el número de clasificadores creados puede tener efectos sobre los resultados finales. El motivo es que, para filtrar un atributo para un clasificador, se tiene en cuenta tanto la frecuencia del atributo para la clase correspondiente, como la frecuencia del atributo para todas las clases. Este último valor es el que puede variar dependiendo de si decidimos construir todos los clasificadores o solo una subconjunto de ellos. Estas variaciones de la frecuencia global de

[2] Es importante tener en cuenta que el tiempo empleado en entrenar todos los clasificadores para un experimento en concreto es bastante considerable. Si además queríamos lanzar un gran número de experimentos, el tiempo total de experimentación era demasiado elevado.

los atributos ocasiona que en algún caso dichos atributos superen la condición de filtrado, y en otros caso no, y sean descartados. El uso de diferentes conjuntos de atributos, aunque esta diferencia sea solo de un mínimo número de atributos, puede provocar resultados dispares. Para estudiar hasta qué punto varían los resultados, diseñamos este experimento, en el cual se genera el conjunto completo de clases semánticas. De nuevo elegimos uno de nuestros experimentos para realizar la comparación con el uso de esta técnica. Seleccionamos aquel experimento que utiliza BLC–20 como repositorio de clases semánticas, y el conjunto de atributos locales IXA. La evaluación la realizamos sobre los corpus SE2 y SE3. En la tabla B.7 podemos ver los resultados de dicho experimento (valor F1).

	Clasif. SensEval–2		**Todos clasif.**	
	Polisémicas	**Todas**	**Polisémicas**	**Todas**
SensEval–2	70,16	78,73	69,36	78,17
SensEval–3	66,16	74,94	64,95	74,04

Tabla B.7. Generación de todos los clasificadores para BLC–20

Como vemos, la generación de todos los clasificadores provoca resultados ligeramente peores, aunque la diferencia es muy pequeña. Muy posiblemente, el uso de menos clasificadores ocasiona que el proceso de filtrado funcione mejor, y se seleccionen atributos más representativos, eliminando aquellos que introduzcan ruido. De cualquier modo, vemos que el resultado es muy similar en ambas aproximaciones, y por tanto podemos emplear aquella que utiliza la lista reducida de clasificadores para realizar nuestros experimentos, ya que las conclusiones que saquemos de ellos también serían aplicables para el desarrollo de un etiquetador general.

B.3 Ajuste de los Pesos de los Atributos

El formato de entrada del paquete SVM que empleamos permite establecer un peso asociado a cada uno de los atributos para cada ejemplo de entrenamiento. En principio no modificamos los pesos de los atributos, y utilizamos siempre 1 como constante para todos los

atributos. En estos experimentos, tratamos de modificar los pesos asociados a cada atributo, para darle más peso a aquellos que consideremos más importantes. Esta importancia la medimos de nuevo en términos de frecuencias. Disponemos de tres frecuencias para un atributo de un ejemplo cualquiera:

1. Frecuencia del atributo para el ejemplo: F_e[3]
2. Frecuencia del atributo para la clase: F_c
3. Frecuencia del atributo para todas las clases: F_g

Como hemos descrito anteriormente, los valores F_c y F_g los utilizamos para filtrar los atributos. Ahora F_e y F_g lo usamos para establecer el peso de un atributo concreto para un ejemplo determinado de una clase. Simplemente se dividen ambos valores (F_e / F_g) para obtener el peso concreto, lo que dará más importancia a aquellos atributos con mayor frecuencia para el ejemplo y menor frecuencia para la clase. Para realizar la comparación utilizamos como referencia el experimento que utiliza BLC–20 como conjunto de clases semánticas para generar los clasificadores y el conjunto de atributos local según la definición descrita en la sección 6.1.2. En la tabla B.8 podemos ver los valores F1 para ambos experimentos, evaluados sobre SE2 y SE3.

	Atrib. Local Peso=1		**Atrib. Local Peso=F_e/F_g**	
	Polisémicas	**Todas**	**Polisémicas**	**Todas**
SE2	70,16	78,73	60,08	71,55
SE3	66,16	74,94	59,94	70,34

Tabla B.8. Valores F1 para los experimentos con ajuste de pesos F_e/F_g

Realizamos otro experimento en la misma línea del anterior, pero variando la forma de calcular el peso para cada atributo. Ahora en lugar de realizar la división entre las frecuencias, obtenemos directamente su producto. De este modo se valora tanto que el atributo sea frecuente para el ejemplo, como que lo sea también para la clase en general. En la tabla B.9 vemos el resultado (medida F1) para este experimento.

[3] Normalmente el valor de frecuencia de un atributo para un ejemplo será 1. En algunos casos, en función de la codificación, este valor puede ser mayor que 1.

	Atrib. Local Peso=1		**Atrib. Local Peso=F_e x F_g**	
	Polisémicas	**Todas**	**Polisémicas**	**Todas**
SE2	70,16	78,73	69,10	77,98
SE3	66,16	74,94	62,97	72,58

Tabla B.9. Medida F1 para experimentos con ajuste de pesos F_e x F_g

Como vemos en las tablas, ninguno de los dos experimentos supone una mejora respecto a usar un valor constante para los atributos. A pesar de esto, se obtiene un mejor resultado en el segundo caso, obteniendo el producto de frecuencias. De cualquier modo, como ya hemos dicho, es mejor opción utilizar un valor constante para todos los atributos y permitir que el propio motor SVM sea quien procese los atributos y decida cual es más relevante. También tenemos que considerar que las técnicas que hemos probado para calcular el peso de los atributos son muy simples, y posiblemente con algunas técnicas más refinadas se podría haber conseguido alguna mejora, lo que queda también como línea de desarrollo posterior.