Universitat d'Alacant
Universidad de Alicante

Sistemas de clasificación de preguntas basados en corpus para la búsqueda de respuestas

David Tomás Díaz

Tesis Doctorales

www.eltallerdigital.com

UNIVERSIDAD de ALICANTE

Tesis Doctoral

Sistemas de clasificación de preguntas basados en corpus para la búsqueda de respuestas

David Tomás Díaz

Director

Dr. José Luis Vicedo González

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

Alicante, 9 de junio de 2009

Agradecimientos

Gracias a José L. Vicedo, amigo y mentor. Gracias a Carlos Pérez, Felipe Sánchez, Borja Navarro, Paloma Moreda, Patricio Martínez, Manuel Palomar, José Manuel Gómez, Sergio Ferrández, Óscar Ferrández, Rubén Izquierdo, Ivens Huertas, Ester Boldrini, Claudio Giuliano, Horacio Saggion, Milen Kouylekov, Bernardo Magnini, Günter Neumann, Paolo Rosso, Horacio Rodríguez, Empar Bisbal, Lidia Moreno, Armando Suárez, José F. Navarro, Yenory Rojas, Elisa Noguera, Lorenza Moreno, Zornitsa Kozareva, Antonio Ferrández, Maximiliano Saiz, Sonia Vázquez, Jesús Peral, Andrés Montoyo, Rafael Muñoz, Fernando Llopis, Rafael Muñoz Jr., Estela Saquete, Juan Antonio Pérez, Paco Moreno, Antonio Pertusa, Jorge Calera, Pedro Pastor, Cristina Cachero, Miguel Ángel Varó, Miguel Ángel Baeza, Vicente Ferri, Eva Cantos, Diego Ibáñez, José Ramón Lillo, Marcel Puchol, Sergio Navarro, María Pardiño, Jesús Hermida, Héctor Llorens, Elena Lloret, Alexandra Balahur, Antonio Toral, José Norberto Mazón, Sandra Roger, Francisco Gallego y Pedro Caselles. Gracias a Alfredo, Merce, Sensi, Esther, Gabriel, Candela, Adela, Adela Jr., Edu, Laura, Edu Jr., Rafa, Mari, Esther, Betty, Jose, Virtu, Valentina, Gaspar, Franci, María, Darío, Papá y Mamá. Gracias a Rose, mi amor, y a quien está por llegar.

Nunca me faltaron motivos para seguir adelante.

Alicante, 9 de junio de 2009 *David Tomás Díaz*

Índice general

Índice de figuras

Índice de tablas

1 Introducción

A comienzo de los años 70 se produjo un cambio en la forma de funcionar de las sociedades. En esta época, los medios de generación de riqueza se trasladaron de los sectores industriales a los sectores de servicios. La mayor parte de los empleos dejaron de asociarse a la fabricación de productos tangibles para centrarse en la generación, almacenamiento y procesamiento de todo tipo de datos. Comenzaba a gestarse la *sociedad de la información*. En este nuevo modelo de sociedad, la información pasó a convertirse en el activo intangible de mayor valor, siendo el motor del desarrollo y avance de las naciones. Surge una *economía del conocimiento* caracterizada por utilizar la información como elemento fundamental para la generación de riqueza.

Esta nueva sociedad ha encontrado en Internet y la World Wide Web (familiarmente, "la Web") su principal medio para el acceso e intercambio de datos, de información, de conocimiento. El sueño tecnológico iniciado por los primeros ordenadores a principios de los años 40, adquiere una nueva dimensión con la llegada de la red de redes: los ordenadores no sólo facilitan la gestión de información digital, sino también su distribución. A finales del siglo XX tuvo lugar la implantación masiva de Internet en la empresa y en los hogares. Se instauró de esta forma un nuevo socialismo documental donde cualquier persona puede acceder a la información, aportar sus propios contenidos y hacerlos accesibles al resto del mundo. Cambió el rol del usuario medio, pasando de mero consumidor a productor de información, dando lugar a una auténtica *explosión documental*. Cualquier cifra que diéramos ahora sobre el tamaño de la Web se vería duplicada en 8 meses.

Esta ventana al mundo ha supuesto un avance decisivo para el conocimiento humano. La posibilidad de acceder de forma sencilla al trabajo de otros ha permitido recoger sus frutos e ir un paso más allá. Sin embargo, la sobreabundancia de información presente en Internet, en lugar de fomentar un mayor conocimiento, consigue en ocasiones todo lo contrario. Tenemos acceso a datos y más datos, pero sin ningún criterio. Ante este maremágnum de información surge un problema obvio: la dificultad de localizar lo que estamos buscando. Ya no es suficiente con guardar los datos, sino que hay que organizarlos para hacerlos accesibles a los usuarios. Pero, ¿cómo acceder a toda esta información digitalizada? ¿Cómo utilizarla y hacerla productiva?

¿Cómo hacer que proporcione el máximo beneficio? La respuesta a estas preguntas se ha convertido en el santo grial de la era de la información.

En este gigantesco universo de información digital, hecho a la medida de las capacidades de cómputo masivo de los ordenadores, ya no existe catalogador o documentalista humano que nos pueda ayudar en nuestras búsquedas. La solución aportada por la comunidad científica a este problema son los sistemas de *recuperación de información* (RI) o *information retrieval* (Baeza-Yates y Ribeiro-Neto, 1999). La RI surgen ante la necesidad de los usuarios de escudriñar grandes cantidades de información digitalizada. Su objetivo es facilitar la materia prima para la obtención de conocimiento por parte del usuario, seleccionando la información relevante para ello. Estos sistemas reciben una petición de información por parte de un usuario y, como resultado de su ejecución, devuelven una lista de documentos. Estos textos se muestran ordenados según un criterio que intenta reflejar en qué medida cada documento contiene información que responde a las necesidades expresadas por el usuario. Los sistemas de RI más conocidos en la actualidad son aquellos que actúan sobre Internet y que permiten localizar información en la Web. Sirvan de ejemplo algunos populares motores de búsqueda como Google,[1] Ask[2] o AltaVista.[3]

¿Y cómo expresamos nuestras necesidades de información a estos sistemas? La forma que tienen los usuarios de un sistema de RI de expresar esta necesidad es mediante una *consulta*. En su forma más simple, una consulta estaría compuesta por una serie de *términos clave*, es decir, palabras o frases que representan un concepto concreto de la materia que estamos buscando (Broder, 2002). La forma "natural" que han adquirido los usuarios de los sistemas de RI para plasmar sus deseos de información es codificarlos en forma de términos clave, desproveyendo a sus consultas de toda estructura gramatical. Una necesidad de información que ante un humano verbalizaríamos como "¿Me podrías decir quién escribió la primera novela de la historia?", ante un sistema de RI acabaría convertida en una sucinta consulta del tipo "escritor primera novela historia". De esta forma, los resultados devueltos por los sistemas de RI son habitualmente secciones de documentos que presentan muchos términos en común con la consulta, pero que puede que no contengan la respuesta esperada.

Estos sistemas, pese a haber demostrado su innegable utilidad y su capacidad para localizar información relevante para los usuarios de la Web, presentan serias carencias. Hagamos el experimento mental de trasladar el comportamiento de un sistema de RI a una situación del mundo real. ¿Qué esperamos que ocurra cuando a una persona le hacemos una pregunta para la que no sabemos la respuesta? Lo normal en esta situación sería que, la

[1] http://www.google.com.
[2] http://www.ask.com.
[3] http://www.altavista.com.

persona a la que se le preguntó, consultara algún tipo de información (por ejemplo, un libro o una revista) para encontrar algún texto que pudiera leer y entender, permitiéndole determinar la respuesta a la pregunta que se le hubiera formulado. Llegado a este punto, podría darnos la respuesta. De igual modo, podría indicarnos dónde la encontró, lo que nos ayudaría a adjudicar un cierto nivel de confianza a esa respuesta. Lo que nunca esperaríamos de la persona consultada, si queremos catalogarla de eficaz, es que simplemente nos devolviera un libro o un montón de documentos en los que piensa que se hallará la respuesta. Desafortunadamente, esta es la situación con la que deben contentarse los usuarios de los sistemas de RI.

Las características que definieron las líneas de investigación en sistemas de RI presentan serios inconvenientes a la hora de facilitar la obtención de respuestas concretas a preguntas precisas formuladas por los usuarios. Una vez que el usuario recibe del sistema la lista de documentos relevantes a su consulta, todavía le queda la ardua tarea de revisar cada uno de éstos para, en primer lugar, comprobar si están realmente relacionados con la información solicitada y, en segundo lugar, leerlos con el fin de localizar está información puntual en su interior.

Estos inconvenientes y el creciente interés en sistemas que afrontaran con éxito la tarea de localizar respuestas concretas en grandes volúmenes de información, dejaron la puerta abierta a la aparición de un nuevo campo de investigación conocido como *búsqueda de respuestas* (BR) o *question answering*. La BR tiene como finalidad encontrar respuestas concretas a necesidades precisas de información formuladas por los usuarios en lenguaje natural, es decir, mediante el lenguaje humano.

Aunque muchos sistemas de RI sugieren que el usuario puede expresar sus necesidades de información mediante lenguaje natural, internamente estos sistemas se encargan de transformar nuestra petición en una lista de términos clave. Para comprobar este hecho podemos hacer un sencillo experimento: introduciendo la pregunta "¿Quién es el más fuerte del mundo?" en un destacado buscador como Google, veremos que en alguna de las primeras entradas se devuelve información sobre el material más fuerte del mundo, o la bebida más fuerte del mundo. El buscador intenta localizar aquellos documentos en los que aparecen términos como "fuerte" o "mundo", ignorando que realmente estamos preguntando por una persona.

Los sistemas de BR, por contra, nos permiten consultar al sistema en nuestro propio idioma, como interrogaríamos a otro ser humano. Ante una necesidad de información concreta no emplearíamos un lenguaje críptico y formal, sino nuestra propia lengua. Ya no es el hombre el que se adapta al lenguaje de la máquina, sino que ésta trata de comprender al hombre en su propio idioma. Conseguir este objetivo implica codificar el conocimiento humano, expresado a través nuestro complejo y ambiguo lenguaje, a un

lenguaje formal[4] que sea capaz de entender el ordenador. Las consultas que se realizan a los sistemas de BR son preguntas sintáctica y semánticamente correctas, cuyo significado se entiende como la petición de una necesidad de información. Las consultas adquieren así un significado propio y muchas veces único en un contexto determinado, expresando de forma más rica nuestra necesidad de información y ofreciendo una menor ambigüedad. Establecer un modelo de comunicación natural con el sistema, exige un esfuerzo extra a los desarrolladores, que deben de lidiar con los problemas inherentes al lenguaje natural, como la ambigüedad y la variación lingüística. Hay muchas formas de preguntar por una misma información. A veces las variaciones son simples reordenaciones de palabras de la pregunta y en otros casos las variaciones pueden depender del contexto en el que se formula la pregunta o del conocimiento del que pregunta.

En los sistemas de BR, un primer paso para poder devolver la respuesta solicitada por el usuario es analizar la pregunta y comprenderla, saber por qué se nos está preguntando: ¿por una persona?, ¿por una fecha?, ¿por una cifra?, ¿por algún tipo de objeto o evento? En el ejemplo anterior de la consulta formulada a Google, saber que se está preguntando por una persona, y no por ninguna otra entidad, permitiría al sistema descartar algunos de los resultados incorrectamente devueltos.

La *clasificación de preguntas* (CP) o *question classification* se ha desmarcado como una tarea en sí misma dentro del mundo del procesamiento del lenguaje natural y de la BR. Su objetivo es identificar de forma automática qué se nos está preguntando, categorizando las preguntas en diferentes clases semánticas en función del tipo de respuesta esperada. Así, ante una pregunta como "¿Quién es el presidente de los Estados Unidos?", un sistema de CP diría que nos están preguntando por una persona; ante una petición como "¿Dónde está la Torre Eiffel?" identificaría que se está preguntando por un lugar; finalmente, la pregunta "¿Cuándo nación Mozart?" esperaría como respuesta una fecha. En estos ejemplos, *persona*, *lugar* o *fecha* representan la clase semántica de la respuesta esperada. Conocerla nos permite acotar el conjunto de respuestas posibles: nadie quiere preguntar "¿Quién protagonizó Toro Salvaje?" y obtener por respuesta "25 de diciembre". La CP se ha convertido en un área de fuerte interés en los últimos tiempos, primordialmente debido a su aplicación en los sistemas de BR.

Nuestro trabajo de tesis se centra en este marco, en el de la CP aplicada a los sistemas de BR. Vamos a presentar una serie de aproximaciones al desarrollo de sistemas de CP flexibles. El concepto de flexibilidad se entiende aquí como la capacidad del sistema para adaptarse de forma sencilla a

[4]Un lenguaje formal es un lenguaje artificial creado por el hombre y que esta formado por símbolos y formulas. Su objetivo fundamental es formalizar la programación de computadoras o representar simbólicamente el conocimiento. Los lenguajes de programación o el lenguaje de la lógica matemática son ejemplos de lenguajes formales.

diferentes idiomas y dominios. Para ello, basaremos nuestros desarrollos en técnicas de aprendizaje automático sobre corpus, que permitan a nuestros sistemas aprender a través de la experiencia.

En lo que resta de capítulo, hablaremos de las motivaciones que impulsan esta tesis y de los objetivos planteados, así como de la metodología seguida y la estructura de este trabajo.

1.1. Motivaciones

Los sistemas de BR se han convertido en objeto de amplio estudio en la última década, gracias en parte a los distintos foros internacionales en este campo: TREC[5] (Voorhees, 2001b), CLEF[6] (Vallin et al., 2006) y NTCIR[7] (Kando, 2005). Estos foros han marcado las pautas de desarrollo de la BR, estableciendo los retos a superar y el marco de evaluación y comparación de las diferentes aproximaciones entre sí. El interés en este campo no ha parado de crecer, abriéndose paso en la sociedad a través de numerosas proyectos e iniciativas, tanto públicas como privadas.

Los sistemas de CP, como parte fundamental de la tarea de BR, resultan de vital importancia en las tecnologías actuales de recuperación de información. En los últimos años, la CP ha adquirido suficiente relevancia como para ser considerada una tarea independiente de los sistemas de BR y evaluable en sí misma. Trabajos como el de Li y Roth (2002) han permitido establecer un marco de trabajo en el que evaluar y comparar estos sistemas entre sí.

Aunque aparentemente la tarea de clasificar preguntas resulte sencilla e intuitiva para un humano, hay diversos factores que afectan al funcionamiento de los sistemas de clasificación automáticos y a su robustez. La superación de las dificultades que enumeraremos en los siguientes párrafos, así como la ya comentada importancia adquirida por los sistemas de CP, han motivado el trabajo desarrollado en esta tesis.

En los siguientes párrafos vamos a plantear las dificultades que deben afrontar los sistemas actuales de CP, cuya superación ha motivado este trabajo:

- **Procesamiento del lenguaje natural**. Los sistemas de CP trabajan con peticiones en lenguaje natural, por lo que deben afrontar los problemas derivados de la variación y la ambigüedad del lenguaje humano.

  La *ambigüedad lingüística* lleva a que algunas palabras tomen distinto significado dependiendo del contexto en el que tengan lugar. Un

[5] Text REtrieval Conference: http://trec.nist.org.
[6] Cross Language Evaluation Forum: http://clef-campaign.org.
[7] NII-NACSIS Test Collection for IR Systems: http://research.nii.ac.jp/ntcir/.

ejemplo de ambigüedad sería la que se da en las preguntas "¿Cuál es el nombre científico del gato común?" y "¿Quién inventó el gato hidráulico?". Aquí, la palabra "gato" tiene dos significados claramente diferentes.[8]

El otro problema citado es la *variación del lenguaje*. Los seres humanos pueden manejar fácilmente diferentes expresiones, ya que tienden a centrarse en el significado de la expresión (semántica), y no en la forma exacta de expresarla (sintaxis). Por lo tanto, se tiende a usar el lenguaje en toda su complejidad para expresar preguntas, sabiendo que cuando son formuladas a otras personas las entenderán y serán capaces de responderlas. Esta variación lingüística lleva a que una misma pregunta pueda presentar estructuras muy diversas: "¿Quién escribió La Divina Comedia?", "¿Quién es el autor de La Divina Comedia?", "Dame el nombre de la persona que escribió La Divina Comedia" y así hasta un sinfín de posibles variantes.

Solucionar los problemas derivados del tratamiento del lenguaje natural es uno de los retos que debe de afrontar cualquier sistema de CP.

- **Tratamiento del multilingüismo**. La mayoría de sistemas de CP desarrollados están orientados al idioma inglés. Las herramientas y recursos que emplean estos sistemas suelen ser dependientes del idioma, no estando disponibles para todos ellos. Esto implica que si se quieren reproducir los resultados de estos sistemas para un idioma diferente, primero ha de resolverse el problema de disponer de las herramientas y recursos apropiados para el nuevo idioma.

  Son muy escasos las aproximaciones a CP que han afrontado el problema de trabajar sobre múltiples idiomas, centrándose los experimentos realizados en corpus pequeños (Solorio et al., 2004). Por ello, otro de los retos que se nos plantea a la hora de desarrollar sistemas de CP es la creación de sistemas que sean capaces de adaptarse fácilmente a diferentes idiomas, sin necesidad de emplear o desarrollar costosos recursos y herramientas.

- **Aplicación a diferentes dominios**. Los estudios realizados en el campo de la CP se han centrado mayoritariamente en un único conjunto de datos de evaluación. Los resultados ofrecidos por estos sistemas están optimizados para el dominio de aplicación, siendo cuestionable su funcionamiento sobre otros conjuntos de datos. La misma problemática que surgía con la aplicación a diferentes idiomas

[8] Este es un ejemplo de ambigüedad léxica pura ya que, aunque el significado de la palabra es diferente, desempeña un mismo papel sintáctico en la oración (como sustantivos).

tiene lugar para la aplicación entre dominios. La portabilidad de los sistemas resulta dificultosa, ya que el uso de muchas de las herramientas y recursos que se emplean para su desarrollo ligan el sistema al dominio de aplicación. Existe una necesidad de desarrollar y evaluar sistemas que sean fácilmente adaptables a diferentes conjuntos de datos y dominios.

- **Taxonomías refinadas**. El objetivo de la CP aplicada a la tarea de BR es asignar una clase semántica a una pregunta dada. Las clases que se asignan vienen prefijadas en una taxonomía. La tendencia actual de los sistemas de BR es que las taxonomías de preguntas se hagan cada vez más grandes y refinadas, para mejorar así la precisión a la hora de localizar la respuesta esperada.

  Esto introduce la necesidad de sistemas de clasificación más precisos. La obtención de este tipo de sistemas es dificultosa y requiere de un gran esfuerzo por parte de los desarrolladores, ya sea para definir reglas manuales de forma precisa o para recopilar grandes conjuntos de datos de entrenamiento. El estudio de aproximaciones de bajo coste (tanto computacional como humano) sobre taxonomías refinadas, es otra de las motivaciones de esta tesis.

- **Credibilidad de los resultados**. En los trabajos desarrollados dentro del campo de la CP, son escasas las ocasiones en las que los resultados obtenidos van acompañados de tests estadísticos que indiquen si las mejoras aportadas son realmente significativas (Sundblad, 2007). Esto nos puede llevar a pensar que, en algunas ocasiones, estas mejoras pueden no ser más que fruto del azar, poniendo en entredicho los resultados obtenidos y las conclusiones que se derivan de ellos.

  Se hace necesario un estudio riguroso de las diferentes aproximaciones y experimentos realizados que nos indique, de forma fehaciente, la validez de los resultados expuestos en este campo de investigación.

1.2. Objetivos

Todas las motivaciones y dificultades expuestas en la sección anterior nos sirven para establecer el principal objetivo de esta investigación: el desarrollo de sistemas de CP fácilmente adaptables a diferentes idiomas y dominios. Alcanzar este objetivo no es un camino sencillo, y nos lleva a establecer dos premisas fundamentales que vamos respetar durante el desarrollo de nuestro trabajo:

- *Los sistemas aprenden por sí mismos.* Para que los sistemas sean rápidamente aplicables a nuevos idiomas y dominios, debemos evitar la dependencia del conocimiento inducido por expertos humanos.

Para cumplir esta premisa basaremos nuestros sistemas en técnicas de aprendizaje automático, es decir, construiremos sistemas que sean capaces de mejorar su funcionamiento de forma automática a través de la experiencia adquirida mediante conjuntos de datos (*corpus*). Queremos de esta manera tener sistemas de propósito general que sean capaces de adaptarse a las circunstancias, en lugar de tener que readaptarlos de forma explícita para cada situación particular que se presente, para cada idioma o dominio al que se quiera aplicar.

- *El aprendizaje no requiere de recursos lingüísticos complejos.* Debemos evitar ser dependientes de herramientas o recursos ligados al idioma o al dominio. Para cumplir esta segunda premisa, nuestros sistemas se basan en características textuales superficiales extraidas de forma automática a partir de corpus etiquetados y no etiquetados. De esta manera evitamos el uso de herramientas o recursos lingüísticos que comprometan la portabilidad de nuestros desarrollos.

Este objetivo principal puede desglosarse en una serie de objetivos intermedios:

- Desarrollar sistemas de CP capaces de adaptarse a diferentes idiomas empleando las mismas características de aprendizaje para todos ellos.
- Evaluar los sistemas de CP sobre diferentes conjuntos de datos obtenidos a partir de fuentes heterogéneas provenientes de distintos dominios.
- Realizar una evaluación empírica extensa para determinar los mejores algoritmos y características de aprendizaje que permitan cumplir nuestros objetivos.
- Establecer los mecanismos necesarios para extraer información a partir de textos no etiquetados, que permita la mejora del proceso de aprendizaje sin recurrir a costosas herramientas o recursos lingüísticos.
- Estudiar las ventajas y desventajas de nuestra aproximación y comparar su rendimiento con respecto a otros sistemas que hacen un uso intensivo de recursos y herramientas lingüísticas.
- Desarrollar una aproximación mínimamente supervisada a la CP sobre taxonomías refinadas. Buscamos así evitar la necesidad de grandes corpus de entrenamiento para llevar a cabo esta tarea.

Otros objetivos secundarios derivados de las motivaciones planteadas previamente son:

- Definir claramente la tarea de CP como una tarea en sí misma, evaluable con independencia de los sistemas de BR.
- Establecer el estado de la cuestión de una disciplina que ha evolucionado notablemente en los últimos años.
- Enumerar otras aplicaciones, más allá de la BR, donde los sistemas de CP resultan de utilidad.
- Desarrollar corpus de preguntas para el entrenamiento y evaluación de sistemas basados en aprendizaje automático. Estos corpus abarcaran diferentes idiomas y dominios para poder evaluar la capacidad de adaptación de los sistemas aquí planteados.
- Que este trabajo sirva como iniciación y acercamiento a la tarea general de clasificación mediante aprendizaje automático, exponiendo los problemas, algoritmos, metodologías y evaluaciones usadas habitualmente en este campo.

1.3. Metodología

Vamos a describir a continuación la metodología seguida para la consecución de los objetivos marcados en el punto anterior. Hemos estructurado nuestro trabajo en tres aproximaciones diferentes, de forma que nos permita abarcar la totalidad de los objetivos planteados en el punto anterior.

Primera aproximación: CP supervisada basada en n-gramas. En este primera aproximación desarrollamos un sistema de CP que aprende de forma automática a partir de información obtenida estrictamente de un corpus de entrenamiento. Ningún otro tipo de herramienta o recurso lingüístico es requerido, dando como resultado un sistema flexible. Este estudio nos va a permitir establecer un sistema de referencia para aquellas situaciones en que únicamente se dispone de un corpus para el aprendizaje. Llevar a cabo esta aproximación implica la realización de diversas tareas:

- Determinar cuál es el algoritmo más apropiado para la tarea de CP.
- Analizar diferentes características de aprendizaje a nivel de palabra obtenidas exclusivamente de los datos de entrenamiento.
- Desarrollar corpus de entrenamiento y evaluación en diferentes idiomas para la tarea de clasificación multilingüe.
- Desarrollar corpus de entrenamiento y evaluación en diferentes dominios para la tarea de clasificación en dominios abiertos y restringidos.

- Evaluar los algoritmos y características definidas sobre los diferentes corpus desarrollados.

Además, vamos a reforzar esta aproximación empleando diversas técnicas de selección de características. Estas técnicas están basadas en información estadística sobre los propios datos de entrenamiento y no afectan, por tanto, a la portabilidad del sistema a diferentes idiomas o dominios.

Segunda aproximación: CP semisupervisada explotando textos no etiquetados. En esta segunda aproximación, vamos a enriquecer el modelo básico definido en nuestra primera aproximación. Para ello, completaremos la información extraida del conjunto de entrenamiento añadiendo información semántica externa en la fase de aprendizaje. Para incorporar esta información, y siguiendo las premisas establecidas en los objetivos, emplearemos únicamente texto no etiquetado que puede ser adquirido de forma automática de la Web. De esta forma, mejoramos la capacidad del sistema empleando datos no etiquetados. Esto da lugar a una aproximación semisupervisada a la CP. Las tareas a realizar en esta aproximación son:

- Desarrollar una aproximación que permita incorporar información semántica partiendo de texto no etiquetado. Esto mantendrá intacta la independencia de nuestro sistema con respecto a otros recursos y herramientas.
- Comparar esta aproximación con otros sistemas que incorporan información semántica proveniente de recursos lingüísticos complejos.
- Evaluar el sistema sobre diferentes idiomas para comprobar su capacidad de adaptación.

Tercera aproximación: CP mínimamente supervisada sobre taxonomías refinadas. El rendimiento de los sistemas basados en corpus, depende de forma crítica del tamaño del conjunto de datos de entrenamiento. Cuando el número de clases posibles que debe de aprender a discriminar el sistema aumenta, resulta más complicado para los corpus de entrenamiento dar una cobertura adecuada a todas las clases posibles. En esta tercera aproximación afrontamos el problema de la CP sobre taxonomías refinadas en ausencia de datos de entrenamiento. A partir de un pequeño conjunto de semillas iniciales definidas por el usuario para cada clase, el sistema aprenderá a discriminar de forma automática entre ellas a partir de información adquirida de forma automática de la Web. De esta forma obtenemos una aproximación mínimamente supervisada para la clasificación sobre taxonomías refinadas, que tradicionalmente requeriría de grandes

corpus de entrenamiento para ofrecer una cobertura adecuada al problema. Las tareas a realizar en esta fase son:

- Definir un modelo para la adquisición automática de muestras de entrenamiento. Evitamos de esta manera la adquisición de grandes conjuntos de datos de entrenamiento.
- Desarrollar un conjunto de datos de evaluación sobre una taxonomía refinada que nos permita medir el rendimiento del sistema.
- Evaluar el sistema sobre diferentes idiomas.
- Comparar nuestra aproximación con los sistemas de aprendizaje existentes, valorando las ventajas aportadas por nuestra propuesta.

1.4. Estructura de la tesis

Vamos a esbozar a continuación la organización y contenido del resto de capítulos que conforman este trabajo de tesis:

- En el **capítulo 2** presentaremos la problemática que afrontan las aplicaciones que trabajan con lenguaje natural. Describiremos formalmente los sistemas de CP y los situaremos dentro del campo de la BR. Se ofrecerán también referencias a otras tareas donde los sistemas de CP resultan de utilidad.
- En el **capítulo 3** describiremos en detalle los sistemas de CP basados en aprendizaje automático y presentaremos el pasado y presente de estos sistemas desde diferentes perspectivas: según los algoritmos, según las características de aprendizaje, según los recursos empleados y según los corpus y taxonomías sobre los que aprenden y clasifican.
- En el **capítulo 4** mostraremos el trabajo realizado para completar la *primera aproximación* de nuestra metodología. Haremos un estudio comparativo de diferentes corpus y características de aprendizaje, seleccionando las mejores configuraciones para la obtención de sistemas de CP flexibles.
- En el **capítulo 5** afrontaremos las tareas necesarias para completar la *segunda aproximación* de la metodología. Plantearemos un método para la incorporación de información semántica al proceso de aprendizaje que mejore la funcionalidad del sistema de CP, todo ello sin comprometer su capacidad de adaptarse a diferentes idiomas y dominios.

- En el **capítulo 6** abarcaremos los retos planteados en la *tercera aproximación*. Presentaremos un sistema para la CP sobre taxonomías refinadas empleando un algoritmo propio. Este algoritmo nos servirá para evitar el desarrollo de grandes conjuntos de datos para el entrenamiento del sistema.
- Por último, en el **capítulo 7** mostraremos las conclusiones, aportaciones y trabajos futuros, así como la producción científica derivada de esta investigación.

2

La clasificación de preguntas

Este capítulo nos va a servir para introducir el problema de la *clasificación de preguntas* (CP). Para ello, comenzaremos dando una definición de lo que es una *pregunta* desde un punto de vista lingüístico. Veremos qué problemas se derivan del tratamiento automático del lenguaje humano en general y de las preguntas en particular, así como las distintas aproximaciones existentes a esta tarea. Haremos una definición formal del problema de la clasificación automática de preguntas y la ubicaremos dentro del contexto de los sistemas de *búsqueda de respuestas* (BR). Completaremos esta descripción hablando de otras tareas en las que los sistemas de CP han demostrado su utilidad.

Esta introducción nos va servirá para establecer las bases de los sistemas de CP que desarrollaremos en los siguientes capítulos y justificar las decisiones de diseño tomadas.

2.1. ¿Qué es una pregunta?

Cualquier ser humano sabe lo que es una pregunta y cómo distinguirla cuando tiene lugar en el discurso. Pero si nos pidieran que explicásemos lo que son las preguntas o cómo responderlas, probablemente tendríamos que pararnos y pensar qué decir. En el habla común, el término *pregunta* se usa en al menos tres formas diferentes que distinguiremos aquí para evitar confusiones (Groenendijk y Stokhof, 1997):

- Es un tipo particular de oración, caracterizada por el orden de las palabras,[1] la entonación, los signos de interrogación y la ocurrencia de pronombres interrogativos. Es lo que conocemos como *oración interrogativa*. "¿Quién ganó el Oscar al mejor actor principal en 2007?" o "¿Entiendes?" son ejemplos de oraciones de este tipo. El estudio

[1] Esto se cumple para idiomas como el inglés o el español, pero no para otros como el chino, donde estas oraciones no sufren ningún cambio de orden en las palabras aunque sí incluyen partículas interrogativas específicas que indican la presencia de este tipo de oración.

de las oraciones interrogativas pertenece a la parte sintáctica de la lingüística.

- Es el acto hablado que se lleva típicamente a cabo al formular oraciones interrogativas. Denota una petición de información por parte del hablante a un destinatario (una petición a responder la pregunta). Este acto del habla se conoce formalmente como *acto interrogativo*. El estudio del acto interrogativo pertenece a la pragmática, y más concretamente a la teoría de los actos del habla (Austin, 1962).

- Hace referencia a la cosa que se está preguntado, y que, como consecuencia, debe ser contestada. Este objeto puede considerarse como el contenido semántico (o sentido) de una oración interrogativa, o como el contenido de un acto interrogativo.

En los sistemas de CP que definiremos formalmente en la sección 2.3, se integran las tres acepciones dadas del término *pregunta*. La entrada de estos sistemas es una oración interrogativa, con la intención de realizar una petición de información. La salida que proporciona es la clase semántica de la pregunta, indicando la *cosa* por la que se ha preguntado.

Hay que señalar que no todas las preguntas toman la forma de una oración interrogativa. Tanto en la teoría lingüística tradicional como en la contemporánea, las formas en las que se puede presentar una pregunta son básicamente dos (Higginbotham, 1995): directas ("¿Quién es Jorge?") o indirectas ("Quiero saber quién es Jorge"). Las *oraciones interrogativas directas* tienen signos de interrogación[2] e incluyen normalmente una partícula interrogativa.[3] Las *oraciones interrogativas indirectas* no llevan signos interrogativos y las preguntas se formulan mediante una oración en forma imperativa o enunciativa, en lugar de usar la correspondiente interrogativo. Este tipo de interrogativas son construcciones en las que la cláusula subordinada está encabezada por un elemento interrogativo (Contreras, 1999).

Los sistemas de CP deben lidiar con preguntas formuladas empleando cualquiera de las múltiples construcciones posibles que ofrece el lenguaje humano. En la siguiente sección vamos a realizar una introducción al tratamiento automático del lenguaje humano y las cuestiones que éste aborda.

[2] Al principio y al final (¿?) en español o sólo al final (?) en idiomas como el inglés, el italiano o el catalán.

[3] Para español tenemos "qué", "dónde", "cómo", "por qué", "quién", "quiénes", "cuál" y "cuáles". Para inglés tenemos "*what*", "*where*", "*how*", "*when*", "*why*", "*who*" y "*which*". Son las conocidas como *wh-words*.

2.2. El procesamiento del lenguaje natural

Ya en el capítulo anterior indicábamos que los sistemas de CP trabajan con preguntas formuladas en lenguaje natural. Cuando hablamos de *lenguaje natural*, nos referimos al lenguaje hablado o escrito, empleado por humanos para propósitos generales de comunicación. Emplearemos este término para diferenciarlo de los *lenguajes formales*, construidos de forma artificial, como aquellos empleados en programación o en lógica formal. La forma habitual de comunicarnos con un ordenador es mediante lenguajes formales, que pueden ser interpretados o compilados hasta transformarse en una secuencia binaria de ceros y unos, la única información que en última instancia es capaz de procesar la circuitería de un ordenador.

El *procesamiento del lenguaje natural* (PLN) (Moreno et al., 1999) es una disciplina que surge para que la comunicación entre el hombre y la máquina resulte más fluida, para que sea la máquina la que se adapte al lenguaje del hombre y no al contrario. El PLN es una parte esencial de la *inteligencia artificial*[4] (IA) que investiga y formula mecanismos computacionales que faciliten la interrelación entre hombres y máquinas, permitiendo una comunicación más fluida y menos rígida que los lenguajes formales.

El PLN afronta tareas que, pese a su aparente simplicidad para los humanos, esconden una elevada complejidad para su resolución de forma automática desde una perspectiva computacional. Su objetivo es el estudio de los problemas derivados de la generación y comprensión automática del lenguaje natural. Lo que busca esta disciplina es construir sistemas y mecanismos que permitan la comunicación entre personas y máquinas mediante lenguaje naturales, de forma que los usuarios puedan llegar a comunicarse con el ordenador de la misma forma que lo harían con otro humano.

2.2.1. Un poco de historia

La investigación en PLN tiene una larga trayectoría, remontándose a finales de los años 40 con la aparición de los primeros sistemas de traducción automática. Estos sistemas fracasaron debido a la escasa potencia de los ordenadores y la poca sofisticación lingüística de los algoritmos empleados (pensaban que podrían resolver el problema a partir de grandes diccionarios y su consulta automática). Es a partir de estas experiencias cuando se hace patente la naturaleza de los problemas del tratamiento del lenguaje humano y las limitaciones tanto teóricas como técnicas de las aproximaciones

[4]La IA es la ciencia que estudia el diseño y desarrollo de algoritmos para máquinas con el fin de imitar el comportamiento y la comprensión humana. La investigación en este campo se centra en la producción de máquinas para la automatización de tareas que requieran un comportamiento inteligente.

existentes. Fue en los años 60 y 70 cuando los estudios en PLN dieron lugar a las primeras interfaces en lenguaje natural a bases de datos, con sistemas como LUNAR (Woods, 1970), que permitía a los geólogos formular preguntas sobre datos de análisis químico de rocas lunares y muestras de suelo. Estos esfuerzos obtuvieron un cierto grado de éxito. Las mejoras en las infraestructuras informáticas y la capacidad de cómputo de los ordenadores dieron pie, durante la década de los 80 y principios de los 90, al resurgir de la investigación en el terreno de la traducción automática.

Son abundantes los estudios realizados en el campo del PLN en los últimos años y numerosas las ramas de investigación surgidas. Entre las principales tareas en este campo podemos destacar:

- *Traducción automática* (Hutchins y Somers, 1992)
- *Recuperación de información* (Baeza-Yates y Ribeiro-Neto, 1999)
- *Reconocimiento del habla* (Junqua y Haton, 1995)
- *Resúmenes automáticos* (Mani, 1999)
- *Interfaces en lenguaje natural a bases de datos* (Androutsopoulos et al., 1995)
- *Detección de autoría* (Juola, 2007)
- *Análisis de sentimientos* (Pang y Lee, 2008)
- *Búsqueda de respuestas* (Paşca, 2003)
- *Extracción de información* (Cardie, 1997)
- *Desambiguación del sentido de las palabras* (Agirre y Edmonds, 2006)
- *Implicación textual* (Dagan et al., 2006)
- *Clasificación de textos* (Sebastiani, 2002)
- *Reconocimiento de entidades* (Palmer y Day, 1997)

Consecuentemente, las publicaciones y foros sobre el tema han sido muy numerosos. Una de las revistas más conocidas en este campo es *Computational Linguistics*, la revista oficial de *The Association for Computational Linguistics*,[5] que desde 1974 está dedicada exclusivamente a investigaciones cobre el diseño y análisis de sistemas de PLN. Ofrece información sobre aspectos computacionales de investigación en lenguaje, lingüística y psicología del procesamiento del lenguaje.

[5] http://www.aclweb.org.

2.2.2. Niveles de análisis del lenguaje

Existen desarrollos de PLN que sobre todos y cada uno de los diferentes niveles de análisis posibles del lenguaje. Vamos a nombrar a continuación estos niveles y algunos ejemplos de herramientas que en ellos se dan:

- **Análisis morfológico**. El conocimiento morfológico proporciona las herramientas para formar palabras a partir de unidades más pequeñas. Abarca el análisis de las palabras para la extracción de raíces, rasgos flexivos y unidades léxicas compuestas, entre otros fenómenos morfológicos. En este nivel encontramos aplicaciones como los *lematizadores* y los *analizadores morfológicos* o *part-of-speech taggers*.

- **Análisis sintáctico**. El conocimiento sintáctico establece cómo se deben combinar las palabras para formar oraciones correctas y estudiar cómo se relacionan éstas entre sí. Lleva a cabo el análisis de la estructura sintáctica de la frase mediante una gramática de la lengua en cuestión. Dentro del campo del PLN este tipo de análisis se acomete desde dos puntos de vista diferentes. Por una parte tenemos los *analizadores sintácticos parciales* o *shallow parsers*, que identifican estructuras como sintagmas nominales, verbales y preposicionales. Permiten recuperar información sintáctica de forma eficiente y fiable a partir de texto no restringido (Abney, 1997). Por otra parte tenemos los *analizadores sintácticos totales* o *deep parsers*, que obtienen el árbol sintáctico completo de la oración. Estos últimos, debido a su complejidad y coste computacional, son menos utilizados que los anteriores.

- **Análisis semántico**. El conocimiento semántico hace referencia al conjunto de significados de las palabras y cómo éstas se combinan para formar el significado completo de una oración. Incluye la extracción del significado de la frase y la resolución de ambigüedades léxicas y estructurales. Dentro de este nivel encontramos herramientas como los *desambiguadores del sentido de las palabras*, los *reconocedores de entidades* o las *bases de datos léxicas*.

- **Análisis pragmático**. El conocimiento pragmático ayuda a interpretar la oración completa dentro de su contexto. Aborda el análisis del texto más allá de los límites de la oración para determinar, por ejemplo, los antecedentes referenciales de los pronombres. Los sistemas de *resolución de la anáfora* son un ejemplo de aplicación de PLN en este nivel de análisis.

2.2.3. El problema de la ambigüedad y la variación lingüística

Cualquier aplicación que trabaje con el lenguaje humano debe afrontar un principio general: la lengua es variable y se manifiesta de modo variable. Este principio supone que los hablantes recurren a elementos lingüísticos distintos para expresar contenidos distintos y, a la vez, que se pueden usar elementos lingüísticos diferentes para decir una misma cosa.

La libertad que ofrece el lenguaje humano da pie a la existencia de dos problemas fundamentales que deben de afrontar la mayoría de sistemas de PLN y que merman su efectividad (Vállez y Pedraza-Jiménez, 2007): la *variación* y la *ambigüedad* lingüística.

La *variación lingüística* hace referencia a la posibilidad de usar diferentes términos o expresiones para comunicar una misma idea, es decir, el fenómeno de la *sinonimia*. El lenguaje humano permite usar elementos lingüísticos distintos para expresar distintos contenidos, así como usar elementos lingüísticos diferentes para referirnos a una misma cosa. Esta variación se puede dar en prácticamente todos los niveles de la lengua, desde el más concreto, como el *fonético-fonológico* al más amplio, como el *discurso*, pasando por la *gramática* y el *léxico*. Términos como "chico", "chaval" o "muchacho" son ejemplos de variación léxica, donde palabras diferentes apuntan a un mismo concepto. Ejemplos como "Estoy que me caigo", "No puedo más" o "Estoy hecho polvo" son expresiones que reflejan una misma idea con una variación lingüística considerable entre ellas. Igualmente, y centrándonos en nuestro objeto de estudio, para expresar nuestros deseos a través de una pregunta las combinaciones posibles de palabras y estructuras son ilimitadas: "¿Cómo se llama el autor de La Regenta?", "¿Quién escribió La Regenta?" o "¿La Regenta es una obra de qué autor?", son preguntas sintácticamente muy distintas pero semánticamente equivalentes.

El segundo problema mencionado, el de la *ambigüedad lingüística*, tiene lugar cuando un término o expresión admite diferentes interpretaciones. Esta ambigüedad puede darse en todos los niveles de análisis lingüístico expuestos anteriormente:

- A nivel léxico, una misma palabras puede tener varios significados. En este campo encontraríamos problemas como la *homonimia*[6] o la *polisemia*.[7] Un ejemplo clásico de polisemia se da en la oración "Me dejé el periódico en el banco", donde "banco" puede hacer referencia a un tipo de asiento o a una entidad bancaria.

[6] Palabras a las que les corresponden, según el contexto, diferentes significados. Para deshacer la ambigüedad hay que apoyarse en el contexto.

[7] Varios sentidos para una sola palabra. Es el resultado de la evolución histórica de una lengua, que con el paso del tiempo va unificando diferentes palabras en una sola forma por evolución fonética.

- A nivel estructural, se requiere información semántica para desambiguar la dependencia de los sintagmas preposicionales que conducen a la construcción de distintos árboles sintácticos. Un ejemplo clásico de este tipo de ambigüedad es "Vi a María por la ventana con el catalejo", que admite una doble interpretación: "La vi mediante un catalejo" o "Vi que llevaba un catalejo".

- A nivel pragmático, las oraciones no siempre significan lo que textualmente se está diciendo. En determinadas circunstancias el sentido de las palabras que forman la oración tiene que interpretarse a un nivel superior, recurriendo al contexto es que se formula la frase. Elementos como la ironía juegan un papel relevante en la interpretación del mensaje. Una expresión tan habitual como "Se moría de risa" no debe interpretarse en sentido literal, sino figurado.

Algunas herramientas de PLN se aplican de forma específica a la resolución de los dos problemas citados. Técnicas como la detección de paráfrasis o la implicación textual afrontan el problema de la variación de la lengua, mientras que los sistemas de desambiguación del sentido de las palabras tratan de resolver el problema de la ambigüedad semántica.

2.2.4. Aproximaciones al tratamiento del lenguaje natural

Las dificultades comentadas en el punto anterior hacen del PLN una disciplina viva y con numerosos retos que afrontar. Existen dos filosofías que pugnan (o cooperan) por resolver los problemas derivados del tratamiento del lenguaje humano. Por una parte está la aproximación estadística al problema, y por otra el enfoque lingüístico. Ambas propuestas difieren considerablemente, aunque en la práctica se suelen utilizar técnicas provenientes de ambos enfoques:

- **Enfoque estadístico**. En este enfoque el texto se contempla como un conjunto de palabras, sin tener en consideración el orden, la estructura, o el significado de las mismas. Desde esta concepción del lenguaje natural, las frecuencias de aparición de las palabras y sus distribuciones son suficiente fuente de información para los sistemas. Es el enfoque conocido como *bolsa de palabras* o *bag-of-words* (BOW) (Manning y Schütze, 1999). La simplicidad y eficacia de estos modelos los han dotado de gran popularidad en numerosas tareas de PLN, como la RI o la traducción automática.

  Este tipo de técnicas suelen aplicarse sobre textos preprocesados, es decir, textos "limpios" en los que se han eliminado etiquetas (*stripping*), normalizado las palabras (mediante conversión a minúsculas, manejo de fechas, números y abreviaturas, eliminación de *palabras*

vacías o *stopwords*,[8] identificación de n-gramas, etc.) o lematizado,[9] con la intención de homogeneizar las estimaciones estadísticas de las palabras del texto.

Esta filosofía es la que siguen sistemas de RI como Google. Bajo este planteamiento, el sistema no necesita "entender" (desde un punto de vista humano) el contenido de una página Web para saber si ésta se ajusta a las necesidades del usuario: si las estadísticas de los enlaces que apuntan a la página dicen que es mejor, entonces es suficiente. No es necesario ningún tipo de análisis semántico. En este caso, la cantidad ingente de información presente en Internet permite inferir "verdades" por pura correlación.

El aumento de capacidad de los ordenadores actuales permite esta forma de afrontar el problema del tratamiento del lenguaje humano. Para algunos investigadores, este avance implica un cambio en la filosofía del método científico. En palabras de Chris Anderson, de la revista Wired:[10]

> "*We can stop looking for models. We can analyze the data without hypotheses about what it might show. We can throw the numbers into the biggest computing clusters the world has ever seen and let statistical algorithms find patterns where science cannot.*"

- **Enfoque lingüístico**. Esta aproximación se basa en la aplicación de diferentes técnicas y reglas que codifiquen de forma explícita el conocimiento lingüístico (Moreno et al., 1999). Los textos son analizados en los diferentes niveles lingüísticos comentados más arriba (léxico, sintáctico, semántico y pragmático) mediante herramientas que se encargan de anotar estos textos.

  Se pueden emplear, por ejemplo, etiquetadores morfológicos que asignen a cada palabra del texto la categoría gramatical a partir de los rasgos morfológicos identificados. Una vez identificados estos rasgos, se puede realizar un análisis sintáctico para identificar cómo se combinan y relacionan las palabras, formando unidades superiores como sintagmas y oraciones. A partir de la estructura sintáctica del texto, es cuando se busca el significado de las oraciones que lo componen mediante herramientas de análisis semántico.

[8] Nombre que reciben las palabras que, pese a ser muy frecuentes en el texto, no aportan significado al mismo y carecen de relevancia en diversas tareas de PLN (como la RI o la clasificación de textos). Los artículos, pronombres y preposiciones son típicos ejemplos de palabras vacías.

[9] En lexicografía, concentrar en un único lema las formas de una palabra variable.

[10] http://www.wired.com/science/discoveries/magazine/16-07/pb_theory.

Todas estas herramientas y procesos sirven para afrontar el desarrollo de sistemas de PLN motivados por la lingüística, donde el enfoque se centra en aprovechar estas herramientas para facilitar la comprensión del texto y la realización de las tareas de procesamiento. Retomando el ejemplo anterior del sistema de RI, un sistema de este tipo que estuviera basado en un enfoque lingüístico podría localizar textos que contuvieran determinadas estructuras gramaticales coincidentes con la petición realizada por el usuario, y no sólo palabras sueltas tratadas de forma individual en cualquier orden (Woods et al., 2000).

2.3. La clasificación automática de preguntas

La *clasificación de preguntas*[11] (CP) es una tarea de PLN que trata de asignar de forma automática una clase, perteneciente a una taxonomía o conjunto cerrado, a una pregunta formulada en lenguaje natural.

De una manera más formal, definimos la CP como una tarea de clasificación multiclase que busca la asignación $g : X \rightarrow c_1, \ldots, c_n$ de una instancia $x \in X$ (en nuestro caso una pregunta) a una de las n clases posibles $c_1, \ldots, c_n$ (Li y Roth, 2005).

Un ejemplo de funcionamiento de estos sistemas sería, dada una pregunta como "¿Quién fue el primer presidente de Taiwan?", identificar que se nos está preguntando por una persona. Otro ejemplo sería asociar a la pregunta "¿Dónde nació Ulises S. Grant?" la clase semántica lugar, o dada la pregunta "¿Cuántos planetas hay en el sistema solar?", identificar que la respuesta esperada es un número. En estos ejemplos, *persona*, *lugar* o *número* serían las clases c_i en las que nuestro sistema automático debe clasificar estas preguntas. Como se puede deducir, antes de clasificar una pregunta es necesario saber cuál es el *conjunto de clases* o *taxonomía* que se pueden asignar a dicha pregunta. Estas clases o categorías son simples etiquetas simbólicas. No se proporciona ninguna información adicional sobre su significado a la hora de construir el clasificador. Esto implica que el texto que constituye la etiqueta (*persona*, *lugar* o *número* en los ejemplos anteriores) no puede ser utilizado como información a la hora de clasificar.

Como en la mayoría de tareas de PLN, las formas de afrontar la resolución automática del problema de la CP son dos: aproximaciones basadas en *conocimiento* o *reglas* (*knowledge-based methods*) y aproximaciones basadas en *corpus* o *aprendizaje automático* (*corpus-based methods*). En la primera aproximación, se utiliza conocimiento lingüístico preexistente, codificado por un experto (por ejemplo, en forma de reglas, diccionarios, tesauros, lexicones, ontologías, etc.), normalmente adquiridos de forma manual. Por contra, en la segunda aproximación el conocimiento se extrae

[11] Algunos de los nombres que recibe esta tarea en la literatura anglosajona son *question classification*, *question categorization* y *answer type recognition*.

a partir de grandes cantidades de ejemplos empleando métodos estadísticos y aprendizaje automático.

A la hora de definir un clasificador de preguntas (en general cualquier tipo de sistema de clasificación automática), la primera gran decisión consiste en establecer el enfoque a seguir. Tanto los sistemas basados en conocimiento como los basados en corpus trabajan sobre la misma premisa: una pregunta nueva será clasificada por el sistema basándose en las preguntas similares vistas durante la construcción del clasificador.

2.3.1. Sistemas basados en conocimiento

La mayoría de los primeros sistemas de CP realizaban esta tarea a partir de reglas heurísticas y patrones definidos de forma manual (Hermjakob, 2001; Voorhees, 1999). Estos sistemas aprovechan gramáticas construidas manualmente y conjuntos de expresiones regulares para analizar las preguntas y determinar su clase. Ejemplos de sistemas de este tipo serían los desarrollados por Durme et al. (2003), Paşca y Harabagiu (2001) y Sasaki et al. (2002).

En ocasiones, la aproximación más sencilla para construir un clasificador consiste en definir un conjunto de reglas manuales. Esta aproximación permite tener rápidamente un clasificador sencillo de baja cobertura (es decir, capaz de clasificar sobre taxonomías con un número pequeño de clases). Si las reglas son escritas por expertos, pueden llegar a tener una gran precisión. Además, los criterios de clasificación pueden ser fácilmente controlados cuando el número de reglas es pequeño. Otros ejemplos de sistemas que siguen esta aproximación son los de Breck et al. (1999), Cooper y Rüger (2000), Cui et al. (2004), Hull (1999), Lee et al. (2000), Prager et al. (1999), Moldovan et al. (1999), Radev et al. (2002) y Singhal et al. (1999). Una trabajo interesante en este campo es el desarrollado por Greenwood (2005), que establece un formalimo para la definición manual de reglas, obteniendo una precisión del 90 % sobre una taxonomía amplia de 67 tipos de pregunta posibles. Otra aproximación seguida por algunos sistemas es la de utilizar métodos estadísticos para obtener de forma automática reglas que se ajusten a los tipos de pregunta. El sistema Javelin (Nyberg et al., 2003) es un ejemplo de sistema que combina reglas definidas manualmente con otras aprendidas de forma automática.

Este tipo de clasificador manual puede requerir muchas horas de trabajo humano para ser construido y probado, garantizando el nivel apropiado de cobertura y precisión. La ventaja que tiene esta aproximación es que se pueden incluir reglas para cubrir preguntas para las cuales únicamente se ha visto un ejemplo previo. Su aplicación es útil en sistemas que trabajan en *dominios restringidos*,[12] donde está muy claro el tipo de preguntas que

[12] En un *dominio restringido* o *restricted domain* las preguntas giran en torno a un tema común, como puede ser el dominio turístico, el dominio político, el dominio médico, etc.

```
What {is|are} <phrase_to_define> ?
What is the definition of <phrase_to_define> ?
Who {is|was|are|were} <person_name(s)> ?
```

Figura 2.1: Conjunto de patrones para la detección de preguntas de tipo *definición*.

nos pueden hacer, siendo el conjunto de clases posibles normalmente más limitado que en sistemas de *dominio abierto*.[13] Algunos tipos de pregunta son especialmente propicios para esta aproximación. Por ejemplo, preguntas que requieren por respuesta una fecha de nacimiento pueden ser reconocidas de forma rápida usando apenas seis reglas bien construidas (Greenwood, 2005).

En general, las expresiones regulares funcionan bien para preguntas estándar ("¿Cuánto cuesta un litro de gasolina?") pero funcionan peor cuando nos desviamos de este estándar ("Me gustaría sabe cuánto me puede costar un litro de gasolina"). La figura 2.1 muestra un ejemplo de reglas manuales empleadas en un sistema de CP para el inglés (Paşca y Harabagiu, 2001) que permite la detección de preguntas de tipo *definición*. Algunas preguntas capturadas por estos patrones serían "*What is platinum?*" o "*Who is Barbara Jordan?*".

Existen dos limitaciones principales en los sistemas basados en reglas manuales. El primer problema es la cantidad de trabajo necesario para formular patrones eficientes que capturen todos los posibles tipos de pregunta. Tal y como indicábamos en la sección 2.2, el lenguaje humano permite utilizar numerosas variantes lingüísticas para expresar una misma idea. La figura 2.2 presenta diferentes reformulaciones de una misma pregunta. A todas estas reformulaciones se les debería asignar la misma clase semántica ya que todas ellas hacen referencia al mismo concepto (un *lugar*). El empleo de diferentes términos y estructuras sintácticas dificultan a un clasificador basado en un conjunto pequeño de reglas la tarea de generalizar de forma adecuada. Esto resulta cierto incluso si se emplean bases de conocimiento externo para crear patrones más generales (Harabagiu et al., 2000; Hermjakob, 2001). Más aún, las reglas manuales ofrecen una baja cobertura debido a la amplia variedad de formas posibles en las que puede manifestarse una misma pregunta, resultando difícil su mantenimiento a medida que el número de estas crece.

La segunda limitación es la falta de flexibilidad debido a la dependencia del idioma y el dominio. El cambio de idioma de trabajo, un cambio en el campo de aplicación o la inclusión de nuevos tipos de pregunta, conlleva generalmente la revisión y posible redefinición de las reglas y heurísticas planteadas inicialmente en el sistema. Las reglas manuales que funcionan

[13] En *dominio abierto* u *open domain* las preguntas pueden versar sobre cualquier tema.

What tourist attractions are there in Reims?
What are the names of the tourist attractions in Reims?
What do most tourists visit in Reims?
What attracts tourists to Reims?
What are tourist attractions in Reims?
What could I see in Reims?
What is worth seeing in Reims?
What can one see in Reims?

Figura 2.2: Distintas reformulaciones de una misma pregunta pertenecientes al conjunto de evaluación de la tarea de BR de la conferencia TREC-9.

bien en un conjunto específico de preguntas pueden dar resultados pobres cuando se aplican a otro conjunto. Por esta razón, las reglas construidas para una taxonomía de clases específica deben ser revisadas antes de ser aplicadas a otra taxonomía diferente.

Teniendo en cuenta estas dificultades, la mayoría de sistemas que usan reglas manuales están forzados a trabajar sobre taxonomías con un número limitado de clases de preguntas. Sin embargo, hay ocasiones en las que son necesarios sistemas más robustos que puedan ser fácilmente adaptados a nuevos idiomas y al manejo de nuevos conjuntos de datos y taxonomías de preguntas. Es en este punto donde entran en juego los sistemas basados en aprendizaje automático.

2.3.2. Sistemas basados en aprendizaje automático

El *aprendizaje automático* o *machine learning* es un campo de la IA relacionado con el diseño y desarrollo de algoritmos y técnicas que permiten a los ordenadores "aprender". Este aprendizaje se lleva a cabo a partir de grandes cantidades de ejemplos (corpus) de los que se extrae el conocimiento. Por esta razón, a esta aproximación se la conoce habitualmente como *basada en corpus*.

Cada instancia del corpus (una pregunta en nuestro caso) es representada mediante un conjunto de *características de aprendizaje* o *features.Es habitual referirse a estas características como* atributos *o* attributes. Estas características simbolizan la información relevante para el aprendizaje. Por ejemplo, en el caso de los sistemas de CP, una característica de aprendizaje interesante que se puede extraer de las preguntas a la hora de clasificarlas es el pronombre interrogativo.

Dependiendo de si las instancias del corpus están etiquetadas o no, es decir, de si conocemos o no la clase correcta a la que pertenecen, tenemos tres tipos fundamentales de aprendizaje (Mitchell, 1997):

- **Aprendizaje supervisado**. Los ejemplos del corpus están etiquetados, siendo el problema fundamental el de encontrar una función que relacione un conjunto de entradas con un conjunto de salidas. Los problemas de *clasificación* y *regresión* entran dentro de este grupo.

- **Aprendizaje no supervisado**. Los ejemplos no están etiquetados, siendo el problema fundamental el de encontrar la estructura subyacente del conjunto de datos. Los problemas de *agrupamiento* (*clustering*) y *compresión de datos* entran dentro de este grupo.

- **Aprendizaje semisupervisado**. Se sitúa a medio camino entre el aprendizaje supervisado y el no supervisado. Tiene lugar cuando se emplean tanto datos etiquetados como no etiquetados para la construcción del modelo (típicamente un pequeño conjunto de datos etiquetados y un conjunto grande de datos sin etiquetar). Técnicas como *co-training* o *expectation-maximization* entran dentro de este grupo (Chapelle et al., 2006).

La tarea de CP entra dentro del campo del aprendizaje supervisado (aunque existen algunas aproximaciones semisupervisadas, como veremos en el capítulo 5). En las aproximaciones basadas en aprendizaje automático a la tarea de CP, el conocimiento del experto que se empleaba en los sistemas manuales es reemplazado por un conjunto suficientemente grande de preguntas etiquetadas con sus correspondientes clases semánticas correctas. A partir de este conjunto de entrenamiento se induce un modelo que permite al clasificador, dada una nueva instancia, predecir la clase a la que pertenece.

Existen numerosos algoritmos de aprendizaje que han demostrado su utilidad en diferentes tareas de PLN. La figura 2.3 muestra algunos de estos algoritmos agrupados en función de la familia a la que pertenecen (Alpaydin, 2004).

Resulta difícil imaginar un clasificador construido manualmente mediante reglas que dependa de miles de características. Sin embargo, los métodos de aprendizaje pueden utilizar un número potencialmente grande de características para generalizar y clasificar de forma automática. La gran promesa de esta aproximación es la de ofrecer al responsable de desarrollar el sistema la posibilidad de centrarse en el diseño de las características y en el desarrollo de datos etiquetados, en lugar de codificar y manetener heurísticas complejas basadas en reglas. De esta forma, los sistemas basados en aprendizaje automático permiten crear aplicaciones más flexibles, que se adapten a cambios en el entorno y aprendan a partir de corpus de entrenamiento. Esto permite superar muchas de las limitaciones de los sistemas basados en reglas manuales. Algunas de las ventajas que proporcionan con respecto a estos últimos son:

Computacionales puros
- Árboles de decisión
- Clasificación del vecino más cercano
- Agrupamiento basado en teoría de grafos
- Reglas de asociación

Estadísticos
- Regresión multivariada
- Discriminación lineal
- Teoría de la decisión Bayesiana
- Redes Bayesianas
- K-means

Computacionales-Estadísticos
- Máquinas de vectores soporte
- AdaBoost

Bio-inspirados
- Redes neuronales
- Algoritmos genéticos
- Sistemas inmunológicos artificiales

Figura 2.3: Principales algoritmos de aprendizaje agrupados por familias.

- No hay necesidad de conocimiento de un experto para la codificación de reglas.

- El clasificador puede ser reentrenado de forma flexible sobre un nuevo conjunto de datos para adaptarse a una nueva taxonomía.

- La tarea de mantenimiento se ve claramente simplificada, ya que únicamente requiere reentrenar el clasificador sobre las nuevas condiciones de trabajo.

- Existen en el mercado numerosos algoritmos de aprendizaje implementados y libremente disponibles, lo que reduce la necesidad de recursos humanos para la construcción de estos sistemas. Dos ejemplos destacados de este tipo de software son Weka (Witten y Frank, 2005) y TiMBL (Daelemans y van den Bosch, 2005).

Actualmente, los resultados conseguidos usando aproximaciones basadas en aprendizaje representan la última tecnología en la tarea de CP. En el siguiente capítulo entraremos en detalle en estos sistemas, haciendo un repaso de las aproximaciones más relevantes en este campo.

2.4. Clasificación de preguntas y clasificación de textos

La *clasificación de textos*[14] es una tarea de largo recorrido dentro del campo del PLN (Sebastiani, 2002). Originada a principios de los años 60, no fue hasta los 90 cuando se convirtió en un área de fuerte interés gracias al desarrollo de las tecnologías informáticas. Estos sistemas tratan de asignar una serie categorías predefinidas a un documento basándose en su contenido. Pese a los puntos en común que ofrece la clasificación de textos con la CP (ambas tratan de asignar una etiqueta a un texto, ya estemos hablando de una pregunta o de un documento), hay una serie de diferencias remarcables que distinguen a una tarea de la otra:

- La cantidad de texto disponible es considerablemente menor en una pregunta que en un documento. Esto hace que el solapamiento entre preguntas (es decir, el número de términos coincidentes) sea muy inferior al que se suele dar entre documentos. De esta forma, las medidas tradicionales para el cálculo de la similitud entre documentos como *cosine similarity*, *Jaccard index* o *Dice coefficient* (Salton, 1989), no resultan adecuadas para la tarea de CP (Jeon et al., 2005).

- En clasificación de textos es habitual la ponderación de los términos aparecidos en determinadas zonas del documento, como pueden ser los párrafos iniciales y finales de un texto (que anticipan los contenidos del mismo), las cabeceras en los correos electrónicos (*autor* y *asunto*) o el título y los términos clave de un artículo de investigación (Manning et al., 2008). Este tipo de información carece de sentido en CP, ya que no se pueden localizar estas zonas de especial interés debido a la escasa cantidad de texto con la que se trabaja.

- En clasificación de textos es habitual ponderar la relevancia de los términos en los documentos en función del número de veces que aparecen en ellos (con medidas como *tf-idf*[15]). Sin embargo, a la hora de clasificar preguntas estas frecuencias no aportan ningún tipo de información, ya que normalmente los términos aparecen una única vez en cada pregunta, siendo escasas las repeticiones (la frecuencia es habitualmente 1).

[14] Conocida en literatura anglosajona como *text categorization*, *text classification* o *topic spotting*.

[15] El valor *tf-idf* (*term frequency-inverse document frequency*) se calcula como la frecuencia del término (*tf*) multiplicada por la inversa de la frecuencia del documento (*idf*), que se obtiene dividiendo el número total de documentos en el corpus por el número de documentos en los que aparece el término.

- El uso que se hace de las *stopwords* es muy diferente entre ambas tareas. Este tipo de palabras son típicamente descartadas en clasificación de textos (Joachims, 1998). Sin embargo, en CP estas palabras juegan un papel muy importante. Los pronombres interrogativos como "quién", "cuándo" o "por qué" pueden ser extremadamente útiles para determinar satisfactoriamente la clase semántica a la que pertenece una pregunta.

Estas diferencias hacen que la tarea de CP presente un grado adicional de dificultad con respecto a la clasificación de textos. Trabajamos con conjuntos de preguntas que son semánticamente similares, pero léxicamente muy diferentes. Como veremos en los siguientes capítulos, para compensar esta falta de información proporcionada por el texto se hace necesaria una representación más sofisticada de las instancias del problema.

2.5. Aplicación a la búsqueda de respuestas

Los sistemas de *búsqueda de respuestas* (BR) o *question answering* se han convertido en un importante punto de interés en el campo del PLN. La BR se define como la tarea automática que tiene como finalidad encontrar respuestas concretas a necesidades precisas de información formuladas por los usuarios (Vicedo, 2003). Estos sistemas son especialmente útiles en situaciones en las que el usuario necesita conocer un dato muy específico y no dispone de tiempo para leer toda la documentación referente al tema objeto de búsqueda.

Los sistemas de CP son una parte fundamental dentro de los sistemas de BR y su evolución ha ido de la mano de éstos. En esta sección vamos a hacer un repaso a la tarea de BR y vamos a ubicar los sistemas de CP dentro de esta tarea.

2.5.1. Orígenes

La primera discusión sobre las características que debería cumplir un sistema de BR, así como la primera aproximación a un sistema funcional (QUALM), fueron introducidos por Wendy Lehnert a finales de los 70 (Lehnert, 1977a, 1980). En estos trabajos se definieron las características ideales que debían poseer los sistemas de este tipo: entender la pregunta del usuario, buscar la respuesta en una base de datos de conocimiento y posteriormente componer la respuesta para presentarla al solicitante. Debían integrar técnicas relacionadas con el entendimiento del lenguaje natural, la búsqueda de conocimiento y la generación de lenguaje humano.

La investigación en sistemas de BR tuvo sus orígenes en el campo de la IA. Inicialmente se consideró requisito indispensable que los sistemas de BR satisficieran todas y cada una de las características ideales citadas

anteriormente. Hasta la fecha, sin embargo, únicamente se han podido obtener buenos resultados a costa de restringir los dominios sobre los que se realizan las consultas.

En la actualidad, la investigación en sistemas de BR se afronta principalmente desde el punto de vista de la comunidad especializada en sistemas de RI. Desde esta perspectiva, el poder desarrollar la tarea sobre dominios no restringidos constituye el requisito básico a cumplir. Partiendo de este requerimiento inicial, las investigaciones se orientan hacia el desarrollo de sistemas que van incorporando progresivamente herramientas más complejas que permitan la convergencia hacia el sistema ideal propuesto por Lehnert. En cualquier caso, se puede considerar que el estado actual de las investigaciones en este campo está todavía en una fase temprana, lejos de la visión y objetivos definidos por Carbonell et al. (2000).

2.5.2. Situación actual

En los últimos años, espoleados por importantes foros internacionales como las conferencias TREC, CLEF y NTCIR, se han desarrollado infinidad de sistemas de BR basados en diversas aproximaciones. Los sistemas actuales afrontan la tarea de búsqueda desde la perspectiva de un usuario casual: un usuario que realiza preguntas simples que requieren como contestación un hecho, situación o dato concreto (John Burger, 2003).

La dificultad de localizar y verificar la respuesta precisa hacen de la tarea de BR una tarea más desafiante que las tareas comunes de RI. Por esta razón, se hace necesario el uso de técnicas avanzadas de lenguaje natural más allá de la simple extracción y expansión de términos. Es habitual el uso de bases de conocimiento, como diccionarios, enciclopedias, bases de datos léxico-semánticas como WordNet (Fellbaum, 1998) y EuroWordNet,[16] y ontologías como SENSUS (Hovy et al., 2000) y Mikrokosmos (Mahesh y Nirenburg, 1995). Otra fuente de información habitual en los sistemas actuales es la Web. Diversas aproximaciones aprovechan la abundancia de información presente en Internet para localizar o justificar las respuestas formuladas a sus sistemas (Brill et al., 2001; Clarke et al., 2001b). Se busca la redundancia de datos para evitar la necesidad de emplear técnicas complejas de análisis lingüístico. La idea subyacente tras estas técnicas es que, suponiendo que tenemos suficientes datos, siempre habrá algún texto que explícitamente muestre la respuesta a la pregunta formulada.

2.5.3. Arquitectura de los sistemas

Pese a la gran variedad de sistemas y aproximaciones que existen a la tarea de BR, la mayoría de éstos presentan una arquitectura común (Magnini et al., 2002; Moldovan et al., 2002; Soubbotin y Soubbotin, 2002; Yang y

[16] http://www.illc.uva.nl/EuroWordNet/.

Chua, 2002), organizando su funcionamiento en tres fases bien diferenciadas: *análisis de la pregunta*, *selección de documentos o pasajes relevantes* y *extracción de la respuesta*. Estas tareas se abordan de forma secuencial, aunque hay excepciones como el sistema FALCON (Harabagiu et al., 2000) que lleva a cabo varias iteraciones durante el proceso de búsqueda. Vamos a describir a continuación cada una de estas fases con mayor detalle.

Análisis de la pregunta

El *análisis de la pregunta* constituye la primera fase de la tarea de BR. En esta fase se recibe como entrada una pregunta en lenguaje natural formulada por el usuario, llevando a cabo dos procesos diferentes: la *clasificación de la pregunta* y la *formación de la consulta*. Es pues, en esta primera fase, donde se ubican los sistemas de CP objeto de estudio en esta tesis.

El objetivo de la CP en los sistemas de BR es detectar el tipo de respuesta esperada para la pregunta. La mayoría de sistemas de BR incluyen una taxonomía de tipos de respuesta, típicamente con un tamaño de entre 10 y 50 clases posibles (Hovy et al., 2002). Un ejemplo sencillo de taxonomía sería la empleada por Ittycheriah et al. (2001), compuesta básicamente de las etiquetas estándar de las conferencias MUC:[17] *Person*, *Location*, *Organization*, *Cardinal*, *Percent*, *Date*, *Time*, *Duration*, *Measure*, *Money*, *Phrase* y *Reason*.

La CP en los sistemas de BR tiene un doble propósito. En primer lugar, proporciona una restricción semántica a las respuestas esperadas que permite filtrar un gran número de éstas durante la fase final de extracción. Por ejemplo, cuando consideramos la pregunta "¿Cuál es la ciudad más grande de Alemania?", detectar que nos están preguntando por un lugar (es decir, clasificar la pregunta como perteneciente a la clase *lugar*), o mejor aún, por una ciudad (dependiendo del grado de refinamiento de la taxonomía de preguntas que se tenga), nos va a permitir descartar un gran número de respuestas candidatas quedándonos sólo con aquellas que sean nombres de lugar o ciudad. Sin esta restricción semántica sería mucho más difícil, o incluso imposible, detectar la respuesta correcta dentro de todas las posibles respuestas candidatas, que necesariamente serían todos los sintagmas nominales o entidades presentes en el texto.

El segundo propósito de la CP es proporcionar información a los procesos subsiguientes del sistema de BR, de forma que les permita determinar la estrategia de selección de respuestas y las bases de conocimiento que el sistema puede necesitar para obtener la respuesta final. Es habitual en los sistemas de BR aplicar diferentes estrategias de extracción de la respuesta dependiendo de la clase asignada a la pregunta. Por ejemplo, una pregunta

[17] *Message Understandig Conference* son una serie de conferencias creadas para promocionar y evaluar la investigación en extracción de información (Grishman y Sundheim, 1996).

como "¿En qué provincia está Villena?" requiere el nombre de una provincia por respuesta. Esto implica que el sistema necesita tener conocimiento sobre nombres de provincias y un reconocedor de entidades capaz de identificarlas en el texto. Por otra parte, para una pregunta como "¿Qué es un prisma?", cuyo objetivo es una *definición*, se pueden emplear estrategias específicas para la detección de definiciones en el texto. Una solución directa sería utilizar patrones predefinidos del tipo {`un`|`el`} `prisma es` ⟨`RESPUESTA`⟩ o `los prismas son` ⟨`RESPUESTA`⟩, donde ⟨`RESPUESTA`⟩ sería una respuesta candidata a la pregunta formulada.

Numerosos trabajos en el área de la BR (Hovy et al., 2001; Moldovan et al., 2003; Roth et al., 2002) han demostrado que, para localizar una respuesta precisa, se deben filtrar primero un amplio rango de respuestas candidatas basándose en algún tipo de taxonomía de tipos de respuesta esperada. Existen estudios previos realizados sobre el impacto de la CP en el resultado global de los sistemas de BR. Radev et al. (2002) demostraron que una clasificación incorrecta del tipo de respuesta esperado por la pregunta, que se emplee como filtro anterior a la fase de extracción de la respuesta, hace que la posibilidad del sistema de responder a la pregunta correctamente sea 17 veces menor. En otro análisis realizado sobre los errores en sistemas de domino abierto, Moldovan et al. (2003) revelaban que más de un 35 % de éstos eran directamente imputables al módulo de CP. El sistema de CP es, por tanto, de notable importancia dentro de un sistema de BR ya que un mejor funcionamiento del clasificador conlleva un mejor funcionamiento del sistema de completo.

El segundo proceso llevado a cabo en la fase de análisis de la pregunta es la *formación de la consulta*. Este proceso permite determinar las *palabras clave* o *keywords* de la pregunta que proporcionan información significativa sobre el objeto buscado. Con estas palabras se forma una consulta que se empleará en la fase siguiente del proceso de BR para recuperar los documentos susceptibles de contener una respuesta. Para recuperar estos documentos se emplea un sistema de RI. La figura 2.4 muestra varios ejemplos de preguntas y sus correspondientes consultas una vez extractados las palabras clave. Una práctica habitual para obtener estos términos clave es eliminar aquellas palabras que aparecen en una lista de *stopwords* confeccionada generalmente de forma manual.

La consulta no sólo puede estar formada por palabras clave, sino que existen diversas técnicas para enriquecerla y extender su semántica con el fin de recuperar el máximo posible de documentos relacionados con la pregunta. Entre estas técnicas destacamos dos: la *realimentación* o *relevance feedback* (Ruthven y Lalmas, 2003) y la *expansión de la consulta* o *query expansion* (Qiu y Frei, 1993). La idea tras la *realimentación* es tomar los resultados devueltos inicialmente por el sistema de RI para la consulta y utilizar la información sobre su relevancia o no para realizar una nueva consulta. Esta nueva consulta estará modificada en función de los términos que aparecen

Pregunta: ¿Quién ganó el premio Nobel de Física en 1927?
Consulta: `ganó premio Nobel Física 1927`

Pregunta: ¿Cuándo tuvo lugar el primer vuelo del Columbia?
Consulta: `primer vuelo Columbia`

Pregunta: ¿Qué película ganó el Oscar en 1945?
Consulta: `película ganó Oscar 1945`

Pregunta: ¿Cuánto cuesta un litro de gasolina?
Consulta: `cuesta litro gasolina`

Figura 2.4: Ejemplos de preguntas y sus correspondientes consultas obtenidas mediante la extracción de palabras clave.

en los documentos que se hayan considerado más relevantes. Una técnica aplicada habitualmente para realizar este proceso es el *algoritmo de Rocchio* (Rocchio, 1971). Por otra parte, la *expansión de la consulta* busca igualmente enriquecer la consulta inicial del usuario empleando para ello estrategias como la inclusión de sinónimos, el empleo de variantes morfológicas o la ponderación de los términos.

La importancia del proceso de selección de términos clave radica en que una consulta mal formada puede provocar que el sistema de RI no devuelva los documentos relevantes que deben contener la respuesta a la pregunta, convirtiendo en inútil el esfuerzo posterior para la extracción de la respuesta.

Recuperación de información

Los sistemas de *recuperación de información* (RI) o *information retrieval* realizan la tarea de seleccionar y recuperar aquellos documentos que son relevantes a una consulta del usuario. Como resultado, estos sistemas devuelven una lista de documentos ordenada en función de ciertos criterios que intentan reflejar la importancia del documento con respecto a la consulta realizada.

El uso de los sistemas de RI dentro del proceso de BR se debe a la necesidad de acotar el área de búsqueda de las posibles respuestas. Localizar respuestas en un documento requiere un grado de análisis y comprensión del texto que no resulta factible sobre grandes corpus[18] debido a su alto coste computacional. Los sistemas de RI resultan determinantes para acotar la búsqueda, detectando los documentos que están más relacionados con la consulta (generalmente unos pocos cientos) y que consecuentemente tienen

[18]La tarea de BR llevada a cabo en el TREC-9 se realizó sobre un corpus de casi un millón de documentos.

más probabilidad de contener una respuesta válida. Hacen la función de filtro sobre el corpus original, permitiendo obtener un subconjunto de documentos sobre el que se realizará la posterior localización y extracción de la respuesta.

Una alternativa a los modelos clásicos de RI es la *recuperación de pasajes* o *passage retrieval* (Callan, 1994). Estos sistemas miden la relevancia de un documento con respecto a una consulta en función de la importancia de los fragmentos contiguos de texto (*pasajes*) que lo conforman. Esta aproximación facilita la detección, dentro de documentos grandes, de aquellos extractos que pueden ser más relevantes para el usuario y que, debido a su inclusión en un documento mayor, podrían pasar desapercibidos a un sistema de RI que considere el documento completo como unidad de información. Aunque estos sistemas resultan computacionalmente más costosos que los tradicionales sistemas de RI, obtienen mejoras de rendimiento que justifican su adopción en la mayoría de situaciones (Kaszkiel y Zobel, 1997).

Aunque la clasificación realizada por los sistemas de CP se emplea principalmente en la fase de extracción de la respuesta, también en la fase de RI resultan de utilidad. Algunos sistemas de BR (Moschitti y Harabagiu, 2004; Pinchak y Lin, 2006; Skowron y Araki, 2004b) emplean la clasificación de la pregunta para filtrar pasajes en la fase de RI, descartando aquellos que no contienen entidades que se ajusten a la clase semántica detectada por el sistema de CP.

Extracción de la respuesta

La última fase llevada a cabo en un sistema de BR es la *extracción de la respuesta* o *answer extraction*. En esta fase se procesa el conjunto de documentos o pasajes obtenidos por el sistema de RI con el objetivo de localizar y extraer la respuesta buscada.

Es en este punto donde resulta de mayor utilidad la clase detectada por el sistema de CP. Esta información permite limitar el conjunto de posibles respuestas a aquellas que coincidan con la clase detectada. Para una pregunta de tipo *persona* como "¿Quién descubrió América?", sólo se tendrían en cuenta aquellas respuestas candidatas que sean nombres de persona, reduciendo drásticamente el conjunto de posibles soluciones. Para que este proceso sea eficaz debe de existir una coordinación entre el módulo de CP y el módulo de extracción de la respuesta. Si el sistema de CP es capaz de clasificar las preguntas en *persona*, *número* y *fecha*, el sistema de extracción de la respuesta debe ser capaz de detectar en el texto nombres de personas, números y fechas. Para realizar esta tarea, es habitual el uso en esta fase de etiquetadores morfológicos y reconocedores de entidades.

Una vez filtradas las respuestas que no son compatibles con la clase detectada por el sistema de CP, existen diversas estrategias para determinar la respuesta candidata final. Por ejemplo, tenemos sistemas que emplean medidas de distancia entre las palabras claves de la pregunta y la respuesta

(Tomás y Vicedo, 2007b), patrones manuales de extracción (Roger et al., 2006), redundancia de la respuesta en la Web (Clarke et al., 2001a) o detección de relaciones semánticas (Vicedo y Ferrández, 2000).

2.5.4. Tipos de pregunta tratados

No todos los tipos de pregunta que se pueden dar en la comunicación humana son de interés para los sistemas de BR, ni por tanto para los sistemas de CP que se integran en ellos. La finalidad con la que fueron concebidos los sistemas de BR es la de cubrir necesidades de información requeridas por un usuario. Ayer (1932) nos ofrece una interesante definición de lo que es una pregunta que encaja perfectamente con el concepto de pregunta tratada por los sistemas de BR:

> "*We enquire in every case, what observations would lead us to answer the question, one way or the other; and, if none can be discovered, we must conclude that the sentence under consideration does not, as far as we are concerned, express a genuine question, however strongly its grammatical appearance may suggest that it does.*"

Una pregunta es algo que puede ser respondido y si no hay forma de darle respuesta, entonces lo que se ha formulado no es una pregunta. Lo importante no son los signos de interrogación, o la idea de querer saber algo. La respuesta no sólo da contestación a la pregunta, sino que confirma que realmente se trataba de una pregunta. Las *preguntas retóricas* son un ejemplo de oraciones interrogativas utilizadas por su efecto persuasivo pero que no esperan una respuesta. Por ejemplo, "¿Por qué siempre me toca a mí?" o "¿Cuántas veces te lo tengo que repetir?" no esperan ningún tipo de respuesta. Estas preguntas son un artefacto usado por el hablante para afirmar o negar algo.

Desde un punto de vista gramatical nos podemos encontrar con cuatro tipos fundamentales de oraciones interrogativas (Contreras, 1999):

- **Interrogativas totales (o absolutas)**. Son aquellas que requieren de una respuesta afirmativa o negativa. Por ejemplo, "¿Tienes dinero?", "¿No tienes dinero?", "¿Nunca vas al cine?", "¿Le diste una propina al camarero?" o "¿Él ya no vive aquí?".
- **Interrogativas disyuntivas**. Son aquellas preguntas que piden una decisión entre dos o más alternativas. Por ejemplo, "¿Quieres libros, revistas o periódicos?", "¿Te gusta la música o la literatura?" o "Tus amigos, ¿estudian o trabajan?".
- **Interrogativas de confirmación**. Como indica su nombre, son preguntas que piden que el interlocutor confirme o niegue una

afirmación. La sintaxis de este tipo de interrogativas es muy simple, constando de expresiones fijas con entonación ascendente que se agregan al final de una afirmación. Por ejemplo, "¿Verdad?", "¿A que sí?" o "¿Ah?".

- **Interrogativas parciales**. Son preguntas que piden algún tipo de información más específica. Las interrogativas parciales siempre incluyen alguno de los siguientes elementos interrogativos: "qué", "quién(es)", "cuál(es)", "cómo", "dónde", "cuándo", "cuánto(s)" o "por qué". Normalmente el elemento interrogativo encabeza la pregunta, como en los ejemplos siguientes: "¿Qué compraste?", "¿Quién escribió ese libro?", "¿Cuál es tu casa?", "¿Cómo se escribe tu nombre?", "¿Dónde vives?", "¿Cuándo naciste?", "¿Cuánto vale ese coche?" o "¿Por qué me miras así?". Las interrogativas parciales pueden contener en ocasiones más de un elemento interrogativo. Se trata en ese caso de *interrogativas múltiples* en las que los elementos interrogativos pueden estar coordinados entre sí ("¿Dónde y con quién andabas?" o "¿Cuándo y cómo piensas pagar ese dinero?") o no ("¿Quién conoce a quién?", "¿Cuándo viene quién?" o "¿Qué le diste a quién?").

El nivel de conocimiento con el que trabajan los actuales sistemas de BR hace que no sean capaces de responder preguntas del tipo *total* o *disyuntivo*. Las interrogativas de tipo *confirmación* no representan requerimiento de información y, por tanto, no son tratadas por los sistemas de BR. Así pues, las *interrogativas parciales* son el tipo de pregunta que realmente nos interesan en nuestro estudio. Aún así, no todas las *interrogativas parciales* son tratadas por los sistemas tradicionales de BR. La mayoría de sistemas se centran en preguntas de tipo *factual* (*factoid*) y *definición*, tal y como se definieron dentro del marco de los sistemas de BR en las conferencias TREC. Las preguntas de tipo *factual* son aquellas que interpelan por un único hecho. Por ejemplo, "¿Cuándo nació Mozart?", "¿Qué altura tiene la Torre Eiffel?" o "¿Dónde está ubicada la sede de Microsoft?" son ejemplos de este tipo de preguntas. Por contra, las preguntas de tipo *definición* requieren una respuesta más compleja, generalmente un fragmento de texto corto que sucintamente defina aquello por lo que se está preguntando. Las respuestas esperadas para este tipo de preguntas serían típicamente las de una entrada enciclopédica. Por ejemplo, si un usuario pregunta por una persona ("¿Quién es Barack Obama?") probablemente querrá saber fechas importantes de su vida, logros más relevantes y cualquier otro ítem destacable por el que sea conocido. Hay otros tipos de pregunta, como las de tipo *instrucción* ("¿Cómo hago una sopa de tomate?") y las de tipo *explicación* ("¿Por qué entró América en la Segunda Guerra Mundial?") que apenas han sido tratadas por los sistemas pasados y presentes de BR (Verberne, 2006).

Cuando un sistema de CP se integra dentro de un sistema completo de BR, la taxonomía de clases sobre la que es capaz de clasificar las preguntas se

limita a aquellas que el sistema de BR es capaz de responder. Sin embargo, cuando tratamos los sistemas de CP de forma aislada para su estudio, las taxonomías de clases no sufren esta limitación. En el siguiente capítulo haremos un repaso por la situación actual de los sistemas de CP. Muchos de estos sistemas se han desarrollado con independencia de los sistemas de BR, por lo que las taxonomías con las que trabajan tienen un mayor grado de libertad que las utilizadas habitualmente en los sistemas de BR. Por ejemplo, Cheung et al. (2004) presentan un sistema de CP sobre una taxonomía que incluye clases como *true-or-false* y *yes-or-no*, dos tipos de pregunta que un sistema actual de BR difícilmente es capaz de responder.

2.6. Otras aplicaciones

Aunque los sistemas de CP son conocidos por su aplicación dentro de los sistemas de BR, existen otras tareas en las que estos sistemas son de utilidad. Los *servicios de referencia virtual* y la *clasificación facetada* son dos ejemplos de estos usos.

2.6.1. Servicios de referencia virtual

Los *servicios de referencia virtual* son un componente esencial en las bibliotecas virtuales para la gestión de información (Pomerantz et al., 2004) y representan otra tarea en la que se puede hacer uso de los sistemas de CP. Estos sistemas pueden emplearse como parte de un sistema automático de prioridades que determine cuándo una pregunta puede ser respondida de forma automática por una sistema de BR o debe ser respondida por un experto humano basándose en el tipo de respuesta esperada.

Una pregunta que se clasifique como perteneciente a una categoría cuya respuesta sea "sencilla", como un nombre de *persona* o una *fecha*, podría ser encaminada de forma automática a un sistema de BR que se encargue de responderla. Por otra parte, si la pregunta se clasifica como perteneciente a una clase "compleja", como aquellas que preguntan por una definición técnica o una explicación detallada, debería ser enviada a un experto humano que se encargue de darle solución. La clase detectada para la pregunta podría incluso ser utilizada por el sistema para determinar de forma automática a qué experto humano debe ser enviada para su contestación.

2.6.2. Búsqueda facetada

Desde los inicios de la Web han habido dos paradigmas principales en cuanto a búsqueda de información. El primero, la *búsqueda por navegación* representada por sitios Web como *Open Directory Project*,[19] ayuda a la gente

[19] http://www.dmoz.org.

a acotar la información que están buscando usando directorios temáticos o taxonomías. El segundo paradigma, la *búsqueda directa* presente en sitios como Google, permite a los usuarios escribir sus propias peticiones como una conjunto de palabras clave en una caja de texto para llevar a cabo la búsqueda de información.

La *búsqueda facetada* (Yee et al., 2003) es una nueva aproximación de reciente aparición. Este paradigma pretende combinar la búsqueda por navegación y directa, permitiendo a los usuarios navegar en un espacio de información multidimensional, combinando la búsqueda textual con una reducción progresiva de opciones de selección en cada dimensión. Los sistemas de búsqueda facetada asumen que la información está organizada en múltiples facetas independientes, en lugar de en una única taxonomía. Por ejemplo, podemos definir para una guía de restaurantes atributos como *cocina*, *ciudad* o *servicios*. Estos atributos son *facetas* que ayudan a los usuarios a navegar a través de ellas seleccionando los valores deseados, como por ejemplo *mejicana* para *cocina*, *madrid* para *ciudad* o *aparcamiento propio* para *servicios*.

Este paradigma se complementa con la *búsqueda por categorías* (Tunkelang, 2006), que no es una búsqueda directa sobre la información guardada, sino una búsqueda en el espacio de valores posibles de las facetas. Mientras que la búsqueda directa devuelve un conjunto de documentos que pueden ser refinadas posteriormente usando una aproximación basada en búsqueda facetada, la búsqueda por categorías proporciona resultados que son en sí mismos puntos de entrada de una búsqueda facetada. En el ejemplo de la guía de restaurantes, un usuario interrogaría al sistema con peticiones como *madrid* o *italiana* para restringir los resultados a restaurantes en esa *ciudad* o con este tipo de *cocina*.

Las interfaces actuales a la búsqueda por categoría están limitados a búsqueda por palabras clave sobre valores de las facetas. En el trabajo que desarrollamos en (Tomás y Vicedo, 2007a) realizamos una novedosa propuesta para la búsqueda por categorías. Afrontamos el reto de identificar valores de facetas presentes de forma implícita en preguntas formuladas en lenguaje natural. El problema se abordó desde el punto de vista de la CP. Mientras que los sistemas tradicionales de CP están limitados a clasificar preguntas sobre una única taxonomía, en este trabajo introdujimos la idea de la CP en múltiples taxonomías. En el contexto de la búsqueda por categorías, nuestro sistema recibe una pregunta y detecta las diferentes facetas (taxonomías) y sus valores (clases) implícitamente presentes en la pregunta. Los valores asignados permiten reducir el conjunto de documentos relevantes a sólo aquellos que pertenecen a las clases y taxonomías identificadas. Siguiendo el ejemplo previo, una pregunta como "Estoy buscando un restaurante turco en Madrid" fijaría el valor de la faceta *cocina* a *turco* y el de *ciudad* a *madrid*, para así devolver sólo documentos relativos a restaurantes que cumplan con estas dos restricciones.

Para llevar a cabo esta tarea, nuestro sistema crea un *modelo de lenguaje*[20] para cada clase (cada valor de cada faceta), basándose en un conjunto de documentos pertenecientes a dicha clase. Para identificar las clases presentes en la pregunta, se calcula la probabilidad de generar dicha pregunta a partir de los distintos modelos de lenguaje. A diferencia de las aproximaciones tradicionales a la búsqueda por categorías, no limitamos nuestra búsqueda a la lista de los posibles valores de las facetas, sino que extendemos estos valores aprovechando la redundancia de términos en los documentos. Volviendo al ejemplo de la guía de restaurantes, los documentos que describen restaurantes con *cocina* de tipo *mejicana*, contendrán probablemente palabras como "burrito", "fajita" o "taco". De igual forma, los documentos que describen restaurantes con *servicios* como *aparcamiento propio*, contendrán habitualmente las palabras "parking", "aparcar" o "coche". De esta forma, nuestro sistema puede interpretar una petición como "Quiero reservar una mesa para comerme un burrito" e inferir que el usuario está preguntando por las facetas *cocina* y *servicios* con los valores *mejicano* (disparada por la palabra "burrito") y *reservas* (disparada por la palabra "reservar") respectivamente.

2.7. Conclusiones

En este capítulo se ha introducido formalmente el problema de la CP. Estos sistemas, al trabajar con lenguaje humano deben afrontar las dificultades inherentes a él. Los sistemas informáticos que afrontan los problemas derivados del tratamiento del lenguaje natural se engloban dentro del PLN, una rama de larga tradición dentro de la IA.

A lo largo del capítulo hemos introducido las distintas aproximaciones seguidas en el desarrollo de sistemas de PLN en general y de sistemas de CP en particular. Atendiendo a las premisas establecidas en el capítulo anterior (*los sistemas aprenden por sí mismos* y *el aprendizaje no requiere de recursos lingüísticos complejos*), vamos a tomar una serie de decisiones de diseño:

- Nuestros sistemas de CP se basarán en aprendizaje automático sobre corpus (descritos en la sección 2.3). De esta manera evitamos el desarrollo de reglas manuales y la dependencia de conocimiento experto, permitiendo al sistema aprender y adaptarse por sí mismo a diferentes situaciones.

- Vamos a seguir una aproximación estadística al tratamiento del lenguaje humano, tal y como se describió en la sección 2.2. Esta aproximación resulta más adecuada para nuestros propósitos que la aproximación lingüística, ya que esta última implica el uso de

[20] En la sección 3.4.2 hablaremos más en detalle sobre estos modelos.

herramientas de análisis para los distintos niveles del lenguaje. El uso de estas herramientas trae consigo la dependencia con respecto al idioma y el dominio para el que fueron desarrolladas.

- Desarrollaremos nuestros clasificadores como módulos independientes, sin ligarlos a ningún sistema de BR concreto. Esto nos va a dar un mayor grado de libertada a la hora de trabajar sobre diferentes taxonomías de preguntas. Cuando la CP se integra dentro de un sistema completo de BR, el tipo de preguntas con los que trabaja está limitado a aquellas que el sistema de BR es capaz de responder. Tratando los sistemas de CP de forma aislada las taxonomías de clases no sufren esta limitación.

En el siguiente capítulo describiremos en detalle la aproximación a la CP basada en aprendizaje automático, prestando atención a los distintos elementos que intervienen en este tipo de sistemas. Aprovecharemos esta descripción para mostrar otros estudios científicos realizados y establecer el estado de la cuestión en esta área.

3

Los sistemas de clasificación de preguntas basados en corpus

En los últimos años se ha producido un claro cambio de tendencia en los trabajos de investigación publicados dentro del campo del PLN. Las ideas provenientes de la estadística y el aprendizaje automático han remplazado los conceptos teóricos del lenguaje para afrontar la construcción de estos sistemas. Las aproximaciones modernas al procesamiento del habla y el lenguaje le deben más a la teoría de la información de Shannon que a las gramáticas generativas de Chomsky. Las primeras aproximaciones a los grandes problemas lingüísticos (como el análisis sintáctico y la interpretación semántica de las oraciones) basadas en lógica e inferencia, han sido reemplazadas ampliamente por una nueva visión más pragmática que tiende a abordar problemas más sencillos, como el etiquetado morfológico o el análisis sintáctico superficial, centrándose en manejar los casos comunes con una alta precisión.

Las aproximaciones a la CP desarrolladas en esta tesis van a basarse en el empleo de técnicas de aprendizaje automático sobre corpus. En este capítulo vamos a describir los principios generales del aprendizaje automático y los componentes que intervienen en su aplicación a la tarea de CP. Para el desarrollo de estos sistemas y en general para cualquier sistema de clasificación automática, hay que definir una serie de elementos:

- Una taxonomía de tipos de pregunta con la que queremos clasificar las entradas que lleguen al sistema.
- Un conjunto de ejemplos (preguntas en nuestro caso) correctamente etiquetados con las clases definidos en la taxonomía anterior. Es lo que se conoce como *corpus de entrenamiento*.
- Un conjunto de características de aprendizaje extraidas del corpus de entrenamiento que refleje la información relevante de cada ejemplo.
- Un algoritmo que aprenda a predecir la clase a la que pertenece cada nueva entrada a partir de las características de aprendizaje.

La construcción de un sistema de CP implica la combinación de los elementos anteriores siguiendo una serie de pasos preestablecidos:

1. Obtener el corpus de entrenamiento. Este conjunto debe ser los suficientemente amplio como para resultar representativo de todos los tipos de pregunta que se puedan dar en el dominio de trabajo. Cada ejemplo de este corpus constará de una pregunta etiquetada con su clase correspondiente.

2. Determinar el conjunto de *características* que describa adecuadamente los ejemplos de entrada del sistema. Cada pregunta quedará representada por un *vector de características* de tamaño fijo, en el que se incluye la clase a la que pertenece. Estos vectores son las *instancias* de entrenamiento del sistema de aprendizaje. La precisión de los sistemas de clasificación depende en gran medida de las características elegidas, por lo que buena parte del éxito de las aproximaciones basadas en aprendizaje automático depende de esta elección.

3. Seleccionar el algoritmo de aprendizaje en función del problema que vayamos a tratar. El número de muestras de aprendizaje, el número de clases posibles y el número de características definidas, son circunstancias que afectan al rendimiento de los algoritmos de clasificación. Seleccionar el más adecuado para una tarea concreta no es una cuestión trivial. En la sección 3.4 veremos en detalle algunos de los algoritmos aplicados de forma más habitual en los sistemas de CP basados en corpus.

4. Ajustar los parámetros del algoritmo y evaluación del mismo sobre un nuevo conjunto de datos de entrada sin clasificar conocido como *corpus de evaluación*. Para evaluar correctamente el rendimiento del clasificador, el conjunto de preguntas de evaluación debe ser diferente del conjunto de preguntas empleadas por el algoritmo durante la fase de entrenamiento.

En las siguientes secciones de este capítulo vamos a hacer un repaso pormenorizado de los distintos componentes que conforman los sistemas de CP basados en aprendizaje automático. Para ello analizaremos los sistemas de CP desde diferentes puntos de vista en función de los componentes anteriormente descritos:

- Según la taxonomía de tipos de pregunta que se pueden dar en el dominio de trabajo.

- Según los corpus de preguntas sobre los que el sistema aprende y clasifica.

- Según las características de aprendizaje definidas.
- Según los algoritmos de aprendizaje utilizados.

De forma paralela haremos una revisión del estado de la cuestión en la tarea de CP, citando los trabajos más importantes realizados en esta área. Completaremos este repaso con algunas aproximaciones especiales que se han realizado en la tarea de CP.

3.1. Taxonomías

Los tipos de pregunta[1] sobre los que es capaz de clasificar un sistema de CP se agrupan en una taxonomía[2] cerrada, estableciendo el conjunto posible de clases que se pueden dar en el corpus y que el clasificador debe de aprender a asignar correctamente a las preguntas. Las taxonomías no son específicas de las preguntas y se eligen en función del dominio del problema.

Las principales características que definen a una taxonomía son: el *tamaño*, la *estructura* y la *cobertura*. Vamos a ver cada una de éstas en detalle:

- **Tamaño**. El tamaño hace referencia al número de clases con que cuenta la taxonomía. Este tamaño da una idea de la dificultad de crear un clasificador que ofrezca un buen rendimiento. Como norma general, un incremento en el número de clases posibles supone un incremento paralelo en la dificultad para distinguir entre ellas. El tamaño de la taxonomía está condicionado por el grado de refinamiento que queramos alcanzar al clasificar. Por ejemplo, a la hora de localizar una respuesta para las preguntas "¿Qué ciudad canadiense tiene más habitantes?" y "¿Qué país regaló a Nueva York la estatua de la libertad?", asignar la clase *ciudad* y *país* respectivamente, puede ser más útil que simplemente saber que están preguntando por un *lugar*. Definir taxonomías más refinadas y con mayor número de clases se hace bajo la creencia de que tener información más precisa sobre la respuesta esperada puede mejorar el rendimiento final de los sistemas de BR (Paşca y Harabagiu, 2001).

  Podemos categorizar las taxonomías empleadas por los sistemas actuales como *pequeñas* (menos de 10 clases), *medianas* (entre 10 y 50 clases) y *grandes* (a partir de 50 clases). Dentro de la categoría de taxonomías pequeñas, encontramos los trabajos de García Cumbreras

[1] En literatura anglosajona es habitual referirse a este concepto como *question type* (Tomuro, 2002), *expected answer type* (Pinchak y Lin, 2006) o *QTarget* (Hovy et al., 2002).

[2] Otros términos habituales para referirse a las taxonomías son *jerarquía* (*hierarchy*) (Li y Roth, 2005) y *ontología* (*ontology*) (Metzler y Croft, 2005).

Person	*Object*
Organization	*Other*
Measure	*Place*
Date	

Figura 3.1: Ejemplo de taxonomía de pequeño tamaño empleada en (Solorio et al., 2004).

et al. (2006) (6 clases), Solorio et al. (2004) (7 clases), Singhal et al. (1999) (8 clases) e Ittycheriah et al. (2000) (9 clases). La figura 3.1 muestra un ejemplo de taxonomía de pequeño tamaño. La definición de una clase *other* es característica de los sistemas de CP en dominio abierto. Esta clase es un cajón de sastre para todas aquellas preguntas que no se ajustan a ninguna de las clases restantes de la taxonomía.

Las taxonomías de tamaño pequeño eran características de los primeros sistemas de CP. En la actualidad, la tendencia es al empleo de taxonomías de tamaño medio, más refinadas que las anteriores. Dentro de estas aproximaciones tenemos los sistemas de Pinto et al. (2002) (16 clases), Breck et al. (1999); Ferret et al. (1999); Radev et al. (2002) (17 clases), Prager et al. (1999) (20 clases), Cheung et al. (2004) (21 clases) y Metzler y Croft (2005) (31 clases). En muchas ocasiones el tamaño de la taxonomía está ligado a las herramientas empleadas por los sistemas de CP. La taxonomía definida por Metzler y Croft (2005), por ejemplo, queda condicionada por el uso del reconocedor de entidades BBN IdentiFinder (Bikel et al., 1999), del que heredan como clases las entidades que éste es capaz de detectar. La figura 3.2 muestra la taxonomía de clases definida en este trabajo.

Por último, existen aproximaciones con taxonomías que podemos definir como de gran tamaño. Destacan en este sentido los trabajos de Li y Roth (2002) (50 clases), Greenwood (2005) (67 clases, aunque en este caso su sistema de CP está basada en conocimiento) y Suzuki et al. (2003b) (68 clases). Hermjakob (2001) definen una taxonomía de 122 clases para el sistema Webclopedia de BR, aunque en este trabajo se limitan a evaluar un analizador sintáctico sobre las preguntas de cara a la tarea de CP, en lugar de evaluar el sistema de clasificación en sí mismo. Hovy et al. (2002) definen un conjunto de 180 clases, cada una de las cuales va acompañada de una serie de patrones para la extracción de respuestas sobre esas clases.[3] Su objetivo era ayudar al desarrollo rápido de sistemas de BR. (Sekine et al., 2002) define una jerarquía de casi 150 tipos de entidades de carácter general. Esta

[3]Tanto la taxonomía como los patrones están disponibles en http://www.isi.edu/natural-language/projects/webclopedia/Taxonomy/.

Animal	*Game*	*Person*
Biography	*Geo-Political Entity*	*Plant*
Cardinal	*Language*	*Product*
Cause/Effect/Influence	*Location*	*Product Description*
Contact Info	*Money*	*Quantity*
Date	*Nationality*	*Reason*
Definition	*Organization*	*Substance*
Disease	*Organization Description*	*Time*
Event	*Other*	*Use*
Facility	*Percent*	*Work of Art*
Facility Description		

Figura 3.2: Ejemplo de taxonomía de tamaño medio empleada en (Metzler y Croft, 2005).

taxonomía fue creada para el etiquetado de entidades en dominio abierto, por lo que su uso encaja perfectamente con el tipos de preguntas que se suelen dar en los sistemas de BR en dominio abierto. Hablaremos más en detalle sobre esta taxonomía en el capítulo siguiente. En la figura 3.3 puede verse un ejemplo de taxonomía de gran tamaño (Greenwood, 2005).

- **Estructura**. A la hora de definir una taxonomía podemos hacerlo en un único nivel, obteniendo una taxonomía *plana*, o a diferentes niveles, obteniendo una taxonomía *jerárquica*. En este último caso tenemos una serie de subclases que dependen de una clase de nivel superior. Organizar las clases de forma jerárquica permite al sistema volver de un tipo de pregunta específico hacia uno más general cuando no lea sea posible clasificar de forma refinada.

  La estructura empleada en la taxonomía está íntimamente relacionada con el tamaño de la misma. Las taxonomías de tamaño pequeño son siempre planas, ya que la definición de pocas clases no da pie a establecer jerarquías. Todas las taxonomías que clasificábamos en el punto anterior como *pequeñas* son de este tipo. Mientras, las taxonomías de tamaño medio pueden ser tanto planas como jerárquicas. Entre las citadas anteriormente, las definidas por Pinto et al. (2002), Radev et al. (2002), Prager et al. (1999) y Cheung et al. (2004) son de tipo plano. Entre las jerárquicas tenemos las de Breck et al. (1999) y Ferret et al. (1999). Por otra parte, las taxonomías grandes suelen ser siempre de tipo jerárquico. La taxonomía de Li y Roth (2002) presenta un primer nivel de 6 clases y un segundo nivel de 50. Greenwood (2005) y Suzuki et al. (2003b) establecen hasta cuatro niveles diferentes en su taxonomía. La jerarquía de Greenwood (2005), mostrada en la figura 3.3, se estructura en cuatro niveles. Presenta

- *Amount*
  - *Money*
  - *Percent*
  - *Measurement*
    - *Time*
      - *How Often*
      - *Age*
    - *Distance*
    - *Speed*
    - *Temperature*
    - *Area*
    - *Mass*
    - *Computer*
    - *Other*
- *Currency Unit*
- *Reward*
  - *Award*
  - *Prize*
  - *Trophy*
  - *Medal*
- *Date*
  - *Month*
- *Proper Name*
- *State Motto*
- *State Bird*
- *State Flower*
- *State Tree*
- *State Nickname*
- *Organization*
- *Planet*
- *God*
  - *Egyptian*
  - *Greek*
  - *Roman*
- *Person*
  - *Male*
  - *Female*
- *Location*
  - *City*
  - *Province*
  - *Country*
  - *National Park*
  - *Lake*
  - *River*
  - *Cemetery*
  - *Continent*
  - *US State*
- *Language*
- *Nationality*
- *National Anthem*
- *Religion*
- *Space Craft*
- *Job Title*
- *Quoted Text*
- *Zodiac Sign*
- *Birthstone*
- *Address*
- *Colour*
- *Colour List*
- *Unknown Proper Name*
- *Chemical Element*
  - *Symbol*
  - *Name*
- *Chemical Compound*
  - *Symbol*
  - *Name*

Figura 3.3: Ejemplo de taxonomía de tamaño grande empleada en (Greenwood, 2005).

algunas clases muy refinadas, como *Zodiac Sign*, cuyo conjunto posible de respuestas está totalmente acotado (a los doce signos del zodíaco). Esta jerarquía está definida a la medida del tipo de preguntas que su sistema de BR es capaz de responder.

En cuanto a los argumentos a favor de emplear una u otra estructura para las taxonomías, Suzuki et al. (2003b) recomiendan la definición de jerarquías de preguntas por su utilidad para que procesos posteriores de los sistemas de BR puedan beneficiarse de esta estructura. Otro trabajo que argumenta a favor de la clasificación jerárquica es el de Manning et al. (2008), donde defienden que este tipo de organización de las taxonomías es prometedora, aunque en la actualidad la mejora que se consigue en clasificación trabajando con una jerarquía en lugar de hacerlo con una taxonomía plana es bastante modesta. Aún así, defienden que la técnica puede ser útil simplemente para mejorar la escalabilidad a la hora de construir clasificadores sobre grandes taxonomías. Li y Roth (2002) definen una jerarquía de clases de preguntas a dos niveles y experimentan con un clasificador jerárquico y un clasificador plano. Los resultados empíricos revelan que el sistema no mejora por usar una aproximación u otra.

- **Cobertura**. A la hora de definir las clases que van a ser incluidas en la taxonomía del clasificador, pueden seguirse dos enfoques diferentes que dependerán de la intención de nuestro desarrollo (Greenwood, 2005): definir clases que cubran la mayoría de las preguntas que pueden darse en el dominio en el que trabajemos, o definir clases en función de los tipos de respuesta que el sistema de BR, en el que se vaya a incluir el sistema de CP, sea capaz de localizar y extraer.

  El primer enfoque nos situaría en la perspectiva de un mundo ideal, donde los sistemas de BR serían capaces de responder cualquier pregunta que se les hiciera. Para que esto ocurra, el conjunto de clases de la taxonomía debe ser lo suficientemente amplio para cubrir la mayoría de tipos de pregunta posibles. Ésta es la visión seguida por Li y Roth (2002) al diseñar una jerarquía de 50 clases a dos niveles con la intención de cubrir las preguntas que típicamente se dan en las conferencias TREC. En el trabajo desarrollado por Hovy et al. (2000) se define una jerarquía similar a la anterior para el sistema de Webclopedia de BR con un total de 94 clases posibles. En un principio estas taxonomías parecen útiles, especialmente cuando se combinan con un conjunto apropiado de ejemplos etiquetados de cada clase para su aprendizaje. El hecho de que un sistema de CP sea capaz de clasificar una pregunta, no significa que automáticamente el sistema de BR en el que se integre sea capaz de responder a preguntas de ese tipo. Por citar un ejemplo, en la taxonomía definida para la Webclopedia

encontramos la clase *C-PROPER-PLANET* (relativa a nombres de planetas), que abarcaría preguntas como "¿Cuál es el sexto planeta desde el sol?". Si un sistema de BR no es capaz de detectar en los documentos los nombres de planetas, este tipo de respuesta esperada sólo servirá para que el sistema admita que es incapaz de responder a la pregunta. Este tema es tratado en profundidad en el trabajo de Ittycheriah et al. (2000).

El segundo enfoque para la construcción de la taxonomía de tipos de pregunta es el de comenzar por estudiar los tipos de respuesta que el sistema de BR, en el que se incluya el sistema de CP, sea capaz de localizar y extraer del texto. Afrontar el problema en esta dirección tiene sus inconvenientes, ya que implica construir una nueva taxonomía para cada sistema de BR en el que se incluya el sistema de CP, no siendo fácil en ocasiones transferir los datos de entrenamiento de un sistema a otro. Si la intención es construir un sistema de CP abierto, que se pueda incorporar en diferentes sistemas de BR, la primera aproximación sería la más adecuada.

Dentro del campo de la CP aplicada a la BR, las clases que habitualmente se dan en las taxonomías actuales son clases semánticas. Se han diseñado para que puedan ser semánticamente separables, en contraposición con las taxonomías conceptuales que se definieron en los primeros sistemas de BR. Es el caso de las taxonomías de Wendy Lehnert y Arthur Graesser. Lehnert definió una taxonomía conceptual para su sistema QUALM (Lehnert, 1977b) compuesta por 13 clases orientadas a la BR en el contexto de la comprensión de textos. La figura 3.4 muestra esta taxonomía con una serie de ejemplos asociados a cada clase. Por otra parte, Graesser extendió la taxonomía original de Lehnert añadiendo nuevas categorías y haciendo un estudio empírico de la cobertura, demostrando su capacidad para ajustarse a todas las preguntas posibles del discurso humano. Esta taxonomía alcanzó su forma final en el trabajo descrito en (Graesser et al., 1994), donde aparece dividida en dos subtipos: preguntas que requieren una respuesta *corta* frente a preguntas que requieren una respuesta *larga*. Esta taxonomía puede verse en la figura 3.5. Las cinco primeras clases (*Verification*, *Disjunctive*, *Concept completion*, *Feature specification* y *Quantification*) esperan una respuesta corta, mientras que el resto esperan una respuesta larga.

Tanto Lehnert como Graesser utilizan estas taxonomías como marco teórico. Dejan en manos de los investigadores su uso para la clasificación de preguntas de usuarios reales. En el contexto de la CP aplicada a los sistemas de BR actuales, las categorías conceptuales propuestas por Lehnert y Graesser no resultan de utilidad. Como ya se comentó en la sección 2.5.4, las preguntas de tipo *Verification* (interrogativas totales) y *Disjunctive* (interrogativas disyuntivas) requieren un conocimiento del mundo que los sistemas de BR difícilmente pueden alcanzar en la actualidad.

Causal Antecedent	Why did John go to New York?
	What resulted in John's leaving?
Goal Orientation	Why did Mary drop the book?
	Mary left for what reason?
Enablement	How was John able to eat?
	What did John need to do in order to leave?
Causal Consequent	What happened when John left?
	What if I don't leave?
Verification	Did John leave?
	Does John think that Mary left?
Disjunctive	Was John or Mary here?
	Is John coming or going?
Instrumental/Procedural	How did John go to New York?
	What did John use to eat?
Concept Completion	What did John eat?
	Who gave Mary the book?
Expectational	Why didn't John go to New York?
	Why isn't John eating?
Judgmental	What should I do to keep her from leaving?
	What should John do now?
Quantification	How many people are there?
	How ill was John?
Feature Specification	What color are John's eyes?
	What breed of dog is Rover?
Request	Would you pass the salt?
	Can you get me my coat?

Figura 3.4: Taxonomía conceptual y ejemplos de preguntas usados en el sistema QUALM de Wendy Lehnert.

Verification	Is a fact true?
	Did an event occur?
Disjunctive	Is X or Y the case?
	Is X, Y, or Z the case?
Concept completion	Who? What? When? Where?
	What is the referent of a noun argument slot?
Feature specification	What qualitative attributes does entity X have?
	What is the value of a qualitative variable?
Quantification	What is the value of a quantitative variable?
	How much? How many?
Definition	What does X mean?
	What is the superordinate category and some properties of X?
Example	What is an example of X?
	What is a particular instance of the category?
Comparison	How is X similar to Y?
	How is X different from Y?
Interpretation	How is a particular event interpreted or summarized?
Causal antecedent	What caused some event to occur?
	What state or event causally led to an event or state?
Causal consequence	What are the consequences of an event or state?
	What causally unfolds from an event or state?
Goal orientation	What are the motives behind an agent's actions?
	What goals inspired an agent to perform an action?
Instrumental/Proc.	How does an agent accomplish a goal?
	What instrument or body part is used when an agent performs an action?
Enablement	What object or resource anables an agent to perform an action?
	What plan of action accomplishes an agent's goal?
Expectational	Why did some expected event not occur?
Judgmental	The questioner wants the answerer to judge an idea or to give advice on what to do.
Assertion	The speaker expresses that he or she is missing some information.
Request/Directive	The speaker directly requests that the listener supply some information.

Figura 3.5: Taxonomía conceptual y ejemplos de preguntas de Arthur Graesser.

Pese a que se han definido numerosas taxonomías de preguntas, no existe ninguna que pueda calificarse de estándar. Tampoco existe un consenso bien definido sobre cómo deberían ser las taxonomías a emplear en los sistemas de CP. El tamaño y la estructura influyen decisivamente en la dificultad de la tarea, haciendo que en numerosas ocasiones resulte difícil comparar el rendimiento de los sistemas entre sí.

3.2. Corpus

En el ámbito de PLN, un *corpus lingüístico* es una colección de textos en soporte electrónico, normalmente amplio, que contiene ejemplos reales de uso de una lengua tal y como es utilizada por los hablantes, con sus errores, peculiaridades y excepciones (Navarro, 2006). Para que un sistema de CP basado en aprendizaje sea capaz de clasificar sobre una taxonomía, es necesario un corpus con preguntas de ejemplo etiquetadas con dicha taxonomía. Los corpus utilizados en los sistemas de clasificación basados en aprendizaje automático cumplen tres funciones:

- **Entrenamiento**. Las preguntas etiquetadas correctamente contenidas en el corpus de entrenamiento permiten al sistema aprender a clasificar nuevas instancias. El conjunto de entrenamiento debe ser representativo de las situaciones con las que se puede encontrar el sistema de CP durante su funcionamiento.

- **Validación**. Si fuera necesario, una porción del conjunto de entrenamiento puede emplearse para ajustar los parámetros del algoritmo de aprendizaje mediante la optimización de su funcionamiento en este subconjunto. Este proceso de optimización se puede también llevar a cabo sobre todo el conjunto de entrenamiento realizando una validación cruzada.[4]

- **Evaluación**. Una vez tenemos el sistema entrenado, el siguiente paso es evaluar su funcionamiento. Esta evaluación se lleva a cabo proporcionando al sistema un nuevo conjunto de ejemplos (preguntas) para los que el sistema determinará la clase a la que pertenecen. La clase que predice el sistema se contrasta con la clase real asignada previamente por un humano (o a través de algún tipo de medición) para obtener el rendimiento del clasificador. Para que la evaluación sea fiable, el conjunto de preguntas de evaluación debe ser diferente al empleado durante el entrenamiento. Al igual que ocurría en el punto anterior, podemos realizar una validación cruzada sobre el corpus de entrenamiento para evitar la necesidad de definir un corpus específico de evaluación. Esta técnica es especialmente útil cuando los corpus de

[4] En la sección 4.4.2 hablaremos más en detalle de esta técnica.

entrenamiento son pequeños y no se desea dedicar una parte de ellos exclusivamente para la evaluación.

Las principales características que definen un corpus son: el *tamaño*, el *dominio* de aplicación y el *idioma*. Veamos cada uno de estos conceptos en detalle:

- **Tamaño**. Esta característica hace referencia a la cantidad de preguntas que se han etiquetado y que conforman el corpus. En los sistemas de aprendizaje automático supervisado, el número de muestras de entrenamiento resulta fundamental para el rendimiento final del sistema. Si el número de características de aprendizaje es muy grande comparado con el número de muestras de entrenamiento, existe riesgo de que se produzca un *sobreajuste* (*overfitting*). El problema del sobreajuste (también conocido como *the curse of dimensionality*) tiene lugar cuando un algoritmo se sobreentrena sobre un conjunto de datos, ajustándose en exceso a las características específicas del corpus de entrenamiento y perdiendo capacidad de generalización. Este efecto hace que los sistemas de aprendizaje se comporten muy bien cuando se clasifican las instancias de entrenamiento, pero pierdan capacidad cuando llega una instancia nueva no vista con anterioridad. Para evitar este problema nos interesa tener un número suficientemente amplio de muestras por cada clase para que el sistema las pueda caracterizar de forma adecuada. Un problema común que suele darse en los corpus de entrenamiento es el de las *clases sesgadas*. Este fenómeno tiene lugar cuando existen clases de la taxonomía para las que tenemos muy pocas muestras. Un ejemplo de este problema se plantea en (Li y Roth, 2002), donde los autores desarrollan un corpus de preguntas en el que el número de muestras de algunas clases se incrementó de forma artificial para evitar el problema del sesgo.

  En los sistemas de CP basados en corpus se han empleado conjuntos de datos de muy diversos tamaños. Existen desarrollos con corpus pequeños, como el descrito por Solorio et al. (2004) y en el que emplean un corpus de 450 preguntas tanto para entrenamiento como para evaluación, realizando para ello una validación cruzada. Existen otros corpus de un tamaño medio, como el creado por Li y Roth (2002) con 5.452 preguntas de entrenamiento y 500 de evaluación. Otros corpus más grandes se han descrito, como el utilizado por Li y Roth (2005) con 21.500 preguntas de entrenamiento y 1.000 más de evaluación.

- **Dominio**. El auge de los sistemas de BR llegó de la mano de las conferencias internacionales TREC, CLEF y NTCIR. En estas competiciones se proporcionaba a los participantes un conjunto de preguntas de evaluación que los sistemas de BR debían ser capaces

de responder. Las preguntas que habitualmente se daban en estas competiciones eran de tipo factual y en dominio abierto. Fueron generadas, revisadas y corregidas de forma artificial sobre los corpus de documentos proporcionados por los organizadores. Son numerosos los sistemas de CP que han trabajado sobre corpus formados por preguntas pertenecientes a estas competiciones o por otras de estilo similar. Por citar algunos ejemplos, Day et al. (2007) emplean preguntas del NTCIR, Solorio et al. (2004) utilizan preguntas del CLEF y Li y Roth (2002) emplean un corpus formado por preguntas del TREC complementadas con otras de estilo similar.

Sundblad (2007) realiza una crítica sobre el empleo de estos corpus artificiales en los sistemas de CP, planteando cuál sería el funcionamiento de estos sistemas ante preguntas de usuarios reales. Para evaluar este punto retoma la taxonomía empleada por Li y Roth (2002) y la utiliza para etiquetar un corpus de preguntas de usuarios reales extraídas de los registros de Answerbus.[5] El rendimiento obtenido sobre este corpus fue prácticamente equivalente al obtenido sobre el corpus de preguntas original. Este resultado rechaza la hipótesis inicial de que el rendimiento de los sistemas de CP puede verse afectado cuando se aplican a preguntas obtenidas de usuarios reales.

Otro ejemplo de corpus obtenido de usuarios reales es el desarrollado por Beitzel et al. (2007) al recopilar preguntas extraídas de los registros del buscador AOL.[6]

- **Idioma**. El idioma en el que estén las preguntas del corpus va a condicionar el idioma sobre el que va a ser capaz de clasificar nuestro sistema. Por ejemplo, para desarrollar un CP para español, el sistema deberá ser entrenado sobre un corpus de preguntas en español. Un corpus de preguntas no contiene necesariamente textos en un sólo idioma. Podemos tener corpus en los que se den dos idiomas (*corpus bilingüe*) o corpus en los que se den varios (*corpus multilingüe*). Esta denominación no implica que las preguntas en los distintos idiomas sean las mismas. A un corpus que presenta las mismas preguntas en diferentes idiomas lo denominaremos *corpus paralelo* (Hallebeek, 1999).

  Los corpus de preguntas desarrollados para CP son mayoritariamente en inglés (Cheung et al., 2004; Hermjakob, 2001; Li y Roth, 2002, 2005). Aunque de forma minoritaria, se han realizado investigaciones en otros idiomas como el finés (Aunimo y Kuuskoski, 2005), el estonio (Hennoste et al., 2005), el francés (Feiguina y Kégl, 2005), el chino (Day et al., 2005; Lin et al., 2006), el japonés (Suzuki et al., 2003b),

[5] http://www.answerbus.com.
[6] http://search.aol.com.

el portugués (Solorio et al., 2005) y el español (García Cumbreras et al., 2005; Tomás et al., 2005).

Existen corpus de preguntas desarrollados directamente para la evaluación de sistemas de CP y otros desarrollados de forma más general para la tarea de BR. Todos ellos resultan igualmente útiles para evaluar la tarea de clasificar preguntas. Vamos a describir brevemente algunos de los corpus de preguntas libremente disponibles que existen en la actualidad:

- *DISEQuA*. Dentro del marco de las conferencias CLEF, los grupos coordinadores[7] de la tarea de BR de 2003 recopilaron un conjunto de preguntas originales para la tarea monolingüe[8] en cuatro idiomas diferentes: holandés, italiano, español e inglés. Este corpus es conocido como DISEQuA[9] (Dutch Italian Spanish English Questions and Answers) (Magnini et al., 2003). Está etiquetado en formato XML y no sólo contiene preguntas, sino también sus correspondientes respuestas. El corpus consta de 450 preguntas y respuestas en los citados idiomas. Las preguntas están etiquetadas mediante una taxonomía plana con 7 clases posibles: *Person*, *Organization*, *Measure*, *Date*, *Object*, *Other* y *Place*. Este corpus se ha empleado en sistemas como los de Solorio et al. (2004) y Tomás y Vicedo (2007a).

- *Webclopedia*. El corpus desarrollado para el sistema Webclopedia[10] de BR es un corpus de preguntas de gran tamaño en inglés (Hermjakob, 2001). Consta de más de 27.000 preguntas provenientes de Answers.com[11] etiquetadas con una taxonomía de 122 clases posibles.

- *UIUC*. El corpus desarrollado por Li y Roth (2002) se ha convertido en el corpus más utilizado para la evaluación de sistemas de CP.[12] El corpus consta de 5.452 preguntas de entrenamiento y 500 más de evaluación en inglés. Las preguntas fueron etiquetadas empleando una jerarquía de clases de dos niveles: un primer nivel de 6 clases y un segundo nivel de 50. Algunos de los sistemas que han trabajado sobre este corpus son los de Li y Roth (2002, 2005), Zhang y Lee (2003), Hacioglu y Ward (2003), Krishnan et al. (2005), Skowron y Araki (2004a) y Nguyen et al. (2008). En el capítulo 5 hablaremos de forma más extensa de este corpus y lo emplearemos en nuestros propios experimentos.

[7] Estos grupos fueron ITC-irst (Italia), UNED (España) e ILLC (Holanda).

[8] Aquella en la cual las preguntas y los textos sobre los que buscar están en el mismo idioma.

[9] Disponible en http://clef-qa.itc.it/2004/down/DISEQuA_v1.0.zip.

[10] Se puede encontrar información adicional sobre el mismo en http://www.isi.edu/natural-language/projects/webclopedia/Taxonomy-data/.

[11] http://www.answers.com.

[12] Disponible en http://l2r.cs.uiuc.edu/~cogcomp/Data/QA/QC/.

- *Multieight-04*. Este corpus[13] está formado por una colección de 700 preguntas que fueron formuladas para la tarea de BR multilingüe del CLEF 2004 (Magnini et al., 2005). Las preguntas del corpus se encuentran en 7 idiomas diferentes (alemán, inglés, español, francés, italiano, holandés y portugués), incluyendo también las respuestas a las preguntas formuladas. Existen dos idiomas más, el búlgaro y el finés, para los que se tradujeron las preguntas pero no las respuestas. El corpus, en formato XML, está etiquetado siguiendo una taxonomía de 8 clases posibles: *LOCATION*, *MANNER*, *MEASURE*, *OBJECT*, *ORGANIZATION*, *PERSON*, *TIME* y *OTHER*.

- *TREC, CLEF y NTCIR*. Existen un gran número de preguntas disponibles provenientes de las diferentes ediciones de estas competiciones. Estas preguntas se desarrollaron para evaluar sistemas de BR y no están etiquetadas, ya que cada sistema de BR usa su propia taxonomía de preguntas. La tarea de BR de las conferencias TREC se viene celebrando desde la octava edición de esta competición (TREC-8), que tuvo lugar en 1999 y se extendió hasta el año 2007 (TREC-16). Entre 1999 y 2003 se liberaron 2.393 preguntas para la tarea principal. Entre 2004 y 2007 las preguntas se estructuraron de manera diferente, agrupándolas en torno a un tópico común (285 tópicos en total).

  Las conferencias CLEF se han caracterizado por el tratamiento de múltiples idiomas. En el año 2003 se introdujo la tarea de BR multilingüe, prolongada hasta el año 2008. Los primeros idiomas tratados fueron el holandés, el italiano y el español para la tarea monolingüe, y el holandés, francés, alemán, italiano y español para la tarea translingüe (*cross-lingual*), en la que la pregunta se formulaba en estos idiomas y la respuesta se buscaba en un corpus de documentos en inglés. En 2008 el conjunto de idiomas se extendió al búlgaro, inglés, francés, alemán, italiano, portugués, rumano, griego, vasco y español. Cada año se han proporcionado 200 preguntas en cada uno de los idiomas con el mismo estilo que las formuladas en el TREC.

  Las conferencias NTCIR han seguido las mismas pautas que las otras dos mencionadas anteriormente pero centrándose en los idiomas japonés, chino e inglés. El primer NTCIR que abordó la tarea de BR fue el de 2003 (NTCIR-3). En él se proporcionaron 1.200 preguntas en japonés con su correspondiente traducción al inglés. Este conjunto se amplió en el NTCIR-4 con 650 preguntas más en japonés y su correspondiente traducción en inglés. En el año siguiente se incorporó la tarea de BR translingüe aportando 200 preguntas más en chino, japonés e inglés. Adicionalmente, para la tarea de BR

[13] Disponible en http://clef-qa.itc.it/2005/down/corpora/multieight-04_v1.2.zip.

monolingüe se crearon 360 preguntas más. Por último, en el NTCIR-6 se crearon 200 preguntas en japonés para la tarea monolingüe y tres conjuntos más de preguntas, 200 en japonés, 150 en chino y 350 en inglés traducidas de las anteriores para la tarea translingüe.

Son diversos los trabajos que han utilizado estas preguntas. Day et al. (2007) emplean los recursos proporcionados en las conferencias NTCIR, obteniendo 2.322 preguntas en chino para entrenamiento y 150 más para la evaluación, etiquetadas con la taxonomía de Li y Roth (2002). Li (2002) obtiene un pequeño conjunto de 693 preguntas del TREC etiquetas con 7 clases diferentes. Radev et al. (2002) utilizan un subconjunto del TREC de 1.200 preguntas etiquetados con 17 tipos posibles de respuesta.

Además de los ya mencionados, podemos citar otros ejemplos de corpus empleados en sistemas de CP. Metzler y Croft (2005) crean un corpus propio llamado MadSci, consistente en 250 preguntas obtenidas de usuarios reales de la Web MadSci[14] y etiquetadas con una versión extendida de las entidades detectadas por IdentiFinder. Suzuki et al. (2003b) definen un corpus de 5.011 preguntas en japonés con una taxonomía de 150 clases. Ittycheriah et al. (2000) emplean un conjunto de 1.900 preguntas específicamente creadas y etiquetadas para la tarea, así como 1.400 preguntas provenientes de una base de datos de juegos de preguntas y respuestas, etiquetadas siguiendo las categorías MUC.

3.3. Características de aprendizaje

La precisión del algoritmo de clasificación depende fuertemente de cómo se representen los ejemplos de entrada. Estos ejemplos se codifican empleando un conjunto de características de aprendizaje que representan ciertos atributos del objeto independientes de la clase a la que pertenezca. Estas características se combinan formando un *vector de características*.

En la mayoría de problemas de clasificación, la obtención de las características que se emplean para el aprendizaje es un arte. Con la excepción de algunas técnicas basadas en redes neuronales y algoritmos genéticos que son capaces de intuirlas de forma automática, la identificación manual de un buen conjunto de características forma las bases de casi todos los problemas de clasificación. Este proceso se conoce habitualmente como *feature engineering* (Scott, 1999). La elección de un buen conjunto de características puede mejorar considerablemente el funcionamiento del clasificador. Para que el aprendizaje sea realmente automático, éstas tienen que poder extraerse de forma automática a partir de las preguntas de entrada. Las características empleadas son independientes del algoritmo de

[14] http://www.madsci.org.

aprendizaje a utilizar. Es frecuente el caso en el que la mejora del sistema es mayor explotando características específicas del dominio que cambiando de un algoritmo de aprendizaje a otro. Un mejor conjunto de características proporciona una mejor representación de la pregunta, traduciéndose en último lugar en un mejor rendimiento del clasificador.

La escasa cantidad de texto presente en una pregunta hacen que la gran mayoría de los trabajos desarrollados en CP opten por utilizar herramientas y recursos lingüísticos para enriquecer el vector de características. Diferentes recursos léxicos, sintácticos y semánticos se han empleado con este propósito. En esta sección revisamos las características de aprendizaje más relevantes empleadas en el desarrollo de sistemas de CP y los recursos de los que se derivan. Para este estudio, vamos a agrupar las características en función del nivel de análisis lingüístico utilizado: *n-gramas*, *características léxicas*, *características sintácticas* y *características semánticas*.

3.3.1. N-gramas

La forma más sencilla de construir el vector de características es mediante las propias palabras que forman la pregunta. Cuando estas palabras se cogen sin tener en cuenta la estructura gramatical ni el orden en el que aparecen en la pregunta, tenemos lo que se conoce como modelo de *bolsa de palabras* o *bag-of-words* (BOW). Esta simplificación a la hora de representar el texto se utiliza en tareas de PLN como la clasificación de textos o la RI. En este modelo de representación, una pregunta como "¿Quién disparó a JFK?" sería totalmente idéntica a "¿A quién disparó JFK?", mientras que no compartiría ninguna similitud con "Dime el nombre del asesino de John F. Kennedy?".

Cuando las palabras en lugar de tomarse de forma individual se toman en secuencias de n elementos, tenemos lo que se conocen como *n-gramas*. Un n-grama es una subsecuencia de n elementos de una secuencia dada. Los n-gramas más utilizados en tareas de PLN son los de tamaño 1 (las propias palabras que comentábamos en el párrafo anterior), los de tamaño 2 y los de tamaño 3, conocidos respectivamente como *unigramas* (1-gramas), *bigramas* (2-gramas) y *trigramas* (3-gramas). En la figura 3.6 podemos ver un ejemplo de estas subsecuencias para la pregunta "¿Cuál es la ciudad más grande de Europa?".

Los n-gramas son la forma más sencilla de explotar las dependencias entre palabras. Representan el orden de las palabras en una ventana de n términos y permiten reflejar mejor la estructura del texto que el modelo BOW. A la hora de decidir el tamaño de n-gramas que se va a emplear como característica de aprendizaje, hay que tener en cuenta que usar n-gramas de mayor orden hace que éstos se dispersen y haya menos repeticiones. Por ejemplo, la probabilidad de que el unigrama "ciudad" aparezca en una pregunta siempre será mayor o igual que la probabilidad de que aparezca el trigrama "dime la ciudad".

¿Cuál es la ciudad más grande de Europa?

unigramas: `Cuál`, `es`, `la`, `ciudad`, `más`, `grande`, `de`, `Europa`

bigramas: `Cuál es`, `es la`, `la ciudad`, `ciudad más`, `más grande`, `grande de`, `de Europa`

trigramas: `Cuál es la`, `es la ciudad`, `la ciudad más`, `ciudad más grande`, `más grande de`, `grande de Europa`

Figura 3.6: Ejemplo de descomposición de una pregunta en unigramas, bigramas y trigramas.

La ventaja de los n-gramas como modelo de representación es la sencillez para obtenerlos a partir de un texto. Sólo es necesario hacer una división en componentes léxicos (*tokens*) de las palabras de la pregunta y agruparlas según el tamaño de n-grama deseado. Por esta razón son numerosos los sistemas de CP que utilizan los n-gramas como características de aprendizaje. Los tamaños más empleados son unigramas (Li y Roth, 2002; Moschitti et al., 2007; Nguyen et al., 2008; Skowron y Araki, 2004a; Suzuki et al., 2003b), bigramas (Day et al., 2007; García Cumbreras et al., 2006; Huang et al., 2007; Li et al., 2005; Metzler y Croft, 2005) y combinaciones de unigramas y bigramas (Hacioglu y Ward, 2003; Krishnan et al., 2005; Pinto et al., 2002). Solorio et al. (2004) utilizan un modelo de unigramas y lo comparan con otro obtenido mediante prefijos de un determinado tamaño (los 4 o 5 primeros caracteres de las palabras), obteniendo mejor rendimiento con este último para los experimentos realizados en español e italiano (no así para inglés). Kocik (2004) sólo tiene en cuenta el primer unigrama y el primer bigrama de la pregunta. García Cumbreras et al. (2006) y Tomás et al. (2005) utilizan de forma explícita el número de unigramas detectados en la pregunta como característica de aprendizaje. Tomás et al. (2005) y Bisbal et al. (2005b) utilizan combinaciones de los cuatro primeros unigramas y bigramas de la pregunta.

Los n-gramas han demostrado ser tremendamente efectivos en la tarea de clasificación de textos. Sin embargo, en la tarea de CP la mayoría de sistemas utiliza otro tipo de herramientas y recursos lingüísticos para enriquecer el espacio de características. Vamos a repasar en los siguientes puntos cuáles son estas otras características usadas de forma habitual en la tarea de CP.

3.3.2. Características léxicas

Se incluyen en este apartado las características obtenidas a través de herramientas que trabajan a nivel de palabra. Entre estas herramientas encontramos los *lematizadores* y los *etiquetadores morfológicos*.

Un *lematizador* es una herramienta capaz de reconocer una palabra y obtener su *lema* o *forma canónica*.[15] Permiten determinar una forma única común a todas las posibles formas de una misma palabra. Otra herramienta que realiza una función similar es el *stemmer*. Esta herramienta permite reducir una palabra a su *raíz* o *tema*. El algoritmo más conocido para realizar este proceso es el de Porter (1997), que ha demostrado su buen funcionamiento sobre textos en inglés. Sin embargo, su rendimiento es más discutible para idiomas como el español, con morfología más rica y mayor flexión verbal. Nguyen et al. (2008) y Pinchak y Lin (2006) utilizan formas lematizadas para normalizar las palabras antes de incluirlas en el vector de características. Bisbal et al. (2005a) realizan una comparación de vectores de características empleando n-gramas y su versión lematizada. Los resultados revelan que no hay mejoras significativas entre el uso de unos y otros en diversos experimentos realizados en inglés y español.

Por otra parte, los *etiquetadores morfológicos* o *part-of-speech taggers* permiten obtener la categoría gramatical (si es un sustantivo, un adjetivo, un verbo, etc.) de cada una de las palabras que forman un texto, eliminando la ambigüedad gramatical que pueda darse. La desambiguación léxica se ha usado en muchos sistemas, siempre como complemento a otras características de aprendizaje. Es el caso de los trabajos desarrollados en (Day et al., 2007; Krishnan et al., 2005; Nguyen et al., 2008; Pinto et al., 2002; Skowron y Araki, 2004a) (AUTOTAG), (Suzuki et al., 2003b) (CaboCha), (García Cumbreras et al., 2006; Li y Roth, 2002) (TreeTagger), (Metzler y Croft, 2005) (MXPost) y (Cheung et al., 2004) (Brill POS tagger). Otros sistemas no incluyen directamente esta información como característica de aprendizaje, sino que la utilizan para, por ejemplo, detectar el primer sustantivo que tiene lugar en la pregunta y usarlo como característica de aprendizaje (Bisbal et al., 2005b).

3.3.3. Características sintácticas

El análisis sintáctico es el núcleo fundamental de muchos sistemas de PLN. Proporcionan una estructura asociada a la oración que refleja las relaciones sintácticas de sus componentes, dando una idea de las dependencias que existen entre las palabras. Este tipo de información

[15] Sería el equivalente a un ítem de entrada en un diccionario. Por ejemplo, para un verbo ("compraron") devolvería su infinitivo ("comprar"), mientras que para un nombre ("gatas") o un adjetivo ("roja") devolvería su forma en masculino singular ("gato" y "rojo").

se ha utilizado de forma habitual en sistemas de CP, principalmente en forma de análisis sintáctico superficial. Una aplicación de estos analizadores es la detección de constituyentes sintácticos (sintagmas), identificando la jerarquía sintáctica entre los elementos de la oración. Otra aplicación es la detección de dependencias sintácticas, identificando las relaciones de dependencia entre los elementos de la oración.

El uso de información sintáctica dentro de los sistemas de CP es controvertido. Los analizadores sintácticos están enfocados a su aplicación sobre textos declarativos. Aunque las preguntas son típicamente más cortas que el texto declarativo (lo cual debería hacer que su análisis resultase más sencillo), el orden de las palabras es marcadamente diferente. Esto hace que los analizadores sintácticos empeoren su rendimiento a la hora de etiquetarlas. Hermjakob (2001) se hace eco de este problema y desarrolla su propio banco de árboles sintácticos (*treebank*) de preguntas. Este corpus, formado por 1.153 preguntas obtenidas del TREC y de diversos recursos de la Web, permite entrenar el analizador sintáctico para que aprenda a etiquetar adecuadamente oraciones interrogativas. De esta forma, emplean el analizador sintáctico CONTEX (Hermjakob y Mooney, 1997) para analizar las preguntas entrenándolo no sólo sobre el Penn Treebank (un corpus constituido por noticias de prensa), sino también sobre el corpus de preguntas desarrollado por ellos. Demuestran una mejora considerable en el rendimiento del analizador sintáctico sobre preguntas, aunque no analizan el impacto de este análisis en la clasificación. Otro analizador sintáctico especializado en preguntas es SUPPLE (Gaizauskas et al., 2005b). Fue empleado dentro de un sistema completo de BR (Gaizauskas et al., 2005a) aunque no hay datos sobre la aportación de esta herramienta sobre el rendimiento final del sistema de CP.

En los experimentos realizados por Li y Roth (2002), el uso de información sintáctica no parece aportar mejoras a la CP. La inclusión en el vector de características de sintagmas (*chunks*) y del primer sintagma nominal de la pregunta (*head chunk*) no mejora el rendimiento del sistema basado en BOW. Por otra parte, Li y Roth (2005) apenas consiguen mejorar la representación mediante n-gramas utilizando la detección de sintagmas. En este mismo sistema emplean las estructuras de dependencia entre palabras detectadas por el analizador sintáctico MiniPar (Lin, 1998), obteniendo mejores resultados con los pares de palabras dependientes que detecta este sistema que con la representación basada en bigramas.

Cheung et al. (2004) usan un analizar sintáctico LR para la extracción de sintagmas nominales y sus núcleos con la intención de agrupar palabras y obtener secuencias con un mayor significado (como "*car batteries*", "*hard time*" o "*human eye*"). García Cumbreras et al. (2006) emplean también los núcleos sintácticos como características de aprendizaje. En estos dos trabajos no hay datos específicos sobre el rendimiento ofrecido por este tipo de información.

3.3.4. Características semánticas

Aunque es posible conseguir buenos resultados usando n-gramas, características léxicas y sintácticas, numerosas aplicaciones han demostrado que el uso de características semánticas mejora la representación de los ejemplos de entrada. Por ejemplo, las preguntas "¿Cuál es el animal más rápido del mundo?" y "¿Cuál es el deportista más rápido del mundo?" tendrían un grado de similitud muy alto si las representáramos únicamente empleando n-gramas, ya que la mayoría de los términos entre una y otra se solapan. Más aún, si empleáramos un etiquetador morfológico o analizáramos el árbol sintáctico, el grado de similitud entre ambas preguntas sería pleno. Sin embargo, el tipo de respuesta esperado por cada pregunta es totalmente diferente, siendo un *animal* en el primer caso y una *persona* en el segundo. Por contra, una pregunta como "¿Qué actor ganó el Oscar en 1945?" presentaría escasa similitud con la segunda pregunta anterior pese a que su respuesta esperada es también *persona*. En estas circunstancias, ser capaz de diferenciar los significados de algunos de los términos de la pregunta, como "animal" y "deportista", o la equivalencia de otros como "deportista" y "actor", es fundamental para una correcta clasificación. Es en este punto donde resultan de utilidad las herramientas y recursos semánticos.

Las características semánticas son las más empleadas y las que mejor resultado han proporcionado en los estudios realizados en el campo de la CP. Trabajos como los de Li y Roth (2002, 2005) han sido fundamentales para que muchos de los desarrollos posteriores se hayan centrado en la incorporación de este tipo de características a los sistemas. En estos trabajos se demuestra que la información semántica puede mejorar notablemente la tarea de clasificación.

Uno de los recursos empleados en estos trabajos son las listas de palabras semánticamente relacionadas y agrupadas por clases (Kocik, 2004; Li y Roth, 2002, 2005). Estas listas creadas de forma manual relacionan cada clase de la taxonomía de preguntas con un conjunto de palabras relacionadas. Por ejemplo, la clase *food* se asocia con palabras como "*apple*", "*beer*", "*butter*", "*cereal*" o "*cook*". Puede verse un extracto de estas listas en la figura 3.7. La presencia de estas palabras en la pregunta es tomada como indicio de que la pregunta pertenece a la clase en cuya lista fueron incluidas. Li y Roth (2005) definen un método para obtener estas listas de forma automática basado en su similitud de distribución (Pantel y Lin, 2002a), obteniendo resultados ligeramente inferiores a los obtenidos con las listas manuales. Otro tipo de listas de palabras manuales que son habituales en CP son las listas de pronombres interrogativos (Metzler y Croft, 2005), empleados como información *a priori* para que el clasificador adopte distintas estrategias dependiendo del interrogativo que se dé en la pregunta. Greenwood (2005) emplea una lista de "palabras buenas" que hacen que determinados términos de la pregunta sean tenidos más en consideración por el clasificador que

animal	*mountain*	*food*
animal	highest	alcoholic
animals	hill	apple
Bear	hills	apples
bear	ledge	ate
bears	ledges	beer
beast	mesa	berry
beasts	mesas	berries
bird	mountains	breakfast
birds	peak	brew
breed	peaks	butter

Figura 3.7: Primeros 10 términos de la lista de palabras relacionadas para tres de las clases semánticas de Li y Roth: *animal*, *mountain* y *food*.

otros, premiando aquellas palabras que indican claramente sobre qué se nos está preguntando (como "ciudad", "año" o "presidente").

Otro tipo de información semántica utilizada habitualmente es el reconocimiento de entidades (*named entity recognition*). Los *reconocedores de entidades* permiten localizar en el texto nombres de personas, organizaciones, lugares, expresiones temporales, cantidades, etc. y sustituirlas por una etiqueta semántica. Por ejemplo, en las preguntas "¿Dónde está París?" y "¿Dónde está Sevilla?", detectar correctamente que los dos nombres de ciudad son entidades nos permitiría sustituir "París" y "Sevilla" por la etiqueta *LUGAR* y alcanzar una representación común para ambas ("¿Dónde está *LUGAR*?"). Este tipo de sustituciones permite homogeneizar la representación de las preguntas, aumentando la similitud entre ellas y facilitando así la tarea de clasificación. Hacioglu y Ward (2003) mejoran la representación básica mediante n-gramas utilizando un reconocedor de 7 tipos de entidades posibles (IdentiFinder), aunque empeoran sus resultados cuando intentan utilizar una versión más refinada del reconocedor sobre 29 tipos posibles. Achacan los resultados al peor rendimiento del etiquetador refinado. Brown (2004) llega a la misma conclusión que el trabajo anterior, mejorando la representación BOW empleando las entidades detectadas por IdentiFinder. En (Suzuki et al., 2003b) también obtienen ligeras mejoras sobre el modelo de n-gramas. Pinchak y Lin (2006) utiliza el etiquetador de entidades ANNIE (Maynard et al., 2002) para compararlo con una aproximación propia basada en el algoritmo de *clustering by comitee* (Pantel y Lin, 2002b) obteniendo mejores resultados con este último, aunque no compara la mejora obtenida con respecto a la representación con n-gramas. Li y Roth (2002, 2005) obtienen también una mejora significativa con respecto al modelo basado en BOW al emplear entidades. En el caso de Metzler y Croft (2005) el reconocimiento de entidades parece empeorar los

resultados del sistema. Lo justifican argumentando que las entidades que aparecen en la pregunta habitualmente no se corresponden con la clase semántica de la pregunta (que aparezca una entidad de tipo *persona* en la pregunta no implica que se nos esté preguntando por una persona). Estos resultados revelan un uso inadecuado de las entidades en el vector de características, ya que el resto de aproximaciones sí que consiguen mejorar el sistema básico. Otros sistemas que han empleado reconocimiento de entidades son los de Nguyen et al. (2008), García Cumbreras et al. (2006) y Blunsom et al. (2006), aunque no ofrecen resultados de las posibles mejoras de rendimiento proporcionadas por esta herramienta.

Las bases de datos léxicas como WordNet y EuroWordnet han sido utilizadas con asiduidad como recurso lingüístico en numerosas aproximaciones. Estas bases de datos agrupan las palabras en conjuntos denominados *synsets*, donde se almacenan las distintas relaciones semánticas que se dan entre palabras. Ejemplos de este tipo de relaciones son la sinonimia, hiperonimia, meronimia y homonimia. La relación más utilizada es la de hiperonimia, que permite utilizar un término general para referirse a la realidad nombrada por un término más particular. Por ejemplo, *persona* sería hiperónimo de *actor* y *escritor*. El uso de hiperónimos permite, al igual que hacía el etiquetado de entidades, homogeneizar la representación de las preguntas. Entre las aproximaciones que usan este tipo de característica de aprendizaje tenemos las de Krishnan et al. (2005), Li et al. (2005), García Cumbreras et al. (2006) y Kocik (2004). Metzler y Croft (2005) emplean WordNet para extraer los hiperónimos de los núcleos de los sintagmas detectados en la fase de análisis sintáctico. Schlobach et al. (2004) además de WordNet utilizan dos sistemas de información de nombres geográficos para mejorar la detección de tipos y la obtención de respuestas para preguntas sobre lugares: GEOnet Names Server[16] y Geographic Names Information System.[17] Blunsom et al. (2006) emplean además de WordNet un diccionario geográfico (*gazeteer*). Suzuki et al. (2003b) usan GoiTaikei, una versión japonesa de WordNet. Day et al. (2007) emplean un diccionario de sinónimos en chino (TongYiCi CiLin) y la base de conocimiento HowNet 2000 (Wai y Yongsheng, 2002) para derivar características semánticas para el aprendizaje.

Otra característica de aprendizaje empleada en diversas aproximaciones es el foco de la pregunta (*question focus*). Moldovan et al. (1999) introducen la idea de foco de la pregunta para la tarea de BR, describiéndolo como una palabra o secuencia de palabras que definen la pregunta y eliminan su ambigüedad. Por ejemplo, en la pregunta "¿Cuál es la ciudad más grande de Alemania?" el foco sería "la ciudad más grande". Skowron y Araki (2004a) emplean heurísticas para localizar el foco de la pregunta para su uso como característica de aprendizaje. García Cumbreras et al. (2006) y Kocik (2004)

[16] http://gnswww.nima.mil/geonames/GNS/index.jsp.
[17] http://geonames.usgs.gov/stategaz/index.html.

también emplean el foco como información adicional para enriquecer la representación de la pregunta. Blunsom et al. (2006) detectan también el foco usando para ello el analizador sintáctico C&C CCG (Clark y Curran, 2004) con un modelo especialmente creado para analizar preguntas. Un concepto similar al foco de la pregunta es el empleado por Krishnan et al. (2005) y Day et al. (2007), donde definen la idea de *informer span*. Éstos son subsecuencias cortas de palabras (de una a tres) de la pregunta que son indicativas de la clase de la pregunta. Por ejemplo, en la pregunta "¿Cuánto pesa un elefante adulto?" el término "pesa" sería el *informer span*. Realizan dos experimentos diferentes para demostrar la utilidad de esta característica para el aprendizaje: el primero etiquetando manualmente estos elementos y el segundo detectándolos automáticamente mediante heurísticas. Los resultados obtenidos revelan una importante mejora en el rendimiento del sistema al incluir este tipo de información.

3.4. Algoritmos de aprendizaje

El *algoritmo de aprendizaje* permite al sistema predecir la clase correspondiente a una nueva pregunta dada a partir del modelo aprendido de un corpus de entrenamiento. Son numerosos los algoritmos empleados en los sistemas de CP basados en corpus. En esta sección vamos a revisar los más utilizados en esta área: *máquinas de vectores de soporte*, *modelos de lenguaje*, *máxima entropía*, *árboles de decisión*, *arquitectura SNoW*, *k-nearest neighbors* y *naive Bayes*.

3.4.1. Máquinas de vectores de soporte

Las *máquinas de vectores de soporte* o *support vector machines* (SVM) han adquirido gran popularidad en los últimos tiempos entre la comunidad dedicada al aprendizaje automático, siendo utilizadas en numerosas aplicaciones de PLN (Cortes y Vapnik, 1995; Vapnik, 1995). Tomando como entrada dos conjuntos de muestras[18] en un espacio de n dimensiones, el objetivo de este algoritmo es encontrar un hiperplano óptimo (frontera) que maximiza el margen entre los dos conjuntos de datos. Para ello se extraen los ejemplos más cercanos a la frontera, a los que se conoce como *vectores de soporte* o *support vectors*. El hiperplano óptimo es aquel que maximiza el margen o distancia entre la frontera y dichos vectores de soporte. En general, cuanto mayor sea el margen de separación menor será el error de generalización del clasificador.

Cada ejemplo se representa como un vector de dimensión n (una lista de n números), siendo el objetivo separar dichos ejemplos con un hiperplano

[18] El algoritmo básico de SVM permite discriminar entre muestras pertenecientes a dos clases posibles. Es lo que se conoce como *clasificador binario*.

de dimensión $n-1$. Es lo que se conoce como *clasificador lineal*. Más formalmente, el corpus de entrenamiento se representa como un conjunto de pares instancia-clase $(\mathbf{x}_i, y_i)$ tal que $i = 1 \ldots m$, siendo m el número de muestras, $\mathbf{x}_i \in \mathbb{R}^n$ el vector de características y $y_i \in \{1, -1\}^m$ la etiqueta que indica si la muestra $\mathbf{x}_i$ pertenece o no a la clase y_i. SVM obtiene la solución al siguiente problema de optimización:

$$\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}\xi_i$$

siendo

$$y_i(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) \geq 1 - \xi_i$$
$$\xi_i \geq 0$$

donde la función $\langle \mathbf{w}, \mathbf{x}_i\rangle + b$ representa el hiperplano buscado, C es un parámetro de compromiso entre el error cometido ξ_i y el margen, y $\mathbf{w}$ es un vector de pesos.[19] Las variables ξ_i fueron introducidas para abordar problemas que no fueran linealmente separables, permitiendo cierto error de clasificación.

SVM fue diseñado originalmente para resolver problemas de clasificación binaria. Para abordar el problema de la clasificación en k clases (como es el caso de la CP) hay que transformar el problema de la clasificación multiclase en múltiples problemas de clasificación binaria (Allwein et al., 2001). Hay dos aproximaciones básicas en este sentido: *uno contra todos* (*one-against-all*), donde se entrenan k clasificadores y cada uno separa una clase del resto, y *uno contra uno* (*one-against-one*), donde se han de entrenar $\frac{k(k-1)}{2}$ clasificadores y cada uno discrimina entre dos de las clases. Es importante notar que la estrategia *uno contra uno*, al trabajar con menos muestras, tiene mayor libertad para encontrar una frontera que separe ambas clases. Respecto al coste de entrenamiento, es preferible el uso de *uno contra todos* puesto que sólo ha de entrenar k clasificadores.

Pese a que en su forma más básica SVM induce separadores lineales, si el conjunto no es linealmente separable puede extenderse el algoritmo mediante una transformación no lineal $\phi(\mathbf{x})$ a un nuevo espacio de características. La función ϕ permite transformar el espacio de características de entrada (*input space*) en un espacio de trabajo de mayor dimensionalidad (*transformed feature space*) donde intentar encontrar de nuevo el hiperplano óptimo. De esta forma se realiza una clasificación lineal en el nuevo espacio, que es equivalente a una clasificación no-lineal en el espacio original.

Las *funciones núcleo* (*kernel functions*, *funciones kernel* o simplemente *kernels*) son un tipo especial de función que permiten hacer la

[19] El *producto escalar* de dos vectores $\mathbf{a} = \{a_1, a_2, \ldots, a_n\}$ y $\mathbf{b} = \{b_1, b_2, \ldots, b_n\}$ se define como $\langle \mathbf{a}, \mathbf{b}\rangle = \sum_{i=1}^{n} a_i b_i = a_1 b_1 + a_2 b2 + \ldots + a_n b_n$.

Lineal

$$k(\mathbf{x}_i, \mathbf{x}_j) = \langle \mathbf{x}_i, \mathbf{x}_j \rangle$$

Polinómico

$$k(\mathbf{x}_i, \mathbf{x}_j) = (\gamma \langle \mathbf{x}_i, \mathbf{x}_j \rangle + r)^d, \gamma > 0$$

Gaussiano o radial basis function (RBF)

$$k(\mathbf{x}_i, \mathbf{x}_j) = exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2), \gamma > 0$$

Sigmoide

$$k(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\gamma \langle \mathbf{x}_i, \mathbf{x}_j \rangle + r)$$

Figura 3.8: Cuatro kernels habituales en tareas de clasificación. γ, r y d son los parámetros de los kernels.

transformación del espacio de características de forma implícita durante el entrenamiento, sin necesidad de calcular explícitamente la función ϕ (Schölkopf y Smola, 2001; Shawe-Taylor y Cristianini, 2004). Es lo que se conoce como *kernel trick*. Una vez que el hiperplano se ha creado, la función kernel se emplea para transformar los nuevos ejemplos al espacio de características para la clasificación. Formalmente, un kernel k es una función simétrica $k(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle = k(\mathbf{x}_j, \mathbf{x}_i)$ que puede ser interpretada como una medida de similitud entre dos vectores de características $\mathbf{x}_i$ y $\mathbf{x}_j$. En la figura 3.8 se pueden ver algunos de los kernels básicos más utilizados en tareas de clasificación. La selección del kernel apropiado es importante ya que es éste el que define el espacio de trabajo transformado donde se llevará a cabo el entrenamiento y la clasificación. En el capítulo 5 retomaremos con más detalle el estudio de los kernels y su utilización con SVM.

SVM ha sido aplicado con éxito a numeroros problemas de PLN, demostrando su buen funcionamiento en espacios de alta dimensionalidad, como el análisis sintáctico superficial (Kudo y Matsumoto, 2001) o la clasificación de documentos (Joachims, 1998; Rennie y Rifkin, 2001; Taira y Haruno, 1999). En esta última tarea, los resultados dados por Dumais et al. (1998) muestran el buen rendimiento de SVM con respecto a otros algoritmos, consolidando su reputación en las tareas de clasificación. Este algoritmo se ha convertido hoy en día en el más utilizado para la CP. Existen diversas implementaciones disponibles para la comunidad científica entre las

que podemos destacar SvmFu,[20] LIBSVM (Chang y Lin, 2001), SVMLight (Joachims, 1999) y Weka (Witten y Frank, 2005).

Hay numerosos sistemas de CP que han usado SVM en su forma más básica mediante el empleo del kernel lineal. Entre los sistemas que emplean esta aproximación podemos destacar los de Hacioglu y Ward (2003), Krishnan et al. (2005), Skowron y Araki (2004a), Nguyen et al. (2008), Day et al. (2007), Solorio et al. (2004), Bisbal et al. (2005b), Bisbal et al. (2005a) y Tomás et al. (2005).

En los últimos tiempos la tendencia seguida en CP ha sido experimentar con diferentes kernels para mejorar los resultados obtenidos con el kernel lineal. Li et al. (2005) emplean los cuatro kernels disponibles en la implementación de LIBSVM y los comparan entre sí: kernel lineal, polinómico, RBF y sigmoideo. El kernel RBF se emplea también en el trabajo desarrollado por Metzler y Croft (2005).

Otro kernel muy extendido en CP es el *tree kernel* (Collins y Duffy, 2001). Este kernel permite medir la similitud de dos árboles contando el número de ramas comunes. Zhang y Lee (2003) emplean este kernel para incorporar la información del árbol de análisis sintáctico de las preguntas al proceso de clasificación. Otra propuesta de este tipo es la desarrollada por Moschitti et al. (2007), donde definen una nueva estructura en forma de árbol basada en información sintáctica y semántica supericial codificada mediante estructuras predicativas (*Predicate-Argument Structures*). Su kernel permite explotar el poder de representación de dichas estructuras mediante un clasificador SVM. Pan et al. (2008) definen un *tree kernel* semántico que aprovecha distintas fuentes de información (relaciones de WordNet, listas manuales de palabras relacionadas y entidades) para incorporar información sobre la similitud semántica entre preguntas.

En el trabajo de Suzuki et al. (2003b) se define un kernel denominado *Hierarchical Directed Acyclic Graph* (HDAG) (Suzuki et al., 2003a). Este kernel está diseñado para manejar con facilidad estructuras lingüísticas en el texto, como los sintagmas y sus relaciones, empleándolas como características de aprendizaje sin necesidad de convertir dichas estructuras a un vector de características de forma explícita.

3.4.2. Modelos de lenguaje

La idea básica tras los *modelos de lenguaje* (*statistical language models*) es que cada fragmento de texto puede ser visto como si fuera generado a partir de un modelo (Jelinek, 1997). Si tenemos dos fragmentos de texto, podemos definir el grado de relevancia entre ellos como la probabilidad de que sean generados por el mismo modelo de lenguaje. Aunque fueron originalmente desarrollado para la tarea de reconocimiento del habla, esta

[20] http://five-percent-nation.mit.edu/SvmFu/.

técnica de aprendizaje ha sido ampliamente utilizada en RI (Ponte y Croft, 1998; Song y Croft, 1999). En esta tarea se construye un modelo de lenguaje para cada documento de forma que, dada una consulta, se pueda decidir cuándo un documento es relevante basándose en la probabilidad de que su modelo de lenguaje genere dicha consulta.

Suponiendo que la consulta Q se compone de m términos $w_1, w_2, \ldots, w_m$, y que podemos calcular la probabilidad de que Q se genere a partir del documento D como

$$P(Q|D) = P(w_1|D) * P(w_2|D, w_1) * \ldots * P(w_n|D, w_1, w_2, \ldots, w_{m-1}).$$

Para construir el modelo de lenguaje de un documento necesitamos estimar las probabilidades de estos términos. Habitualmente se asumen modelos de *n-gramas* para simplificar la estimación:

$$P(w_i|D, w_1, w_2, \ldots, w_{i-1}) = P(w_i|D, w_{i-(n-1)}, w_{i-(n-2)}, \ldots, w_{i-1}),$$

donde la probabilidad de que w_i ocurra en el documento D dependerá únicamente de los $n-1$ términos previos (Manning y Schütze, 1999).

Estimar la probabilidad de los n-gramas puede ser difícil en un documento, ya que los sintagmas u oraciones pueden tener una longitud arbitraria y algunas secuencias de n-gramas no son observadas durante el entrenamiento del modelo de lenguaje. Es lo que se conoce como el problema de la dispersión de datos o *data sparseness*. Para evitar este problema, se usan habitualmente técnicas de suavizado. Algunos de los métodos de suavizado empleados habitualmente son *Good-Turing*, *Jelinek–Mercer*, *Dirichlet Priors* y *Absolute Discounting* (Zhai y Lafferty, 2004).

Los modelos de lenguaje se han aplicado también a la tarea de CP de manera similar a como se han utilizado en la tarea de RI. En primer lugar se construye un modelo de lenguaje para las preguntas de entrenamiento pertenecientes a cada clase c. Cuando llega una nueva pregunta q, se calcula la probabilidad $P(q|c)$ para cada clase c y se elige la que obtenga mayor probabilidad. Una importante ventaja de este marco de trabajo es su capacidad para modelar no sólo documentos sino también consultas a través de modelos de lenguaje estadísticos. Esto hace posible establecer los parámetros de recuperación de forma automática y mejorar el funcionamiento de la recuperación mediante el uso de métodos de estimación estadísticos. Algunas de las herramientas existentes para el modelado de lenguaje son SRILM (Stolcke, 2002), CMU-Cambridge Statistical Language Modeling Toolkit (Clarkson y Rosenfeld, 1997) y Lemur Toolkit.[21]

Entre los sistemas de CP que emplean esta aproximación encontramos el de Li (2002). En este trabajo se emplea la aproximación basada en

[21] http://www.lemurproject.org.

modelos de lenguaje para complementar una aproximación basada en reglas manuales, obteniendo mejores resultados que el sistema puramente basado en reglas. Otra aproximación es la seguida por Jeon et al. (2005) para clasificar preguntas empleando diferentes medidas de similitud entre respuestas utilizadas en el campo de la RI. Computan la similitud mediante *tf-idf* y modelos de lenguaje, obteniendo mejores resultados con estos últimos. En el trabajo de Murdock y Croft (2002) desarrollan una aproximación especial empleando modelos de lenguaje, entre otras técnicas, para distinguir entre preguntas sobre hechos ("¿Cuánto se tarda en conseguir el pasaporte?") y preguntas sobre procedimientos ("¿Cómo puedo conseguir el pasaporte?"). Pinto et al. (2002) desarrollan un sistema completo de BR especializado en extraer información contenida en tablas de páginas Web, creando modelos de lenguaje para cada uno de los tipos de pregunta. Otra aproximación basada en modelos de lenguaje es la desarrollada por Brown (2004), obteniendo mejores resultados que SVM para los experimentos que plantea. Otro trabajo que emplea esta técnica es el desarrollado por Tomás y Vicedo (2007a). En esta ocasión los modelos de lenguaje son empleados para afrontar la tarea de clasificar preguntas sobre múltiples taxonomías, tal y como se describió en la sección 2.6.2.

3.4.3. Máxima entropía

El modelado con *máxima entropía* (ME) permite integrar información de diversas fuentes heterogéneas para tareas de clasificación (Berger et al., 1996). Un clasificador basado en ME consta de un conjunto de parámetros o coeficientes obtenidos por un proceso de optimización. Cada coeficiente se asocia a una característica observada en el conjunto de entrenamiento. El principal propósito es obtener la distribución de probabilidad que maximiza la entropía, asumiendo la máxima ignorancia sin tener en cuenta nada más allá del corpus de entrenamiento.

Suponiendo un conjunto de contextos X y un conjunto de clases C, la función $cl : X \rightarrow C$ elige la clase c con mayor probabilidad condicional en el contexto $x : cl(x) = \arg\max_c p(c|x)$. Cada atributo se calcula mediante una función asociada a una clase particular c' que tiene la forma

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

donde $cp(x)$ es una característica observable en el contexto. La probabilidad condicional $p(c|x)$ se define como

$$p(c|x) = \frac{1}{Z(x)} \prod_{i=1}^{K} \alpha_i^{f_i(x,c)},$$

donde α_i es el peso del atributo i, K el número de atributos definidos y $Z(x)$ una constante para asegurar que la suma de todas las probabilidades condicionales para este contexto es igual a 1.

Los primeros trabajos con este modelo se desarrollaron en el área del reconocimiento del habla, aunque posteriormente se ha extendido su uso a otras tareas de PLN como el análisis morfológico, el etiquetado morfosintáctico, la desambiguación de sintagmas preposicionales, la segmentación de oraciones, el análisis sintáctico superficial y general, la clasificación de documentos y la traducción automática. Ratnaparkhi (1998) proporciona una buena introducción a los métodos de ME y una revisión de las diversas aplicaciones existentes. Entre las implementaciones realizadas de ME tenemos la proporcionada en el paquete de software Weka.

La tarea de CP no es una excepción, existiendo diversas aproximaciones que han utilizado este algoritmo. Un ejemplo es el trabajo desarrollado por Bisbal et al. (2005a), en el que emplean una implementación propia del algoritmo *MaxEnt* (Suárez y Palomar, 2002) utilizando como procedimiento de estimación de coeficientes GIS (*Generalized Iterative Scaling*) sin aplicar suavizado ni selección de atributos. Ittycheriah et al. (2000, 2001) usan ME para un sistema completo de BR, empleando una mezcla de características de aprendizaje sintácticas y semánticas. Blunsom et al. (2006) obtienen muy buenos resultados con este clasificador, aprendiendo a partir de información léxica y sintáctica obtenida de un analizador sintáctico especialmente entrenado para etiquetar preguntas. El trabajo de Kocik (2004) muestra un estudio completo y comparativo entre naive Bayes y ME, mostrando las ventajas de este último.

3.4.4. Árboles de decisión

Los métodos de aprendizaje supervisado basados en *árboles de decisión* son uno de los métodos más populares dentro del área de la IA para tratar el problema de la clasificación (Quinlan, 1986). Estos algoritmos permiten la extracción y representación de reglas de clasificación obtenidas a partir de un conjunto de ejemplos. Estos árboles pueden ser representados como un conjunto de reglas de tipo *si-entonces* para hacerlas más fácilmente legibles por humanos. En cada nodo del árbol se realiza la evaluación de alguna de las características de la instancia, y cada rama descendiente de ese nodo se corresponde a uno de los posibles valores para esa característica. Una nueva instancia se clasifica comenzando desde el nodo raíz del árbol, evaluando la característica especificada por ese nodo y descendiendo por la rama correspondiente al valor de la característica presente en la instancia dada. Este proceso se repite para el nuevo nodo alcanzado hasta llegar a las hojas, donde finalmente se proporciona la clase asignada a la instancia.

Entre los algoritmos más conocidos pertenecientes a esta familia tenemos ID3, C4.5 (una extensión de ID3) y C5.0 (una versión extendida de C4.5

de carácter comercial). En cuanto a implementaciones, en Weka podemos encontrar los algoritmos ID3 y C4.5. También está libremente disponible la propia implementación realizada por Quinlan del algoritmo C4.5.[22]

El uso de los árboles de decisión se ha extendido por diversas disciplinas como la estadística, la ingeniería (reconocimiento de formas), la teoría de la decisión (programación de tablas de decisión) y el procesamiento de la señal. Su aplicación a problemas de PLN se ha extendido a prácticamente todos los niveles de tratamiento del lenguaje, incluyendo el análisis sintáctico (Magerman, 1995), la desambiguación semántica (Pedersen, 2001) y la clasificación de documentos (Yang, 1999).

En la tarea de CP también se han aplicado estos algoritmos principalmente en su variante C4.5. Es una creencia bastante extendida que para aplicaciones de clasificación de textos, los métodos de aprendizaje basados en árboles de decisión no resultan apropiados, ya que se basan en probar un pequeño conjunto de características únicamente (se evalúa una única característica en cada nodo del árbol). Cheung et al. (2004) usan este algoritmo para generar árboles de decisión, obteniendo mejores resultados con esta aproximación que con el algoritmo naive Bayes. Sundblad (2007) y Zhang y Lee (2003) utilizan también este algoritmo para compararlo con otros como SVM, obteniendo peores resultados que éste.

3.4.5. Arquitectura SNoW

La *arquitectura de aprendizaje SNoW* (*Sparse Network of Winnows*) consiste en una red de baja conectividad de separadores lineales sobre un espacio de características predefinido o aprendido de forma incremental (*online*) (Carlson et al., 1999). SNoW es una arquitectura de aprendizaje multiclase, donde cada una de las clases se representa como un único nodo objetivo, aprendido como una función lineal sobre el espacio de características de aprendizaje o como una combinación de varias de ellas. La arquitectura tiene forma de red neuronal donde diferentes algoritmos como winnow, perceptron o naive Bayes, de los que hereda sus capacidades de generalización, se pueden integrar en los nodos.

La variante más utilizada es aquella que emplea un nodo con el algoritmo de winnow para cada clase, donde cada uno de estos nodos aprende a discriminar entre una clase particular y el resto (Màrquez, 2001). Winnow es un algoritmo incremental para problemas de clasificación binarios que aprende un conjunto de pesos para las características del espacio de entrada, indicando el grado de importancia de éstos. Para clasificar un nuevo ejemplo, el algoritmo realiza una combinación lineal del conjunto de características (típicamente una suma ponderada de los pesos asociados a las características que aparecen en el ejemplo a clasificar), asignando la clase positiva si el

[22] http://www.rulequest.com/Personal/c4.5r8.tar.gz.

resultado supera un determinado umbral de confianza o la clase negativa en caso contrario.

Cuando SNoW clasifica una nueva entrada se comporta de manera similar a una red neuronal, tomando las activaciones de las características de entrada y determinando como clase de salida aquella que corresponde al nodo de mayor activación. Las características de entrada pueden tomar valores binarios, indicando que la característica está activa, o valores reales reflejando su importancia. Los nodos objetivo están enlazados a las características de entrada que son relevantes para su clase a través de arcos ponderados. Esta relevancia y su peso se establecen dinámicamente durante el aprendizaje por la propia arquitectura SNoW.

Sea $A_t = i_1, \ldots, i_m$ el conjunto de características que están activas en un ejemplo y que están enlazadas con el nodo objetivo t, el nodo está activo si y sólo si

$$\sum_{i \in A_t} w_i^t > \theta_t,$$

donde w_i^t es el peso del arco que conecta la i-ésima característica al nodo objetivo t, siendo θ_t su umbral. Las decisiones tomadas por SNoW pueden ser tanto binarias, indicando cuál de las clases se predice para un ejemplo dado, como continuas, indicando un grado de confianza en la predicción.

SNoW ha demostrado su buen funcionamiento en dominios de elevada dimensionalidad dentro del PLN, como la clasificación de documentos (Zhang, 2000), la corrección ortográfica sensible al contexto (Golding y Roth, 1999), el procesamiento visual (Yang et al., 2000) y la clasificación de opiniones (Nigam y Hurst, 2004) y de sentimientos (Cui et al., 2006). La implementación de la arquitectura SNoW está libremente disponible para fines investigadores.[23]

La CP es otra de las áreas donde este algoritmo ha conseguido popularidad en los últimos años, debido en gran medida al trabajo desarrollado por Li y Roth (2002). En este trabajo emplean la arquitectura SNoW para desarrollar un clasificador jerárquico capaz de etiquetar preguntas sobre una taxonomía de clases de dos niveles. El trabajo desarrollado por Li y Roth (2005) es una expansión del trabajo anterior, aumentando el corpus de experimentación y ofreciendo un estudio exhaustivo de la aportación de diferentes recursos semánticos a la clasificación. Zhang y Lee (2003), siguiendo los experimentos anteriores, ofrecen una comparativa de SNoW con otros algoritmos: SVM, naive Bayes, k-nearest neighbors y árboles de decisión. Empleando una aproximación basada en BOW y n-gramas, obtiene mejores resultados con SVM y árboles de decisión que con SNoW.

[23] http://l2r.cs.uiuc.edu/~cogcomp/download.php?key=SNOW.

3.4.6. k-nearest neighbors

El *aprendizaje basado en ejemplos* o *memory-based learning*[24] es un tipo de aprendizaje supervisado, basado en la hipótesis de que las tareas cognitivas se llevan a cabo contrastando la similitud entre las situaciones nuevas y las situaciones pasadas almacenadas. En el proceso de aprendizaje se memorizan todos los ejemplos en su forma original, sin necesidad de intentar generalizar ninguna regla ni representación más concisa. En el proceso de clasificación de nuevas instancias se obtiene de la memoria de ejemplos el conjunto de ellos más parecido al que se está intentando clasificar, asignándole la categoría mayoritaria encontrada.

El método más básico de aprendizaje basado en ejemplos es *k-nearest neighbors* (k-NN). Este algoritmo es una conocida aproximación estadística que ha sido estudiada de forma amplia en reconocimiento de formas durante más de cuatro décadas (Dasarathy, 1990) y que ha sido aplicado a tareas como la clasificación de textos desde sus orígenes (Sebastiani, 2002).

El algoritmo k-NN es sencillo: dada una instancia a clasificar, el sistema encuentra los k vecinos más cercanos entre las instancias de entrenamiento, usando las categorías de estos k vecinos para dar peso a las categorías candidatas. El valor de similitud de cada instancia vecina a la nueva instancia a clasificar se asigna como peso a la categoría de dicha instancia vecina. Si varios de los k vecinos más cercanos comparten categoría, entonces los pesos de esta categoría se acumulan y el peso resultante se emplea como el valor de probabilidad de esa categoría con respecto a la nueva instancia. Ordenando las puntuaciones de las categorías candidatas se obtiene una lista ordenada para cada nueva instancia. Estableciendo un umbral sobre estos marcadores se obtienen asignaciones binarias. La regla de decisión de k-NN se define como

$$y(\vec{x}, c_j) = \sum_{\vec{d} \in k-NN} sim(\vec{x}, \vec{d_i}) y(\vec{d_i}, c_j) - b_j,$$

donde $y(\vec{d_i}, c_j) \in \{0, 1\}$ es la clasificación para la instancia $\vec{d_i}$ con respecto a la categoría c_j (1=sí y 0=no), $sim(\vec{x}, \vec{d_i})$ es la similitud entre la nueva instancia $\vec{x}$ y la instancia de entrenamiento $\vec{d_i}$, y b_j es el umbral específico para las decisiones binarias. Existen diversas funciones para medir la similitud entre dos instancias, siendo la medida del coseno entre los vectores una de las más utilizadas (Yang y Liu, 1999). Entre las implementaciones que existen de este algoritmo tenemos el programa TiMBL (*Tilburg Memory-based Learning Environment*) (Daelemans et al., 2007) y el paquete Weka.

[24] Esta familia de algoritmos recibe otras denominaciones en diferentes áreas de la IA como *lazy learning*, *instance-based learning*, *similarity-based learning*, *case-based learning* o *exemplar-based learning*.

Aparte de la ya citada tarea de clasificación de textos, el uso de este algoritmo se extiende a otras muchas aplicaciones de PLN, como el análisis sintáctico superficial y general (Argamon et al., 1998) y la desambiguación semántica (Ng y Lee, 1996).

En CP también se ha empleado este algoritmo, principalmente para realizar comparativas con otras aproximaciones. Es el caso de los trabajos realizados por Zhang y Lee (2003), Bisbal et al. (2005a) y Sundblad (2007). En estos experimentos demuestra un rendimiento inferior a otros algoritmos como SVM.

3.4.7. Naive Bayes

Naive Bayes (NB) es un clasificador estocástico usado de forma habitual en aprendizaje automático (Mitchell, 1997). Este algoritmo fue descrito originalmente por Duda y Hart (1973). La idea básica en las aproximaciones basadas en NB es usar las distribuciones conjuntas de las palabras y categorías para estimar las probabilidades de las categorías dada una instancia.

La parte *naive* o "ingenua" de NB es la asunción de independencia de las palabras, es decir, la probabilidad condicional de una palabra dada una clase se asume que es independiente de la probabilidad condicional de otras palabras dada esa clase. Esta asunción hace la computación de un clasificador NB mucho más eficiente que la complejidad exponencial de los clasificadores *bayesianos no-naive*, ya que no usa combinaciones de palabras como predictores.

De manera más formal, sea $\{1 \dots K\}$ el conjunto de clases posibles y $\{x_{i,1}, \dots, x_{i,m}\}$ el conjunto de valores de las características del ejemplo x_i, el algoritmo NB selecciona la clase que maximiza $P(k|x_{i,1}, \dots, x_{i,m})$:

$$\arg\max_x P(k|x_{i,1}, \dots, x_{i,m}) \approx \arg\max_x P(k) \prod_j P(x_{i,j}|k),$$

donde $P(k)$ y $P(x_{i,j}|K)$ son probabilidades que se estiman a partir del corpus de aprendizaje mediante las frecuencias relativas.

Este algoritmo ha sido usado ampliamente por la comunidad de PLN. A pesar de su simplicidad ha obtenido resultados destacables en numerosas tareas, entre las que podemos citar el etiquetado morfosintáctico (Peshkin et al., 2003), la desambiguación semántica (Escudero et al., 2000) y la clasificación de textos (Mccallum y Nigam, 1998).

Una de las debilidades conocidas de NB es el problema para trabajar en espacios de gran dimensionalidad, es decir, con un gran número de características de aprendizaje. Los trabajos que emplean este algoritmo en la tarea de CP se centran en utilizarlo como algoritmo de referencia (*baseline*), empleándolo para su comparación con otros algoritmos que ofrecen mejor rendimiento en esta tarea. Kocik (2004) hace una comparativa de NB frente

a otra aproximación basada en ME, tomando el primero como algoritmo de referencia y demostrando la superioridad de ME. Cheung et al. (2004) emplean NB como referencia para compararlo con otro algoritmo basado en árboles de decisión, obteniendo mejores resultados con este último. Zhang y Lee (2003) lo emplean para comparar sus resultados con otros algoritmos como k-NN, SNoW y SVM, obteniendo nuevamente peores resultados con NB.

3.4.8. Otros algoritmos

Aunque los algoritmos citados anteriormente son los de uso más común, no son los únicos empleados en los desarrollos dentro del campo de la CP.

Huang et al. (2007) emplean un algoritmo *pasivo-agresivo* (PA) (Crammer et al., 2006), una familia de algoritmos incrementales para clasificación binaria. Este algoritmo trabaja de forma similar a SVM, y puede ser visto como una variante incremental de éste. El algoritmo PA intenta encontrar un hiperplano que separe las instancias de entrenamiento en dos. El margen de un ejemplo es proporcional a la distancia del ejemplo al hiperplano. Cuando comete errores en la predicción de los ejemplos, el algoritmo PA utiliza el margen para modificar el actual clasificador. La ventaja de este algoritmo con respecto a SVM es que, al ser incremental, actualiza el clasificador tras ver el siguiente ejemplo y asegura que la actualización no empeorará el rendimiento del sistema. Realiza una comparación de este algoritmo con SVM y SNoW, obteniendo mejores resultados con su aproximación basada en el algoritmo PA.

Otra aproximación especial es la seguida por Radev et al. (2002) induciendo reglas de decisión usando una versión extendida de RIPPER (Cohen, 1996) y combinándolas con reglas manuales. Emplean esta técnica para desarrollar un sistema completo de BR en la Web.

Prager et al. (2002) emplean un algoritmo propio que trata de determinar la clase semántica del foco de la pregunta, basado en la coocurrencia del foco en cada clase. Cada clase recibe un peso que es comparado con el peso esperado para esa clase. Cuanto más se desvía positivamente este peso de su valor esperado, mayor es el grado de asociación positiva entre el foco y la clase, y mayor también su consideración como clase candidata.

Por último, Greenwood (2005) realiza una aproximación basada en RI. Usa un motor de RI para indexar un gran conjunto de preguntas de ejemplo previamente etiquetadas con la clase semántica correcta. Dada una nueva pregunta, que en este caso haría la función de consulta del sistema de RI, devuelve la pregunta o preguntas etiquetadas más similares, que harían la función de documentos indexados. A la nueva pregunta se le asigna el tipo o tipos de las preguntas más similares devueltas por el sistema de RI. Esta aproximación a la tarea de CP es, de hecho, una extensión del algoritmo k-NN. Normalmente, en k-NN los vecinos más cercanos a una nueva instancia

se definen en función de la *distancia euclídea* estándar. En la aproximación realizada en este trabajo, los vecinos más cercanos se definen en función de la similitud asignada por el motor de RI.

3.5. Aproximaciones especiales

Vamos a citar en este punto algunos sistemas de CP que han aportado un enfoque diferente a la tarea o se han centrado en algún aspecto particular de ésta. Por ejemplo, Moschitti y Harabagiu (2004) buscan una forma de integrar la tarea de BR con la clasificación de textos. Para ello desarrollan un sistema de CP que en lugar de asociar una pregunta con un tipo de respuesta esperada, asocia a cada pregunta un tipo de documento esperado. Emplean un conjunto de preguntas de entrenamiento relacionadas con cinco categorías del corpus Reuters-21578, empleado como referencia habitual para evaluar sistemas de clasificación de textos. Esta aproximación les permite clasificar la pregunta en estas categorías, filtrando así todas las respuestas que ocurran en documentos que no pertenezcan a la categoría detectada.

Pinchak y Bergsma (2007) realizan una aproximación centrada en preguntas sobre cantidades. Usando la Web como fuente de información obtienen de forma automática unidades de medida apropiadas para cada pregunta. Por ejemplo, para la pregunta "¿Cuánto pesa un oso Grizzly?" detectan que la respuesta esperada debe ir acompañada de términos como "kilogramos", "libras" o "toneladas". La intención es mejorar el rendimiento de los sistemas de BR para este tipo de preguntas cuantificables, descartando como respuestas posibles aquellas cantidades numéricas que no vayan acompañadas de las unidades de medida esperadas.

Verberne (2006) realiza una aproximación a BR basada en la estructura del discurso y centrada en la resolución de preguntas clarificadoras de tipo "¿Por qué?". En el módulo de CP definen cuatro clases posibles para este tipo de preguntas: *cause*, *motivation*, *circumnstance* y *purpose*. Hacen una aproximación basada en características sintácticas para la clasificación.

Grivolla et al. (2005) presentan un método para predecir de forma automática el grado de relevancia que tiene un conjunto de documentos devueltos por un sistema de RI en respuesta a una consulta. Esta predicción se afronta desde el punto de vista de la BR. Para ello desarrollan un sistema de CP capaz de discriminar entre preguntas *fáciles* y *difíciles*, asignadas en función de la precisión media obtenida por el sistema de BR sobre la colección de preguntas. Experimentan con diferentes características y algoritmos de aprendizaje, como SVM y los árboles de decisión.

Tamura et al. (2005) afrontan la tarea de clasificación de preguntas formadas por múltiples oraciones pero con una única respuesta. Preguntas como "El icono para volver al escritorio ha desaparecido. ¿Cómo puedo recuperarlo?" son muy habituales en la Web. Este trabajo presenta un

componente de extracción para obtener la oración que contiene la pregunta a partir del conjunto de oraciones dado ("¿Cómo puedo recuperarlo?"), descartando aquellas que no son relevantes a la hora de detectar el tipo esperado de respuesta ("El icono para volver al escritorio ha desaparecido."). Realizan experimentos con preguntas en japonés obtenidas de usuarios reales de la Web.

Pinchak y Lin (2006) presentan un método no supervisado para construir dinámicamente un modelo de tipos de respuesta probabilístico para cada pregunta. De esta forma evitan tener una taxonomía predefinida de clases de pregunta. Para afrontar la tarea de BR obtienen contextos candidatos (relaciones de dependencia entre palabras de una oración) del corpus de documentos y del corpus de preguntas. Calculan la probabilidad de que los contextos de la repuesta coincidan con los de la pregunta. En este trabajo, el sistema de CP se emplea para filtrar los documentos devueltos por el sistema de RI.

Por último, García Cumbreras et al. (2006) abordan el tema de la clasificación en diferentes idiomas. Clasifican preguntas en español con un sistema de CP entrenado sobre un corpus de preguntas en inglés. Para ello traducen al inglés las preguntas en español usando un traductor automático y aplicando luego el sistema de CP en inglés. Prueban diferentes combinaciones de características y sistemas de traducción en la Web.

3.6. Conclusiones

En este capítulo se ha hecho un repaso a los distintos componentes que intervienen en la tarea de CP basada en aprendizaje automático, ofreciendo paralelamente una revisión del estado de los sistemas actuales con respecto a cada uno de estos componentes.

En primer lugar hemos revisado distintas taxonomías empleadas en los sistemas de CP, clasificándolas en función de su tamaño, estructura y cobertura. Veíamos como la estructura (plana o jerárquica) de la taxonomía estaba íntimamente ligada con el número de clases posibles. Diseñar una taxonomía jerárquica quedaba justificado por la escalabilidad y organización de las clases que proporciona. Sin embargo, la mejora de rendimiento que teóricamente pudiera tener desarrollar un clasificador jerárquico sobre estas taxonomías no se ve justificada en la actualidad desde el punto de vista empírico. Por otra parte discutíamos dos puntos de vista a la hora de definir los tipos de preguntas: que fueran lo suficientemente amplios para cubrir todas las preguntas que se pudieran dar en el dominio, o que fueran los apropiados para que un determinado sistema de BR fuera capaz de detectar y extraer respuestas de esos tipos en el texto. El objetivo de este trabajo de tesis es el desarrollo y evaluación de sistemas de CP. Estos desarrollos deben realizarse con independencia de cualquier sistema de BR para que

la evaluación no se vea afectada por otros aspectos de esta tarea. Por esta razón, las taxonomías con las que trabajaremos no estarán limitadas por las capacidades de los sistemas de BR para responder a determinados tipos de preguntas, tratando de cubrir el mayor rango posible de consultas que se puedan dar en el dominio de trabajo.

Después de las taxonomías hemos fijado nuestra atención en los corpus de preguntas empleados en esta tarea. Los hemos caracterizado por su tamaño, el dominio de aplicación y el idioma en el que se hayan las preguntas. Veíamos como las conferencias TREC, CLEF y NTCIR habían influido decisivamente en el tipo de preguntas a las que se enfrentan los sistemas de BR, y por consiguiente, los sistemas de CP contenidos en ellos: preguntas factuales en dominio abierto. Para el buen rendimiento de los sistemas de clasificación, estos corpus han de tener el suficiente número de muestras para evitar el problema de las clases sesgadas y del sobreajuste del clasificador durante el entrenamiento. En el capítulo 6 vamos a abordar el problema de la CP en ausencia de corpus. Propondremos un algoritmo mínimamente supervisado para la clasificación sobre taxonomías refinadas sin necesidad de muestras de entrenamiento.

Revisando el estado de la cuestión se aprecia que son muchos y muy diferentes los corpus y taxonomías empleados en la tarea de CP, concluyendo que no existe un estándar para el entrenamiento y evaluación de estos sistemas. No obstante, gracias a sus características intrínsecas y a su libre disponibilidad, el corpus UIUC (descrito en la sección 3.2) ha sido empleado en numerosos trabajos investigadores en esta área. En el capítulo 5 lo emplearemos para comparar nuestra aproximación con otros desarrollos actuales. Por lo que respecta al idioma, veíamos que la gran mayoría de estudios realizados se centraban en el idioma inglés, siendo muy escasos los trabajos realizados y los corpus disponibles en otros idiomas. Más aún, el tema del multilingüismo apenas ha sido tratado en estos sistemas. En el capítulo 4 desarrollaremos nuestros propios corpus de preguntas en distintos idiomas y dominios para poder entrenar y evaluar adecuadamente nuestro trabajo.

Después de revisar los corpus hemos hecho un repaso de las diferentes características de aprendizaje que utilizan los sistemas de CP. Muchos de ellos emplean una representación inicial mediante n-gramas que luego complementan con características léxicas, sintácticas y semánticas. De entre todas ellas, la información semántica demostraba ser especialmente útil para mejorar el rendimiento de los clasificadores. El problema de incorporar información a partir de análisis lingüístico es la necesidad de recursos y herramientas para su obtención. Por ejemplo, incorporar al vector de características de la pregunta información sobre la categoría gramatical de sus términos, palabras sinónimas o las entidades que contiene, implica tener disponibles etiquetadores morfológicos, bases de datos léxicas y reconocedores de entidades. Estas herramientas no siempre son fáciles de

encontrar en todos los idiomas y su uso limita la portabilidad a otros idiomas y dominios. Otro problema que plantea el uso de algunas herramientas lingüísticas es el coste computacional del proceso asociado. A la hora de entrenar un clasificador, el tiempo requerido para formalizar el vector de características puede no ser importante. Sin embargo, a la hora de clasificar una nueva pregunta en un sistema de CP/BR en tiempo real, el compás de espera para el usuario debe ser mínimo. Éste es otro argumento a tener en cuenta a la hora de decidir qué tipo de características de aprendizaje se van a emplear en el clasificador. En el capítulo 4 vamos a plantear un primer desarrollo basado en n-gramas para su aplicación sobre diferentes idiomas y dominios. Emplearemos diversos métodos estadísticos para modificar el espacio de características y mejorar el rendimiento del clasificador. Los capítulos 5 y 6 incidirán en la incorporación de información semántica de manera no supervisada o mínimamente supervisada para enriquecer el modelo de n-gramas, evitando en todo momento el uso de herramientas lingüísticas que limiten la capacidad de adaptación del sistema de clasificación.

Por último se han descrito los algoritmos de aprendizaje más comúnmente utilizados en la tarea de CP: SVM, modelos de lenguaje, ME, árboles de decisión, arquitectura SNoW, k-NN y NB. De entre estos algoritmos destaca SVM por ser el más utilizado y el que mejor rendimiento ha proporcionado en diversos estudios comparativos (Bisbal et al., 2005a; Sundblad, 2007; Zhang y Lee, 2003). La posibilidad de emplear diferentes kernels en SVM ha centrado la atención de los últimos avances realizados en el campo de la CP. Trabajos como los de Suzuki et al. (2003b), Zhang y Lee (2003), Moschitti et al. (2007) y Pang y Lee (2008) han demostrado la utilidad de estos kernels en la clasificación. En el capítulo 5 vamos a definir nuestro propio kernel para la incorporación de información semántica al clasificador SVM de manera no supervisada.

4

Clasificación de preguntas supervisada basada en n-gramas

En esta primera aproximación proponemos un sistema de CP basado exclusivamente en información obtenida directamente del corpus de entrenamiento. Para ello emplearemos como características de aprendizaje diferentes combinaciones de n-gramas obtenidas de las preguntas del corpus, de forma que no sea necesario el uso de herramientas o recursos lingüísticos adicionales. Esto nos va a permitir obtener de forma directa un sistema de CP fácilmente adaptable a diferentes idiomas y dominios, ya que únicamente depende del corpus sobre el que se entrena. Tal y como comentábamos en el capítulo anterior, los corpus de preguntas en idiomas diferentes al inglés son escasos y de reducido tamaño. Para poder evaluar esta primera aproximación sobre diferentes idiomas hemos desarrollado un corpus paralelo de preguntas en inglés, español, italiano y catalán. Estas preguntas pertenecen al dominio abierto. Para evaluar el rendimiento sobre diferentes dominios, hemos desarrollado un corpus propio de preguntas formuladas por usuarios reales en un dominio restringido (el dominio turístico).

En este capítulo vamos a experimentar con diferentes combinaciones de n-gramas para obtener una configuración que nos proporcione el mejor rendimiento posible dado los escasos recursos puestos en juego. A fin de mejorar este sistema básico, hemos empleado distintas técnicas estadísticas de selección de características que permiten determinar los atributos de aprendizaje más relevantes para el entrenamiento del clasificador. Estas técnicas están basadas únicamente en estimaciones estadísticas obtenidas a partir del corpus de entrenamiento, no afectando a la adaptabilidad del sistema. Aunque la selección de características ha sido ampliamente utilizada en diversas tareas de PLN, no existen estudios previos sobre su aplicación a la tarea de CP.

De manera adicional, este capítulo nos servirá para describir los procedimientos y medidas habituales de evaluación en el campo de los sistemas de CP.

4.1. Corpus y taxonomías

Para entrenar el sistema y evaluar su adaptabilidad y su robustez vamos a desarrollar dos corpus diferentes de preguntas. El primero de ellos, que hemos denominado *corpus TREC*, nos permitirá evaluar el sistema sobre diferentes idiomas. El segundo, denominado *corpus QALL-ME*, nos permitirá contrastar el rendimiento del sistema sobre diferentes dominios. Ambos corpus presentan marcadas diferencias en cuanto a tamaño, origen, taxonomía y estilo subyacente.

4.1.1. Corpus TREC

En el capítulo anterior se describieron distintos corpus desarrollados para la tarea de CP. Aunque algunos de ellos estaban disponibles en diferentes idiomas, como DISEQuA o Multieight-04, su reducido tamaño (450 y 700 preguntas respectivamente) los hace poco apropiados para su utilización en sistemas basados en corpus. Por ello hemos desarrollado nuestro propio corpus de entrenamiento y evaluación para diferentes idiomas. De ahora en adelante nos referiremos a este conjunto de datos como *corpus TREC*.

El primer paso para formalizar este corpus fue la recopilación de las preguntas que lo componen. Partimos de las preguntas de evaluación en inglés definidas para la tarea BR de las conferencias TREC, desde el año 1999 (TREC-8) hasta el 2003 (TREC-12).[1] Fue precisamente en el TREC-8 donde se abordó por primera vez la tarea de BR (Voorhees, 1999). Las 200 preguntas que se facilitaron a los participantes fueron obtenidas de los registros de FAQ Finder (Burke et al., 1997) y de diversos asesores y participantes. En el TREC-9 se proporcionaron 500 preguntas obtenidas a partir de los registros de Encarta[2] y Excite.[3] A este conjunto inicial se añadieron 193 preguntas más, resultado de la reformulación de 54 de las preguntas originales. Se buscaba de esta manera comprobar la robustez de los sistemas de BR a la variación del lenguaje. Un ejemplo de pregunta y sus variantes la podemos ver en la figura 4.1.

En el TREC-10 (Voorhees, 2001a) se propusieron 500 preguntas filtrando los registros de MSNSearch[4] y AskJeeves.[5] En el TREC-11 (Voorhees, 2002) se proporcionaron otras 500 preguntas, nuevamente a partir de los registros de MSNSearch y AskJeeves. En el TREC-12 (Voorhees, 2003) se dividió el conjunto total de 500 preguntas en tres tipos diferentes. El primero constaba de 413 preguntas factuales obtenidas de los registros de AOL[6] y MSNSearch. El segundo conjunto lo componían 37 preguntas

[1] Están disponibles en: http://trec.nist.gov/data/qa.html.
[2] http://es.encarta.msn.com.
[3] http://www.excite.com.
[4] http://www.msn.com.
[5] http://www.ask.com.
[6] http://www.aol.com.

Original: What is the tallest mountain?

Variantes: What is the world's highest peak?
What is the highest mountain in the world?
Name the highest mountain.
What is the name of the tallest mountain in the world?

Figura 4.1: Ejemplo de pregunta del TREC-9 y sus correspondientes variantes (Voorhees, 2000).

de tipo *lista*, que esperan como resultado más de una respuesta. Por ejemplo "*List the names of chewing gums*", donde los asesores detectaron un total de 16 posibles respuestas ("Stimorol", "Orbit", "Trident", etc.) en el corpus AQUAINT.[7] Estas preguntas de tipo lista fueron construidas por los asesores de las conferencias. Diferentes preguntas tenían diferente número de posibles respuestas, desde 3 ("*What Chinese provinces have a McDonald's restaurant?*") hasta 44 ("*Which countries were visited by first lady Hillary Clinton?*"). Por último se obtuvieron 50 preguntas más de tipo definición de los mismos registros empleados para las preguntas factuales. Como ejemplos de este tipo de preguntas tenemos "*Who is Colin Powell?*", "*What is mold?*" y "*What is Ph in biology?*". Aunque estos conjuntos representan preguntas reales, fueron revisadas por los asesores de las conferencias para corregir errores ortográficos, de puntuación y gramaticales.

Una vez recopiladas las preguntas en inglés pasamos a traducir el corpus a otros idiomas. Esta traducción se llevó a cabo de forma manual. Para el español se partió de las traducciones de las preguntas del TREC-8, TREC-9, TREC-10 y TREC-11 realizadas por el Grupo de Procesamiento del Lenguaje Natural de la UNED.[8] Para obtener el mismo corpus que en inglés se tradujeron las preguntas del TREC-12 y se revisaron todas las anteriores a fin de obtener una traducción uniforme. Para ampliar el conjunto de idiomas de trabajo se realizó la traducción a italiano y catalán, obteniendo finalmente un corpus paralelo de 2.393 preguntas en cuatro idiomas con el que entrenar y evaluar nuestro sistema.

El siguiente paso consistió en etiquetar manualmente las preguntas con su correspondiente clase semántica. Las preguntas del TREC no vienen etiquetadas con su clase, ya que son los propios participantes de estas conferencias los que deciden la taxonomía de clases que más conviene a su sistema de BR. Al no existir ninguna taxonomía estándar en el campo de la BR y los sistemas de CP, decidimos definir una propia tomando como base

[7] Este corpus se compone de noticias periodísticas en inglés obtenidas de tres fuentes: *Xinhua News Service*, *the New York Times News Service* y *the Associated Press Worldstream News Service* (http://www.ldc.upenn.edu/Catalog/docs/LDC2002T31/).

[8] http://nlp.uned.es.

la jerarquía extendida de entidades nombradas de Sekine[9] (Sekine et al., 2002). Esta jerarquía fue diseñada para cubrir aquellas entidades que, dicho de manera informal, aparecen de forma habitual en textos periodísticos. Fue definida partiendo del conjunto de entidades nombradas empleado en las conferencias MUC y del conjunto de entidades desarrollado para el proyecto IREX (Sekine et al., 2000). Esta taxonomía se definió con el objetivo de dar cobertura a entidades nombradas más específicas que las dabas habitualmente en los sistemas de extracción de información. No intenta cubrir ningún dominio particular, sino abordar el etiquetado de entidades en textos de carácter general en dominio abierto.

La propuesta final de Sekine se traduce en una jerarquía de cerca de 150 tipos de entidades nombradas de carácter general. La intención al diseñarla de forma jerarquía era que pudiera ajustarse fácilmente a diferentes tareas según el grado de refinamiento de las entidades a detectar. La jerarquía fue desarrollada en tres fases. En primer lugar se diseñaron tres jerarquías, obtenidas a partir de artículos periodísticos, de sistemas de PLN existentes y de tesauros como WordNet y Roget (Roget, 1987). En una segunda fase se unieron las tres jerarquías en una sola, empleándose para etiquetar diversos corpus de carácter general. Este etiquetado sirvió para refinar la jerarquía anterior y obtener la versión definitiva.

Entre los sistemas que inspiraron el diseño de esta jerarquía están aquellos empleados en la tarea de BR del TREC. Este detalle hace que la cobertura que proporciona esta taxonomía sea especialmente adecuada para etiquetar las preguntas que tienen lugar en nuestro corpus TREC. Para este etiquetado hemos utilizado como base la jerarquía descrita por Sekine, centrándonos en las etiquetas que aparecen en el primer nivel. Sobre esta base se han añadido las clases *definición* y *acrónimo*. Éstas no existían originalmente en la jerarquía de Sekine (ya que ésta se centra en entidades) pero se han incluido para aumentar la cobertura de la taxonomía, ya que son dos tipos de pregunta que se dan de forma habitual en las conferencias TREC.

Una vez definida la taxonomía de clases se pasó al etiquetado de las 2.393 preguntas por parte de dos revisores, obteniendo un *índice kappa* o *kappa agreement*[10] de 0,87. El acuerdo esperado se calculó según la descripción de (Fleiss, 1971) tomando como igual para los revisores la distribución de proporciones sobre las categorías. En caso de no haber acuerdo entre ambos revisores, una tercera persona intervino en la decisión final.

La tabla 4.1 muestra la distribución de preguntas por clase. Se puede observar en la tabla que para la clase *título* no existe ninguna pregunta en el

[9] Existe una versión actualizada en http://nlp.cs.nyu.edu/ene/.

[10] Es una medida estadística que proporciona el grado de acuerdo entre anotadores más allá de las coincidencias fortuitas.

Clase	Preguntas
nombre propio	362
organización	89
lugar	411
instalación	58
producto	131
enfermedad	9
evento	7
título	0
idioma	7
religión	2
objeto natural	73
color	16
temporal	324
número	333
definición	563
acrónimo	8
Total	2.393

Tabla 4.1: Número de preguntas para cada una de las clases de la taxonomía de Sekine en el corpus TREC.

corpus, por lo que en los experimentos no se tendrá en cuenta, dando lugar a una taxonomía final de 15 clases.

4.1.2. Corpus QALL-ME

Este corpus (Boldrini et al., 2009) fue desarrollado dentro del marco del proyecto QALL-ME.[11] Está compuesto por 4.500 preguntas en español con su versión paralela en inglés. Es el resultado de la grabación de preguntas realizadas por 150 usuarios reales de un sistema de BR en dominio turístico. Esta colección de preguntas se creó para tener una muestra de preguntas en lenguaje natural de potenciales usuarios de un sistema de BR. En la figura 4.2 puede verse una muestra de las preguntas recopiladas.

A cada pregunta se le asignó una clase semántica perteneciente a la ontología desarrollada específicamente para el proyecto QALL-ME. Se emplearon un total de 40 clases posibles en el etiquetado (ver tabla 4.2),

[11] QALL-ME es un proyecto financiado por la Unión Europea para establecer una infraestructura para la BR multilingüe y multimodal en el dominio turístico. El sistema QALL-ME (`http://qallme.fbk.eu`.) permite a los usuarios enviar preguntas en lenguaje natural en diferentes idiomas (tanto de forma textual como oral) usando diferentes dispositivos de entrada (como teléfonos móviles), devolviendo una lista de respuestas específicas con el formato más apropiado (pequeños textos, mapas, vídeos o fotos).

¿Cuál es la dirección de la farmacia Manuel Giménez Canales?
¿Qué película de animación puedo ver este sábado?
¿Podría facilitarme el teléfono del local Baikal?
Me gustaría saber el horario del restaurante Terra del Cid.
¿A qué hora cierra el restaurante Davis de Alicante?
Quisiera saber el nombre de algún hotel en Alicante.
¿Qué tipo de comida tienen en la sidrería Aurrerá?
¿Tiene aparcamiento el restaurante Boutique del Mar en Alicante?
¿Cuántas habitaciones tiene el hotel Camposol Park?
¿Aceptan tarjetas de crédito en el hotel Cimbel?

Figura 4.2: Conjunto de preguntas pertenecientes al corpus QALL-ME.

todas ellas relacionadas con el dominio turístico. Para el cálculo del acuerdo entre anotadores se siguió la misma estrategia definida en el punto anterior. El valor obtenido para este acuerdo fue de 0,89. Pese a que el número de clases posibles es mucho mayor en el corpus QALL-ME (40) que en el corpus TREC (15), el acuerdo entre anotadores conseguido fue mayor. Este resultado no es de extrañar ya que, al tratarse de un dominio cerrado, los conceptos sobre los que se puede preguntar están mucho más delimitados y la taxonomía presenta una cobertura más precisa del conjunto de preguntas posibles.

4.2. Características de aprendizaje

Cada instancia del problema debe codificarse mediante un vector de características a partir del cual aprenderá el algoritmo de clasificación. Para mantener la independencia de nuestro sistema con respecto a otras herramientas o recursos lingüísticos, vamos a emplear como únicas características de aprendizaje los n-gramas (ver sección 3.3.1 para más detalles) obtenidos del propio corpus de entrenamiento:

- **Unigramas** (1-gramas). Se emplean los términos extraídos de la pregunta como componentes del vector de características. Esta característica permite que todas las palabras vistas en las instancias de entrenamiento estén representadas durante la fase de aprendizaje.
- **Bigramas** (2-gramas). Representan todas las combinaciones de términos adyacentes en una pregunta como una secuencia de dos palabras. Con respecto a los unigramas, los bigramas tienen la ventaja de añadir información más rica al conjunto de características: saber que dos palabras ocurren una junto a otra proporciona más información sobre la estructura del texto que saber que ocurren de forma independiente

Clase	Preguntas
actividad	14
alojamiento	14
aparcamiento	162
área deportiva	6
bar	7
biblioteca	3
bus	6
cabaña	3
camping	14
cine	90
cocina	1
contacto	5
definición	1.541
descripción de espacio natural	1
destino	1
día de la semana	141
dirección postal	361
director	34
espacio natural	3
farmacia	93
fecha	4
hostal	54
hotel	375
numérico	122
otro	119
patrimonio cultural	5
película	101
periodo	1
periodo de tiempo	401
periodo temporal	413
precio de habitación	203
precio de ticket	105
rango de precio	1
restaurante	292
sala de conferencias	81
servicios	83
servicios en hotel	72
servicios para minusválidos	67
tarjeta de crédito	266
temporal	446
Total	4.500

Tabla 4.2: Número de preguntas por cada una de las clases en el corpus QALL-ME.

dentro de la pregunta. Por ejemplo, saber que la pregunta contiene el bigrama "qué autor" proporciona más información sobre el tipo de respuesta esperada que los unigramas "qué" y "autor" de forma separada.

- **Trigramas** (3-gramas). Los trigramas son similares a los bigramas, aunque en este caso tenemos tuplas de tres palabras. Esta característica representa todas las combinaciones de palabras adyacentes en una pregunta como una secuencia de tres palabras.

- **Combinaciones de n-gramas**. Emplearemos combinaciones de los distintos tamaños de n-gramas anteriores para ver si éstas son capaces de obtener mejores resultados que sus componentes individuales por separado. Vamos a experimentar con combinaciones de unigramas y bigramas (1+2-gramas), unigramas y trigramas (1+3-gramas), bigramas y trigramas (2+3-gramas) y, en último lugar, unigramas, bigramas y trigramas (1+2+3-gramas).

Aunque el tamaño de los n-gramas puede ser ilimitado, en las tareas de PLN no suelen utilizarse n-gramas de tamaño superior a 3 (Manning y Schütze, 1999), ya que esto da lugar al problema de la dispersión de datos comentado en la sección 3.3.1.

Los componentes del vector de características serán cada uno de los posibles n-gramas que se puedan dar en la pregunta. En nuestros experimentos vamos a utilizar vectores binarios, donde la aparición de una característica se representa con un 1 y la no aparición con un 0. En la tarea de clasificación de textos se emplea habitualmente la frecuencia de aparición del n-grama o el *tf-idf* para representar a cada término del vector indicando su peso en el documento. En la tarea de CP carece de sentido usar este tipo de representación para indicar el peso de los n-gramas en la pregunta, ya que la frecuencia de los términos raramente es superior a 1. Por supuesto, ésta es una simplificación ya que en una pregunta como "¿Cuál es el nombre del autor de el nombre de la rosa?", los unigramas "el" y "nombre" aparecen repetidos. Esta circunstancia puntual carece de efecto sobre los algoritmos de aprendizaje y se puede simplificar de forma general la representación empleando vectores binarios.

La obtención de los n-gramas de las preguntas del corpus exige un preproceso. Los corpus con los que trataremos aquí se encuentran en *texto plano*, es decir, carecen de todo formato o etiqueta lingüística que nos informe sobre el papel o significado de las palabras que en él tienen lugar. La única información adicional a cada pregunta es la clase a la que pertenece, asignada manualmente por un revisor para poder entrenar el sistema y evaluar su precisión a la hora de clasificar. La obtención de los n-gramas exige una división previa de la pregunta en *tokens*, donde cada uno de éstos será una palabra, un número o un signo de puntuación. Algunos signos de

puntuación se emplean habitualmente en distintas tareas de PLN,[12] mientras que otros se han descartado al no aportar ningún tipo de información relevante. En el caso de los corpus aquí tratados, las preguntas vienen perfectamente delimitadas por lo que no resulta necesario mantener signos de puntuación para identificarlas. Para la obtención de n-gramas nos basaremos en el concepto práctico de *palabra gráfica* (*graphic word*) tal y como fue definida por Kučera y Francis (1967):

> "[...] *a string of contiguous alphanumeric characters with space on either side; may include hyphens and apostrophes, but no other punctuation marks.*"

Además de la eliminación de signos de puntuación, hay otra serie de transformaciones que se suelen aplicar de forma habitual a los corpus textuales empleados en tareas de aprendizaje. Una sencilla transformación que vamos a utilizar es la de reemplazar todos los caracteres en mayúsculas del texto por su equivalente en minúsculas (Witten et al., 1994). Esta simple transformación permite que términos equivalentes en mayúsculas y minúsculas sean reconocidos como iguales, aumentando la posibilidad de que estas características resulten útiles en la clasificación. Por ejemplo, términos como "Ciudad", "ciudad" o "CIUDAD", convergen de esta manera en una misma característica.

Además de la transformación en minúsculas existen otras opciones comunes para la normalización de textos. Una de estas opciones es la obtención de la raíz de las palabras mediante un *stemmer*, convirtiendo palabras como "oxigenar", "oxigenado" u "oxigenación" a una forma común "oxigen". Este proceso se puede llevar a cabo con un sencillo algoritmo como el de Porter. Sin embargo, existen evidencias empíricas de que el funcionamiento de este algoritmo empeora notablemente con idiomas que presentan una morfología y flexión verbal más rica que el inglés, como es el caso del español, el italiano o el catalán. Por esta razón no vamos a emplear este tipo de normalización en nuestro trabajo.

Otro proceso que permite la normalización del texto es la utilización del lema o forma canónica de los términos. Incluir esta característica implica el uso de herramientas para la lematización, lo que entraría en conflicto con nuestro objetivo de utilizar sistemas que no dependan de otras herramientas lingüísticas. Más aún, en estudios previos realizados en sistemas de CP empleando lematizadores tanto para inglés como para español (Bisbal et al., 2005a) no existen evidencias empíricas de que este tipo de normalización mejore el rendimiento del sistema. Por estas razones, en nuestros experimentos tampoco realizaremos este tipo de normalización.

[12] Es el caso del punto o el signo de interrogación para marcar los límites de las oraciones en el texto, o las comillas para la detección de entidades.

Para obtener los n-gramas de las preguntas hemos utilizado el CMU-Cambridge Statistical Language Modeling Toolkit, un conjunto de herramientas para facilitar la construcción de modelos de lenguaje.[13] Este conjunto de herramientas permiten llevar a cabo tareas como la obtención de frecuencias de aparición de palabras, conteos y estadísticas sobre n-gramas, así como otras tareas para el modelado del lenguaje como el cálculo de la perplejidad.

4.3. Algoritmo de aprendizaje

En el capítulo anterior hacíamos un repaso de algunos de los algoritmos más comunes en la tarea de CP. De entre todos ellos destacaba SVM por su buen rendimiento en esta tarea. Este algoritmo se engloba dentro de los métodos de la teoría del aprendizaje computacional, una disciplina que pretende determinar cuáles son los límites teóricos del aprendizaje (Màrquez, 2001). Desarrolla teorías para interrelacionar conceptos como la probabilidad de tener éxito en el aprendizaje, el tamaño de la muestra de aprendizaje, la complejidad del espacio de hipótesis, el grado de acierto con que se puede aproximar el concepto objetivo o la forma de presentación de los ejemplos de aprendizaje. Otros algoritmos que se engloban dentro de esta familia son winnow, SNoW (ambos mencionados en el capítulo anterior) y AdaBoost (Freund y Schapire, 1997).

SVM posee una serie de propiedades que lo hacen especialmente atractivo para esta tarea:

- Trabaja adecuadamente en espacios de aprendizaje un gran número de características, tratando adecuadamente el problema del sobreajuste.
- Muy tolerante a la presencia de atributos redundantes e irrelevantes.
- Pese a ser un clasificador binario es fácilmente adaptable a problemas multiclase.

Estas propiedades, junto con los estudios previos realizados en este campo, hacen que nos decantemos por este algoritmo para su inclusión en nuestro sistema de clasificación. Para los experimentos realizados en este apartado se ha utilizado la implementación de SVM proporcionada por el conjunto de herramientas de aprendizaje automático Weka. Esta implementación utiliza la técnica de *uno contra uno* para abordar los problemas de clasificación multiclase. El algoritmo de optimización implementado en Weka para el entrenamiento de los SVM es el *Sequential Optimization Algorithm* de Platt (1998). Tras diversos experimentos con los kernels definidos en la figura 3.8, obtuvimos los mejores resultados con el kernel *lineal* y el parámetro

[13] Está disponible en http://svr-www.eng.cam.ac.uk/~prc14/toolkit.html.

de penalización $C = 1$. Cuando el número de características de aprendizaje es grande (como es nuestro caso), puede no ser necesario transformar el vector de entrada a un espacio de mayor dimensión ya que la transformación no-lineal no mejora el rendimiento del clasificador. Esto justifica el buen funcionamiento del kernel lineal, siendo únicamente necesario en este caso ajustar el parámetro C (Hsu et al., 2003).

4.4. Evaluación del sistema

El rendimiento de los sistemas de CP influye decisivamente en el rendimiento global de los sistemas de BR. Sin embargo, muchos otros factores influyen en la habilidad final de los sistemas de BR para obtener una respuesta correcta a una pregunta dada. Se ha demostrado que mejoras en paralelo de la precisión del sistema de CP, del sistema de RI, del reconocimiento de entidades y de la extracción de la respuesta, son necesarias para mejorar el funcionamiento de un sistema de BR (Ittycheriah et al., 2000). Por esta razón, en nuestros experimentos evaluamos los sistemas de CP de forma aislada a los sistemas de BR, para que esta evaluación no se vea afectado por el resto de componentes que intervienen en el proceso completo de búsqueda.

La evaluación de los sistemas de CP, al igual que en los sistemas de clasificación de textos o los sistemas de RI, se lleva a cabo de forma empírica en lugar de hacerlo de forma analítica. Para ello se emplean distintos conjuntos de datos que permiten contrastar su funcionamiento de forma experimental. Para probar que un sistema es correcto y completo de forma analítica, necesitaríamos una especificación formal del problema que el sistema intenta resolver (Sebastiani, 2002). Sin embargo, la idea central de la tarea de CP (la pertenencia de una pregunta a una clase) es, debido a su carácter subjetivo, inherentemente no formalizable.

4.4.1. Medidas de rendimiento

La métrica más común a la hora de evaluar los sistemas de CP es la *precisión*. Formalmente, dado un conjunto de M preguntas (previamente clasificadas) y una lista ordenada de pesos asignados por el sistema de CP a cada clase, definimos la *precisión* P como

$$P = \frac{1}{M} \sum_{i=1}^{M} \delta(ranking_i, 1),$$

donde δ es la función delta de Kronecker, una función de dos variables que vale 1 si éstas son iguales y 0 si no lo son

$$\delta(a,b) = \begin{cases} 1 & \text{si } a = b \\ 0 & \text{en otro caso} \end{cases},$$

y $ranking_i$ es el orden que ocupa la clase correcta en la lista devuelta por el clasificador.

Esta definición tradicional de precisión sólo tiene en cuenta si la clase correcta aparece la primera en la lista. Una medida menos común es la métrica $P_{\leq n}$ (Li y Roth, 2002), una versión generalizada de la anterior. Esta medida tiene en cuenta que el clasificador puede asignar hasta n clases posibles a cada pregunta:

$$P_{\leq n} = \frac{1}{M} \sum_{k=1}^{n} \sum_{i=1}^{M} \delta(ranking_i, k).$$

Esta forma generalizada ofrece una versión relaja de la métrica tradicional P, donde se cumple que

$$P = P_{\leq 1}$$
$$P \leq P_{\leq n}$$
$$P_{\leq n} \leq P_{\leq n+1}, \forall n \geq 1$$

Li y Roth (2002, 2005) definen un clasificador capaz de asignar múltiples etiquetas a una única pregunta, justificando este funcionamiento por la ambigüedad que se produce a la hora de clasificar algunas instancias. En su caso, una pregunta como "¿Qué comen los murciélagos?" podría estar preguntando por una *comida*, una *planta* o un *animal*. En sus experimentos utilizan la medida generalizada $P_{\leq 5}$ que permite valorar hasta cinco clases posibles asignadas a una misma pregunta. A la hora de evaluar nuestro sistema vamos a centrarnos en utilizar la medida tradicional, mucho más difundida que la anterior, donde únicamente se asigna una clase posible por pregunta.

4.4.2. Validación cruzada

Como se comentó en la sección 3.2, cuando se trabaja con sistemas de aprendizaje automático los corpus de trabajo se dividen en dos conjuntos: un conjunto de entrenamiento (y opcionalmente de validación) y un conjunto de evaluación. El conjunto de entrenamiento se utiliza para construir el clasificador, mientras que el conjunto de evaluación se emplea para valorar su rendimiento. El sistema asigna una clase a cada nueva pregunta de entrada del conjunto de evaluación en función de lo aprendido a partir del conjunto de entrenamiento. Esta predicción se compara con la clase "real" asignada manualmente a cada pregunta durante el desarrollo del corpus. Las instancias del corpus de evaluación no pueden ser empleados durante la fase de entrenamiento del clasificador.

En conjuntos de entrenamiento de tamaño limitado no resulta conveniente extraer un subconjunto de éste y dedicarlo exclusivamente a la evaluación del sistema, ya que estaríamos reduciendo el tamaño del posible corpus de aprendizaje. Una alternativa a esta forma de dividir el corpus es lo que se conoce como validación cruzada en k particiones (*k-fold cross-validation*). Este particionado permite utilizar el corpus completo tanto para entrenar como para evaluar. Para ello se divide aleatoriamente[14] el corpus inicial en k conjuntos disjuntos de igual tamaño, empleando de forma iterativa una de las particiones como evaluación y las k-1 restantes como entrenamiento. De esta forma se llevan a cabo k experimentos de clasificación, obteniendo como valor final de precisión la media de las precisiones individuales de cada experimento. El valor empleado de forma más habitual para k es 10 (*10-fold cross validation*). Este número se basa en evidencias teóricas y estudios realizados sobre numerosos conjuntos de datos y diferentes algoritmos (Witten y Frank, 2005).

Otra variante habitual de particionado es *leave-one-out*, donde para un corpus con n muestras de entrenamiento se realizan n experimentos diferentes, seleccionando una muestra para evaluación y n-1 para entrenamiento. Esta medida es difícil de poner en práctica con la mayoría de algoritmos de aprendizaje debido al coste computacional de entrenar tantos clasificadores como muestras haya en el corpus.[15]

En nuestros experimentos con el corpus TREC y el corpus QALL-ME hemos optado por emplear la validación cruzada equilibrada con 10 particiones. De esta forma no es necesario dedicar una parte en exclusiva del corpus para la evaluación, pudiendo entrenar y evaluar con todo el conjunto de preguntas.

4.4.3. Credibilidad de los resultados

A la hora de comparar el funcionamiento de distintos sistemas de clasificación para un problema particular interesa saber fehacientemente cuando uno es mejor que otro, minimizando la posibilidad de que la diferencia de precisión obtenida pueda ser fortuita a causa del conjunto de datos empleado durante el proceso de experimentación. Certificar la validez de los resultados es un reto para los investigadores en aprendizaje automático. Tal y como indica Sundblad (2007), estas técnicas de corroboración de resultados no se han utilizado con asiduidad en los trabajos realizados en el campo de la CP. Resulta difícil en estas condiciones certificar si las supuestas mejoras aportados por algunas de las aproximaciones en este campo son reales o no.

[14] Esta división puede ser equilibrada (*stratified k-fold cross-validation*) para que el número de muestras resultantes en las particiones conserve la proporción de muestras por clase del corpus original.

[15] Esta técnica es factible en modelos de aprendizaje basados en ejemplos, ya que al no intentar generalizar se computan de forma rápida.

La verificación de los resultados se consigue empleando test estadísticos (*significance tests*). El objetivo de estos tests es determinar cuándo un sistema, por término medio, ofrece mejor o peor precisión que otro a lo largo de todos los posibles conjuntos de entrenamiento y evaluación que puedan ser obtenidos en el dominio de trabajo.

Uno de los mecanismos estadísticos más conocidos para llevar a cabo la validación de resultados es el *t-test* [16] (Dietterich, 1998). Este test se basa en la media y varianza de precisión obtenida por el clasificador. Esto permite que un sistema no se considere mejor que otro simplemente mirando la media, sino teniendo en cuenta también la varianza de sus resultados. Sólo si el test estadístico rechaza la posibilidad de una diferencia accidental entre ambos sistemas, podremos decir de forma confiada que uno es mejor que otro. Para llevar a cabo este test se realiza una validación cruzada sobre diferentes particiones del conjunto de datos, evaluando cada uno de los sistemas sobre dichas particiones. Cuando las particiones sobre las que se evalúan los sistemas son las mismas (tenemos el mismo conjunto de entrenamiento y evaluación en cada una de las iteraciones) se puede emplear una versión más precisa del *t-test* conocida como *paired t-test*. El *t-test* estándar puede generar demasiadas diferencias significativas debido a variaciones en las estimaciones, en particular, cuando simplemente se realiza una validación cruzada de k particiones. Por ello, se han propuesto diversas modificaciones de este test para evitar algunos de los problemas que presenta. En nuestros experimentos hemos utilizado el *corrected resampled t-test* (Nadeau y Bengio, 2003), empleando una prueba bilateral (*two-tailed test*) dado que, a priori, no sabemos si la media de uno de los sistemas a comparar es mayor que la media del otro o viceversa.

Para valorar los resultados de este tipo de test estadísticos se aporta un grado de confianza representado por el valor p. El grado de confianza que se considera aceptable en experimentación suele ser de $p < 0,05$ o $p < 0,01$, indicando que la diferencia obtenida entre los sistemas no se debe al azar con una seguridad del 95 % o del 99 % respectivamente. A medida que reducimos el límite p incrementamos la confianza en las conclusiones obtenidas.

4.5. Experimentación

Hemos llevado a cabo una serie de experimentos empleando las características descritas anteriormente sobre los dos corpus presentados. En primer lugar vamos a realizar diversos experimentos con el corpus TREC para ver el comportamiento de las distintas características sobre cada uno de los cuatro idiomas del corpus. A continuación, el corpus QALL-ME nos permitirá contrastar en un dominio restringido los resultados anteriores obtenidos por el clasificador sobre dominio abierto. Por último,

[16] También conocido *Student's t-test* en homenaje a su autor.

Característica	Inglés	Español	Italiano	Catalán
1-gramas	81,64	80,97	**79,62**	80,54
2-gramas	76,78	75,33	72,68	76,00
3-gramas	54,55	58,10	56,76	62,89
1+2-gramas	**81,70**	**81,17**	79,45	**81,03**
1+3-gramas	80,07	79,66	78,49	79,94
2+3-gramas	75,18	72,70	70,47	74,34
1+2+3-gramas	80,36	80,07	78,39	80,43

Tabla 4.3: Precisión obtenida por cada una de las características sobre el corpus TREC. Los mejores valores de precisión para cada idioma se muestran en negrita.

completaremos nuestro estudio aplicando diferentes técnicas de selección de características con la intención de reducir la dimensionalidad del espacio de aprendizaje, eliminar ruido y mejorar así el rendimiento del sistema.

4.5.1. Comparación entre idiomas

En este apartado vamos a centrarnos en experimentar con SVM y con las distintas características definidas en la sección 4.2 sobre los cuatro idiomas del corpus TREC. La tabla 4.3 muestran los resultados obtenidos por cada una de las características en los distintos idiomas.

Estos valores revelan que los resultados obtenidos para cada uno de los idiomas son bastante similares. En el caso de los 1-gramas, los resultados para inglés (81,64 %) son ligeramente superiores al resto de idiomas, siendo el italiano el que peor resultados ofrece (79,62 %). Esta tendencia se repite de nuevo para los 2-gramas. En los experimentos con 3-gramas, sin embargo, se obtienen los peores resultados para el inglés (54,55 %), siendo los mejores resultados los de catalán (62,89 %). En los experimentos combinando características, de forma similar a los experimentos individuales, se obtienen mejores resultados para inglés que para el resto, siendo el italiano el idioma para el que peor precisión ofrece el sistema. La única excepción es la combinación de 1+2+3-gramas, donde son los experimentos en catalán los que mejores resultados ofrecen. Una explicación para estas ligeras diferencias de rendimiento es el grado de flexión verbal y nominal de cada uno de los idiomas tratados. Este grado de flexión hace que el número de n-gramas diferentes que nos podemos encontrar sea mayor para idiomas como el español, el italiano y el catalán, que para el inglés, lo cual dificulta la tarea de clasificación. La tabla 4.4 refleja el tamaño del vector de aprendizaje (número de n-gramas diferentes) para cada una de las características tratadas en cada uno de los cuatro idiomas. En el caso de los 1-gramas, el valor mostrado coincide con el tamaño del vocabulario del corpus. Para el inglés, por ejemplo, existen un total de 3.764 términos diferentes en el corpus de

Característica	Inglés	Español	Italiano	Catalán
1-gramas	3.764	4.164	4.315	4.190
2-gramas	8.465	8.578	8.644	8.625
3-gramas	10.015	10.358	9.842	10.391
1+2-gramas	12.229	12.742	12.959	12.815
1+3-gramas	13.779	14.522	14.157	14.581
2+3-gramas	18.480	18.936	18.486	19.016
1+2+3-gramas	22.244	23.100	22.801	23.206

Tabla 4.4: Tamaño del vector de aprendizaje para cada una de las características sobre el corpus TREC.

preguntas, mientras que para el italiano este número es de 4.315, lo que supone casi un 15 % de incremento con respecto al anterior.

En la tabla 4.5 se muestra la comparación de rendimiento de las distintas características mediante *t-test* para cada idioma. Los símbolos "$\gg$" y "$>$" indican que la *característica 1* es significativamente mejor que la *característica 2* con un grado de confianza $p < 0,01$ y $p < 0,05$ respectivamente. De forma equivalente, "$\ll$" y "$<$" indican que la *característica 1* es significativamente peor que la *característica 2*. El símbolo "$=$" indica que no hay una diferencia significativa de funcionamiento entre las dos aproximaciones.

Los resultados del *t-test* reflejan que, de entre las características individuales (1-gramas, 2-gramas y 3-gramas), los resultados obtenidos por los 1-gramas son significativamente mejores que los obtenidos con los n-gramas de mayor tamaño para todos los idiomas. Conforme aumentamos el tamaño de los n-gramas reducimos la posibilidad de encontrar repeticiones de éstos en el texto (el problema de la dispersión de datos), haciendo que su aportación a la clasificación tenga escasa relevancia en muchos de los casos. Adicionalmente, el tamaño del vocabulario aumenta de forma ostensible tal y como puede apreciarse en la tabla 4.4. Para el inglés, por ejemplo, la precisión obtenida con 1-gramas (81,64 %) es considerablemente mayor que para 2-gramas (76,78 %), haciéndose más notable cuando se compara con los 3-gramas (54,55 %). Este comportamiento se repite en todos los idiomas estudiados.

Para las combinaciones de características, los valores que se obtienen con la combinación 1+2-gramas obtienen mejores resultados que los 1-gramas por sí solos (excepto para el catalán, donde se obtiene 79,62 % para 1-gramas y 79,45 % para 1+2-gramas). Sin embargo, la tabla 4.5 revela que estas diferencias no son realmente significativas. Por otra parte, las combinaciones que emplean 3-gramas desmuestran obtener peores resultados que los

Caract. 1	Caract. 2	Inglés	Español	Italiano	Catalán
1-gramas	2-gramas	$\gg$	$\gg$	$\gg$	$\gg$
1-gramas	3-gramas	$\gg$	$\gg$	$\gg$	$\gg$
1-gramas	1+2-gramas	$=$	$=$	$=$	$=$
1-gramas	1+3-gramas	$\gg$	$>$	$>$	$=$
1-gramas	2+3-gramas	$\gg$	$\gg$	$\gg$	$\gg$
1-gramas	1+2+3-gramas	$>$	$=$	$=$	$=$
2-gramas	3-gramas	$\gg$	$\gg$	$\gg$	$\gg$
2-gramas	1+2-gramas	$\ll$	$\ll$	$\ll$	$\ll$
2-gramas	1+3-gramas	$\ll$	$\ll$	$\ll$	$\ll$
2-gramas	2+3-gramas	$\gg$	$\gg$	$\gg$	$\gg$
2-gramas	1+2+3-gramas	$\ll$	$\ll$	$\ll$	$\ll$
3-gramas	1+2-gramas	$\ll$	$\ll$	$\ll$	$\ll$
3-gramas	1+3-gramas	$\ll$	$\ll$	$\ll$	$\ll$
3-gramas	2+3-gramas	$\ll$	$\ll$	$\ll$	$\ll$
3-gramas	1+2+3-gramas	$\ll$	$\ll$	$\ll$	$\ll$
1+2-gramas	1+3-gramas	$\gg$	$\gg$	$=$	$>$
1+2-gramas	2+3-gramas	$\gg$	$\gg$	$\gg$	$\gg$
1+2-gramas	1+2+3-gramas	$\gg$	$\gg$	$>$	$=$
1+3-gramas	2+3-gramas	$\gg$	$\gg$	$\gg$	$\gg$
1+3-gramas	1+2+3-gramas	$=$	$=$	$=$	$=$
2+3-gramas	1+2+3-gramas	$\ll$	$\ll$	$\ll$	$\ll$

Tabla 4.5: Comparación entre características sobre el corpus TREC.

Característica	Inglés	Español
1-gramas	94,39	94,96
2-gramas	94,28	94,14
3-gramas	92,19	90,20
1+2-gramas	**95,46**	**95,65**
1+3-gramas	95,23	95,30
2+3-gramas	93,66	93,15
1+2+3-gramas	95,07	95,39

Tabla 4.6: Precisión obtenida por cada una de las características sobre el corpus QALL-ME. Los mejores valores de precisión para cada idioma se muestran en negrita.

sistemas que no los emplean (1-gramas $\gg$ 1+3-gramas, 2-gramas $\gg$ 2+3-gramas, 1+2-gramas $\gg$ 1+2+3-gramas). Esto demuestra que los 3-gramas únicamente introducen ruido en el sistema al tratarse de datos muy dispersos (apenas hay repeticiones entre preguntas). Aumentan considerablemente el tamaño del vector de características y obtienen peor rendimiento que otras características tanto de forma individual como combinada. Resulta recomendable, por tanto, limitar el tamaño de los n-gramas a 2 a la hora de construir el clasificador.

4.5.2. Comparación entre dominios

Repetir los experimentos anteriores sobre el corpus QALL-ME nos permitirá analizar si los comportamientos detectados para el sistema sobre dominio abierto se repiten para los experimentos sobre dominio restringido, empleando una taxonomía diferente a la anterior.

En la tabla 4.6 pueden verse los resultados obtenidos con cada una de las características de aprendizaje sobre el corpus QALL-ME. La precisión media obtenida por el sistema con las distintas características es notablemente mayor que para los experimentos sobre el corpus TREC. Para inglés, por ejemplo, la precisión media obtenida es 94,33 % frente al 75,75 % obtenida en el corpus TREC. La taxonomía con la que se etiquetó el corpus QALL-ME presentaba un total de 40 clases posibles, frente a las 15 con las que fue etiquetado el corpus TREC. Esto podría sugerir que la tarea de clasificar preguntas para la taxonomía del corpus QALL-ME sería más dificultosa. Sin embargo, los resultados revelan todo lo contrario.

Hay dos razones principales que justifican los resultados obtenidos. En primer lugar, el tamaño del corpus QALL-ME (4.500 preguntas) es ostensiblemente mayor que el corpus TREC (2.393 preguntas). Es sobradamente conocido que el rendimiento de los clasificadores basados en corpus es directamente proporcional al número de muestras del conjunto de datos de entrenamiento. El segundo factor determinante en esta diferencia

Característica	Inglés	Español
1-gramas	2.741	2.825
2-gramas	7.997	8.199
3-gramas	12.944	12.934
1+2-gramas	10.738	11.024
1+3-gramas	15.685	15.759
2+3-gramas	20.941	21.133
1+2+3-gramas	23.681	23.958

Tabla 4.7: Tamaño del vector de aprendizaje para cada una de las características sobre el corpus QALL-ME.

de rendimiento es el dominio de aplicación. En un dominio cerrado, como el representado por el corpus QALL-ME, el número posible de preguntas y reformulaciones de éstas es limitado. Cuando describíamos los corpus en la sección 4.1, veíamos que el acuerdos entre anotadores humanos a la hora de asignar clases a las preguntas era mayor durante el etiquetado del corpus QALL-ME. Este resultado revela que la ambigüedad presente en las preguntas que se dan en este corpus es menor que en un corpus de dominio abierto como el corpus TREC. Esto se hace patente también cuando observamos el tamaño de los vectores de características en la tabla 4.7. Pese a que el corpus QALL-ME es considerablemente mayor que el corpus TREC (casi el doble de instancias), el número de 1-gramas y 2-gramas diferentes que tienen lugar es inferior tanto para inglés como para español. Esto refleja que, en dominio cerrado, el tamaño del vocabulario es más reducido que en dominio abierto.

La comparativa entre las distintas características puede verse en la tabla 4.8. En este caso la diferencia entre 1-gramas, 2-gramas y 3-gramas es menor que para el caso de dominio abierto. Esto refleja que, en dominio restringido, las reformulaciones de una misma pregunta son menores, provocando que los n-gramas de mayor tamaño sean más frecuentes. En inglés no existe una diferencia significativa entre los 1-gramas y los 2-gramas, aunque sí entre los 1-gramas y los 3-gramas. En el caso del español, los 1-gramas se comportan significativamente mejor que 2-gramas y 3-gramas. En el caso de la combinación de características, cabe destacar el buen rendimiento en inglés de las combinaciones 1+2-gramas, 1+3-gramas y 1+2+3-gramas. En español, la combinación de 1+2-gramas y 1+2+3-gramas son las que ofrecen los mejores resultados, funcionando significativamente mejor que el resto. Estos resultados reflejan que la contribución de los 3-gramas en dominio restringido es más relevante.

Característica 1	Característica 2	Inglés	Español
1-gramas	2-gramas	$=$	$>$
1-gramas	3-gramas	$\gg$	$\gg$
1-gramas	1+2-gramas	$\ll$	$<$
1-gramas	1+3-gramas	$<$	$=$
1-gramas	2+3-gramas	$=$	$\gg$
1-gramas	1+2+3-gramas	$<$	$=$
2-gramas	3-gramas	$\gg$	$\gg$
2-gramas	1+2-gramas	$\ll$	$\ll$
2-gramas	1+3-gramas	$\ll$	$\ll$
2-gramas	2+3-gramas	$\gg$	$\gg$
2-gramas	1+2+3-gramas	$\ll$	$\ll$
3-gramas	1+2-gramas	$\ll$	$\ll$
3-gramas	1+3-gramas	$\ll$	$\ll$
3-gramas	2+3-gramas	$\ll$	$\ll$
3-gramas	1+2+3-gramas	$\ll$	$\ll$
1+2-gramas	1+3-gramas	$=$	$\gg$
1+2-gramas	2+3-gramas	$\gg$	$\gg$
1+2-gramas	1+2+3-gramas	$>$	$=$
1+3-gramas	2+3-gramas	$\gg$	$\gg$
1+3-gramas	1+2+3-gramas	$=$	$=$
2+3-gramas	1+2+3-gramas	$\ll$	$\ll$

Tabla 4.8: Comparación entre características sobre el corpus QALL-ME.

4.5.3. Selección de características

En muchas situaciones prácticas que se dan al abordar la tarea de clasificación hay demasiadas características de aprendizaje que manejar por parte de los algoritmos. Algunas de estas características, quizás la inmensa mayoría, son claramente irrelevantes o redundantes para el aprendizaje. Es el caso de las características obtenidas a partir de los n-gramas del texto, donde gran parte de éstas no aporta información útil al proceso de aprendizaje. En la tarea de CP, los n-gramas que representan nombres propios, cifras, fechas o aquellos que aparecen con escasa frecuencia, difícilmente pueden dar indicios sobre la clase semántica de la pregunta.

Son numerosos los algoritmos de aprendizaje que tratan por ellos mismos de seleccionar las características más apropiadas para la clasificación, ignorando aquellas que sean irrelevantes o redundantes. Winnow es un ejemplo de algoritmo eficiente a la hora de buscar características relevantes en espacios de gran dimensionalidad.[17] Sin embargo hay algoritmos como NB que no realizan ningún tipo de selección de características, de forma que todas ellas, independientemente de su relevancia, son utilizadas en el proceso de clasificación. En la práctica, el funcionamiento de los algoritmos de aprendizaje puede ser mejorado frecuentemente mediante un proceso de preselección de características o *feature selection* (Cardie, 1996; Wang y He, 2004; Yang y Pedersen, 1997). Numerosos experimentos han demostrado que añadir características irrelevantes o redundantes provoca que el funcionamiento de algoritmos como los árboles de decisión, los clasificadores lineales, los sistemas de aprendizaje basado en ejemplos o los métodos de agrupamiento se deteriore.

Realizar una selección previa de características aporta diversos beneficios al proceso posterior de aprendizaje. Uno de ellos es la reducción de dimensionalidad que se produce al descartar características irrelevantes. Hay muchos algoritmos que no son capaces de manejar espacios de alta dimensionalidad y trabajar con vectores de características de gran tamaño. Esta reducción de tamaño puede hacer que el problema sea asequible para algoritmos que de otra forma no podrían afrontarlo, además de mejorar de forma general la velocidad de cómputo a la hora de entrenar y evaluar el clasificador. Otra ventaja de la selección de características es la eliminación de *ruido*[18] que se produce al descartar características que realmente no contribuyen al proceso de clasificación. Esta reducción permite minimizar el problema del sobreajuste.

Siempre que se posea un profundo entendimiento del problema de aprendizaje y del significado de las características, la mejor manera de seleccionar estas características es hacerlo de forma manual. Un ejemplo de

[17] A este tipo de algoritmos se los conoce como *attribute-efficient learner*.

[18] Una característica introduce *ruido* cuando al ser añadida al vector de aprendizaje incrementa el error de clasificación sobre nuevos datos.

...	**what**	whereby
we	**what's**	wherein
we'd	whatever	whereupon
we're	**when**	wherever
we've	whence	whether
welcome	whenever	**which**
well	**where**	while
went	**where's**	whither
were	whereafter	**who**
weren't	whereas	...

Figura 4.3: Extracto de la lista de 572 *stopwords* del sistema SMART. Se han destacado en negrita algunos términos que resultan fundamentales en la tarea de CP.

este tipo de selección para el problema de CP sería emplear únicamente las dos primeras palabras (el primer bigrama) de la pregunta como característica de aprendizaje (Kocik, 2004). Este tipo de característica, totalmente enfocada a la tarea de CP, resalta la importancia del inicio de la pregunta ("¿Cuánto mide...?", "¿Quién es...?" o "¿Qué color...?") a la hora de determinar la clase semántica a la que pertenece. Realizar la selección de esta forma presenta el problema de ser excesivamente dependientes de la tarea y del idioma tratado.

Otra forma de reducir la dimensionalidad del espacio de aprendizaje es el uso de listas de *stopwords*. La utilización de estas listas está muy extendida en las tareas de RI, ya que permiten descartar términos que aparecen habitualmente en los textos y que no aportan información útil sobre los mismos. Estas listas incluyen términos como "es", "la" o "son". Existen también estudios que demuestran la utilidad de estas listas para la reducción de dimensionalidad y subsiguiente mejora del rendimiento de los sistemas de clasificación de textos (Joachims, 1998). En la tarea de CP, sin embargo, estas listas no resultan de utilidad. La figura 4.3 muestra un extracto de la lista de *stopwords* utilizada por el sistema de recuperación de información SMART (Rocchio, 1971). En esta tabla, se puede observar que existen términos como "*who*", "*when*" o "*where*" que son fundamentales para detectar el tipo de respuesta esperada. Por esta razón, además de por ser un recurso lingüísticos creado manualmente y totalmente dependiente del idioma, en nuestros experimentos no vamos a realizar este tipo de reducción de dimensionalidad.

Frente a las aproximaciones manuales existen técnicas basadas en la teoría de la información que permiten determinar estadísticamente cuáles son las características que aportan más información al proceso de aprendizaje, dando la posibilidad de descartar aquellas que no resulten de utilidad. Para llevar a cabo este proceso se emplea una función que mide la

importancia de cada característica en la tarea de clasificación. La selección se lleva a cabo manteniendo un subconjunto de las características consideradas más relevantes por dicha función. De entre estas funciones podemos destacar el *umbral de frecuencia* (*frequency thresholding*), *information gain* (IG), *mutual information* (MI) y χ^2. Vamos a centrarnos en el estudio del *umbral de frecuencia*, IG y χ^2, ya que resultados experimentales previos han demostrado que MI funciona significativamente peor a la hora de seleccionar características en tareas de clasificación (Yang y Pedersen, 1997). En ese mismo estudio demostraron que técnicas sofisticadas como IG o χ^2 pueden reducir la dimensionalidad del espacio de características por un factor de 100 sin pérdida (e incluso con pequeñas ganancias) en la efectividad del clasificador.

Estas técnicas de selección de características han sido ampliamente estudiadas en la tarea de clasificación de textos (Forman, 2003; Gabrilovich y Markovitch, 2004; Lewis, 1992; Rogati y Yang, 2002; Scott, 1999; Taira y Haruno, 1999). Sin embargo, no existen estudios previos sobre el efecto de la selección de características en la tarea de CP. En esta sección experimentaremos el rendimiento de las distintas combinaciones de n-gramas y los métodos de selección de características sobre el corpus TREC.

Umbral de frecuencia

La selección basada en *umbral de frecuencia* consiste en mantener aquellos n-gramas que aparecen al menos un número determinado de veces en el corpus. Este umbral puede referirse tanto al número de preguntas en las que aparece el n-grama como al número total de veces que aparece en la colección. La asunción detrás de esta técnica es que los n-gramas que ocurren de forma escasa en el corpus no son relevantes para la predicción de las clases o no resultan relevantes para el rendimiento global del sistema. La eliminación de estos n-gramas poco frecuentes reduce la dimensionalidad del espacio de características, con lo cual es posible conseguir una mejora en el funcionamiento del clasificador si los términos eliminados estaban introduciendo ruido en la construcción del modelo.

La cantidad de texto de la que disponemos en la tarea de CP es escasa, por lo que la frecuencia de los n-gramas en la colección es habitualmente baja. Por esta razón no es conveniente establecer un umbral elevado en la eliminación de términos en función de su frecuencia, ya que la reducción podría ser demasiado drástica. En la figura 4.4 se muestra la evolución del número de 1-gramas, 2-gramas y 3-gramas en el corpus TREC en inglés dependiendo del umbral de frecuencia. Se puede observar que, simplemente eliminando aquellos n-gramas que aparecen una única vez en el corpus, la reducción de dimensionalidad obtenida es más que notable. Vamos a centrarnos por ello en estudiar cómo afecta al sistema la eliminación de

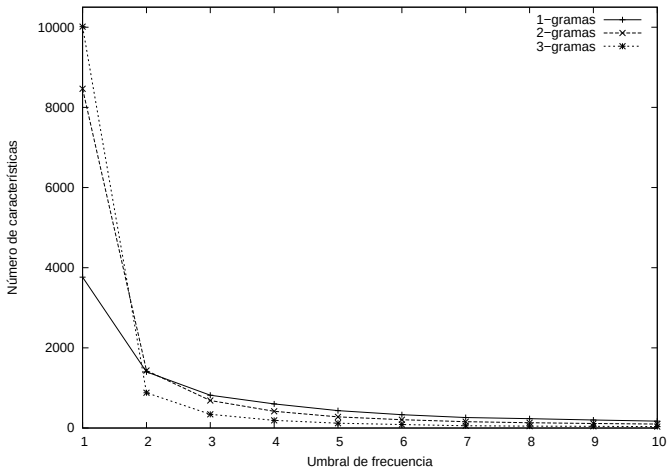

Figura 4.4: Número de 1-gramas, 2-gramas y 3-gramas para inglés en el corpus TREC dependiendo del umbral de frecuencia de corte.

aquellos n-gramas que aparezcan una única vez, conocidos como *hapax legomena*.

El número de n-gramas que ocurren de forma habitual en un corpus es mucho menor que aquellos n-gramas que ocurren una única vez.[19] En tareas de clasificación de textos es habitual la eliminación de los *hapax legomena* para reducir el vocabulario del problema y, consecuentemente, el tamaño del espacio de características (Dumais et al., 1998). En la tabla 4.9 se muestra cómo varía el número de 1-gramas, 2-gramas y 3-gramas al eliminar los *hapax legomena* del corpus de preguntas. Los tamaños para las combinaciones de n-gramas no se muestran en esta tabla ya que son simplemente la suma de los vocabularios de sus componentes individuales. Esta tabla revela que la reducción de características para 1-gramas en los diferentes idiomas se sitúa entre el 62 % y el 68 %, para 2-gramas entre el 80 % y el 83 %, y para 3-gramas entre el 88 % y el 91 %. Se constata que al aumentar el tamaño del n-grama se reduce la frecuencia de aparición de éstos y se incrementa el problema de la dispersión de datos.

Para comprobar el efecto de esta eliminación hemos realizado una serie de experimentos comparativos con los distintos tamaños de n-gramas sobre el conjunto original y sobre el conjunto obtenido tras esta reducción. Los resultados pueden verse en la tabla 4.10. La comparación de estos resultados con los valores originales presentes en la tabla 4.3 se puede ver en la tabla 4.11. Esta comparación revela que la reducción de dimensionalidad obtiene mejoras para todos los tamaños de n-gramas y sus combinaciones, excepto para los 1-gramas. Esto se justifica porque con n-gramas de mayor tamaño

[19] Es lo que se conoce como Ley de Zipf (Zipf, 1949).

Caract.	Inglés	Español	Italiano	Catalán
1-gramas	1.400 (62,81 %)	1.411 (66,11 %)	1.374 (68,16 %)	1.417 (66,18 %)
2-gramas	1.435 (83,05 %)	1.681 (80,40 %)	1.501 (82,64 %)	1.622 (81,19 %)
3-gramas	879 (91,22 %)	1.131 (89,08 %)	965 (90,20 %)	1.155 (88,88 %)

Tabla 4.9: Número de características y porcentaje de reducción sobre el espacio original tras la eliminación de los *hapax legomena*.

Característica	Inglés	Español	Italiano	Catalán
1-gramas	80,68	80,47	**78,56**	**80,32**
2-gramas	76,96	74,28	73,13	75,93
3-gramas	59,41	60,57	59,79	65,57
1+2-gramas	**81,74**	**80,50**	78,37	80,12
1+3-gramas	81,38	80,04	78,23	79,94
2+3-gramas	77,25	73,20	72,61	74,90
1+2+3-gramas	81,49	80,11	78,37	79,83

Tabla 4.10: Precisión obtenida por cada una de las características con la eliminación de *hapax legomena*.

(y las combinaciones de éstos) el número de características que introducen ruido en la clasificación aumenta de forma considerable. Eliminar aquellos n-gramas poco frecuentes permite la depuración del vector de características y la mejora en la precisión del clasificador. Tenemos, por ejemplo, que para la combinación 1+2+3-gramas en inglés, la precisión sufre una mejora significativa ($p < 0,01$) pasando de 80,36 % a 81,49 % y reduciendo el tamaño del vector de 22.244 componentes a tan sólo 3.714 (una reducción del 83,39 %).

El método del umbral de frecuencia suele ser considerado como una técnica *ad hoc* para mejorar la eficiencia del clasificador, pero no como un criterio fundamentado para la selección de características relevantes. De hecho, no suele usarse como una técnica agresiva de selección, ya que los términos poco frecuentes son considerados como informativos (al menos en las tareas de RI y clasificación de textos).

χ^2

χ^2 se emplea en estadística para evaluar la independencia de dos eventos. En selección de características, los dos eventos son la ocurrencia de la característica c y la ocurrencia de la clase i. Da una medida de cuánto se desvía la frecuencia esperada de la frecuencia observada. Un valor grande de χ^2 indica que la hipótesis de independencia, que implica que los valores observados y esperados son similares, es incorrecta. Desde un punto de vista

Característica	Inglés	Español	Italiano	Catalán
1-gramas	$<$	$=$	$=$	$=$
2-gramas	$=$	$=$	$=$	$=$
3-gramas	$\gg$	$\gg$	$\gg$	$\gg$
1+2-gramas	$=$	$=$	$=$	$=$
1+3-gramas	$>$	$=$	$=$	$=$
2+3-gramas	$\gg$	$=$	$\gg$	$=$
1+2+3-gramas	$=$	$=$	$=$	$=$

Tabla 4.11: Comparación estadística de la precisión entre el experimento original y el experimento de eliminación de *hapax legomena*.

estadístico, este método puede ser problemático: cuando un test estadístico se emplea muchas veces, la probabilidad de obtener al menos un error se incrementa. En cualquier caso, en la clasificación de preguntas o textos raramente importa que unos pocos términos se añadan al conjunto de características o que sean eliminados. La estadística χ^2 tiene un valor natural de 0 si c e i son independientes. La selección se lleva a cabo ordenando las características en función del valor de χ^2 y escogiendo los m mejor valorados.

La figura 4.5 (a) muestra la precisión obtenida por el sistema en inglés, empleando 1-gramas y distintas variantes del número m de características seleccionadas. Se observa que cualquier reducción llevada a cabo sobre el conjunto original, mostrado en la tabla 4.4, provoca un deterioro en el rendimiento del sistema. La única reducción que no provoca una pérdida significativa en la precisión se produce para m=3.500. En la figura 4.5 (b) vemos los resultados para 2-gramas. En este caso, con m=7.000 se obtiene mejor resultado que en el original (76,85 % frente a 76,78 %) aunque esta mejora no es significativa. Para el resto de casos se observa un comportamiento similar al de los 1-gramas. Con m=5.000 se obtiene peor precisión que en el caso original, aunque esta diferencia no es estadísticamente significativa ($p < 0,05$). Por tanto, se puede reducir hasta m=5.000 sin que haya una pérdida de precisión significativa en el sistema. La figura 4.5 (c) muestra los resultados para 3-gramas. Aquí se obtiene un mejor valor para m=8.000 (55,50 %) aunque este valor no es significativamente mejor que para el caso original (54,55 %). Para reducciones mayores, el sistema tiene una bajada de rendimiento significativo con respecto al original. En este sentido el comportamiento es bastante similar al caso de 1-gramas y 2-gramas. Con el resto de idiomas tampoco se obtienen mejoras significativas empleando esta técnica sobre los conjuntos de características mencionados.

En la figura 4.6 (a) vemos los resultados para 1+2-gramas. Aquí se obtiene un mejor valor para m=7.000 (81,82 %) aunque este valor no es

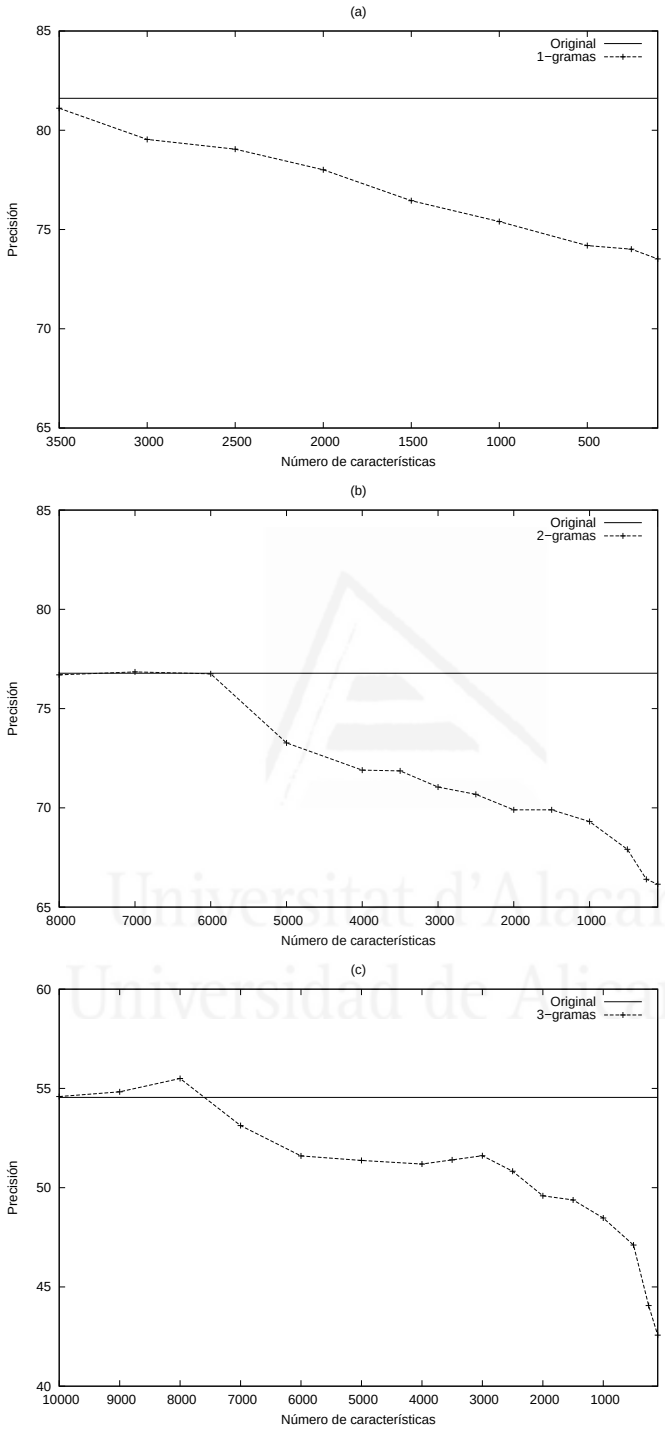

Figura 4.5: Precisión obtenida con χ^2 para distintas reducciones de dimensionalidad en inglés para (a) 1-gramas, (b) 2-gramas y (c) 3-gramas. *Original* representa la precisión obtenida en el experimento original sin reducción.

significativamente mejor que para el caso original (81.70 %). Se puede reducir hasta m=5.000 sin que el resultado sea significativamente peor ($p < 0,01$). Los resultados para la combinación de 1+3-gramas se puede ver en la figura 4.6 (b). Aquí se obtiene un mejor valor para m=9.000 (80,92 %) aunque este valor no es significativamente mejor que para el original (80,08 %). Se puede reducir hasta m=6.000 sin que el resultado sea significativamente peor ($p < 0,01$). En la figura 4.6 (c) tenemos los resultados para 2+3-gramas. Aquí se obtiene un mejor valor para m=5.000 (76,81 %), siendo significativamente mejor que para el caso original (75,18 %). Se puede reducir hasta m=3.000 sin que el resultado sea significativamente peor ($p < 0,01$).

La última combinación de n-gramas, la que se obtiene mediante 1+2+3-gramas, merece un estudio en detalle. En la figura 4.7 (a) se muestra la precisión obtenida para esta combinación en inglés. Se obtiene un máximo para m=11.000 (menos de la mitad de características del conjunto original) obteniendo una precisión de 81,96 %, significativamente mayor ($p > 0,01$) que el experimento original. En la figura 4.7 (b) podemos ver los resultados para el corpus en español con estas mismas características. Al igual que sucedía con el corpus en inglés, esta combinación de características es la que más se beneficia de la aplicación de χ^2 en lo que a ganancia de rendimiento se refiere. Se obtiene un máximo de precisión (81,28 %) con m=12.000 características, siendo este valor significativamente mejor ($p < 0,01$) que con el conjunto original de 23.100 características (80,05 %). Para el resto de resultados podemos destacar que con m=5.000 la precisión del sistema no es significativamente peor que para el caso original habiendo reducido el tamaño del vector un 78,35 %. Los resultados de este mismo conjunto de características para italiano se pueden ver en la figura 4.7 (c). En este caso la aportación de la selección es menos efectiva. La máxima precisión se alcanza con m=12.000 (79,47 %). Este valor, sin embargo, no es significativamente mejor que el original (78,39 %). La única mejora significativa ($p < 0,05$) se consigue para m=13.000, obteniendo una precisión de 79,43 %. Podemos reducir hasta m=8.000 sin pérdida significativa ($p < 0,05$) de precisión. Por último, la figura 4.7 (d) muestra los resultados para catalán. La máxima precisión se alcanza con m=11.000 (81,25 %), obteniendo una mejora significativa ($p < 0,05$) con respecto al valor original (80,43 %). Se puede conseguir reducir el número de características hasta m=5.000 sin que haya una pérdida significativa de precisión ($p < 0,05$) con respecto al original.

Information Gain

Information gain (IG) es otra técnica empleada frecuentemente como criterio para la selección de términos relevantes en el campo del aprendizaje automático (Mitchell, 1997). Mide el número de bits de información

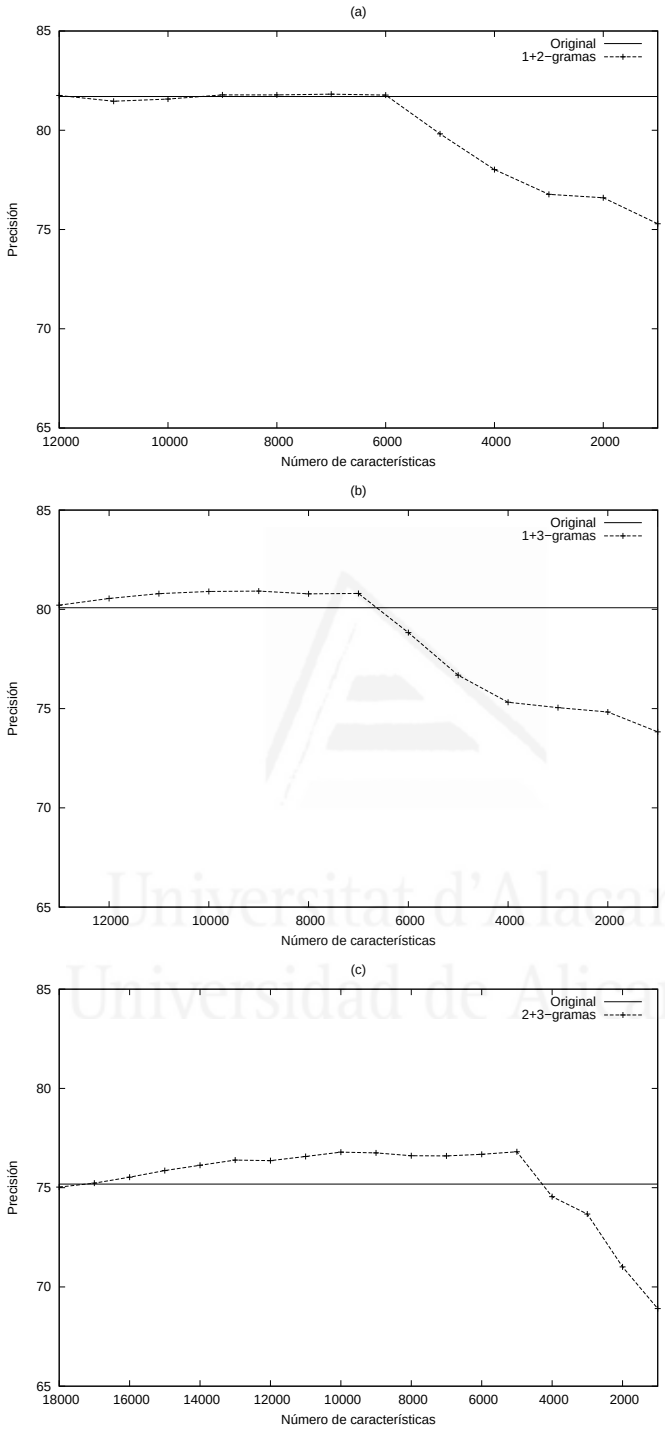

Figura 4.6: Precisión obtenida con χ^2 para distintas reducciones de dimensionalidad en inglés para (a) 1+2-gramas, (b) 1+3-gramas y (c) 2+3-gramas. *Original* representa la precisión obtenida en el experimento original sin reducción.

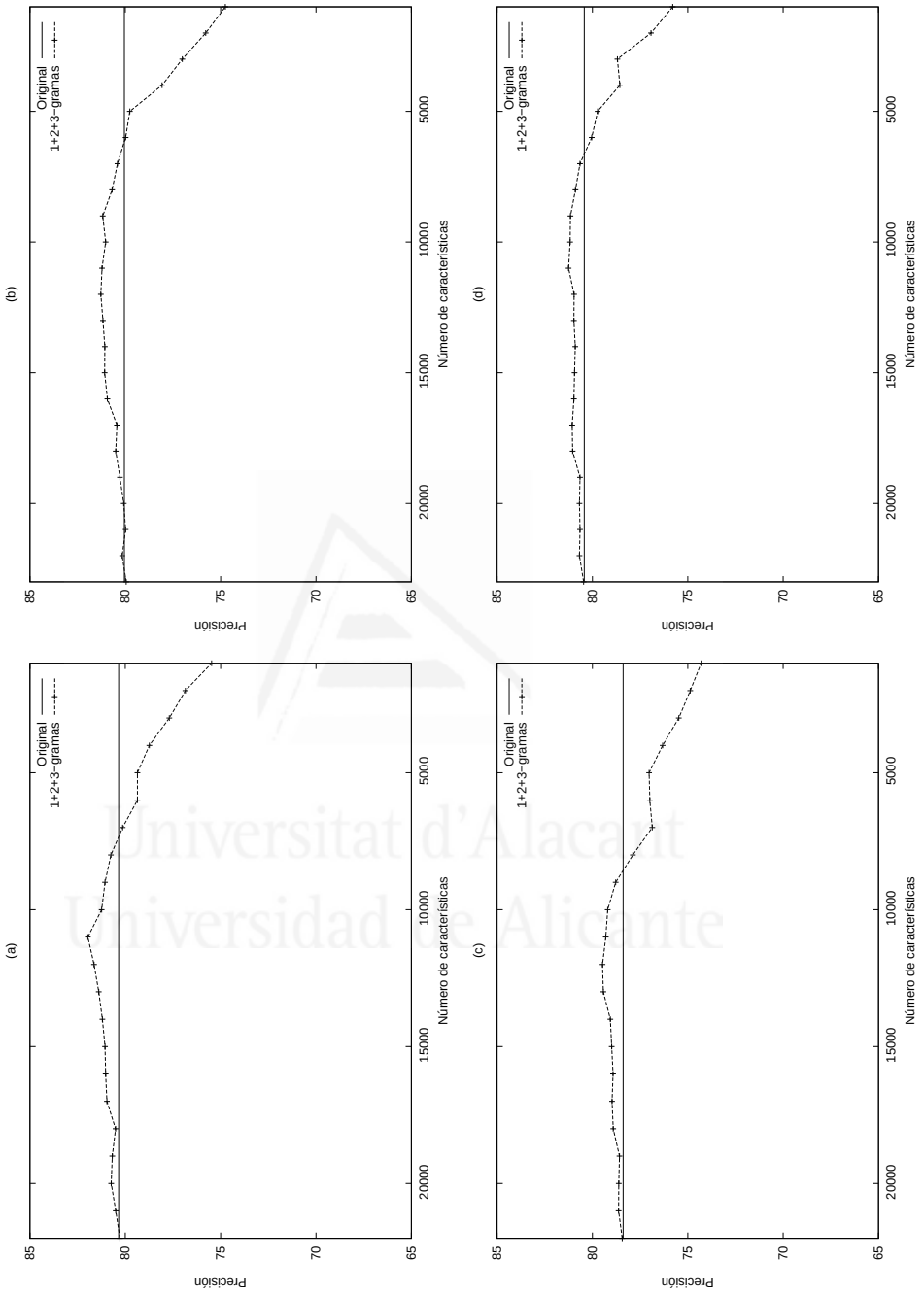

Figura 4.7: Precisión obtenida con χ^2 para distintas reducciones de dimensionalidad para la combinación 1+2+3-gramas en (a) inglés, (b) español, (c) italiano y (d) catalán. *Original* representa la precisión obtenida en el experimento original sin reducción.

obtenidos para la predicción de clases conociendo la presencia o la ausencia de una característica en una instancia.

Para poder definir esta técnica de forma precisa hay que especificar previamente el concepto de *entropía*, una medida comúnmente empleada en teoría de la información. La entropía caracteriza la cantidad de ruido presente en una colección arbitraria de ejemplos, dando una medida de la incertidumbre asociada con una variable aleatoria. Dado un corpus S, definimos la entropía de dicho conjunto de datos $E(S)$ como

$$E(S) = \sum_{i=1}^{n} -p_i log_2 p_i,$$

donde p_i es la proporción de S que pertenece a la clase i, mientras que n representa el número total de clases posibles. A partir de la entropía podemos definir una medida de la efectividad de una característica a la hora de clasificar los datos de entrenamiento. Esta medida, conocida como IG, nos da una idea de la reducción de entropía esperada cuando se particionan los ejemplos en función de una determinada característica. De forma más precisa, el valor $IG(S, c)$ de una determinada característica c con relación a una colección de ejemplos S viene dado por

$$IG(S, c) = E(S) - \sum_{v \in \|c\|} \frac{\|S_v\|}{\|S\|} E(S_v),$$

donde $\|c\|$ representa todos los posibles valores que puede tomar la característica c y S_v es el subconjunto de S para el cual la característica c tiene el valor v (es decir, $S_v = \{s \in S \mid c(s) = v\}$). Al igual que sucedía con χ^2, vamos a emplear IG para decidir qué características son más relevantes para la clasificación. Computaremos IG para cada una de las características, ordenando la lista resultante y eligiendo las m características mejor valoradas.

Tras repetir los experimentos realizados con χ^2, el comportamiento ofrecido por IG resulta prácticamente equivalente al método anterior. Como ejemplo, la figura 4.8 muestra la precisión obtenida por el sistema sobre el corpus en inglés empleando la combinación 1+2+3-gramas. Se observa un máximo para m=11.000, obteniendo una precisión de 81,96 %, significativamente mayor ($p < 0,01$) que el experimento original. Se puede observar que la gráfica obtenida con IG es prácticamente ientica a la obtenida con χ^2 en la figura 4.7 (a), solapándose para la mayoría de valores de m. En ningún caso existe una diferencia estadísticamente significativa.

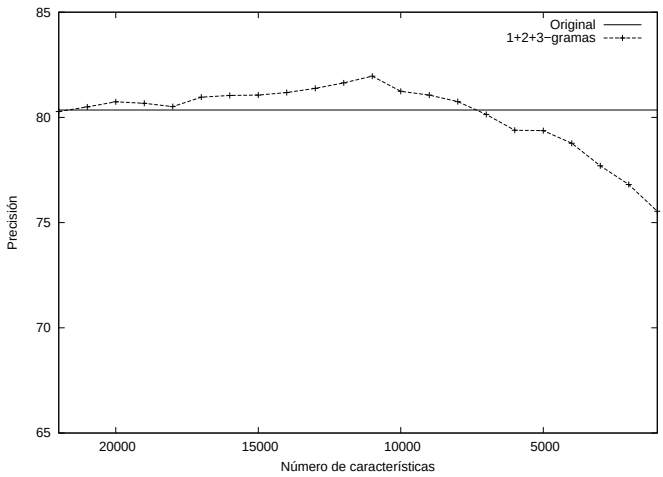

Figura 4.8: Precisión obtenida con IG para distintas reducciones de dimensionalidad en inglés para la combinación de 1+2+3-gramas. *Original* representa la precisión obtenida en el experimento original sin reducción.

4.6. Conclusiones

En este capítulo hemos presentado una primera aproximación a la tarea de CP basada en corpus. El objetivo era construir un sistema fácilmente adaptable a diferentes idiomas y dominios. Por esta razón, el sistema está basado únicamente en n-gramas extraidos del propio corpus de aprendizaje. De esta forma evitamos el uso de herramientas y recursos externos que liguen el sistema a un determinado idioma o dominio.

Hemos empleado dos corpus de preguntas para poder evaluar adecuadamente el sistema en diferentes contextos. El primero de ellos, el corpus TREC, consta de 2.393 preguntas en dominio abierto sobre cuatro idiomas: inglés, español, italiano y catalán. Este corpus fue desarrollado para evaluar la adaptabilidad del sistema de CP a diferentes idiomas. El segundo corpus, el corpus QALL-ME, se compone de 4.500 preguntas en inglés y castellano sobre un dominio restringido (turismo). Este corpus se empleó para evaluar la adaptabilidad del sistema a diferentes dominios.

Como características de aprendizaje hemos experimentado con distintos tamaños de n-gramas y sus combinaciones: 1-gramas, 2-gramas, 3-gramas, 1+2-gramas, 1+3-gramas, 2+3-gramas y 1+2+3-gramas.

Para construir nuestro sistema hemos empleado SVM como algoritmo de aprendizaje. Hay motivos bien fundados para la toma de esta decisión. En primer lugar, existen evidencias empíricas derivadas de comparativas previas realizadas en el campo de la CP (Bisbal et al., 2005a; Sundblad, 2007; Zhang y Lee, 2003). Estas investigaciones revelan que SVM funciona mejor para esta tarea que algoritmos como NB, k-NN, árboles de decisión, SNoW o

ME. Otros motivos para seleccionar SVM son su tolerancia a la presencia de atributos redundantes e irrelevantes, y su capacidad para manejar espacios con un gran número de características evitando el problema del sobreajuste. Tras una experimentación previa decidimos utilizar un kernel lineal que demostró funcionar mejor que los kernels polinómico, gaussiano y RBF.

En los experimentos llevados a cabo en el corpus TREC, los 1-gramas demostraron un mejor funcionamiento que los n-gramas de mayor tamaño. Esto revela que, aunque los n-gramas de mayor tamaño ofrecen una mejor representación de la estructura de la pregunta, sus cualidades estadísticas son inferiores a los 1-gramas (aparecen de forma más dispersa), por lo que resultan menos atractivos desde el punto de vista del aprendizaje automático. Sólo la combinación 1+2-gramas mejoró la representación básica mediante 1-gramas. En cualquier caso, la diferencia de rendimiento no resultó significativa. A la hora de construir un sistema final de CP sería más ventajoso emplear únicamente 1-gramas para el aprendizaje. En este caso el vector de características sería menor que para 1+2-gramas, siendo inferior el coste computacional de llevar a cabo el entrenamiento y la evaluación. Por otra parte, el rendimiento de las distintas características fue muy similar en todos los idiomas estudiados, existiendo pequeñas diferencias achacables al grado de flexión verbal de los idiomas estudiados. La representación del espacio de características mediante n-gramas resulta apropiada para las aproximaciones multilingües, demostrando un comportamiento homogéneo para todos los idiomas.

Los experimentos realizados sobre el corpus QALL-ME nos han permitido comparar el rendimiento del sistema en dominio restringido. Los resultados obtenidos con diferentes vectores de características son extremadamente buenos, situándose entre un 92,19 % para 3-gramas y un 95,46 % para 1+2-gramas en inglés. Resultados muy similares, incluso ligeramente superiores, se obtuvieron para español. Estos resultados son consecuencia de la menor variación lingüística de las preguntas en dominio restringido. El resultado es un menor tamaño de los vectores de aprendizaje, una menor dispersión de los datos y una mejor estimación estadística en todos los tamaños de n-gramas que facilitan la clasificación. Los resultados reflejan que se puede construir un sistema de CP de alto rendimiento en dominios restringidos basándonos únicamente en los n-gramas de las preguntas.

Para intentar mejorar el rendimiento del clasificador experimentamos con diversas técnicas estadísticas de selección de características. Estos métodos permiten reducir la dimensionalidad del espacio de aprendizaje y el coste computacional de la tarea. Habitualmente, esta reducción conlleva la eliminación de características que introducen ruido en la clasificación, mejorando de esta forma la precisión del clasificador. La selección de características que hemos llevado a cabo se realiza de forma automática mediante estimaciones estadísticas sobre el corpus de aprendizaje. De esta forma se evita tener que determinar manualmente cuáles son las

características más relevantes para la clasificación. Aunque su aplicación en otras tareas de clasificación es sobradamente conocida, no existen estudios previos sobre la aplicación de estas técnicas de selección para la tarea de CP.

La primera técnica de selección estudiada consiste en establecer un umbral de frecuencia que elimine las características que aparecen de forma limitada en el corpus. Concretamente, se estableció el umbral para eliminar aquellos n-gramas que aparecieran una única vez en el conjunto de preguntas (*hapax legomena*). Pese a su simplicidad, esta técnica consigue reducir considerablemente la dimensionalidad del espacio nativo de características, mejorando el rendimiento del sistema para todos los vectores de características, con excepción de los 1-gramas. Por ejemplo, el mejor resultado obtenido para inglés (81,74 % para 1+2-gramas) son mejores que el original (81,70 %) aunque esta diferencia no es estadísticamente significativa. En cualquier caso, esta forma de selección consigue descartar el 76.82 % de las características del vector original. Esta reducción implica un menor coste computacional tanto en el entrenamiento como en la clasificación.

Hemos empleado otras dos técnicas de selección, χ^2 e IG, más sofisticadas que la anterior. Los experimentos realizados demuestran que ambas obtienen resultados muy similares, mejorando los valores obtenidos con la técnica del umbral de frecuencia. Los resultados fueron especialmente buenos para la combinación 1+2+3-gramas (81,96 %). Obtuvimos una mejora significativa para todos los idiomas con respecto a los experimentos originales, reduciendo el tamaño del vector de características a menos de la mitad del inicial. La mejora obtenida sugiere que estos métodos de selección consiguen efectivamente mantener los mejores n-gramas para la clasificación mejorando el rendimiento final del sistema. Pese a que los mejores resultados con IG y χ^2 (81.96 % en inglés) superan a los mejores resultados originales (81.70 %), esta diferencia no es significativa. Como ya mencionamos más arriba, la selección de características proporciona dos beneficios. En primer lugar, elimina características ruidosas y mejora el rendimiento del sistema. Aunque esta mejora es patente en los experimentos realizados no es estadísticamente significativa con respecto a los resultados originales. En segundo lugar, la selección de características reduce el número de características de aprendizaje y el coste computacional del entrenamiento y evaluación del clasificador. A diferencia del umbral de frecuencia, IG y χ^2 son procesos costosos, por lo que debe valorarse cuándo el coste de llevar a cabo la selección compensa la reducción de coste que se produce durante el aprendizaje y la clasificación.

5

Clasificación de preguntas semisupervisada explotando textos no etiquetados

En el capítulo anterior desarrollamos un sistema de CP basado en n-gramas. Estas características demostraron un rendimiento muy similar para todos los idiomas tratados. Los resultados revelaron que se puede conseguir una enorme efectividad en dominio restringido (95,46 % en inglés y 95,65 % en español). Aunque los resultados en dominio abierto fueron buenos (81,96 % en inglés, 81,28 % en español, 79,47 % en italiano y 81,25 % en catalán) existe margen para la mejora debido a que la aproximación basada en n-gramas es muy sensible al problema de la variación y ambigüedad del lenguaje. Estos problemas fueron descritos en la sección 2.2.3.

La variación del lenguaje hace referencia a la posibilidad que proporciona el lenguaje natural para utilizar dos o más formas equivalentes, o casi equivalentes, para expresar una información o conseguir un objetivo comunicativo determinado. Es esta variación del lenguaje la que provoca que la misma pregunta pueda formularse de maneras muy diferentes. En la figura 5.1 (a) puede verse un ejemplo de esta situación. Más aún, podemos encontrar preguntas semánticamente similares que son completamente diferentes desde el punto de vista léxico. Por ejemplo, la figura 5.1 (b) muestra tres preguntas que esperan como respuesta un nombre de *animal*. Estas tres preguntas no tienen ningún término en común. Si hiciéramos una representación de estas preguntas en el *modelo de espacio vectorial*[1] o *vector space model* (VSM) (Salton et al., 1975), la similitud[2] obtenida al compararlas entre sí sería 0.

Por otra parte, el problema de la ambigüedad surge cuando una palabra tiene múltiples significados diferentes. En las preguntas de la figura 5.1 (c), la palabra "*country*" presenta dos significados completamente diferentes, el primero refiriéndose a una localización y el segundo refiriéndose a un

[1]Se conoce como *modelo de espacio vectorial* a un modelo algebraico para la representación de documentos textuales en forma de vectores de identificadores. Su utilización se extiende al filtrado, recuperación e indexado de información, permitiendo calcular su relevancia. Fue empleado por primera vez para la tarea de RI en el sistema SMART (Buckley, 1985).

[2]Una forma de obtener la similitud entre dos textos en el VSM es calculando el coseno entre los vectores de identificadores que los representan.

(a)

q_1: What was the name of The Muppets creator?
q_2: Name the creator of The Muppets?

(b)

q_3: What was the first domesticated feline?
q_4: Name a tiger that is extinct?
q_5: Which mammal lives, breeds, eats and sleeps underground?

(c)

q_6: What is the smallest country in Africa?
q_7: What country and western singer is known as The Silver Fox?

Figura 5.1: Ejemplos de preguntas extraidas del corpus UIUC.

estilo musical. En el modelo de representación del VSM, la similitud entre las dos preguntas podría ser considerada incorrectamente alta debido a la ambigüedad de la palabra "*country*".

La representación basada en n-gramas que aplicábamos en el capítulo anterior es incapaz de manejar este tipo de problemas de variación y ambigüedad lingüística. Por esta razón, la tarea de CP puede beneficiarse de forma clara de una representación más rica del espacio de características. Esto ha propiciado que muchas aproximaciones a la tarea de CP utilicen herramientas y recursos para incorporar información lingüística durante el proceso de aprendizaje y clasificación. En el trabajo desarrollado por Li y Roth (2002, 2005) se demuestra la importancia de incluir información semántica durante el aprendizaje. La mejora proporcionada por este tipo de información claramente sobrepasaba a las mejora obtenidas empleando información sintáctica en los experimentos que llevaron a cabo.

En este capítulo nos vamos a centrar en mejorar el rendimiento del sistema de CP incorporando información semántica sobre el modelo de representación basado en n-gramas. Existen diversos recursos utilizados de forma habitual por los sistemas de clasificación para incorporar información semántica al espacio de características:

- Listas manuales de términos relacionados semánticamente (Kocik, 2004; Li y Roth, 2002, 2005; Pan et al., 2008), listas de palabras "buenas" (Greenwood, 2005) y listas de interrogativos (Metzler y Croft, 2005).

- Etiquetado de entidades (Blunsom et al., 2006; Brown, 2004; García Cumbreras et al., 2006; Hacioglu y Ward, 2003; Kocik, 2004; Li y Roth, 2002; Nguyen et al., 2008; Pan et al., 2008; Pinchak y Lin, 2006; Suzuki et al., 2003b).

- Bases de datos léxicas como WordNet, EuroWordNet, GoiTaikei y tesauros como Roget (García Cumbreras et al., 2006; Kocik, 2004; Krishnan et al., 2005; Li et al., 2005; Pan et al., 2008; Suzuki et al., 2003b).

- Bases de datos específicas de nombres geográficos (Blunsom et al., 2006; Schlobach et al., 2004).

- Diccionarios de sinónimos como TongYiCi CiLin (Day et al., 2007).

Todos estos recursos tienen un punto en común: están ligados al idioma para el que fueron concebidos. Además, la mayoría de ellos fueron adquiridos de forma manual a expensas del conocimiento de expertos humanos. Esto provoca que su disponibilidad no sea inmediata en cualquier idioma. Su utilización, por tanto, entra en conflicto con las premisas establecidas al comienzo de esta tesis.

A fin de incorporar información semántica en nuestro sistema, sin comprometer por ello la flexibilidad de nuestro clasificador, hemos desarrollado una propuesta para inducir la información semántica a partir de texto no etiquetado. Nuestra propuesta se basa en el *análisis de la semántica latente* (Deerwester et al., 1990), una técnica ampliamente estudiada en diversas tareas de PLN. Vamos a utilizar esta técnica estadística para incorporar información semántica a partir de texto plano obtenido de la Web. De esta forma, nuestra aproximación no va a depender de ningún recurso o anotación manual, permitiendo obtener información semántica sin comprometer la adaptabilidad de nuestro sistema.

Para incorporar la información semántica en nuestro sistema seguiremos una aproximación semisupervisada basada en kernels (Schölkopf y Smola, 2001). Vamos a extender la representación tradicional basada en bolsa de palabras ofreciendo una forma efectiva de integrar información semántica externa en el proceso de CP mediante *kernels semánticos* (Cristianini y Shawe-Taylor, 2000). Emplearemos una función kernel basada en información semántica latente obtenida a partir de texto sin etiquetar, incorporando este kernel en el clasificador SVM. Nuestra propuesta permite inducir de forma automática relaciones semánticas a partir de texto no etiquetado. De esta forma afrontamos no sólo el problema de la variación del lenguaje, sino también el problema de la ambigüedad, ya que estos kernels han demostrado obtener muy buenos resultados en la tarea de desambiguación del sentido de las palabras (Gliozzo et al., 2005). Combinaremos esta información con una

aproximación básica basada en bolsa de palabras, usando para ello kernels compuestos (*composite kernels*).

Una de las ventajas de nuestra aproximación, cuando se compara con otros métodos tradicionales de aprendizaje semisupervisado (Nguyen et al., 2008), es que en nuestro caso no es necesaria una colección de preguntas sin etiquetar, que habitualmente son difíciles de obtener. En nuestro caso empleamos un corpus genérico de textos (concretamente documentos de la Wikipedia[3]) para mejorar la parte supervisada del algoritmo. El resultado es un sistema fácilmente adaptable a diferentes idiomas y dominios. Haremos una evaluación en inglés y español para corroborar esta afirmación.

Vamos a describir a continuación los fundamentos de LSA. Posteriormente mostraremos como incluir esta información en nuestro clasificador mediante una función kernel.

5.1. Análisis de la semántica latente

El *análisis de la semántica latente* o *latent semantic analysis* (LSA) es una teoría y un método para extraer y representar el significado contextual de las palabras mediante computación estadística aplicada a un corpus amplio de textos (Landauer y Dumais, 1997; Landauer et al., 1998). La idea subyacente es que el conjunto de todos los contextos en los que una palabra aparece o no, proporciona un conjunto de restricciones mutuas que determinan ampliamente la similitud de significado de palabras y conjuntos de palabras entre sí. LSA representa el significado de una palabra como un promedio del significado de todos los textos en los que aparece, y el significado de un texto como un promedio de todas las palabras que contiene.

LSA es una técnica matemática/estadística totalmente automática para extraer e inferir relaciones sobre el uso contextual esperado para las palabras del discurso. No es una técnica tradicional de PLN o de IA. No usa diccionarios construidos manualmente, bases de datos, redes semánticas, gramáticas, analizadores sintácticos o morfologías. La única entrada que requiere es texto plano fragmentado en palabras. LSA emplea como datos iniciales no sólo las coocurrencias de palabras contiguas, sino patrones detallados de ocurrencia de muchas palabras sobre un número muy grande de contextos de significado locales (como frases, párrafos o documentos) tratados como un todo unitario. De esta manera, no se basa en cómo el orden de las palabras produce el significado de una oración, sino que captura cómo las diferencias en la elección de las palabras y la diferencia en el significado de las palabras están relacionadas. Es importante dejar constancia de que las estimaciones de similitud derivadas mediante LSA no son simples frecuencias de aparición, número de coocurrencias o correlación en el uso, sino que dependen de un potente análisis matemático que es capaz

[3] http://www.wikipedia.org.

de inferir de forma correcta relaciones más profundas (de ahí la noción de *semántica latente*). Como consecuencia producen una mejor predicción de los juicios de significado humanos que los que se obtienen de las contingencias superficiales mediante modelos de n-gramas.

Como método práctico para la caracterización del significado de las palabras, LSA produce medidas de la relación entre *palabra-palabra*, *palabra-documento* y *documento-documento*. Las representaciones del significado de palabras y documentos derivadas mediante LSA han demostrado ser capaces de simular una amplia variedad de fenómenos cognitivos humanos, desde la adquisición de vocabulario a la clasificación de palabras o la comprensión del discurso. Uno de los ejemplos más sobresalientes de LSA es su aplicación a recuperación de información[4] (Dumais, 1994). En esta tarea, LSA ha demostrado mejorar significativamente la recuperación automática de información, permitiendo que las peticiones de información de los usuarios encuentren texto relevante incluso cuando éste no contiene ninguno de los términos usados en la consulta.

El primer paso para la aplicación de LSA es representar el texto como una matriz $\mathbf{M}$, en la que cada fila representa un único término y cada columna representa un documento u otro contexto, como pasajes o preguntas. Las celdas de esta matriz expresan la importancia de la palabra en el contexto particular (peso local) o en el dominio de discurso en general (peso global). Para expresar esta importancia es habitual emplear el modelo booleano (si el término aparece o no en el contexto), la frecuencia del término (número de veces que aparece el término en el contexto), *tf-idf* o valores de entropía.

Una vez creada la matriz, el siguiente paso consiste en aplicar una *descomposición en valores singulares* o *singular value decomposition* (SVD) de la matriz para construir un espacio vectorial semántico que representa asociaciones conceptuales entre términos y documentos. Expresado de manera formal, dado un corpus de m documentos y n términos diferentes, se obtiene una matriz $\mathbf{M}$ de tamaño $n \times m$ y rango r, cuya descomposición SVD expresa dicha matriz como el producto de tres matrices

$$\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T, \tag{5.1}$$

donde $\mathbf{\Sigma}$ es una matriz diagonal $r \times r$ que contiene los valores singulares de $\mathbf{M}$, $\mathbf{U}$ es una matriz ortogonal[5] de tamaño $n \times r$ que describe los términos como los vectores singulares izquierdos, y $\mathbf{V}$ es una matriz ortogonal de tamaño $r \times m$ que describe los documentos como los vectores singulares derechos. Cuando estas tres matrices se multiplican se obtiene la matriz original.[6]

[4] Cuando LSA se aplica a la tarea de RI se conoce con el nombre de *latent semantic indexing* (LSI).

[5] $\mathbf{U}^T\mathbf{U} = \mathbf{I}$.

[6] Hay una prueba matemática de que cualquier matriz puede ser descompuesta perfectamente usando no más factores que la menor dimensión de la matriz original.

Una vez realizada la descomposición en valores singulares, LSA propone considerar únicamente los k valores singulares mayores de la descomposición anterior, teniendo como objetivo el reducir la dimensionalidad del espacio de trabajo. LSA asume que la elección de la dimensionalidad en la cual todos los contextos locales de las palabras son simultáneamente representados puede ser de gran importancia, y que reducir la dimensionalidad (el número de parámetros por el cual una palabra o un documento se describe) de los datos observados desde el número de contextos iniciales hacia un número mucho más pequeño, producirá mejores aproximaciones a las relaciones cognitivas humanas.

Es este paso de reducción de la dimensionalidad, la combinación de información superficial en una abstracción más profunda, el que captura las implicaciones mutuas de las palabras y los documentos. De esta forma, un componente importante de la aplicación de esta técnica es encontrar la dimensionalidad óptima para la representación final. El mecanismo que hace que LSA resuelva el problema del aprendizaje inductivo, por el cual la gente adquiere mucho más conocimiento del que parece estar disponible en la experiencia, consiste simplemente en acomodar un número muy grande de relaciones coocurrentes de forma local simultáneamente en un espacio de la dimensionalidad correcta.

Se busca, pues, obtener una nueva matriz del mismo tamaño que la matriz original, pero reduciendo la dimensionalidad del espacio sobre el cual se encontraba definida la matriz $\mathbf{M}$. La nueva matriz será una proyección de la original sobre un espacio de inferior dimensionalidad, que produce el efecto de añadir a los documentos términos semánticamente relacionados que no aparecían en los textos originales sobre los que se construyó la matriz $\mathbf{M}$. Determinar el número de valores singulares a considerar no es una tarea inmediata, y supone la experimentación con el corpus sobre el que se aplique LSA con el fin de obtener resultados óptimos que reflejen correctamente las características intrínsecas y la semántica del corpus de trabajo. Definimos $\mathbf{M}_k$ como la matriz de rango k que LSA interpreta como una aproximación de $\mathbf{M}$ donde los vectores columna (los documentos) de la matriz original son proyectados en el espacio de dimensión k. Formalmente:

$$\mathbf{M}_k = \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{V}_k^T, \tag{5.2}$$

donde $\mathbf{U}_k$ y $\mathbf{V}_k$ contiene las primeras k columnas de las matrices $\mathbf{U}$ y $\mathbf{V}$ respectivamente, mientras que $\mathbf{\Sigma}_k$ contiene los k mayores valores singulares. Cuando se utilizan menor número de factores de los necesarios, la matriz reconstruida es la que mejor ajusta a los mínimos cuadrados. Se puede reducir la dimensionalidad de la solución simplemente borrando coeficientes en la matriz diagonal, comenzando normalmente con los más pequeños. SVD se emplea para condensar la matriz original en una más pequeña, típicamente de entre 100 y 500 dimensiones (Deerwester et al., 1990).

La similitud entre dos términos en el nuevo espacio semántico se puede calcular como el producto escalar de los vectores fila de dichos términos en $\mathbf{M}_k$. La matriz $\mathbf{M}_k\mathbf{M}_k^T$ es una matriz cuadrada que contiene todos los productos escalares entre los términos en el nuevo espacio. Ya que $\mathbf{\Sigma}_k$ es diagonal y $\mathbf{V}_k$ es ortonormal, se verifica que

$$\mathbf{M}_k\mathbf{M}_k^T = \mathbf{U}_k\mathbf{\Sigma}_k^2\mathbf{U}_k^T, \tag{5.3}$$

donde las filas de $\mathbf{U}_k\mathbf{\Sigma}_k$ pueden emplearse como *vectores término*. La similitud entre dos documentos se puede comparar de manera parecida, aunque en este caso se calcula el producto escalar entre dos vectores columna de la matriz $\mathbf{M}_k$, que indica hasta qué punto dos documentos contienen un perfil de términos similar. La matriz $\mathbf{M}_k^T\mathbf{M}_k$ contiene los productos escalares entre todos los documentos. Se verifica que

$$\mathbf{M}_k^T\mathbf{M}_k = \mathbf{V}_k\mathbf{\Sigma}_k^2\mathbf{V}_k^T, \tag{5.4}$$

donde las filas de $\mathbf{V}_k\mathbf{\Sigma}_k$ pueden ser empleadas como los *vectores documento*. La medida del coseno o la distancia euclídea entre los vectores término o los vectores documento corresponde a su similitud estimada en el nuevo espacio.

Un nuevo vector de términos puede ser representado como un *pseudo-documento*, es decir, como una suma ponderada de sus vectores término en el nuevo espacio vectorial. Por ejemplo, el vector $\mathbf{q}_k$ definido como

$$\mathbf{q}_k = \mathbf{q}^T\mathbf{U}_k\mathbf{\Sigma}_k^{-1}, \tag{5.5}$$

es una representación del vector $\mathbf{q}$ en el nuevo espacio semántico.

5.2. Kernels para la clasificación de preguntas

En la sección 3.4.1 introdujimos el concepto de kernel, una aproximación popular de aprendizaje automático dentro de la comunidad de PLN. Veíamos como numerosas aproximaciones a la tarea de CP se basaban en la definición de kernels para la inclusión de información sintáctica y/o semántica en el clasificador. La estrategia seguida por las aproximaciones basadas en kernels consiste en dividir el problema de aprendizaje en dos partes. La primera traslada los datos de entrada a un espacio de características adecuado, mientras que la segunda usa un algoritmo lineal (como SVM o perceptron) para descubrir patrones no lineales en el espacio de características transformado. Esta transformación del espacio se lleva a cabo de forma implícita mediante una función kernel. Estas funciones proporcionan una medida de similitud entre los datos de entrada que depende exclusivamente del tipo de datos específico de entrada y del

dominio. Una función de similitud típica es el *producto escalar* entre vectores de características.

Un kernel es una función $k : X \times X \rightarrow \mathbb{R}$ que toma como entrada dos objetos (por ejemplo, vectores, textos, árboles sintácticos o preguntas como en nuestro caso) y obtiene como salida un número real caracterizando su similitud, con la propiedad de que la función es simétrica. Para todo vector de características $\mathbf{x}_i$ y $\mathbf{x}_j \in X$, definimos un kernel como

$$k(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle \tag{5.6}$$

donde $\phi : X \rightarrow \mathcal{F}$ es una proyección explícita de X (el espacio de características original) a un nuevo espacio de características $\mathcal{F}$.

En las siguientes secciones definimos y combinamos diferentes funciones kernel que calculan la similitud de las preguntas entre sí. Éstas funciones son el único elemento específico del dominio de nuestro sistema de CP, mientras que el algoritmo de aprendizaje es un componente de propósito general. Potencialmente, cualquier función kernel puede trabajar con cualquier algoritmo basado en kernels, como SVM o perceptron. Al igual que en nuestra primera aproximación, emplearemos SVM para la incorporación de estas funciones al clasificador.

5.2.1. Kernel bolsa de palabras

El método más simple para estimar la similitud entre dos preguntas es calcular el producto escalar de su representación vectorial en el VSM. Formalmente, definimos un espacio de dimensionalidad N, $\mathbb{R}^N$, en el que cada dimensión se asocia con una palabra del vocabulario, y la pregunta $\mathbf{q}$ es representada por un vector fila

$$\phi(\mathbf{q}) = (f(t_1, \mathbf{q}), f(t_2, \mathbf{q}), \ldots, f(t_N, \mathbf{q})), \tag{5.7}$$

donde la función $f(t_i, \mathbf{q})$ indica cuándo una palabra particular t_i es empleada en $\mathbf{q}$. Usando esta representación, definimos el kernel *bolsa de palabras* o *bag-of-words* entre dos preguntas como

$$k_{BOW}(\mathbf{q}_i, \mathbf{q}_j) = \langle \phi(\mathbf{q}_i), \phi(\mathbf{q}_j) \rangle = \sum_{l=1}^{N} f(t_l, \mathbf{q}_i), f(t_l, \mathbf{q}_j). \tag{5.8}$$

Este tipo de kernel no afronta adecuadamente el problema de la variación y la ambigüedad del lenguaje. Por ejemplo, pese a que dos preguntas como "¿Cuál es la capital de California?" y "¿En qué ciudad está el aeropuerto McCarren?" esperan como respuesta el mismo tipo semántico (*ciudad*), al no tener palabras en común su similitud sería 0 empleando este kernel. Para afrontar las deficiencias de este tipo de representación, en la siguiente sección introducimos los kernels semánticos y mostramos cómo definir un VSM semántico efectivo utilizando conocimiento externo no etiquetado.

5.2.2. Kernels semánticos

En el contexto de los métodos kernel, la información semántica puede ser integrada considerando transformaciones lineales del tipo $\phi(\mathbf{q})\mathbf{S}$, donde $\mathbf{S}$ es una matriz que puede ser diagonal, cuadrada o, en general, de tamaño $N \times k$ (Shawe-Taylor y Cristianini, 2004). Nos referiremos a esta matriz como la *matriz semántica*. Usando esta transformación, el kernel correspondiente adquiere la forma

$$k(\mathbf{q}_i, \mathbf{q}_j) = \phi(\mathbf{q}_i)\mathbf{S}\mathbf{S}^T\phi(\mathbf{q}_j)^T. \tag{5.9}$$

Diferentes elecciones de la matriz $\mathbf{S}$ darán lugar a diferentes representaciones del espacio de características. Esta matriz puede ser reescrita como $\mathbf{S} = \mathbf{W}\mathbf{P}$, donde $\mathbf{W}$ es una matriz diagonal que indica el peso o relevancia de los términos, mientras que $\mathbf{P}$ es la matriz de proximidad de palabras (*word proximity matrix*) que captura las relaciones semánticas entre los términos en el corpus.

La matriz de proximidad $\mathbf{P}$ puede ser definida incluyendo pesos no nulos entre aquellas palabras cuya relación semántica sea inferida a partir de una fuente externa de conocimiento sobre el dominio. En los siguientes puntos describimos dos aproximaciones alternativas para definir la matriz de proximidad. La primera hace uso de una lista de palabras relacionadas semánticamente. Esta lista se construyó de forma manual y será empleada aquí únicamente con propósito de comparación experimental. La segunda propuesta hace uso de datos no etiquetados y representa la principal contribución de este capítulo.

Kernel semántico explícito

Las listas construidas a mano de palabras relacionadas semánticamente proporcionan una forma simple y efectiva de introducir información semántica en el kernel. Para definir un kernel semántico a partir de estos recursos, podemos construir explícitamente la matriz de proximidad $\mathbf{P}$ fijando sus entradas para reflejar la proximidad semántica entre los términos t_i y t_j en el recurso léxico específico.

En nuestro caso hemos definido $\mathbf{P}$ empleando las listas de palabras semánticamente relacionadas construidas manualmente por Li y Roth (2002).[7] Para la construcción de estas listas, los autores analizaron cerca de 5.500 preguntas y extrajeron manualmente una serie de palabras relacionadas con cada una de las clases semánticas empleadas en sus experimentos. En este sentido, esta fuente de información es similar a las empleadas en algunas aproximaciones previas (Harabagiu et al., 2000; Hermjakob, 2001). En la figura 3.7 veíamos un ejemplo de estas listas de palabras.

[7] Disponibles en http://l2r.cs.uiuc.edu/~cogcomp/Data/QA/QC/.

	city	*location*	*people*	*sport*	...
town	1	1	0	0	
capital	1	0	0	0	
champions	0	0	1	1	
artists	0	0	1	0	
⋮					⋱

Figura 5.2: Un fragmento de la matriz de proximidad definida usando listas de palabras construidas manualmente y relacionadas con una clase semántica.

La figura 5.2 muestra un fragmento de la matriz semántica resultante $\mathbf{P}$ cuyas filas están indexadas por las palabras y cuyas columnas están indexadas por la clase semántica. La entrada (i, j)-ésima es 1 si la palabra w_i está contenida en la lista l_j o 0 en caso contrario. El kernel correspondiente, que llamaremos *kernel semántico explícito*, se define como

$$k_{SE}(\mathbf{q}_i, \mathbf{q}_j) = \phi(\mathbf{q}_i)\mathbf{P}\mathbf{P}^T\phi(\mathbf{q}_j)^T, \tag{5.10}$$

donde ϕ es la proyección definida en la ecuación 5.7.[8]

Estas listas son muy útiles para superar el problema de la variación del lenguaje. Sin embargo, no pueden afrontar el problema de la ambigüedad, ya que estas características se obtienen a partir de términos tratados de forma individual sin tener en cuenta el contexto en el que aparecen.

Kernel semántico latente

Una aproximación alternativa para definir la matriz de proximidad es mirar a la coocurrencia de información en un corpus de gran tamaño. Para ello emplearemos la representación basada en LSA, que nos permite considerar que dos palabras están semánticamente relacionadas si coocurren de forma frecuente en los mismos contextos. Esta segunda aproximación es más atractiva que la anterior basada en listas manuales, ya que nos permite definir de forma automática modelos semánticos para diferentes lenguajes y dominios.

Por ejemplo, pese al hecho que las preguntas mostradas en la figura 5.1 (b) pertenecen a la misma clase semántica (*animal*), su comparación en el estándar VSM es 0 al no tener ningún término en común. Por contra, en el espacio LSA su similitud es mayor que 0 ya que las palabras "*mammal*", "*feline*" y "*tiger*" coocurren frecuentemente en los mismos textos.

Vamos a emplear SVD para definir automáticamente la matriz de proximidad $\mathbf{P}$ a partir de textos de la Wikipedia. La selección de un corpus representativo es una parte importante del proceso de definir un espacio

[8] En este kernel se omite la matriz de pesos, ya que $\mathbf{W} = \mathbf{I}$.

	q_3	q_4	q_5
q_3	1	0,75	0,65
q_4		1	0,73
q_5			1

Figura 5.3: Comparación entre las preguntas de la figura 5.1 (b) en el espacio de semántica latente usando la proyección ρ. Los valores de la diagonal inferior se omiten al tratarse de una comparación simétrica.

semántico. El uso de Wikipedia nos permite definir un modelo estadístico de dominio abierto. La matriz $\mathbf{P}$ será representada a partir de una matriz de términos y documentos $\mathbf{M}$, donde la entrada $\mathbf{M}_{i,j}$ indica la frecuencia del término t_i en el documento d_j. SVD permite descomponer la matriz $\mathbf{M}$ en tres matrices $\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, donde $\mathbf{U}$ y $\mathbf{V}$ son matrices ortogonales cuyas columnas son los vectores propios de $\mathbf{M}\mathbf{M}^T$ y $\mathbf{M}^T\mathbf{M}$ respectivamente, y $\mathbf{\Sigma}$ es la matriz diagonal que contiene los valores singulares de $\mathbf{M}$.

Bajo estas condiciones, definimos la matriz de proximidad $\mathbf{P}$ como

$$\mathbf{P} = \mathbf{U}_k\mathbf{\Sigma}_k^{-1}, \tag{5.11}$$

donde $\mathbf{U}_k$ es la matriz que contiene las primeras k columnas de $\mathbf{U}$ y k es la dimensionalidad del espacio semántico latente y puede ser fijada previamente. Utilizando un pequeño número de dimensiones podemos definir una representación muy compacta de la matriz de proximidad y, consecuentemente, reducir los requerimientos de memoria mientras preservamos la mayoría de la información.

La matriz $\mathbf{P}$ se usa para definir una transformación lineal $\rho : \mathbb{R}^N \rightarrow \mathbb{R}^k$ que transforma el vector $\phi(\mathbf{q})$ del espacio de características de entrada al espacio semántico latente. Formalmente ρ se define como

$$\rho(\phi(\mathbf{q})) = \phi(\mathbf{q})(\mathbf{W}\mathbf{P}), \tag{5.12}$$

donde $\mathbf{W}$ es una matriz diagonal de tamaño $N \times N$ que determina los pesos de las palabras. Las celdas de esta matriz se calculan como $\mathbf{W}_{i,i} = idf(w_i)$, siendo $idf(w_i) = \log \frac{N}{df(w_i)}$ el valor inverso de la frecuencia del documento (idf) de w_i. N representa el número total de documentos en el corpus y $df(w_i)$ el número de documentos en los que aparece w_i. La figura 5.3 muestra un ejemplo de la comparación entre las preguntas de la figura 5.1 (b) en el espacio semántico latente usando la proyección ρ.

Finalmente, el *kernel semántico latente* queda definido como

$$k_{LS}(\mathbf{q}_i, \mathbf{q}_j) = \langle \rho(\phi(\mathbf{q}_i)), \rho(\phi(\mathbf{q}_j)) \rangle, \tag{5.13}$$

donde ϕ es la proyección definida en la ecuación 5.7 y ρ es la transformación lineal definida en la ecuación 5.12.

5.2.3. Kernels compuestos

Podemos emplear la *propiedad de clausura*[9] de las funciones kernel para definir *kernels compuestos* (Briggs y Oates, 2005) que permiten combinar y extender los kernels individuales. Formalmente, definimos un kernel compuesto como

$$k_C(\mathbf{q}_i, \mathbf{q}_j) = \sum_{l=1}^{n} \frac{k_l(\mathbf{q}_i, \mathbf{q}_j)}{\sqrt{k_l(\mathbf{q}_j, \mathbf{q}_j) k_l(\mathbf{q}_i, \mathbf{q}_i)}}, \tag{5.14}$$

donde k_l es un kernel individual válido. Los kernels individuales están normalizados, lo que juega un papel importante a la hora de permitirnos integrar información proveniente de espacios de características heterogéneos. Trabajos recientes han demostrado empíricamente la efectividad de combinar kernels de esta manera, mejorando de forma sistemática el rendimiento de los kernels individuales (Giozzo y Strapparava, 2005; Giuliano et al., 2006; Moschitti, 2004; Zhao y Grishman, 2005).

Para mostrar la efectividad del modelo semántico propuesto, hemos empleado tres kernels compuestos para la tarea de CP que quedan completamente definidos por los n kernels individuales que los componen siguiendo la ecuación anterior:

$k_{BOW} + k_{LS}$ combina la bolsa de palabras (k_{BOW}) con información semántica adquirida automáticamente de la Wikipedia (k_{LS}).

$k_{BOW} + k_{SE}$ combina la bolsa de palabras (k_{BOW}) con la información semántica adquirida a partir de las listas de palabras semánticamente relacionadas (k_{SE}).

$k_{BOW} + k_{LS} + k_{SE}$ combina la bolsa de palabras (k_{BOW}) con la información semántica automáticamente adquirida de la Wikipedia (k_{LS}) y la información semántica adquirida a partir de las listas de palabras semánticamente relacionadas (k_{SE}).

5.3. Experimentos

En esta sección describiremos el marco de evaluación y los resultados obtenidos con nuestra aproximación. El propósito de estos experimentos es evaluar el efecto de los diferentes kernels semánticos en la clasificación, así como la robustez de esta aproximación a la hora de tratar con diferentes idiomas.

[9] Esta propiedad indica que cuando se combinan dos elementos de un conjunto mediante adición o multiplicación, el resultado está también incluido en el conjunto.

5.3.1. Descripción del conjunto de datos

El corpus UIUC, previamente descrito en la sección 3.2, se ha convertido en un estándar *de facto* para la evaluación de los sistemas de CP, siendo utilizado por numerosas aproximaciones a esta tarea. Este hecho se debe a tres factores: estar libremente disponible, tener un conjunto amplio de preguntas de entrenamiento y evaluación, y ofrecer una taxonomía de clases atractiva y desafiante desde el punto de vista de la tarea. Utilizar este corpus nos va a permitir comparar nuestra aproximación con otros trabajos dentro de este área, tal y como veremos en la sección 5.4.

Fue descrito por primera vez por Li y Roth (2002) y contiene un conjunto de entrenamiento en inglés de 5.452 preguntas, complementado con un conjunto de evaluación de 500 preguntas más. Estas preguntas fueron etiquetadas empleando una jerarquía de clases de dos niveles. El primer nivel consta de 6 clases gruesas que son subclasificadas en un segundo nivel de 50 clases finas. Esta jerarquía se ideó con la intención de clasificar preguntas a diferentes niveles de granularidad y dar la posibilidad de aprovechar la naturaleza jerárquica de la taxonomía en el proceso de aprendizaje. En la figura 5.4 puede verse la taxonomía completa.

Hemos evaluado nuestro sistema sobre ambos niveles de granularidad (grueso y fino). La clasificación de grano fino es una piedra de toque para las aproximaciones basadas en aprendizaje automático, ya que el número de muestras por clase se ve drásticamente reducido con respecto a la clasificación gruesa. Es en este marco de trabajo donde podemos realmente evaluar la capacidad de generalización de nuestra propuesta y el efecto de la información inducida por los kernels semánticos. Por otra parte, para medir la capacidad de adaptación a diferentes idiomas de nuestra aproximación hemos traducido a español de forma manual el corpus completo UIUC. De esta forma hemos podido calcular el rendimiento del sistema en inglés y en español.

5.3.2. Configuración de los experimentos

Todos los experimentos fueron llevados a cabo usando el paquete LIBSVM,[10] adaptado para incluir nuestros propios kernels en el algoritmo SVM, y la librería LIBSVDC[11] para llevar a cabo la descomposición SVD.

Para definir las matrices de proximidad en inglés y español realizamos la descomposición SVD utilizando 400 dimensiones[12] ($k = 400$) en la matriz de términos y documentos. Esta matriz se obtuvo a partir de 50.000 páginas elegidas al azar de las páginas en inglés y español (según el idioma del experimento) de la Wikipedia. La elección al azar de estos

[10] http://www.csie.ntu.edu.tw/~cjlin/libsvm/.
[11] http://tedlab.mit.edu/~dr/svdlibc/.
[12] Este valor se obtuvo de forma empírica.

ABBR	*ENTY*
abbreviation	animal
expansion	*body*
DESC	*color*
definition	*creation*
description	*currency*
manner	*disease/medical*
reason	*event*
LOC	*food*
city	*instrument*
country	*language*
mountain	*letter*
other	*other*
state	*plant*
NUM	*product*
code	*religion*
count	*sport*
date	*substance*
distance	*symbol*
moeny	*technique*
order	*term*
other	*vehicle*
percent	*word*
period	*HUM*
speed	*description*
temperature	*group*
size	*individual*
weight	*title*

Figura 5.4: Jerarquía de preguntas de Li y Roth compuesta por 6 clases de primer nivel (gruesas) y 50 clases de segundo nivel (finas).

Kernel	Gruesa	Fina
k_{BOW}	86,40	80,80
k_{LS}	70,40	71,20
$k_{BOW} + k_{LS}$	90,00	83,20
$k_{BOW} + k_{SE}$	89,40	84,00
$k_{BOW} + k_{LS} + k_{SE}$	**90,80**	**85,60**

Tabla 5.1: Resultados de los distintos kernels individuales y compuestos en inglés. Los mejores valores de precisión para la clasificación gruesa y fina se muestran en negrita.

documentos obedece al hecho de querer construir un modelo general para dominio abierto, sin centrarse en ningún tipo concreto de documento. Para los experimentos en inglés se tomó como entrada la versión preprocesada de la Wikipedia liberada por Atserias et al. (2008), sin considerar la anotación sintáctica ni semántica realizada en ese trabajo. Las páginas en español se obtuvieron mediante un sencillo programa que obtiene texto plano de la Wikipedia eliminando de las páginas las etiquetas HTML y la metainformación.

5.3.3. Resultados experimentales

La tabla 5.1 muestra la precisión para los experimentos en inglés. Los resultados obtenidos empleando únicamente el kernel de semántica latente k_{LS} (70,40 % para clasificación gruesa y 71,20 % para fina) evidencian que la información semántica inducida por este kernel no es suficientepara la tarea de CP. La importancia que los pronombres interrogativos y las palabras vacías tienen en CP no pueden ser capturadas con este modelo.

Usando el kernel compuesto $k_{BOW} + k_{LS}$ mejoramos los resultados del kernel de referencia k_{BOW}, alcanzando 90,00 % para clases gruesas y 83,20 % para clases finas. Esta diferencia es estadísticamente significativa ($p < 0{,}01$) tanto para clases gruesas como finas. En este caso, el kernel compuesto permite complementar de forma satisfactoria la información de la bolsa de palabras con el conocimiento semántico obtenido mediante el kernel semántico latente k_{LS}.

Por otra parte, la diferencia entre $k_{BOW} + k_{LS}$ y $k_{BOW} + k_{SE}$ no es estadísticamente significativa para las clases gruesas ni para las finas. Esto significa que la mejora alcanzada con ambos recursos es equivalente. La ventaja de la aproximación con el kernel compuesto $k_{BOW} + k_{LS}$ con respecto a $k_{BOW} + k_{SE}$ es que con el primero no necesitamos ningún recurso etiquetado manualmente.

Finalmente combinamos ambos recursos semánticos en el kernel $k_{BOW} + k_{LS} + k_{SE}$. Este kernel compuesto mejora aún más los resultados obtenidos con los kernels anteriores a un nivel significativo $p < 0{,}05$ para la

Kernel	Gruesa	Fina
k_{BOW}	80,20	73,40
k_{LS}	73,20	58,40
$k_{BOW} + k_{LS}$	**85,00**	**75,60**

Tabla 5.2: Resultados de los distintos kernels individuales y compuestos en español. Los mejores valores de precisión para la clasificación gruesa y fina se muestran en negrita.

clasificación de grano fino, obteniendo una precisión del 85,60 %. Este resultado revela que k_{LS} y k_{SE} capturan diferentes relaciones semánticas y pueden complementarse el uno al otro.

La figura 5.5 muestra las curvas de aprendizaje para los experimentos con el corpus en inglés, tanto para la taxonomía gruesa como para la fina. Para obtener esta curva hemos empleado los 5 subconjuntos del corpus UIUC aportados por los propios creador de este recurso.[13] Estos subconjuntos constan de 1.000, 2.000, 3.000 y 4.000 preguntas del corpus original tomadas al azar, permitiendo evaluar el rendimiento del sistema en función del número de muestras de aprendizaje. El conjunto de evaluación es el original de 500 preguntas. Se puede observar en la gráfica que el uso del kernel compuesto $k_{LS} + k_{BOW}$ incrementa las capacidades de generalización del sistema. De media, este kernel compuesto alcanza la misma precisión que el kernel k_{BOW} con sólo la mitad de los ejemplos de aprendizaje.

Para evaluar el funcionamiento del sistema en diferentes idiomas, repetimos los experimentos previos con el corpus en español. En este caso no construimos el kernel k_{SE}, ya que implicaría la traducción de la lista completa de palabras creadas por Li y Roth. La tabla 5.2 muestra la precisión del clasificador en español.

Estos resultados confirman de nuevo que el kernel compuesto $k_{BOW} + k_{LS}$ (85,00 % para clases gruesas y 75,60 % para clases finas) obtiene una mejora estadísticamente significativa con respecto al kernel k_{BOW} para la clasificación gruesa ($p < 0{,}01$) y fina ($p < 0{,}05$).

En los experimentos en español hemos obtenido de forma generalizada unos resultados más bajos que en sus equivalentes en inglés. Una de las principales causas de esta diferencia se puede achacar a los documentos en español obtenidos de la Wikipedia que se emplearon para construir la matriz de proximidad. Este corpus se obtuvo empleando heurísticas sencillas para la depuración de los documentos, dando como resultado una fuente de información ruidosa.

La figura 5.6 muestra las curvas de aprendizaje para el sistema en español. Comparado con los resultados obtenidos en inglés, se obtiene una mejora inferior en términos de reducción de los datos de entrenamiento para

[13] Están disponibles en http://l2r.cs.uiuc.edu/~cogcomp/Data/QA/QC/.

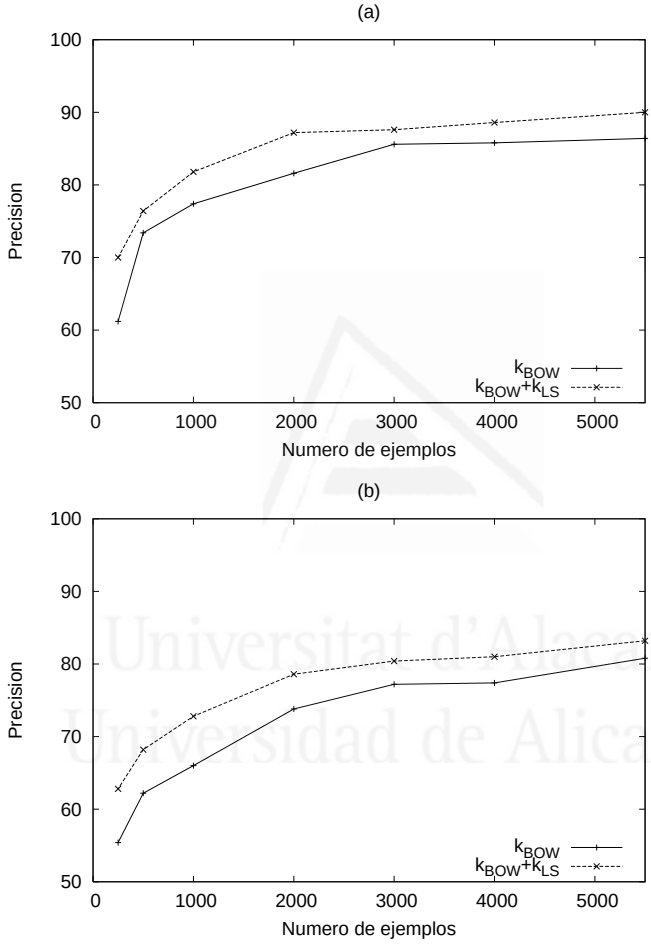

Figura 5.5: Curvas de aprendizaje sobre las clases (a) gruesas y (b) finas para la clasificación en inglés.

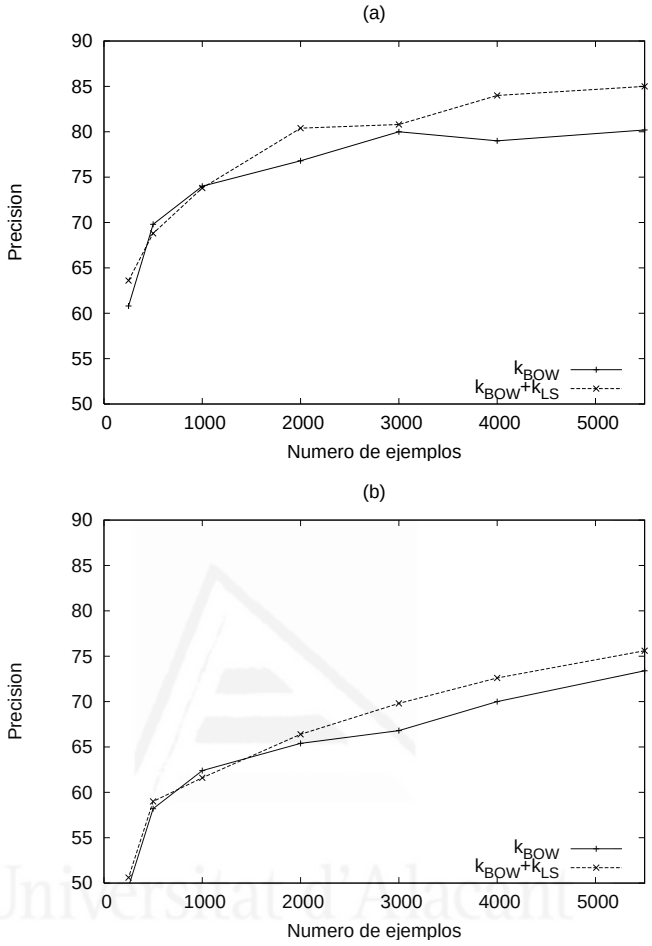

Figura 5.6: Curvas de aprendizaje para clases (a) gruesas y (b) finas para la clasificación en español.

las clases finas. De igual forma, la diferencia entre k_{BOW} y $k_{BOW} + k_{LS}$ es insignificante cuando se emplea un número limitado de ejemplos de entrenamiento.

5.4. Comparación con otros sistemas

Existen diversos trabajos previos que han empleado el corpus UIUC para entrenar y evaluar sistemas de CP, lo que nos permite comparar nuestra aproximación con otros desarrollos. Este corpus fue empleado por primera vez en (Li y Roth, 2002). En este trabajo definieron un clasificador jerárquico basado en SNoW utilizando diversas fuentes para obtener un espacio de características lingüísticamente rico, incluyendo la detección de sintagmas

nominales, reconocimiento de entidades y listas manuales de palabras semánticamente relacionadas. Su sistema alcanzaba un 91 % de precisión para las clases gruesas y un 84,2 % para las clases finas. La diferencia de precisión entre este sistema y el nuestros es mínima y aparentemente no significativa. Nuestra aproximación presenta la ventaja de no necesitar de otras herramientas, como analizadores sintácticos y reconocedores de entidades.

En Zhang y Lee (2003) emplearon *tree kernel* para aprovechar las estructuras sintácticas de las preguntas, obteniendo un 90 % de precisión para clases gruesas. Este kernel no mejoraba los resultados obtenidos empleando n-gramas para las clases finas, concluyendo que el árbol sintáctico no contiene la información requerida para distinguir entre varias categorías finas dentro de una clase gruesa. Nuestro sistema mejora estos resultados, presentando la ventaja adicional de no necesitar de analizadores sintácticos para su funcionamiento.

El trabajo desarrollado en (Hacioglu y Ward, 2003) fue un primer intento de emplear SVD para la reducción de dimensionalidad en CP, aunque no obtuvieron ninguna mejora con esta técnica en sus experimentos. Alcanzaron una precisión de 79,80 % para clases finas con SVD y 2.000 dimensiones, obteniendo peores resultados que la representación original con n-gramas. El problema con estos experimentos fue que aplicarion *análisis de componentes principales* (*principal component analysis*) sobre un pequeño corpus de preguntas para construir el modelo. La mayor diferencia con nuestra aproximación es que ellos construyeron el modelo estadístico usando un corpus muy pequeño de preguntas en lugar de explotar un conjunto grande de documentos sin etiquetar. Aunque el corpus era representativo del problema, su pequeño tamaño comprometió los resultados. Nuestro sistema obtiene una precisión significativamente mayor, demostrando un uso más adecuado de la descomposición SVD.

Otra propuesta que empleó el corpus UIUC fue la desarrollada por Moschitti et al. (2007). En este trabajo obtuvieron 91,80 % de precisión para clases gruesas empleando bolsa de palabras y un *tree kernel* con información sintáctica. No llevaron a cabo ningún experimento con clasificación de grano fino. Si bien el resultado sobre clases gruesas es superior al nuestro, esta diferencia no resulta considerable. Por lo que respecta a la clasificación fina, el no incluir resultados de su sistema hace pensar que no obtuvieron buen rendimiento. Los problemas descritos por Zhang y Lee (2003) empleando un kernel similar a éste pueden ser la causa de esta omisión. Nuevamente, nuestro sistema presenta la ventaja de no requerir de ninguna herramienta lingüística, además de obtener mejoras significativas también para la clasificación de grano fino.

Pan et al. (2008) definieron un *tree kernel* semántico sobre SVM, alcanzando un 94 % de rendimiento sobre la clasificación de grano grueso. No realizaron experimentos sobre las clases finas. Su kernel semántico incluía

información de diversas fuentes como listas de palabras semánticamente relacionadas, información de WordNet y entidades nombradas. Los resultados obtenidos para grano grueso con este sistema lo sitúan como el que mejor rendimiento ofrece para esta taxonomía. Los problemas que presenta esta aproximación son la enorme cantidad de recursos y herramientas empleados (entre otros, analizadores sintácticos, listas de palabras manuales, bases de datos léxicas y reconocedores de entidades), así como la falta de resultados para la clasificación de grano fino, probablemente al no obtener mejoras significativas para esta tarea.

Nguyen et al. (2008) presentó una aproximación a la CP semisupervisada usando el algoritmo Tri-training (Zhou y Li, 2005) y características basadas en etiquetado morfológico. Llevaron a cabo experimentos sobre los subconjuntos del corpus UIUC comentados anteriormente (con 1.000, 2.000, 3.000 y 4.000 preguntas), mientras que el resto de las instancias de entrenamiento se empleaban como datos no etiquetados para el proceso semi-supervisado. Consiguen una precisión de 81.4 % para clases refinadas con 4.000 preguntas de entrenamiento en lugar de usar el corpus completo de 5.432 preguntas. Estos subconjuntos son los mismos que presentábamos en la figura 5.5 (b) para nuestros experimentos. Los resultados obtenidos con nuestro kernel $k_{BOW} + k_{LS}$ mejoran sus experimentos para todos los subconjuntos definidos, demostrando nuevamente que nuestra aproximación semisupervisada utilizando textos de la Wikipedia supera a las aproximaciones que utilizan conjuntos de preguntas sin etiquetar.

Blunsom et al. (2006) obtuvo 92.6 % de precisión empleando *log-linear models* sobre las clases gruesas. Este sistema aprendía a partir de información léxica y sintáctica a partir de un analizador sintáctico especialmente entrenado para etiquetar preguntas. Para la clasificación de grano fino emplearon características extraídas de WordNet, entidades nombradas y *gazeteers*, obteniendo un 86.6 % de precisión. Estos resultados mejoran ligeramente el rendimiento obtenido por nuestros kernels. Sin embargo, sería tremendamente difícil adaptar este sistema a otros idiomas o dominios teniendo en cuenta el elevado uso de recursos y herramientas lingüísticas que hace. Dependiendo de las circunstancias y el propósito del sistema de CP, cabría plantear si la mejora de rendimiento obtenida justifica el número de recursos utilizados y las dependencias adquiridas.

Fuera del campo de la CP, Giozzo y Strapparava (2005) empleó un kernel semántico muy similar al nuestro, llamado *domain kernel*, para la tarea de clasificación de textos. En este trabajo utilizan LSA para inducir modelos de dominio (clusters de términos) a partir de texto no etiquetado. Con este kernel mejoraron los resultados obtenidos con un kernel basado en bolsa de palabras. Comparado con sus resultados, nuestro sistema obtiene una mejora inferior con respecto al kernel bolsa de palabras. Esto refleja las diferencias existentes entre la clasificación de textos y la de preguntas, debido a la mayor dificultad que para los sistemas de aprendizaje presenta esta última tarea.

5.5. Conclusiones y trabajo futuro

A fin de mejorar la representación básica mediante n-gramas que realizábamos en el capítulo anterior, en esta segunda aproximación hemos empleado kernels compuestos para incorporar relaciones semánticas entre palabras y extender la representación mediante bolsa de palabras. Definimos un kernel basado en semántica latente para obtener una función de similitud generalizada entre preguntas. El modelo se obtuvo a partir de documentos no etiquetados de la Wikipedia, dando como resultado un sistema fácilmente adaptable a diferentes idiomas y dominios. Conseguimos mejorar aún más el sistema incluyendo un kernel basado en semántica explícita que emplea listas de palabras semánticamente relacionadas.

El sistema se evaluó en el corpus UIUC, un corpus en inglés de preguntas utilizado por diversos sistemas de CP. Tradujimos el corpus entero a español, obteniendo de esta forma un corpus paralelo para evaluar el sistema sobre distintos idiomas. En los experimentos realizados sobre la taxonomía refinada, nuestro kernel semántico superó a las aproximaciones previas basadas en *tree kernels* e información sintáctica. En la clasificación gruesa, nuestros resultados son comparables a los obtenidos por los sistemas que ofrecen mejor rendimiento, superando muchas otras aproximaciones que hacen uso intensivo de recursos lingüísticos y herramientas. Tanto para la taxonomía gruesa como fina, nuestra aproximación mejora significativamente la representación mediante bolsa de palabras. Al igual que sucedía con los experimentos en inglés, obtuvimos una mejora significativa sobre el corpus en español al incluir el kernel de semántica latente.

Nuestro kernel $k_{BOW} + k_{LS}$ emplea documentos no etiquetados de la Wikipedia para mejorar la representación basada en bolsa de palabras. Esta aproximación semisupervisada ha demostrado funcionar mejor que otras aproximaciones semisupervisadas a la CP basadas en corpus de preguntas no etiquetadas. Nuestra aproximación evita la necesidad de grandes corpus de preguntas (difíciles de obtener) para esta tarea, empleando en su lugar texto plano de fácil acceso en la Web.

El uso de Wikipedia en esta aproximación nos ha permitido definir de forma satisfactoria un modelo estadístico en dominio abierto. Gracias al uso limitado de recursos que realizamos en nuestra aproximación, esta propuesta es capaz de afrontar la tarea de clasificación en diferentes idiomas.

6

Clasificación de preguntas mínimamente supervisada sobre taxonomías refinadas

Tal y como dijimos en capítulos previos, las aproximaciones al aprendizaje automático supervisado requieren de corpus preclasificados de preguntas. El rendimiento de estos sistemas está fuertemente ligado al tamaño del corpus de aprendizaje y al número de instancias por clase. El esfuerzo requerido para anotar un corpus grande de entrenamiento no es trivial. Ya que los corpus de preguntas en lenguaje natural son difíciles de recopilar, las actuales aproximaciones a CP se centran en taxonomías de grano grueso. El tamaño de estas taxonomías varía desde menos de 10 hasta 50 categorías posibles (Sundblad, 2007). Dadas dos preguntas como "¿Quién escribió Dublineses?" y "¿Quién inventó el teléfono?", los sistemas actuales de CP asignarían la categoría *persona*. Sin embargo, para un sistema de BR podría ser más interesante identificar que estas preguntas pertenecen a la categoría *escritor* e *inventor* respectivamente. Existen estudios que dan soporte a la idea de que una clasificación refinada por parte de los sistemas de CP puede mejorar el funcionamiento global de los sistemas de BR (Paşca y Harabagiu, 2001).

En este capítulo afrontamos el desafió de la CP sobre taxonomías de grano fino. En el trabajo de Li y Roth (2002) se pone de manifiesto la dificultad de construir clasificadores sobre este tipo de taxonomías. La solución que dieron para tratar algunas de las clases refinadas de su taxonomía fue añadir manualmente nuevas preguntas al corpus de entrenamiento. En este capítulo vamos a presentar una aproximación que evita la necesidad de corpus de preguntas para el entrenamiento. Para construir el clasificador, el usuario únicamente necesita definir un pequeño conjunto de *semillas* para cada una de las clases refinadas de la taxonomía. Estas semillas se emplean para recuperar fragmentos de texto de la Web. Nuestro sistema obtiene automáticamente a partir de estos fragmentos un conjuntos de términos[1] ponderados para cada una de las clases refinadas de la taxonomía. La presencia de estos términos asociados

[1] En nuestros experimentos extendemos el concepto de término a unigramas, bigramas y trigramas.

a una clase en la pregunta es indicativo de que la pregunta pertenece a dicha clase. Por ejemplo, dada una clase como *inventor*, queremos obtener de forma automática palabras relacionadas semánticamente como "inventor", "invento", "inventó", "descubrir" o "patente", que ayuden al sistema de CP a identificar la clase en la pregunta.

A fin de llevar a cabo la extracción de términos y la estimación de pesos hemos definido un algoritmo llamado DC2 (*dual corpus divergence comparison*). En este algoritmo los términos son ponderados con respecto a cada clase de acuerdo a la disimilitud entre probabilidades de distribución sobre cada una de las clases del dominio. DC2 está basado únicamente en estimaciones estadísticas obtenidas a partir de texto plano, por lo que no requiere ningún tipo de análisis lingüístico. El resultado es una aproximación fácilmente adaptable a diferentes idiomas y dominios. Para dar soporte a esta afirmación hemos llevado a cabo experimentos sobre preguntas en inglés y español.

6.1. Fundamentos estadísticos

Antes de describir el algoritmo subyacente al sistema de clasificación, vamos a introducir algunas ideas y conceptos estadísticos que se emplearán más tarde en este trabajo.

6.1.1. Distribución de Poisson

El modelo probabilístico estándar para la distribución de un cierto tipo de evento en unidades de tamaño fijo es la *distribución de Poisson*. Esta distribución de probabilidad discreta expresa la probabilidad de que un evento t ocurra un número de veces k en unidades de tamaño fijo, si estos ocurren con una tasa media conocida λ. Formalmente se define como

$$P_t(k;\lambda) = e^{-\lambda}\frac{\lambda^k}{k!}. \tag{6.1}$$

En el modelo más común de la distribución de Poisson en el campo de PLN, λ es el número medio de ocurrencias de un término t por documento, es decir $\lambda = cf(t)/N$, donde $cf(t)$ es el número de veces que aparece t en la colección y N es el número total de documentos (Manning y Schütze, 1999). La distribución de Poisson puede ser utilizada para estimar la probabilidad de que un documento tenga exactamente k ocurrencias de un término.

6.1.2. Divergencia de Jensen-Shannon

Diversas medidas se han propuesto para cuantificar la diferencia entre dos distribuciones de probabilidad (Dagan et al., 1999). Una de estas medidas es la *entropía relativa*, también conocida como *divergencia de Kullback-Leibler* (KL), que proporciona una medida estándar de la disimilitud entre dos funciones de probabilidad. La divergencia de KL entre dos distribuciones de probabilidad P y Q se define como

$$D(P\|Q) = \sum_{x \in X} P(x) log \frac{P(x)}{Q(x)}.$$

$D(P\|Q)$ es no negativa, y es 0 sí y sólo sí $P(x) = Q(x) \forall x$. La divergencia de KL no es simétrica y no obedece la desigualdad triangular.[2] Una medida relacionada es la *divergencia de Jensen-Shannon* (JS), que puede definirse como la media de la divergencia de KL de cada una de las dos distribuciones con respecto a su distribución media (Lin, 1991):

$$JS(P\|Q) = \frac{1}{2} D(P\|M) + \frac{1}{2} D(Q\|M), \tag{6.2}$$

donde

$$M = \frac{1}{2}(P + Q).$$

La divergencia de JS es una versión simetrizada de la divergencia de KL. Esta medida y diversas generalizaciones de la misma han sido utilizadas en diferentes contextos, como por ejemplo en el análisis de secuencias simbólicas, en el estudio de textos literarios y en la discriminación de estados cuánticos. Si bien la extensión de su definición a distribuciones continuas de probabilidad es directa, la aplicación de la divergencia de JS ha estado restringida, en general, al estudio de secuencias discretas.

Esta medida supera los problemas de la divergencia de KL, ya que es simétrica y no negativa. $JS(P\|Q)$ está siempre bien definida y acotada, variando entre 0 para distribuciones idénticas y $\log 2$ para distribuciones totalmente diferentes. En sus experimentos, Dagan et al. (1999) demostraron que la divergencia de JS consistentemente funcionaba mejor que otras medidas de similitud entre distribuciones como la divergencia de KL o la *norma L1*.

6.2. Descripción del sistema

Esta sección describe la aproximación en tres fases llevada a cabo para construir el sistema de CP sobre taxonomías de grano fino. La primera

[2] El teorema de desigualdad triangular se define como $\|x + y\| \leq \|x\| + \|y\|$.

	actor	*inventor*	*pintor*
s_1	`Tom Cruise`	`Guglielmo Marconi`	`Pablo Picasso`
s_2	`Tim Robbins`	`Samuel Morse`	`Edvard Munch`
s_3	`Christina Ricci`	`Alfred Nobel`	`Gustav Klimt`
s_4	`Scarlett Johansson`	`Thomas Edison`	`Claude Monet`
s_5	`Robert De Niro`	`Johann Gutenberg`	`Vincent van Gogh`

Figura 6.1: Conjunto de 5 semillas para las clases *actor*, *inventor* y *pintor*.

fase está relacionada con la definición de la taxonomía y de las semillas necesarias para obtener de forma automática fragmentos de texto de la Web. La segunda fase se centra en DC2, el algoritmo que obtiene listas de términos ponderados relacionados con las clases previamente definidas. Finalmente, la tercera fase lleva a cabo la CP propiamente dicha, aprovechando los términos extraídos en el paso anterior de forma similar a como se hace en los sistemas clásicos de RI. Dado que las dos primeras fases se llevan a cabo una sola vez, el proceso final de CP se realiza de forma eficiente desde el punto de vista computacional.

6.2.1. Generación del conjunto de datos

En primer lugar debe definirse el dominio del problema. Es necesario para ello especificar la taxonomía de clases de grano fino C que nuestro sistema de CP será capaz de asignar. En los experimentos presentados en este capítulo, vamos a subcategorizar las preguntas pertenecientes a la clase *persona*. Refinaremos esta clase en 14 subclases diferentes: *actor*, *arquitecto*, *astronauta*, *diseñador*, *director*, *dios*, *inventor*, *asesino*, *monarca*, *músico*, *pintor*, *presidente*, *deportista* y *escritor*. Estas subclases han sido elegidas por su frecuencia y utilidad en sistemas de BR en dominio abierto.

Tal y como dijimos anteriormente, la clasificación en clases finas implica la necesidad de corpus de preguntas de gran tamaño. Para solventar este problema y evitar la necesidad de construir estos corpus, extraeremos automáticamente de la Web un conjunto de fragmentos de textos relacionados con cada clase del problema. Para cada clase $c_i \in C$ se define un conjunto de semillas S. Este conjunto de semillas es la única entrada manual requerida por el sistema. Una semilla $s \in S$ asignada a una clase c, consiste en una entidad representado una posible respuesta a una pregunta perteneciente a dicha clase. La figura 6.1 muestra un conjunto de 5 semillas definidas en nuestros experimentos para tres de las clases mencionadas anteriormente.

Una vez definido el conjunto de semillas para cada clase, el sistema interroga a un motor de búsqueda de información en la Web con dichas semillas para recuperar fragmentos relevantes. De esta manera obtenemos un corpus de fragmentos de texto asociado a cada semilla. Estos fragmentos

```
<SEMILLA> wikipedia the free encyclopedia often regarded as one of
the greatest actors of all time <SEMILLA> has and learned how to box
for his portrayal of jake lamotta in raging bull

<SEMILLA> <SEMILLA> who is thought of as one of the greatest actors
of his time visit imdb for photos filmography discussions bio news
awards agent

<SEMILLA> biography from who2com the most celebrated american film
actor of his era <SEMILLA> won an oscar as best supporting actor for
the godfather part ii in 1974
```

Figura 6.2: Tres fragmentos de muestra obtenidos de la Web. La etiqueta `<SEMILLA>` representa la ocurrencia de la semilla `Robert De Niro` en el texto.

se normalizan transformando a minúsculas el texto, eliminando los signos de puntuación y eliminando todas las ocurrencias en el texto de la semilla empleada. En la figura 6.2 pueden verse tres fragmentos de texto extraídos para la semilla `Robert De Niro` tras la normalización.

Todo el proceso posterior de extracción de términos depende de la calidad de los fragmentos recuperados en esta fase. La definición de las semillas es una tarea no trivial. Hay dos criterios principales que deben seguirse a la hora de seleccionar las semillas. En primer lugar, el número de pasajes recuperado de la Web por cada semilla debe ser lo suficientemente amplio para que las estimaciones estadísticas posteriores sean adecuadas para caracterizar cada una de las clases. Por ejemplo, si queremos caracterizar la clase *actor*, resulta más adecuado seleccionar como semilla alguna estrella de fama mundial que un actor de teatro local. El segundo criterio a tener en cuenta es evitar el uso de semillas transversales a varias clases. Si queremos caracterizar tres clases como *pintor*, *inventor* y *escultor*, no sería adecuado emplear `Leonardo da Vinci` como semilla para ninguna de ellas, ya que puede devolver fragmentos relacionados con cualquiera de las tres clases y dar como resultado una mala caracterización de las mismas.

6.2.2. Extracción de términos y estimación de pesos

El objetivo de este módulo es obtener para cada clase una lista de términos ponderados que estén semánticamente relacionados con la clase. Para ello, todos los términos (unigramas, bigramas y trigramas en nuestros experimentos) son extraidos de los fragmentos recuperados en la fase anterior. En este punto se computan dos valores estadísticos para cada clase c_i: el número total de fragmentos devueltos N_i y el número de ocurrencias de un término t en estos fragmentos, también conocido como la *frecuencia en el corpus* o *corpus frequency* ($cf_i(t)$). Para esta tarea empleamos el CMU-Cambridge Statistical Language Modeling Toolkit.

Para extraer y ponderar los términos relacionados con una clase hemos definido un algoritmo en dos pasos llamado DC2 (*dual corpus divergence comparison*). El primer paso de este algoritmo intenta identificar qué términos son representativos de una clase. Para ello seguimos la hipótesis de que los términos que se distribuyen de forma similar en los fragmentos de las diferentes semillas de una clase están relacionados con esa clase. El segundo paso intenta determinar qué términos son más discriminatorios entre clases. En este caso seguimos la hipótesis de que los términos cuya distribución claramente varía de una clase a otra pueden ser determinantes para distinguir entre dichas clases. El propósito final del algoritmo es identificar términos que al mismo tiempo estén altamente relacionados y sean altamente discriminatorios para cada clase.

Paso 1

Este primer paso trata de identificar términos representativos de una clase. Seguimos la siguiente hipótesis: *los términos que están distribuidos de forma similar en fragmentos recuperados por diferentes semillas de una clase, son representativos de dicha clase*. Por ejemplo, un unigrama como "película" es susceptible de aparecer en fragmentos recuperados por semillas como `Robert De Niro` o `Tim Robbins`, siendo considerada como altamente relacionada con la clase *actor*. Por otra parte, el bigrama "toro salvaje" aparece frecuentemente asociado a `Robert De Niro`,[3] pero con toda probabilidad no será tan frecuente en el resto de las semillas. Este bigrama, por tanto, no debe de considerarse como representativo de la clase *actor*.

Con la intención de medir la relevancia de un término para una clase, calculamos la divergencia de Poisson de dicho término para cada una de las semillas de una clase. Esta distribución de probabilidad ha sido empleada en trabajos previos dentro del campo del PLN para la discriminación entre términos informativos y no informativos en RI (Church y Gale, 1995; Manning y Schütze, 1999; Roelleke, 2003). Estos trabajos se basan en la desviación entre la distribución real de los términos y la distribución calculada mediante Poisson. En nuestro caso vamos a asumir que los términos siguen una distribución de Poisson en los fragmentos y vamos a medir la divergencia de estas distribuciones para ver cómo varía su comportamiento.

En este primer paso, cada clase se trata de forma aislada (la comparación entre clases tiene lugar en el segundo paso). Para medir cómo varía para un término su distribución de Poisson de una semilla a otra, calculamos la divergencia de JS entre distribuciones. Para todas las clases $c_i \in C$, tomamos cada término t para construir una matriz de disimilitud $X1_{t,c_i}$, de tamaño $|S| \times |S|$ (done S es el conjunto de semillas de cada clase), con los valores de

[3] "Toro salvaje" es una de sus películas más aclamadas.

$$X1_{movie,actor} = \begin{array}{c} \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \end{array} \begin{array}{c} \begin{array}{ccccc} s_1 & s_2 & s_3 & s_4 & s_5 \end{array} \\ \begin{pmatrix} 0,0000 & 0,0000 & 0,0015 & 0,0605 & 0,0065 \\ & 0,0000 & 0,0014 & 0,0598 & 0,0063 \\ & & 0,0000 & 0,0444 & 0,0018 \\ & & & 0,0000 & 0,0291 \\ & & & & 0,0000 \end{pmatrix} \end{array}$$

$$X1_{martin,actor} = \begin{array}{c} \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \end{array} \begin{array}{c} \begin{array}{ccccc} s_1 & s_2 & s_3 & s_4 & s_5 \end{array} \\ \begin{pmatrix} 0,0000 & 1,0000 & 1,0000 & 1,0000 & 1,0000 \\ & 0,0000 & 0,0000 & 0,0002 & 0,1284 \\ & & 0,0000 & 0,0002 & 0,1285 \\ & & & 0,0000 & 0,1233 \\ & & & & 0,0000 \end{pmatrix} \end{array}$$

Figura 6.3: Matrices de disimilitud para los términos "*movie*" y "*martin*" en la clase *actor* para los experimentos en inglés. El elemento $x1_{ij}$ representa la divergencia de JS de la distribución de Poisson del término en las semillas s_i y s_j. Los valores de la diagonal inferior se omiten al ser simétrica $X1$.

divergencia de JS de la distribución de Poisson del término en los fragmentos recuperados por las diferentes semillas de cada clase. Dado un término t, el elemento $x1_{jk}$ de la matriz $X1_{t,c_i}$ contiene la divergencia de JS de la distribución de Poisson de t en los fragmentos recuperados por las semillas s_j y s_k en la clase c_i. Todos los valores en la diagonal son 0 (indicando máxima similitud) ya que son el resultado de comparar la distribución de t en una semilla consigo mismo. Como la divergencia de JS es una medida simétrica, la matriz obtenida también lo es. Cuando t no aparece en los fragmentos recuperados por una semilla se asume la máxima disimilitud $\log 2$.[4]

La figura 6.3 muestra las matrices de disimilitud para los términos "*movie*" y "*martin*" en la clase *actor* con las cinco semillas definidas en la figura 6.1 para los experimentos en inglés. Los valores de $x1_{jk}$ en $X1_{movie,actor}$ son bajos en general, indicando que "*movie*" se distribuye de manera similar a lo largo de todas las semillas de la clase *actor*. Por contra, los valores en $X1_{martin,actor}$ son significativamente mayores, revelando una distribución irregular de "*martin*" en las distintas semillas.

El resultado de de este primer paso es un peso $W1_{t,c_i} \in [0,1]$ que representa cómo de variable es la distribución de un término t en las diferentes semillas de la clase c_i. Estos pesos se obtienen computando la divergencia de JS media en las semillas de la clase:

[4] En nuestros experimentos hemos empleado logaritmos en base 2, por lo que la máxima disimilitud será 1.

Clase	$\mathbf{W1}_{movie}$	$\mathbf{W1}_{actor}$	$\mathbf{W1}_{martin}$	$\mathbf{W1}_{the}$
actor	0,0211	0,0163	0,4380	0,0315
arquitecto	0,9002	1,0000	0,7006	0,0358
asesino	0,4060	0,7126	0,9047	0,0015
astronauta	0,7010	0,9001	1,0000	0,0161
deportista	0,7009	0,7014	0,9001	0,0218
dios	0,4016	1,0000	1,0000	0,0482
director	0,0098	0,0036	0,7013	0,0035
diseñador	0,9000	1,0000	0,7000	0,0095
escritor	0,0028	0,7001	0,9000	0,0111
inventor	0,7002	1,0000	0,7003	0,0222
monarca	1,0000	1,0000	0,9000	0,0239
músico	0,0004	1,0000	0,4003	0,0231
pintor	0,0011	1,0000	0,4011	0,0098
presidente	0,7017	1,0000	0,9001	0,0616

Tabla 6.1: Pesos $W1$ obtenidos en el *Paso 1* para diferentes términos en las 14 clases definidas en el problema.

$$W1_{t,c_i} = \frac{1}{|S|^2 - |S|} \sum_{j=1}^{|S|} \sum_{k=1}^{|S|} x1_{jk} \tag{6.3}$$

donde $|S|^2 - |S|$ representa el número de elementos en la matriz $X1$ sin tener en cuenta los elementos de la diagonal. Cuanto mayor sea el valor de $W1_{t,c_i}$, menor será la relevancia de t en c_i, ya que implica que el término no está distribuido de forma regular en las distintas semillas de la clase. La tabla 6.1 muestra los valores de $W1$ para cuatro términos en cada una de las 14 clases del problema en inglés. Términos como "*movie*" y "*actor*" presentan pesos bajos para clases como *actor* y *director*, pero valores más altos para clases como *arquitecto*, *astronauta* o *diseñador*. El término "*martin*" obtiene valores más altos en todas las clases. Esto revela que la distribución de este término es irregular en las semillas de todas las clases. Por otra parte, el peso asignado a "*the*" es muy bajo en todas las clases indicando que está regularmente distribuido en las semillas de estas clases.

A pesar de la distribución regular de una palabra vacía como "*the*" en las clases, este término es claramente no representativo de estas clases ya que no proporciona ningún indicio para distinguir entre ellas. Será en el segundo paso del algoritmo DC2 donde discriminaremos los términos representativos de una clase, como "película" o "protagonizar" para la clase *actor*, de aquellos no representativos, como son las palabras vacías.

Paso 2

Este segundo paso trata de determinar qué términos son más discriminatorios entre clases. Seguimos la siguiente hipótesis: *los términos cuya distribución varía entre clases son interesantes para caracterizar dichas clases*. Es decir, si un término está igualmente distribuido a lo largo de diferentes clases no puede ser considerado como representativo para ninguna de ellas. A diferencia del *Paso 1*, ahora nos interesan aquellos términos con elevada disimilitud de distribución entre clases. Por ejemplo, "película" debería ser discriminatorio en muchos contextos, ayudando a caracterizar una clase como *actor* cuando se intenta diferenciar de clases como *inventor* o *escritor*.

El primer paso a llevar a cabo es obtener un único corpus de fragmentos de texto para cada clase, agrupando los fragmentos individuales obtenidos para cada una de las semillas de una clase. Para cada término se computa la frecuencia en el corpus $cf_i(t)$ y el número total de fragmentos N_i en el conjunto de fragmentos de la clase c_i. Al igual que en el *Paso 1*, calculamos la distribución de Poisson de los términos y la divergencia de JS de estas distribuciones. Obtenemos de esta manera una matriz de disimilitud $X2_t$ de tamaño $|C| \times |C|$ (donde C es el conjunto de clases) para cada término t. Los elementos $x2_{ij}$ de esta matriz representan la disimilitud entre las clases c_i y c_j. En este caso estamos interesados en aquellos términos con valores grandes de $x2_{ij}$, indicando que el término se distribuye de forma diferente entre las clases.

En este segundo paso asignamos un peso $W2_{t,c_i}$ para cada término t en cada clase c_i. El peso del término en la clase depende de los valores de disimilitud de la matriz $X2_t$. Calculamos la media de la divergencia de JS para la clase:

$$W2_{t,c_i} = \frac{1}{|C|-1} \sum_{j=1}^{|C|} x2_{ij} \tag{6.4}$$

En este caso, valores altos de $W2_{t,c_i}$ indican que el término t está claramente relacionado con la clase c_i pero no con las otras clases. En este punto tenemos dos pesos diferentes para cada término t en una clase c_i: $W1_{t,c_i}$, que indica cómo es de representativo un término para la clase, y $W2_{t,c_i}$ que indica cómo es de discriminatorio el término entre clases. Para obtener el peso final asignado a un término, combinamos estos dos valores y la frecuencia $cf_i(t)$ del término en la clase:

$$W_{t,c_i} = \log(cf_i(t))(W2_{t,c_i} - W1_{t,c_i}) \tag{6.5}$$

De esta forma, el peso final depende de dos factores:

- $\log(cf_i)$. Esta primera parte de la ecuación permite tener en cuenta la frecuencia de aparición del término en los fragmentos de la clase

($cf_i(t)$). El logaritmo se emplea para suavizar la importancia de este valor en el peso final, evitando que un término sea considerado excesivamente importante sólo porque sea muy frecuente (como es el caso de las palabras vacías).

- $W2_{t,c_i} - W1_{t,c_i}$. A esta diferencia entre los pesos $W1$ y $W2$ la llamamos el *factor de divergencia*. El peso $W2_{t,c_i} \in [0,1]$ nos indica cómo de variable es la distribución de t entre la clase c_i y el resto. Un término que sea representativo de una clase tendrá un peso $W2$ elevado. Por otra parte, el peso $W1_{t,c_i} \in [0,1]$ nos indica si un término t se distribuye de forma equivalente entre todas las semillas de la clase c_i. Un término que sea representativo de una clase tendrá un peso $W1$ bajo. Al restar estos dos términos obtenemos un valor en el intervalo $[0,1]$ que será mayor cuanto más discriminatorio sea el término entre clases ($W2$ grande) y más similar sea su distribución en las semillas de la clase tratada ($W1$ pequeño). El factor de divergencia nos sirve para modificar el valor asignado por $\log(cf_i)$ para que el peso final W dependa no sólo de la frecuencia de aparición de un término, sino también de su distribución en las semillas de una clase y entre las distintas clases.

La tabla 6.2 muestra la lista de los diez unigramas, bigramas y trigramas mejor ponderados en inglés según este algoritmo para las clases *asesino*, *dios* e *inventor*.

6.2.3. Asignación de la clase

Las listas de términos ponderados creadas por DC2 nos servirán para detectar la clase semántica de una pregunta. La presencia en la pregunta de términos como "actor", "actriz", "película" o "protagonizó" son claramente indicativos de la clase *actor*. La clasificación en taxonomías refinadas se apoya en estas listas de términos siguiendo un algoritmo simple de comparación, similar a la aproximación tradicional que se emplea en RI para detectar la relevancia de los documentos con respecto a una consulta. En nuestro caso, medimos la relevancia de una clase con respecto a una pregunta. Dada una nueva pregunta de entrada, el proceso de asignación de la clase se realiza en cuatro pasos:

1. Extraer todos los términos de la pregunta.
2. Recuperar el peso asignado por DC2 a estos términos en cada clase.
3. Añadir el peso del término al marcador final de cada clase.
4. Asignar a la pregunta la clase que haya obtenido mayor puntuación.

asesino					
unigramas		**bigramas**		**trigramas**	
assassination	8,5650	the assassination	7,8028	the assassination of	7,4940
assassinated	8,1681	assassination of	7,7667	the murder of	6,4164
assassin	8,1017	was assassinated	7,0586	was assassinated by	6,1211
killer	6,3198	was shot	6,8680	shot and killed	5,7189
murder	5,8475	the murder	6,8386	was shot by	4,9441
kill	5,6345	who killed	6,6764	he was shot	4,8937
killing	5,5340	assassinated by	6,5584	he was assassinated	4,7956
conspiracy	5,4323	murder of	6,1600	was assassinated on	4,6391
murdered	5,3867	assassin of	5,9850	was shot and	4,6297
killed	5,1626	shot and	5,8744	for the murder	4,5154

dios					
unigramas		**bigramas**		**trigramas**	
gods	7,8954	god of	7,7475	the god of	6,3864
deity	6,3680	the god	7,4992	god of the	6,1908
divine	5,1958	of god	6,2498	of the god	5,2870
worship	5,0550	of gods	6,1361	god in the	4,3870
sacred	4,8635	god is	6,0066	of the gods	3,8410
almighty	4,7452	names of	5,9160	lord of the	3,1911
wisdom	4,4145	god in	5,4831	the king of	2,7939
messenger	4,2371	god the	5,4043	the name of	2,7811
attributes	3,8056	the lord	5,3945	is the god	2,6704
religious	3,7853	the names	4,8895	the names of	2,5415

inventor					
unigramas		**bigramas**		**trigramas**	
inventor	9,0321	inventor of	8,0905	inventor of the	6,9656
invented	8,8907	invention of	7,7024	invention of the	6,9645
invention	8,3085	invented the	7,6152	the inventor of	6,8992
inventions	7,0231	the inventor	7,0673	the invention of	6,8260
invent	6,2951	the invention	6,9605	who invented the	5,8015
inventing	5,8205	who invented	6,7703	his invention of	4,5172
developed	5,6863	invented by	5,9789	with the invention	4,1667
inventors	5,5976	invented in	5,7925	did not invent	4,1605
patented	5,5481	his invention	5,4434	the father of	4,0188
practical	5,2480	and inventor	5,2642	as the inventor	3,9960

Tabla 6.2: Diez términos con mayor relevancia en inglés para las clases *asesino*, *dios* e *inventor*. Al lado de cada n-grama aparece el peso final asignado por el algoritmo DC2.

La tabla 6.3 muestra un ejemplo de este proceso. Términos como "*actress*" y "*starred*" se comportan de la forma esperada, obteniendo la mayor puntuación para la clase *actor*. Términos como "*what*", "*in*", "*the*" y "*lion*", obtienen valores muy bajos para todas las clases ya que DC2 no los considera relevantes para ninguna de ellas. Una situación especial sucede con "*winter*", ya que este término obtiene un valor alto para la clase *diseñador*. Este resultado no es sorprendente si consideramos que los nombres de estaciones del año son bastante comunes cuando hablamos sobre moda y colecciones.

6.3. Experimentos y evaluación

Hemos evaluado nuestra aproximación a la CP sobre taxonomías refinadas en el escenario descrito en la sección 6.2.1, con 14 subclases posibles para la clase *persona*. Para obtener el conjunto de preguntas de evaluación revisamos el corpus TREC descrito en la sección 4.1.1, seleccionando aquellas preguntas que se ajustaban a las 14 clases definidas. Un total de 205 preguntas fueron etiquetadas manualmente con la clase refinada esperada. En el apéndice A puede verse el corpus completo en inglés. Para contrastar el funcionamiento del sistema en diferentes idiomas lo evaluamos sobre las preguntas en inglés y en español. La tabla 6.4 muestra la distribución de preguntas por clase, mientras que la figura 6.4 muestra diversas preguntas incluídas en el conjunto de evaluación en inglés.

El propósito de los experimentos llevados a cabo en esta sección es doble. Primero, evaluar el funcionamiento de nuestra aproximación para la CP sobre taxonomías refinadas. Segundo, comparar esta aproximación con los actuales sistemas supervisados de aprendizaje. Para contrastar estos dos puntos se llevaron a cabo cinco experimentos:

- *DC2*: este experimento evalúa el funcionamiento de nuestra aproximación basada en el algoritmo DC2. Llevamos a cabo tres experimentos diferentes: *DC2_1-gramas* evalúa DC2 empleando únicamente listas de unigramas, *DC2_1+2-gramas* evalúa DC2 extrayendo listas de unigramas y bigramas, y *DC2_1+2+3-gramas* evalúa DC2 con listas de unigramas, bigramas y trigramas.

- *CF*: el peso W asignado por DC2 a un término t para una clase c_i depende de la frecuencia del término en el corpus ($\log(cf_i(t))$) y del *factor de divergencia* ($W2_{t,c_i} - W1_{t,c_i}$). Para medir la importancia del factor de divergencia en W, vamos a reformular la ecuación 6.5 para que tenga sólo en cuenta la frecuencia del término ($W_{t,c_i} = \log(cf_i)$). Al igual que hicimos en el experimento anterior, llevamos a cabo tres evaluaciones diferentes: *CF_1-gramas* tiene en cuenta sólo las listas de unigramas generadas por nuestro algoritmo, *CF_1+2-gramas* tiene en

Clase	*what*	*actress*	*starred*	*in*	*the*	*lion*	*in*	*winter*	**Total**
actor	0,0000	**2,9459**	**4,6468**	0,0149	0,0192	0,0000	0,0149	0,0000	**7,6269**
arquitecto	0,0000	0,0000	0,0000	0,0497	0,0000	0,0000	0,0497	0,1621	0,2118
asesino	0,0000	0,0000	0,0000	0,0368	0,1792	0,0000	0,0368	0,0000	0,2160
astronauta	0,0000	0,0000	0,0000	0,0000	0,2046	0,0000	0,0000	0,0000	0,2046
deportista	0,0018	0,0000	0,0000	0,0000	0,0203	0,0000	0,0000	0,0000	0,0222
dios	0,0000	0,0000	0,0000	0,0000	0,0504	**0,1000**	0,0000	0,1260	0,2764
director	0,0071	0,0000	1,2150	0,1105	0,2030	0,0000	0,1105	0,0000	1,5356
diseñador	0,0043	0,0000	0,0000	**0,1280**	**0,5341**	0,0000	**0,1280**	**2,9857**	3,6521
escritor	0,0000	0,0000	0,0000	0,0026	0,1002	0,0000	0,0026	0,1874	0,2902
inventor	**0,0155**	0,0000	0,0000	0,0000	0,3696	0,0000	0,0000	0,0000	0,3851
monarca	0,0085	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0085
músico	0,0000	0,0000	0,0000	0,0380	0,0000	0,0000	0,0380	0,0000	0,0380
pintor	0,0000	0,0000	0,0000	0,0376	0,2565	0,0000	0,0376	0,0000	0,2941
presidente	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000

Tabla 6.3: Proceso de clasificación de la pregunta "*What actress starred in the lion in winter?*" en las 14 clases posibles. Sólo los unigramas se muestran en este ejemplo. El mayor valor para cada términos aparece en negrita. El peso final para una clase dada c_i se obtiene sumando los valores en la fila correspondiente.

Clase	Preguntas
actor	20
arquitecto	4
asesino	11
astronauta	10
deportista	17
dios	5
director	5
diseñador	2
escritor	24
inventor	30
monarca	13
músico	18
pintor	3
presidente	43
Total	205

Tabla 6.4: Número de preguntas para cada una de las clases del conjunto de evaluación en inglés.

actor	What actor first portrayed James Bond?
arquitecto	Who were the architects who designed the Empire State Building?
asesino	Who shot Billy the Kid?
astronauta	Who was the second man to walk on the moon?
deportista	Who was the first black heavyweight champion?
dios	Who was the Roman god of the sea?
director	Who directed the film "Fail Safe"?
diseñador	What costume designer decided that Michael Jackson should only wear one glove?
escritor	What author wrote under the pen name "Boz"?
inventor	Who made the first airplane that could fly?
monarca	Who is the Queen of Holland?
músico	Who composed "The Messiah"?
pintor	Who painted "Sunflowers"?
presidente	What president is on the thousand dollar bill?

Figura 6.4: Ejemplos de preguntas y su correspondiente clase pertenecientes al conjunto de evaluación en inglés.

cuenta las listas de unigramas y bigramas, y *CF_1+2+3-gramas* tiene en cuenta las listas de unigramas, bigramas y trigramas.

- *SVM*: este experimento nos permite comparar el funcionamiento de nuestra aproximación con un sistema tradicional de aprendizaje supervisado. Empleamos SVM entrenando y evaluando con el corpus de 205 preguntas mediante *10-fold cross validation*. Empleamos como características de aprendizaje la representación basada en bolsa de palabras.

- *SVM2*: tal y como se describió en la sección 6.2.1, nuestro sistema obtiene de forma automática un corpus de fragmentos de texto de la Web a partir de un conjunto de semillas. Podemos considerar a estos fragmentos como el "corpus de entrenamiento" de nuestro algoritmo, aunque disten mucho de ser preguntas bien formadas en lenguaje natural. DC2 emplea este corpus para construir las listas de términos relevantes para cada clase. Gran parte de la dificultad de construir estas listas reside en descartar todos aquellos términos de los fragmentos que no están relacionados con la clase a la que deberían representar. En este experimento entrenamos SVM sobre este mismo corpus de fragmentos que emplea nuestro algoritmo DC2. De esta forma podemos comparar el rendimiento de nuestro algoritmo con una aproximación tradicional supervisada cuando se entrena sobre este corpus ruidoso obtenido de forma automática.

- *SVM2+CHI*: este experimento compara DC2 con χ^2 a la hora de seleccionar los términos más relevantes del corpus. Para ello hacemos un refinamiento sobre el experimento *SVM2*. El corpus de fragmentos recuperado de la Web mediante semillas incluye muchos términos no relacionados con la clase a la que deberían representar. Aunque SVM ha demostrado funcionar bien en problemas con un gran número de características de aprendizaje, hay diversas aproximaciones a la selección de características que ayudan a descartar características no relevantes (como vimos en la sección 4.5.3). En este experimento repetimos la evaluación realizada en *SVM2* empleando χ^2 para seleccionar previamente las características (términos) más relevantes. Esto nos permite comparar DC2 y χ^2 a la hora de seleccionar los términos que sean relevantes para la clasificación y más representativos de cada clase.

6.3.1. Configuración del sistema

La única entrada requerida por nuestro sistema es un conjunto de semillas. En nuestros experimentos obtuvimos que 5 semillas era el tamaño óptimo para este conjunto. En la figura 6.1 mostrábamos un ejemplo de estas

semillas. El conjunto de semillas completo para los experimentos en inglés pueden verse en el apéndice B. Una vez definidas estas semillas se consulta a Google para recuperar un número de fragmentos por semilla para cada clase. Empíricamente fijamos el número de fragmentos recuperados por semilla a 700. Después de recuperar los fragmentos, el algoritmo DC2 obtiene una lista de términos ponderados para cada clase. En este punto, el sistema es capaz de clasificar las preguntas de evaluación a partir de estas listas tal y como se describió en la sección 6.2.3.

6.3.2. Resultados experimentales

La tabla 6.5 muestra los resultados finales obtenidos en los experimentos en inglés y español. Los resultados empleando DC2 muestran que el mejor rendimiento se obtiene empleando unigramas y bigramas (*DC2_1+2-gramas*) en inglés y con unigramas en español (*DC2_1-gramas*). Para inglés este resultado resulta muy prometedor, alcanzando una precisión final de 85,37 %. Los resultados en español (79,02 %), aunque más bajos que en inglés, pueden ser considerados como buenos cuando se compara con la aproximación supervisada (70,73 % para *SVM*). Dos circunstancias justifican estas diferencias entre idiomas. Primero, el español es un idioma más flexivo que el inglés. Hay numerosas variantes para un término que dependen del género, la persona, el número, etc. Estas variantes afectan las estimaciones estadísticas de los términos. En segundo lugar, la cantidad de información en la Web para español es considerablemente menor que para inglés. Por esta razón es más difícil devolver buenos fragmentos de la Web para cada clase. Podemos cuantificar esta diferencia observando el número medio de fragmentos recuperados por semilla: 485,37 para inglés y 398,87 para español. Por tanto, resulta más sencillo obtener buenas estimaciones estadísticas a partir de los fragmentos en inglés.

Los resultados obtenidos empleando sólo la frecuencia para ponderar los términos (*CF_1-gramas*, *CF_1+2-gramas* y *CF_1+2+3-gramas*) demuestran la importancia del *factor de divergencia* en la ecuación 6.5. Los mejores valores obtenidos en este experimento, 74,63 % en inglés y 69,01 % en español con unigramas y bigramas, son significativamente peores que los resultados obtenidos con la versión completa del algoritmo.

La precisión obtenida con el experimento *SVM* (70,73 % para inglés y 65,37 % en castellano) demuestra que nuestro sistema supera las aproximaciones supervisadas tradicionales cuando el tamaño del corpus de entrenamiento es pequeño (205 muestras), comparado con el número de clases posibles (14). El segundo experimento, *SVM2*, obtuvo 76,10 % de precisión para inglés y 72,68 % para castellano. Aunque esta aproximación mejora los resultados obtenidos con el experimento *SVM*, el rendimiento obtenido sigue estando lejos de los resultados conseguidos habitualmente con las aproximaciones basadas en SVM y bolsa de palabras. Por ejemplo,

Experimento	Inglés	Español
DC2_1-gramas	79,02	**79,02**
DC2_1+2-gramas	**85,37**	78,05
DC2_1+2+3-gramas	80,98	74,63
CF_1-gramas	72,68	71,21
CF_1+2-gramas	74,63	70,73
CF_1+2+3-gramas	74,15	69,75
SVM	70,73	65,37
SVM2	76,10	72,68
SVM2+CHI	79,51	77,07

Tabla 6.5: Resultados finales obtenidos en los experimentos para los conjuntos en inglés y español. Los mejores resultados para cada idioma se muestran en negrita.

en nuestros experimentos del capítulo 4 obteníamos valores cercanos al 82 % en los experimentos en dominio abierto. Este hecho evidencia que el corpus de fragmentos recuperado de forma automática de la Web no es apropiado para los sistemas tradicionales de aprendizaje supervisado, ya que la cantidad de términos no relevantes en estos fragmentos es alto. Sin embargo, nuestro algoritmo DC2 es capaz de obtener información útil de este corpus y alcanzar un elevado rendimiento, demostrando su habilidad para discriminar los términos relevantes de una fuerte de información ruidosa. El mismo comportamiento se da en los experimentos sobre el corpus en español, aunque en este caso se obtuvieron resultados inferiores al inglés.

En la figura 6.5 (a) se muestra una comparación del rendimiento obtenido para cada una de las 14 clases en los experimentos *DC2_1+2-gramas*, *SVM* y *SVM2* en inglés. Los mismos resultados en español se pueden ver en la figura 6.5 (b).

El último experimento, *SVM2+CHI*, demuestra que se puede mejorar el funcionamiento de SVM en el experimento *SVM2* utilizando un método estadístico de selección de características como χ^2. La figura 6.6 resume la precisión obtenida aplicando diferentes umbrales de selección de características, comparando los resultados con la precisión obtenida en los experimentos previos. La tabla 6.5 muestra los mejores resultados para inglés (79,51 % con 40.000 términos) y español (77,07 % con 10.000 términos). Aunque estos resultados mejoran el experimento *SVM* y *SVM2*, su rendimiento sigue siendo significativamente inferior al obtenido con el algoritmo DC2.

6.4. Trabajos relacionados

Fuera del campo de la CP, diversos sistemas han realizado la extracción automática de términos semánticamente relacionados para su aplicación a

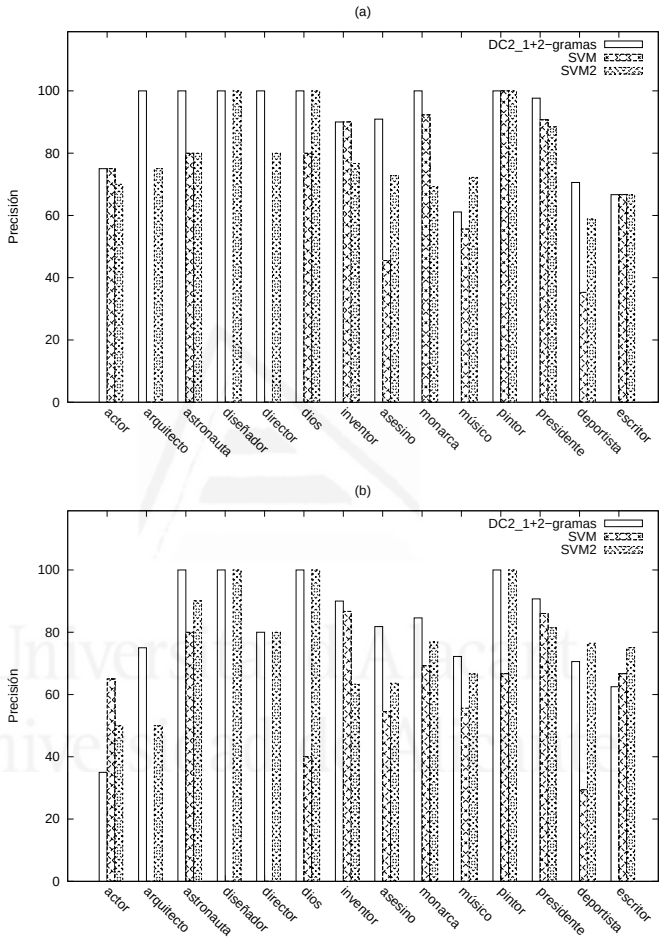

Figura 6.5: Resultados obtenidos para las distintas clases en los experimentos *DC2_1+2-gramas*, *SVM* y *SVM2* en (a) inglés y (b) español.

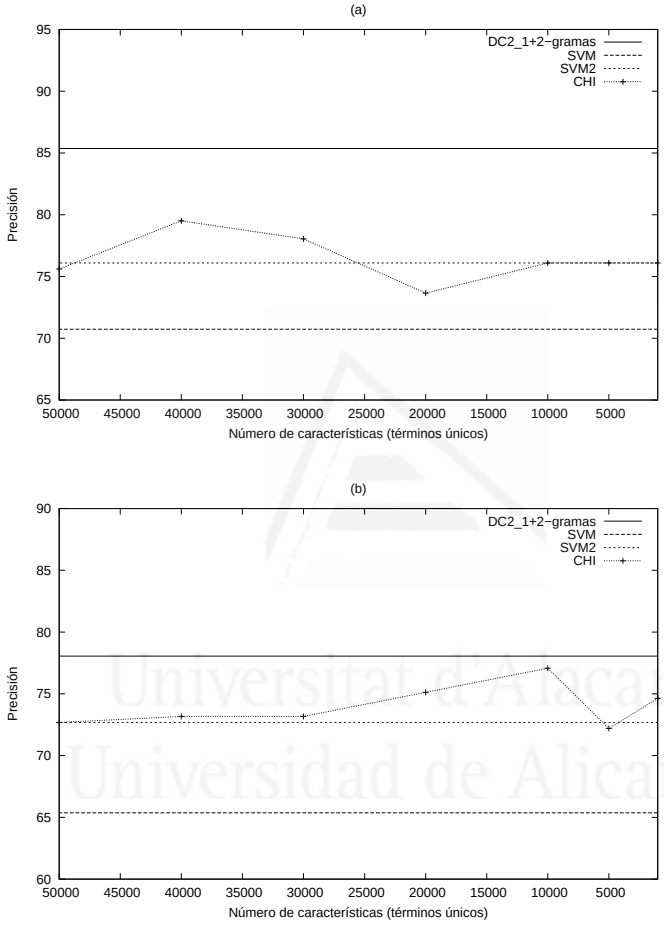

Figura 6.6: Resultados para los experimentos en (a) inglés y (b) español. La gráfica muestra la precisión obtenida con χ^2 (*CHI*) para diferentes umbrales de selección. Se incluyen los resultados obtenidos con *DC2_1+2-gramas*, *SVM* y *SVM2* por motivos de comparación.

diferentes tareas de PLN. En (Lin y Pantel, 2001) definieron el concepto de *topic signature*. Estos *topics* son familias de términos relacionadas con un concepto, como *energía solar* o *seguridad informática*. A diferencia de nuestra aproximación, los términos fueron extraidos a partir de corpus de documentos previamente clasificados. Emplearon la información extraida para llevar a cabo resúmenes de textos.

An et al. (2003) obtuvieron automáticamente un corpus de entidades nombradas. Emplearon como semillas diferentes entidades y recuperaron fragmentos de la Web que incluían a estas entidades. Emplearon diversas herramientas, como un etiquetador gramatical para detectar los límites de las entidades en el texto.

Riloff y Shepherd (1997) experimentaron con un método basado en corpus para construir lexicones semánticos, usando un corpus textual y un pequeño conjunto de semillas por cada categoría para identificar otras palabras pertenecientes también a la categoría. Emplearon estadísticas sencillas y un mecanismo de *bootstrapping* para generar listas ordenadas de palabras. El último proceso era llevado a cabo de forma manual, ya que un revisor humano seleccionaba las palabras mejor ponderadas para incluirlas en el diccionario.

La extracción de relaciones es otro campo que se ha beneficiado de estas aproximaciones automáticas. Brin (1999) considera el problema de la extracción de relaciones en la Web. En este trabajo se presenta el sistema DIPRE, que explota la dualidad entre conjuntos de patrones y relaciones para aumentar las muestras de la relación objetivo comenzando por un pequeño conjunto de semillas.

Centrados en el campo de BR, Girju (2003) analizó preguntas de tipo *causa-efecto* proporcionando una aproximación basada en aprendizaje inductivo para el descubrimiento automático de restricciones léxicas y semánticas, necesarias en la desambiguación de relaciones causales. Ravichandran y Hovy (2001) exploraron el poder de los patrones textuales superficiales para la extracción de respuestas en preguntas de dominio abierto. Desarrollaron un método para aprender dichos patrones de forma automática a partir de un conjunto pequeño de pares de preguntas y respuestas mediante *bootstrapping*. Los patrones fueron automáticamente extraidos de los documentos devueltos y estandarizados. Estos patrones fueron aplicados para encontrar respuestas a nuevas preguntas. Fleischman et al. (2003) presentaron una estrategia alternativa en la cual los patrones adquiridos fueron usados para extraer información relacional muy precisa, creando un repositorio de información que fue empleado para responder preguntas de forma eficiente.

6.5. Conclusiones

En este capítulo hemos presentado una aproximación a la CP sobre taxonomías refinadas. Si un problema de clasificación consiste en un pequeño número de categorías bien separadas, entonces muchos algoritmos de clasificación funcionarán correctamente. Pero en la mayoría de problemas de clasificación reales hay un gran número de categorías, a veces muy similares. La clasificación precisa sobre grandes conjuntos de clases relacionadas es inherentemente difícil.

Nuestra aproximación es mínimamente supervisada, ya que únicamente necesitamos un conjunto de semillas (5 en nuestros experimentos) para caracterizar las clases del problema. Empleamos estas semillas para recuperar los fragmentos de texto de la Web para las clases refinadas. Hemos definido un algoritmo propio, llamado DC2, que permite extraer y ponderar términos altamente relacionados con las clases y altamente discriminatorios entre ellas. Este proceso está basado en la divergencia de JS de la distribución de Poisson de los términos en diferentes fragmentos asociados a las clases.

Los términos extraidos fueron empleados para identificar la clase de la pregunta siguiendo un algoritmo de asignación tradicionalmente empleado para medir la relevancia de documentos y consultas en los sistemas de RI. El algoritmo está totalmente basado en información estadística y no requiere de recursos lingüísticos o herramientas. Más aún, ya que el texto se extrae de forma automática de la Web, no se requiere de un corpus de entrenamiento.

Para evaluar el sistema hemos refinado la clase *persona* en 14 subclases posibles, obteniendo del corpus TREC un subconjunto de más de 200 preguntas pertenecientes a estas subclases. Llevamos a cabo diversos experimentos para comparar nuestra aproximación con otras aproximaciones tradicionales basadas en aprendizaje supervisado como SVM, obteniendo mejoras significativas con nuestro sistema en todas las comparaciones llevadas a cabo. Los resultados mostraron que nuestra aproximación resulta altamente efectiva para esta tarea (85,37 % de precisión en inglés), manejando espacios de aprendizaje con numerosas características irrelevantes mejor que SVM y que técnicas estadísticas de selección de características como χ^2. Para evaluar la independencia del lenguaje de nuestra aproximación llevamos a cabo experimentos en inglés y español. Aunque los resultados en español son inferiores a los obtenidos en inglés (79.02 %), nuestro sistema obtuvo mejoras significativas con respecto al rendimiento de la aproximación tradicional supervisada empleando SVM.

7

Conclusiones y trabajo futuro

El principal objetivo de esta tesis es el desarrollo de sistemas de CP fácilmente adaptables a diferentes idiomas y dominios. Para conseguir este objetivo establecimos dos premisas esenciales:

- *Los sistemas aprenden por sí mismos.* Hemos basado nuestra aproximación en técnicas de aprendizaje automático. Esto implica que los sistemas pueden mejorar a través de la experiencia mediante corpus.
- *El proceso de aprendizaje no depende de recursos lingüísticos complejos.* Nuestras aproximaciones están basadas en características textuales superficiales automáticamente extraídas del corpus de aprendizaje o de texto no etiquetado.

Hemos definido tres aproximaciones diferentes para cumplir este objetivo. Primero, desarrollamos un sistema básico basado en n-gramas obtenidos del corpus de aprendizaje. Segundo, mejoramos la representación básica basada en n-gramas con información semántica obtenida a partir de texto no etiquetado. Finalmente, afrontamos el problema de la ausencia de corpus de aprendizaje y la clasificación sobre taxonomías refinadas. Estas tres aproximaciones pueden ser consideradas como sistemas de "bajo coste", ya que la intervención humana y los recursos y herramientas requeridos son mínimos. En la primera aproximación, el sistema sólo requiere un corpus de preguntas de entrenamiento. En la segunda aproximación, el sistema necesita un corpus de preguntas de entrenamiento y un conjunto de documentos no etiquetados fácilmente accesibles gracias a la Web. En la tercera aproximación sólo se requiere por parte del usuario la definición de un conjunto de semillas para cada clase en la taxonomía del problema. En este caso ni tan siquiera es necesario un corpus para entrenar el sistema.

Aunque esta tesis se ha centrado en el desarrollo y evaluación de sistemas de CP de forma independiente, el objetivo final es su aplicación a sistemas finales de BR para evaluar su influencia en el rendimiento global de esta tarea. La primera aproximación es adecuada cuando sólo se dispone de un corpus de preguntas de entrenamiento. Esta aproximación demostró funcionar de forma adecuada en dominio abierto, siendo altamente

efectiva en dominios restringidos. Si tenemos la oportunidad de adquirir un corpus de documentos no etiquetados (la Web es una fuente inagotable de ellos), resulta interesante la utilización de nuestra segunda aproximación para mejorar el modelo básico sobre n-gramas. Con respecto a la tercera aproximación, su utilización es adecuada para taxonomías refinadas sobre las que no se disponga de corpus adecuados para el entrenamiento de un sistema supervisado. Estudios previos avalan los beneficios sobre el sistema de BR de la clasificación y detección de entidades refinadas (Paşca y Harabagiu, 2001).

De forma adicional a las aproximaciones definidas, en este trabajo hemos revisado el estado de la cuestión en este campo y hemos desarrollado diversos corpus para entrenar y evaluar los sistemas de CP en diferentes idiomas y dominios.

En el resto de este capítulo presentamos las conclusiones alcanzadas para cada una de las aproximaciones y discutimos diversas direcciones futuras de investigación. Finalmente presentamos las principales publicaciones derivadas de este trabajo de investigación.

7.1. Clasificación de preguntas supervisada basada en n-gramas

En esta primera aproximación definimos un sistema básico de CP puramente basado en n-gramas extraídos del corpus de entrenamiento. De esta manera evitamos el uso de herramientas externas y recursos que liguen el sistema a un dominio o idioma en particular. Estudiamos el funcionamiento de diferentes características para la CP. Adicionalmente desarrollamos un conjunto de corpus para entrenar y evaluar sistemas de CP en diferentes idiomas y dominios.

Empleamos SVM como algoritmo de aprendizaje para construir el clasificador. Sus características intrínsecas y diversos estudios previos señalan a este algoritmo como la mejor opción para sistemas de CP basados en corpus. La representación del espacio de características mediante n-gramas nos permitió obtener un sistema fácilmente adaptable a diferentes idiomas, demostrando un comportamiento muy similar en todos los idiomas estudiados. Llevamos a cabo experimentos en inglés, español, italiano y catalán. Los resultados fueron ligeramente mejores en inglés que en el resto de idiomas. Esta diferencia refleja que el español, el italiano y el catalán son idiomas más flexivos, afectando a las estimaciones estadísticas sobre los n-gramas durante el proceso de aprendizaje.

También llevamos a cabo experimentos sobre diferentes dominios. En dominio abierto, las características basadas en n-gramas demostraron obtener buenos resultados en todos los idiomas tratados. La mejor combinación de características se consiguió combinando unigramas y bigramas, obteniendo

una precisión del 81,70 % para inglés, 81,17 % para español, 79,45 % para italiano y 81,03 % para catalán. La diferencia de rendimiento no resultaba estadísticamente significativa al compararla con los resultados obtenidos empleando únicamente unigramas. A la hora de construir un sistema de CP para una aplicaciónreal, resulta más interesante seleccionar vectores de características de menor longitud (unigramas en este caso) ya que el coste computacional depende directamente del tamaño del espacio de características. Por otra parte, los bigramas y trigramas obtuvieron un rendimiento significativamente inferior a los unigramas. Aunque los n-gramas de mayor tamaño definen una mejor representación de la estructura de la oración, demuestran tener unas características estadísticas inferiores para la tarea de CP. De esta forma, a la hora de construir un sistema de CP es recomendable limitar su vector de aprendizaje al uso de unigramas o combinaciones de unigramas y bigramas.

Para mejorar estos resultados experimentamos con diversas técnicas estadísticas de selección de características. Estas técnicas están puramente basadas en información estadísticas y no afectan a la flexibilidad de nuestra aproximación. Estas técnicas no habían sido previamente estudiades en el campo de la CP.

La primera técnica empleada fue el *umbral de frecuencia*, un método que elimina las características poco frecuentes en el corpus. A pesar de su simplicidad, esta técnica reduce considerablemente la dimensionalidad del espacio de características nativo y mejora los resultados de todos los vectores de características, a excepción de los unigramas. Por ejemplo, el mejor resultado obtenido en inglés (81,74 %) era mejor que el original (81,70 %), aunque esta diferencia no era estadísticamente significativa. En cualquier caso, la selección de característica permite eliminar el 76,82 % de las características del vector de aprendizaje original. La reducción del espacio de características decrementa el coste computacional requerido para entrenar y evaluar el sistema. Por tanto, esta forma de selección resulta interesante para sistemas en tiempo real.

Dos técnicas más sofisticadas de selección estudiadas fueron χ^2 e IG. Los experimentos demostraron que ambas técnicas obtenían un rendimiento equivalente en los test llevados a cabo, alcanzando mejor precisión que el *umbral de frecuencia*. Los resultados fueron especialmente buenos para la combinación de unigramas, bigramas y trigramas (81,96 % en inglés). Obtuvimos una mejora significativa para todos los vectores de características (excepto para unigramas) y todos los idiomas con respecto a los experimentos originales sin selección. Esta mejora demostró que el proceso de selección permitió filtrar características ruidosas. Más aún, esta selección redujo el tamaño del espacio de características a menos de la mitad del conjunto original. Aunque los mejores resultados con IG y χ^2 (la combinación mencionada más arriba de unigramas, bigramas y trigramas) fueron mejores que los originales (por ejemplo, la combinación de unigramas y bigramas en

inglés alcanzó 81,70 %), esta diferencia no fue estadísticamente significativa. A diferencia del *umbral de frecuencia*, IG y χ^2 son procesos costosos. A la hora de aplicar estas técnicas debe considerarse si el esfuerzo de emplearlas compensa la mejora obtenida.

Pese a que los resultados obtenidos con n-gramas son buenos, queda aún lugar para la mejora de los sistemas de CP en dominio abierto. Esta mejora puede ser alcanzada mediante recursos y herramientas lingüísticas, pero esto sin duda afecta a la portabilidad del sistema. La segunda aproximación, descrita en la siguiente sección, afronta este problema.

Además de los experimentos en dominio abierto, hicimos una evaluación sobre dominio restringido empleando un corpus construido a base de preguntas relacionadas con información turística. Los resultados obtenidos fueron extremadamente buenos en este caso, alcanzando un 95,46 % para la combinación de unigramas y bigramas en inglés, y un 95,65 % en español. Estos resultados reflejan que, en dominio restringido, la variabilidad de los términos y expresiones es menor que en dominio abierto. Por tanto, podemos desarrollar un sistema de CP de alto rendimiento en este tipo de dominios basándonos puramente en información extraída del corpus de entrenamiento. En este caso, la inclusión de información o herramientas externas debe ser claramente justificable, ya que hay poco espacio para la mejora.

Trabajo futuro

- *Investigar otras características de aprendizaje derivadas del corpus.* Algunas aproximaciones previas a CP han obtenido diversas características del corpus que, en algunas circunstancias, demostraron mejorar el rendimiento de los sistemas de CP: dar un peso extra a las dos primeras palabras del corpus, combinaciones de los primeros n-gramas de la pregunta, la longitud (en número de palabras) de la oración, la presencia de letras en mayúsculas o el número de caracteres en la pregunta. Al obtenerse directamente del corpus de entrenamiento, estas características podrían ser incorporadas a nuestro sistema sin poner en peligro su capacidad de adaptación a otros contextos.

- *Aplicar otros métodos de selección y extracción de características.* Además de los métodos descritos en esta primera aproximación para la selección de características, hay otros métodos estadísticos que se pueden aplicar de forma previa al proceso de aprendizaje. Por ejemplo, planeamos emplear *principal component analysis* (PCA) para llevar a cabo la extracción de características en el espacio nativo de aprendizaje. El problema de aplicar PCA es su elevado coste computacional. Una posible solución para poder ser aplicado adecuadamente sería reducir previamente el espacio de características aplicando la selección mediante el umbral de frecuencias.

7.2. Clasificación de preguntas semisupervisada explotando textos no etiquetados

En la aproximación previa, la representación mediante n-gramas de las preguntas no permitía a los sistemas de CP afrontar de forma apropiada el problema de la ambigüedad y la variación lingüística en dominio abierto. Estudios previos en el campo de la CP han demostrado la importancia de la información semántica en esta tarea. En esta segunda aproximación, enriquecíamos el modelo básico de n-gramas incluyendo información semántica. Para preservar la flexibilidad de esta aproximación, la información fue adquirida por entero a partir de texto no etiquetado obtenido de la Web.

Empleamos kernels compuestos para incorporar relaciones semánticas entre palabras y extender la representación del espacio de características basada en bolsa de palabras. Definimos un kernel basado en semántica latente (LSA) para obtener una función generalizada de similitud entre preguntas. El modelo fue obtenido íntegramente a partir de documentos no etiquetados de la Wikipedia, dando como resultado un sistema flexible y fácilmente adaptable a diferentes idiomas y dominios. El uso de Wikipedia en este trabajo nos permitió definir de forma satisfactoria un modelo estadístico en dominio abierto.

Evaluamos esta aproximación en inglés y español. Obtuvimos una mejora significativa con respecto al modelo básico de bolsa de palabras. El sistema fue evaluado sobre el corpus UIUC, un conjunto de preguntas en inglés sobre dominio abierto ampliamente utilizado en el campo de la CP. Tradujimos el corpus completo al español para poder repetir los experimentos en otro idioma. Este corpus nos permitió comparar nuestra aproximación con otros sistemas lingüísticamente motivados. Nuestro sistema obtuvo mejor rendimiento que numerosas aproximaciones previas que hacían un uso intensivo de recursos lingüísticos y herramientas.

Nuestra aproximación combina el aprendizaje supervisado a partir de corpus con la adquisición no supervisada de información semántica a partir de texto no etiquetado. Esta aproximación semisupervisada demostró mejorar aproximaciones previas a la CP semisupervisada que empleaban corpus de preguntas no etiquetadas. Nuestra aproximación, además de la mejora de rendimiento, resulta más factible en la práctica al no requerir de corpus de preguntas sin etiquetar (difíciles de obtener) sino simplemente documentos para construir la matriz de términos y documentos en el espacio semántico latente. En nuestros experimentos, empleamos un conjunto de 50.000 documentos de la Wikipedia. En cualquier caso, dada la vasta cantidad de documentos electrónicos en la Web, resulta más fácil obtener un conjunto amplio de documentos en cualquier idioma que un conjunto amplio de preguntas.

Trabajo futuro

- *Investigar el efecto de variar el corpus, el número de documentos y las dimensiones empleadas para definir el espacio semántico.* Queremos evaluar cómo el uso de texto plano obtenido de la Web, en lugar de páginas de la Wikipedia, puede afectar el rendimiento de esta aproximación. Además, queremos evaluar el efecto de seleccionar diferentes dimensiones para obtener el espacio semántico latente. Es sobradamente conocido que la calidad del espacio semántico depende directamente del número de dimensiones empleados. La variación del número de documentos empleados para construir el espacio semántico latente también merece un estudio más detallado.

- *Evaluar nuestra aproximación en dominios restringidos.* Aunque los modelos de n-gramas han demostrado obtener una alta precisión en dominios restringidos, es interesante comprobar si es posible mejorar aún más estos sistemas. Planeamos definir el espacio semántico usando corpus específicos sobre un dominio restringido, o empleando secciones concretas de la Wikipedia.

- *Extender la evaluación a otros idiomas.* El espacio semántico fue construido obteniendo 50.000 documentos de la Wikipedia. Existen casi 40 idiomas diferentes en la Wikipedia con más de 50.000 artículos disponibles.[1] Todos estos artículos son candidatos potenciales para modelar el espacio semántico y extender nuestra evaluación a otros idiomas relevantes.

7.3. Clasificación de preguntas mínimamente supervisada sobre taxonomías refinadas

En esta tercera aproximación afrontamos la necesidad de sistemas de CP sobre taxonomías refinadas. Los sistemas de CP basados en corpus dependen enormemente del tamaño del corpus de entrenamiento. A medida que el número de clases en la taxonomía aumenta, hay necesidad de más ejemplos de entrenamiento para mantener el rendimiento del sistema. Definimos para ello una aproximación mínimamente supervisada, evitando la necesidad de grandes corpus de entrenamiento para obtener sistemas de CP sobre taxonomías refinadas. En esta aproximación sólo es necesario un conjunto de semillas (5 en nuestros experimentos) para caracterizar las clases del problema. Empleamos estas semillas para recuperar fragmentos de texto de la Web relacionados con estas clases. Definimos un algoritmo propio, llamado DC2, que extraía y ponderaba términos altamente relacionados con las clases y altamente discriminatorios entre ellas. Este algoritmo estaba basado en la

[1] http://meta.wikimedia.org/wiki/List_of_Wikipedia.

divergencia de JS de la distribución de Poisson de los términos en diferentes fragmentos asociados a las clases.

Los términos extraidos fueron empleados para identificar las clases de las preguntas de evaluación siguiendo un algoritmo sencillo, tradicionalmente empleado para medir la relevancia de las consultas y los documentos en sistemas de RI. Esta aproximación está completamente basada en información estadística y no requiere de recursos lingüísticos o herramientas adicionales. Más aún, como los textos fueron obtenidos automáticamente de la Web, ni tan sólo resulta necesario un corpus de entrenamiento.

Llevamos a cabo un experimento sobre un refinamiento de 14 subclases de *persona*. Para evaluar la independencia del idioma de esta aproximación llevamos a cabo experimentos tanto en inglés como en español. Obtuvimos resultados prometedores en esta tarea (una precisión de 85,37 % en inglés y 79,02 % en español). El algoritmo demostró manejar de forma adecuada características ruidosas, obteniendo mejores resultados que SVM en esta tarea (70,73 % en inglés y 65,37 % español). También obtuvimos mejores resultados que otro método estadístico de selección de características como χ^2 (79,51 % en inglés y 77,07 % en español).

Aunque nuestro algoritmo demostró funcionar adecuadamente en entornos con numerosas características irrelevantes, la cantidad y calidad de los fragmentos recuperados de la Web claramente afecta el rendimiento del sistema. Por ejemplo, el número de fragmentos por semilla recuperadas para inglés fue un 20 % superior que para español, lo cual sin duda afectó al rendimiento final obtenido por DC2 en uno y otro idioma. En cualquier caso, este rendimiento no sólo depende de la cantidad de fragmentos sino de la calidad de los mismos. Esta calidad está fuertemente relacionada con las semillas empleadas para recuperar los fragmentos. No siempre es posible definir con facilidad las semillas y obtener un conjunto adecuado para caracterizar una clase.

En esta tercera aproximación, el esfuerzo requerido para construir el sistema de CP radica en definir un conjunto representativo de semillas. Una vez que este conjunto se define, el sistema se construye automáticamente.

Trabajo futuro

- *Ampliar el escenario de evaluación.* Hemos experimentado con una taxonomía refinada sobre la clase *persona*. Planeamos extender los experimentos a otras taxonomías refinadas para proporcionar una cobertura amplia en diferentes escenarios. Por ejemplo, la clase *lugar* resulta apta para este tipo de refinamiento (*ciudad*, *país*, *región*, *provincia*,...).

- *Mejorar la calidad de las fuentes de datos.* Hemos definido un conjunto de semillas para recuperar de forma automática fragmentos de la Web.

El rendimiento final del algoritmo depende directamente de la calidad de estos fragmentos. La noción de semilla que empleamos es demasiado relajada: una entidad que representa una posible respuesta a una pregunta. Por ejemplo, empleamos la semilla `Robert De Niro` para caracterizar a la clase *actor*. Interrogar la Web con esta semilla puede proporcionar muchos fragmentos relacionados con Robert De Niro como actor pero asociados a otras facetas de su vida (como aspectos personales que den como resultado fragmentos ruidosos). Planeamos expandir la definición de semilla para incluir más de una entidad (por ejemplo, juntar un actor y una película suya (`Robert De Niro` y `Toro salvaje`) como una forma de recuperar fragmentos más específicos de la clase que queremos caracterizar.

- *Emplear DC2 en otras tareas de PLN.* Aunque empleamos DC2 para clasificar preguntas, podemos considerarlo como un algoritmo de propósito general para la caracterización de clases. El resultado del algoritmo DC2 es una lista de términos ponderados semánticamente relacionados con un conjunto de clases. Este tipo de lexicones semánticos han sido empleados en diversas tareas de PLN como IR (Voorhees, 1993), BR (Fleischman et al., 2003) o la extracción de relaciones(Brin, 1999). Planeamos emplear esta información para la mejora de otras tareas de PLN.

7.4. Principales aportaciones

El trabajo desarrollado en esta tesis ha sido parcialmente publicado en diversos congresos y foros. Enumeramos en las siguientes líneas las principales contribuciones.

Algoritmos y características de aprendizaje

- A multilingual SVM-based question classification system. *4th Mexican International Conference on Artificial Intelligence, MICAI 2005* (Bisbal et al., 2005b). Este trabajo presenta un sistema multilingüe de CP basado en SVM. Empleamos características superficiales para entrenar y evaluar el sistema en inglés y español.

- Multilingual question classification based on surface text features. *Frontiers in Artificial Intelligence and Applications, IOS Press, 2005* (Bisbal et al., 2005a). Este trabajo presenta una aproximación multilingüe a la CP basado en técnicas de aprendizaje automático. Comparamos el rendimiento de diversas características textuales superficiales empleando tres algoritmos diferentes (SVM, IB1 y ME). Llevamos a cabo experimentos en inglés y español.

- Automatic feature extraction for question classification based on dissimilarity of probability distributions. *5th International Conference on NLP, FinTAL 2006* (Tomás et al., 2006a). En este trabajo presentamos un método para ponderar de forma automática la relevancia de los n-gramas que aparecen en la pregunta. A cada palabra de la pregunta se le asigna un peso dependiendo de su distribución en un corpus de preguntas y en un corpus de documentos. Llevamos a cabo una selección de características basados en estos pesos para reducir la dimensionalidad del espacio de características. El sistema se evaluó tanto en inglés como en español.

Desarrollo de corpus

- Desarrollo de un corpus de entrenamiento para sistemas de Búsqueda de Respuestas basados en aprendizaje automático. *Procesamiento del Lenguaje Natural, 2006* (Tomás et al., 2006c). Este trabajo describe el desarrollo de un corpus en inglés de pares de preguntas/respuestas factuales. El corpus obtenido consistía en más de 70.000 muestras conteniendo cada uno la siguiente información: una pregunta, su clase, una respuesta exacta a la pregunta, los diferentes contextos (oración, párrafo y documento) donde la respuesta tiene lugar dentro del documento, y una etiqueta indicando si la respuesta era correcta (muestra positiva) o no (muestra negativa). El corpus fue diseñado para entrenar en cada fase de un sistema de BR basado en corpus: CP, RI, extracción y validación de la respuesta.

- A parallel corpus labeled using open and restricted domain ontologies. *10th International Conference, CICLing 2009* (Boldrini et al., 2009). Este trabajo describe el desarrollo de un corpus paralelo de 4.500 preguntas en inglés y español sobre el dominio turístico obtenidas a partir de usuarios reales. Las preguntas fueron etiquetadas con el tipo de respuesta esperada, empleando para ello dos taxonomías diferentes: una taxonomía en dominio abierto y otra en dominio restringido sobre turismo.

Aproximación semisupervisada

- A semi-supervised approach to question classification. *17th European Symposium on Artificial Neural Networks, ESANN 2009* (Tomás y Giuliano, 2009). En este trabajo presentamos una aproximación semisupervisada a CP. Empleamos una función kernel basada en información semántica latente obtenida a partir de texto no etiquetado. Este kernel permitió incluir conocimiento semántico externo en el proceso supervisado de aprendizaje. Obtuvimos de esta manera un

CP altamente efectivo combinando este conocimiento con una aproximación basada en bolsa de palabras por medio de kernels compuestos. Ya que la información semántica fue adquirida a partir de texto no etiquetado, este sistema puede ser fácilmente adaptado a diferentes idiomas y dominios. Los experimentos se realizaron sobre preguntas en inglés y castellano.

Aplicación a la BR

- An XML-based system for Spanish question answering. *6th Workshop of the Cross-Language Evaluation Forum, CLEF 2005* (Tomás et al., 2006b). Este trabajo describe nuestra participación en la campaña QA@CLEF 2005. Desarrollamos un sistema modular basado en XML para la integración, combinación y evaluación de diferentes módulos de BR. Evaluamos un sistema de CP basado en SVM y diversas heurísticas para la extracción de respuestas. Tomamos parte en la tarea monolingüe en español.
- Re-ranking passages with LSA in a question answering system. *7th Workshop of the Cross-Language Evaluation Forum, CLEF 2006* (Tomás y Vicedo, 2007b). Este trabajo describe nuestra participación en la campaña QA@CLEF 2006. Extendimos nuestra participación previa con un sistema de reordenación de pasajes basado en LSA sobre el módulo de RI. Tomamos parte en la tarea monolingüe en español y en la translingüe en español-inglés.

Aplicación de la CP a otras tareas

- Multiple-taxonomy question classification for category search on faceted information. *Text, Speech and Dialogue, 10th International Conference, TSD 2007* (Tomás y Vicedo, 2007a). Este trabajo presentó una aproximación novedosa a la CP, afrontando el desafío de asignar categorías en múltiples taxonomías a preguntas formuladas en lenguaje natural. Aplicamos este sistema a la búsqueda de categorías sobre información facetada. El sistema proporciona un interfaz en lenguaje natural sobre información facetada, detectando las categorías solicitadas por el usuario y reduciendo la búsqueda sobre documentos a aquellos pertenecientes a los valores de las facetas identificadas. Los modelos necesarios para detectar categorías fueron directamente inferidos de un corpus de documentos.

7.5. Proyectos de investigación

Esta tesis se ha desarrollado en el marco de los siguientes proyectos de investigación:

- R2D2: Recuperación de respuestas en documentos digitalizados (*Ministerio de Ciencia y Tecnología, TIC2003-07158-C04-01*). El objetivo principal de este proyecto es el desarrollo y evaluación de sistemas de búsqueda de respuestas y recuperación de documentos en escenarios multilingües.
- TEXT-MESS: Minería de Textos Inteligente, Interactiva y Multilingüe basada en Tecnología del Lenguaje Humano (*Ministerio de Educación y Ciencia, TIN2006-15265-C06-01*). El objetivo principal de este proyecto es mejorar el acceso a la información textual mediante el uso de técnicas de procesamiento del lenguaje natural en ámbitos como la búsqueda de respuestas, la minería de datos o la recuperación de información.
- QALL-ME: Question Answering Learning technologies in a multi-Lingual and Multimodal Environment (*Sexto Programa Marco de Investigación de la Unión Europea, FP6-IST-033860*). El principal objetivo de este proyecto es establecer una infraestructura para la búsqueda de respuestas multilingüe y multimodal en dominio abierto para dispositivos móviles.

Glosario de acrónimos

BOW	Bolsa de palabras (*bag-of-words*)
BR	Búsqueda de respuestas
CLEF	*Cross Language Evaluation Forum*
CP	Clasificación de preguntas
DC2	*Dual Corpus Divergence Comparison*
IA	Inteligencia artificial
IG	*Information gai*n
***k*-NN**	*k-nearest neighbors*
LSA	Análisis de la semántica latente (*latent semantic analysis*)
LSI	*Latent semantic indexing*
ME	Máxima entropía
MI	*Mutual information*
NB	*Naive Bayes*
NTCIR	*NII-NACSIS Test Collection for IR Systems*
PLN	Procesamiento del lenguaje natural
RI	Recuperación de información
SVD	Descomposición en valores singulares (*singular value decomposition*)
SVM	Máquinas de vectores de soporte (*support vector machines*)
TREC	*Text REtrieval Conference*
VSM	Modelo de espacio vectorial (*vector space model*)

A

Corpus de preguntas DC2

1 Who was the lead actress in the movie Sleepless in Seattle?
2 Who played the part of the Godfather in the movie, The Godfather?
3 Who portrayed Jake in the television show, Jake and the Fatman?
4 Who portrayed Fatman in the television show, Jake and the Fatman?
5 Who portrayed Rosanne Rosanna-Dana on the television show Saturday Night Live?
6 Who portrayed the man without a face in the movie of the same name?
7 What's the name of the actress who starred in the movie, Silence of the Lambs?
8 Who played the teacher in Dead Poet's Society?
9 What actress starred in The Lion in Winter?
10 What actor first portrayed James Bond?
11 Which comedian's signature line is Can we talk?
12 Who is the actress known for her role in the movie Gypsy?
13 Who won the Oscar for best actor in 1970?
14 Who starred in The Poseidon Adventure?
15 Who holds the record as the highest paid child performer?
16 What name is horror actor William Henry Pratt better known by?
17 Who did Scott Bakula play in American Beauty?
18 What actor has a tattoo on his right wrist reading Scotland forever?
19 What actress has received the most Oscar nominations?
20 Who has been nominated at least twice for an Academy Award but has never won?
21 What actress played Betsy in Betsy's Weeding?
22 Who was the architect of Central Park?
23 Who built the first pyramid?
24 Who were the architects who designed the Empire State Building?
25 Who was the first American in space?
26 Name the first private citizen to fly in space.
27 Who was the second man to walk on the moon?
28 What was the name of the first Russian astronaut to do a spacewalk?
29 Who was the first woman in space?
30 Name the first Russian astronaut to do a spacewalk.
31 Who was the first Russian astronaut to walk in space?
32 Who was the first Russian to do a spacewalk?
33 Who was the first American to walk in space?
34 List female astronauts or cosmonauts.

35 Name the designer of the shoe that spawned millions of plastic imitations, known as jellies.
36 What costume designer decided that Michael Jackson should only wear one glove?
37 What engineer designed the Erie Canal?
38 Who is the director of the international group called the Human Genome Organization (HUGO) that is trying to coordinate gene-mapping research worldwide?
39 Who is the director of intergovernmental affairs for the San Diego county?
40 Who directed the film Fail Safe?
41 Who is the evil H.R. Director in Dilbert?
42 Who is the Greek God of the Sea?
43 Name one of the major gods of Hinduism?
44 What is the name of a Greek god?
45 Who was the Roman god of the sea?
46 Who was the ancient Egyptian god that was the patron of music and pleasure?
47 Who invented the road traffic cone?
48 Who released the Internet worm in the late 1980s?
49 Who invented the paper clip?
50 Who invented the electric guitar?
51 Who invented television?
52 Who made the first airplane?
53 Who made the first airplane that could fly?
54 Who invented baseball?
55 Who invented the game Scrabble?
56 Who invented basketball?
57 Who was considered to be the father of psychology?
58 Who invented the radio?
59 Who created The Muppets?
60 Who created the character of Scrooge?
61 Who created the character James Bond?
62 Who created the comic strip, Garfield?
63 Who invented The Muppets?
64 Who developed the vaccination against polio?
65 Who developed the Macintosh computer?
66 Who discovered x-rays?
67 Who invented the calculator?
68 Who discovered radium?
69 Who invented the hula hoop?
70 Who invented the slinky?
71 Who invented the telephone?
72 Who invented the instant Polaroid camera?
73 Who discovered oxygen?
74 Who invented Trivial Pursuit?
75 What person developed COBOL?
76 Who was the first person to make the helicopter?
77 Who invented the fishing reel?

78 Who invented the cotton gin?
79 Who created the literary character Phineas Fogg?
80 Who killed Lee Harvey Oswald?
81 Who killed Martin Luther King?
82 Who killed Caesar?
83 What was the man's name who was killed in a duel with Aaron Burr?
84 Who assassinated President McKinley?
85 Who shot Billy the Kid?
86 Who killed John F. Kennedy?
87 What is the Boston Strangler's name?
88 Who stabbed Monica Seles?
89 List Hezbollah members killed or apprehended by Israeli forces.
90 Who was responsible for the killing of Duncan in Macbeth?
91 Who is the Queen of Holland?
92 What is the name of the longest ruling dynasty of Japan?
93 Who was the first king of England?
94 Who is the emperor of Japan?
95 What king was forced to agree to the Magna Carta?
96 Who is the monarch of the United Kingdom?
97 What monarch signed the Magna Carta?
98 Which king signed the Magna Carta?
99 Who was the king who was forced to agree to the Magna Carta?
100 What king signed the Magna Carta?
101 Who was the king who signed the Magna Carta?
102 What daughter of Czar Nicholas II is said to have escaped death in the Russian revolution?
103 What Chinese Dynasty was during 1412-1431?
104 Who wrote the song, Stardust?
105 Who wrote the song, Silent Night?
106 What is the real name of the singer, Madonna?
107 Who wrote the song, Boys of Summer?
108 Who's the lead singer of the Led Zeppelin band?
109 Who was the founding member of the Pink Floyd band?
110 What American composer wrote the music for West Side Story?
111 Who wrote the hymn Amazing Grace?
112 Which vintage rock and roll singer was known as The Killer?
113 Who was the lead singer for the Commodores?
114 Who composed The Messiah?
115 What is the full name of conductor Seiji?
116 Name singers performing the role of Donna Elvira in performances of Mozart's Don Giovanni.
117 What artist recorded the song At Last in the 40's or 50's?
118 What country singer's first album was titled Storms of Life?
119 Who were the great opera tenors of the twentieth century?
120 What country artist is nicknamed Tater?
121 What composer wrote Die Gotterdammerung?
122 Who painted Olympia?
123 Who painted the ceiling of the Sistine Chapel?
124 Who painted Sunflowers?

125 Who was the first Taiwanese President?
126 Who is the president of Stanford University?
127 Who was President of Costa Rica in 1994?
128 Who followed Willy Brandt as chancellor of the Federal Republic of Germany?
129 Who is the President of Ghana?
130 Who won the first general election for President held in Malawi in May 1994?
131 Who is the president of the Spanish government?
132 Who was the Democratic nominee in the American presidential election?
133 Who was President of Afghanistan in 1994?
134 Who was the 16th President of the United States?
135 Who was the president of Vichy France?
136 Who was the first U.S. president ever to resign?
137 Who was the 33rd president of the United States?
138 Who was the 21st U.S. President?
139 Who is the president of Bolivia?
140 Who was the oldest U.S. president?
141 Who was the tallest U.S. president?
142 Which U.S. President is buried in Washington, D.C.?
143 Who was elected president of South Africa in 1994?
144 Which president was unmarried?
145 Who was the 23rd president of the United States?
146 Who was the first U.S. president to appear on TV?
147 Who is the only president to serve 2 non-consecutive terms?
148 Who was the first vice president of the U.S.?
149 Who was the first US President to ride in an automobile to his inauguration?
150 Which U.S.A. president appeared on Laugh-In?
151 Who was president in 1913?
152 Who was the 22nd President of the US?
153 What female leader succeeded Ferdinand Marcos as president of the Philippines?
154 Who succeeded Ferdinand Marcos?
155 What president is on a quarter?
156 Who was the U.S. president in 1929?
157 What president declared Mothers'Day?
158 Who is the president pro tempore of the US Senate?
159 Who was the first woman to run for president?
160 Which president was sworn into office on an airplane?
161 Who was elected President of South Africa in 1994?
162 Who was the first president to speak on the radio?
163 What president served 2 nonconsecutive terms?
164 Which U.S. presidents have died while in office?
165 What president is on the thousand dollar bill?
166 What 20th century American president died at Warm Springs, Georgia?
167 What president created social security?
168 Who may be best known for breaking the color line in baseball?
169 Who won two gold medals in skiing in the Olympic Games in Calgary?

170 Who is the fastest swimmer in the world?
171 Name a female figure skater.
172 Who was the first African American to play for the Brooklyn Dodgers?
173 Who was the first person to run the mile in less than four minutes?
174 What was the first name of the Gehrig who played for the New York Yankees?
175 Who was the only golfer to win the U.S. and British Opens and amateurs in the same year?
176 Who was the baseball player given the nickname Mr. October?
177 Which baseball star stole 130 bases in 1982?
178 Who was the first Triple Crown Winner?
179 Who has the most no hitters in major league baseball?
180 Who was the first black heavyweight champion?
181 Which past and present NFL players have the last name of Johnson?
182 What Boston Red Sox infielder was given his father's first name, with the letters reversed?
183 What player has been the home run champion of the national league seven times?
184 What major league baseball player has 511 pitching victories?
185 Who wrote The Pines of Rome?
186 Who wrote Dubliners?
187 Who wrote Hamlet?
188 Who wrote the Farmer's Almanac?
189 Who was the author of the book about computer hackers called The Cuckoo's Egg: Tracking a Spy Through the Maze of Computer Espionage?
190 Who was Samuel Johnsons's friend and biographer?
191 Who coined the term cyberspace in his novel Neuromancer?
192 Who wrote the book, The Grinch Who Stole Christmas?
193 Who wrote the book, Song of Solomon?
194 Who wrote the book, Huckleberry Finn?
195 Who wrote The Scarlet Letter?
196 Who won the nobel prize in literature in 1988?
197 Who wrote The Pit and the Pendulum?
198 Who wrote An Ideal Husband?
199 Who is a German philosopher?
200 Who wrote The Devine Comedy?
201 Who was the first African American to win the Nobel Prize in literature?
202 What famous Spanish poet died in Spain's Civil War?
203 What author wrote under the pen name Boz?
204 Who wrote Fiddler on the Roof?
205 Who is the author of the poem The Midnight Ride of Paul Revere?

B

Conjunto de semillas DC2

actor	Tom Cruise	*diseñador*	John Galliano
	Kathy Bates		Christian Lacroix
	Christina Ricci		Karl Lagerfeld
	Andy Kaufman		Gianni Versace
	Robert De Niro		Ralph Lauren
arquitecto	Le Corbusier	*escritor*	Jack Kerouac
	Santiago Calatrava		Henry Miller
	Frank Gehry		Franz Kafka
	Norman Foster		William Blake
	Renzo Piano		Michael Ende
asesino	John Wilkes Booth	*inventor*	Guglielmo Marconi
	Lee Harvey Oswald		Samuel Morse
	Mark David Chapman		Alfred Nobel
	James Earl Ray		Thomas Edison
	Nathuram Godse		Johann Gutenberg
astronauta	Neil Armstrong	*monarca*	Carl XVI Gustaf
	Buzz Aldrin		Juan Carlos I
	Yuri Gagarin		Elizabeth II
	Pedro Duque		Mohammed VI
	Valentina Tereshkova		Akihito
deportista	Michael Jordan	*músico*	George Frederic Handel
	Carl Lewis		Wolfgang Amadeus Mozart
	Ty Cobb		Bruce Springsteen
	Tiger Woods		Johnny Cash
	Joe Montana		Bob Dylan
dios	Ganesha	*pintor*	Pablo Picasso
	Allah		Edvard Munch
	Ra		Gustav Klimt
	Jehovah		Claude Monet
	Zeus		Vincent van Gogh
director	Milos Forman	*presidente*	Joseph Estrada
	Martin Scorsese		Vladimir Putin
	Francis Ford Coppola		George Bush
	Stanley Kubrick		Nicolas Sarkozy
	Steven Spielberg		Thabo Mbeki

Bibliografía

Abney, S. (1997). Part-of-speech tagging and partial parsing. In *Corpus-Based Methods in Language and Speech*, pages 118–136. Kluwer Academic Publishers. 2.2.2

Agirre, E. y Edmonds, P., editors (2006). *Word Sense Disambiguation: Algorithms and Applications*, volume 33 of *Text, Speech and Language Technology*. Springer. 2.2.1

Allwein, E. L., Schapire, R. E. y Singer, Y. (2001). Reducing multiclass to binary: a unifying approach for margin classifiers. *The Journal of Machine Learning Research*, 1:113–141. 3.4.1

Alpaydin, E. (2004). *Introduction to Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press. 2.3.2

An, J., Lee, S. y Lee, G. G. (2003). Automatic acquisition of named entity tagged corpus from world wide web. In *ACL '03: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 165–168, Morristown, NJ, USA. Association for Computational Linguistics. 6.4

Androutsopoulos, L., Ritchie, G. D. y Thanisch, P. (1995). Natural language interfaces to databases - an introduction. *Journal of Natural Language Engineering*, 1:29–81. 2.2.1

Argamon, S., Dagan, I. y Krymolowski, Y. (1998). A memory-based approach to learning shallow natural language patterns. In *Proceedings of the 17th international conference on Computational linguistics*, pages 67–73, Morristown, NJ, USA. Association for Computational Linguistics. 3.4.6

Atserias, J., Zaragoza, H., Ciaramita, M. y Attardi, G. (2008). Semantically annotated snapshot of the english wikipedia. In (ELRA), E. L. R. A., editor, *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. 5.3.2

Aunimo, L. y Kuuskoski, R. (2005). Reformulations of finnish questions for question answering. In *Proceedings of the 15th NODALIDA conference*, pages 12–21. 3.2

Austin, J. L. (1962). *How to Do Things With Words*. Harvard University Press. 2.1

Ayer, A. J. (1932). *Language, Truth and Logic*. Gollancz, London. 2.5.4

Baeza-Yates, R. y Ribeiro-Neto, B. (1999). *Modern Information Retrieval*. ACM Press/Addison Wesley. 1, 2.2.1

Beitzel, S. M., Jensen, E. C., Lewis, D. D., Chowdhury, A. y Frieder, O. (2007). Automatic classification of web queries using very large unlabeled query logs. *ACM Transactions on Information Systems*, 25(2):9. 3.2

Berger, A. L., Pietra, V. J. D. y Pietra, S. A. D. (1996). A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71. 3.4.3

Bikel, D. M., Schwartz, R. L. y Weischedel, R. M. (1999). An algorithm that learns what's in a name. *Machine Learning*, 34(1-3):211–231. 3.1

Bisbal, E., Tomás, D., Moreno, L., Vicedo, J. L. y Suárez, A. (2005a). *Artificial Intelligence Research and Development*, volume 131 of *Frontiers in Artificial Intelligence and Applications*, chapter Multilingual Question Classification based on surface text features, pages 255–261. IOS Press, Amsterdam, The Netherlands. 3.3.2, 3.4.1, 3.4.3, 3.4.6, 3.6, 4.2, 4.6, 7.4

Bisbal, E., Tomás, D., Moreno, L., Vicedo, J. L. y Suárez, A. (2005b). A multilingual svm-based question classification system. In Gelbukh, A. F., de Albornoz, A., and Terashima-Marín, H., editors, *MICAI 2005: Advances in Artificial Intelligence, 4th Mexican International Conference on Artificial Intelligence*, volume 3789 of *Lecture Notes in Computer Science*, pages 806–815. Springer. 3.3.1, 3.3.2, 3.4.1, 7.4

Blunsom, P., Kocik, K. y Curran, J. R. (2006). Question classification with log-linear models. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 615–616, New York, NY, USA. ACM. 3.3.4, 3.4.3, 5, 5.4

Boldrini, E., Ferrández, S., Izquierdo, R., Tomás, D. y Vicedo, J. L. (2009). A parallel corpus labeled using open and restricted domain ontologies. In Gelbukh, A. F., editor, *Computational Linguistics and Intelligent Text Processing, 10th International Conference, CICLing 2009*, volume 5449 of *Lecture Notes in Computer Science*, pages 346–356. Springer. 4.1.2, 7.4

Breck, E., Burger, J. D., Ferro, L., House, D., Light, M. y Mani, I. (1999). A sys called qanda. In TREC-8 (1999), pages 499–506. 2.3.1, 3.1, 3.1

Briggs, T. y Oates, T. (2005). Discovering domain-specific composite kernels. pages 732–738. AAAI Press / The MIT Press. 5.2.3

Brill, E., Lin, J., Banko, M., Dumais, S. y Ng, A. (2001). Data-intensive question answering. In TREC-10 (2001), pages 393–400. 2.5.2

Brin, S. (1999). Extracting patterns and relations from the world wide web. In *WebDB '98: Selected papers from the International Workshop on The*

World Wide Web and Databases, pages 172–183, London, UK. Springer-Verlag. 6.4, 7.3

Broder, A. (2002). A taxonomy of web search. *SIGIR Forum*, 36(2):3–10. 1

Brown, J. (2004). Entity-tagged language models for question classification in a qa system. Technical report, IR Lab. 3.3.4, 3.4.2, 5

Buckley, C. (1985). Implementation of the smart information retrieval system. Technical Report TR85-686, Ithaca, NY, USA. 1

Burke, R. D., Hammond, K. J., Kulyukin, V. A., Lytinen, S. L., Tomuro, N. y Schoenberg, S. (1997). Question answering from frequently asked question files: Experiences with the faq finder system. Technical report, Chicago, IL, USA. 4.1.1

Callan, J. P. (1994). Passage-level evidence in document retrieval. In *SIGIR '94: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 302–310, New York, NY, USA. Springer-Verlag New York, Inc. 2.5.3

Carbonell, J., Hovy, E., Harman, D., Maiorano, S., Prange, J. y Sparck-Jones, K. (2000). Vision statement to guide research in question & answering (Q&A) ans text summarization. Technical report, NIST. 2.5.1

Cardie, C. (1996). Automatic feature set selection for case-based learning of linguistic knowledge. In *Conference on Empirical Methods in Natural Language Processing*, pages 113–126. 4.5.3

Cardie, C. (1997). Empirical methods in information extraction. *AI magazine*, 18:65–79. 2.2.1

Carlson, A., Cumby, C., Rosen, J. y Roth, D. (1999). The snow learning architecture. Technical Report UIUCDCS-R-99-2101, UIUC Computer Science Department. 3.4.5

Chang, C. C. y Lin, C. J. (2001). *LIBSVM: a library for support vector machines*. 3.4.1

Chapelle, O., Schökopf, B. y Zien, A., editors (2006). *Semi-Supervised Learning*. MIT Press, Cambridge, MA. 2.3.2

Cheung, Z., Phan, K. L., Mahidadia, A. y Hoffmann, A. G. (2004). Feature extraction for learning to classify questions. In Webb, G. I. and Yu, X., editors, *AI 2004: Advances in Artificial Intelligence, 17th Australian Joint Conference on Artificial Intelligence*, volume 3339 of *Lecture Notes in Computer Science*, pages 1069–1075, Cairns, Australia. Springer. 2.5.4, 3.1, 3.1, 3.2, 3.3.2, 3.3.3, 3.4.4, 3.4.7

Church, K. W. y Gale, W. A. (1995). Poisson mixtures. *Natural Language Engineering*, 1:163–190. 6.2.2

Clark, S. y Curran, J. R. (2004). Parsing the wsj using ccg and log-linear models. In *ACL '04: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 103, Morristown, NJ, USA. Association for Computational Linguistics. 3.3.4

Clarke, C. L. A., Cormack, G. V. y Lynam, T. R. (2001a). Exploiting redundancy in question answering. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 358–365, New York, NY, USA. ACM. 2.5.3

Clarke, C. L. A., Cormack, G. V., Lynam, T. R., Li, C. M. y McLearn, G. L. (2001b). Web reinforced question answering (multitest experiments for trec 2001). In TREC-10 (2001), pages 673–679. 2.5.2

Clarkson, P. y Rosenfeld, R. (1997). Statistical language modeling using the CMU–cambridge toolkit. In *Proceedings Eurospeech '97*, pages 2707–2710, Rhodes, Greece. 3.4.2

Cohen, W. W. (1996). Learning trees and rules with set-valued features. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence and Eighth Innovative Applications of Artificial Intelligence (AAAI/IAAI)*, pages 709–716. The MIT Press. 3.4.8

Collins, M. y Duffy, N. (2001). Convolution kernels for natural language. In *Advances in Neural Information Processing Systems (NIPS14)*, pages 625–632. MIT Press. 3.4.1

Contreras, H. (1999). *Gramática descriptiva de la lengua española*, volume 2, chapter Relaciones entre las construcciones interrogativas, exclamativas y relativas, pages 1931–1964. Real Academia Espa nola / Espasa Calpe, Madrid. 2.1, 2.5.4

Cooper, R. J. y Rüger, S. M. (2000). A simple question answering system. In TREC-9 (2000), pages 249–255. 2.3.1

Cortes, C. y Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3):273–297. 3.4.1

Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S. y Singer, Y. (2006). Online passive-aggressive algorithms. *The Journal of Machine Learning Research*, 7:551–585. 3.4.8

Cristianini, N. y Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press. 5

Cui, H., Li, K., Sun, R., seng Chua, T. y yen Kan, M. (2004). National university of singapore at the trec 13 question answering main task. In *Thirteenth Text REtrieval Conference*, volume 500-261 of *NIST Special Publication*, pages 34–42, Gaithersburg, USA. National Institute of Standards and Technology. 2.3.1

Cui, H., Mittal, V. O. y Datar, M. (2006). Comparative experiments on sentiment classification for online product reviews. In *AAAI*. AAAI Press. 3.4.5

Daelemans, W. y van den Bosch, A. (2005). *Memory-Based Language Processing*. Studies in Natural Language Processing. Cambridge University Press, New York. 2.3.2

Daelemans, W., Zavrel, J., van der Sloot, K. y van den Bosch, A. (2007). Timbl: Tilburg memory based learner, version 6.1, reference guide. Technical Report 07-07, ILK Research Group. 3.4.6

Dagan, I., Glickman, O. y Magnini, B. (2006). The pascal recognising textual entailment challenge. In nonero Candela, J. Q., Dagan, I., Magnini, B., and dÁlché Buc, F., editors, *Machine Learning Challenges*, volume 3994 of *Lecture Notes in Computer Science*, pages 177–190. Springer-Verlag. 2.2.1

Dagan, I., Lee, L. y Pereira, F. C. N. (1999). Similarity-based models of word cooccurrence probabilities. *Machine Learning*, 34(1-3):43–69. 6.1.2, 6.1.2

Dasarathy, B. V. (1990). *Nearest neighbor (NN) norms: NN pattern classification techniques*. IEEE Computer Society Press. 3.4.6

Day, M.-Y., Lee, C.-W., Wu, S.-H., Ong, C.-S. y Hsu, W.-L. (2005). An integrated knowledge-based and machine learning approach for chinese question classification. *Proceedings of 2005 IEEE International Conference on Natural Language Processing and Knowledge Engineering. IEEE NLP-KE '05*, pages 620–625. 3.2

Day, M.-Y., Ong, C.-S. y Hsu, W.-L. (2007). Question classification in english-chinese cross-language question answering: An integrated genetic algorithm and machine learning approach. *IEEE International Conference on Information Reuse and Integration, 2007. IRI 2007*, pages 203–208. 3.2, 3.3.1, 3.3.2, 3.3.4, 3.4.1, 5

Deerwester, S. C., Dumais, S. T., Landauer, T. K., Furnas, G. W. y Harshman, R. A. (1990). Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407. 5, 5.1

Dietterich, T. G. (1998). Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10:1895–1923. 4.4.3

Duda, R. O. y Hart, P. E. (1973). *Pattern Classification and Scene Analysis*, pages 98–105. John Wiley and Sons. 3.4.7

Dumais, S., Platt, J., Heckerman, D. y Sahami, M. (1998). Inductive learning algorithms and representations for text categorization. In *CIKM '98: Proceedings of the seventh international conference on Information and knowledge management*, pages 148–155, New York, NY, USA. ACM. 3.4.1, 4.5.3

Dumais, S. T. (1994). Latent semantic indexing (lsi) and trec-2. In *The Second Text REtrieval Conference (TREC-2*, pages 105–115. 5.1

Durme, B. V., Huang, Y., Kupść, A. y Nyberg, E. (2003). Towards light semantic processing for question answering. In *Proceedings of the HLT-NAACL 2003 workshop on Text meaning*, pages 54–61, Morristown, NJ, USA. Association for Computational Linguistics. 2.3.1

Escudero, G., Màrquez, L. y Rigau, G. (2000). Naive bayes and exemplar-based approaches to word sense disambiguation revisited. In *ECAI 2000, Proceedings of the 14th European Conference on Artificial Intelligence*. 3.4.7

Feiguina, O. y Kégl, B. (2005). Learning to classify questions. In *CLiNE 05: 3rd Computational Linguistics in the North-East Workshop*. 3.2

Fellbaum, C., editor (1998). *WordNet: An Electronic Lexical Database (Language, Speech, and Communication)*. The MIT Press. 2.5.2

Ferret, O., Grau, B., Illouz, G., Jacquemin, C. y Masson, N. (1999). Qalc - the question-answering program of the language and cognition group at limsi-cnrs. In TREC-8 (1999), pages 465–474. 3.1, 3.1

Fleischman, M., Hovy, E. y Echihabi, A. (2003). Offline strategies for online question answering: answering questions before they are asked. In *ACL '03: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 1–7, Morristown, NJ, USA. Association for Computational Linguistics. 6.4, 7.3

Fleiss, J. L. (1971). Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382. 4.1.1

Forman, G. (2003). An extensive empirical study of feature selection metrics for text classification. *The Journal of Machine Learning Research*, 3:1289–1305. 4.5.3

Freund, Y. y Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139. 4.3

Gabrilovich, E. y Markovitch, S. (2004). Text categorization with many redundant features: using aggressive feature selection to make svms competitive with c4.5. In *ICML '04: Proceedings of the twenty-first international conference on Machine learning*, pages 321–328, New York, NY, USA. ACM. 4.5.3

Gaizauskas, R., Greenwood, M. A., Harkema, H., Mepple, M., Saggion, H. y Sanka, A. (2005a). The university of sheffield's trec 2005 q&a experiments. In *Fourteenth Text REtrieval Conference*, volume 500-266 of *NIST Special Publication*, Gaithersburg, USA. National Institute of Standards and Technology. 3.3.3

Gaizauskas, R., Hepple, M., Saggion, H., Greenwood, M. A. y Humphreys, K. (2005b). SUPPLE: A practical parser for natural language engineering applications. In *Proceedings of the 9th International Workshop on Parsing Technologies (IWPT 2005)*, pages 200–201, Vancouver. 3.3.3

García Cumbreras, M. A., López, L. A. U. y Santiago, F. M. (2006). Bruja: Question classification for spanish. using machine translation and an english classifier. In *EACL 2006 Workshop On Multilingual Question Answering - MLQA06*, pages 39–44. 3.1, 3.3.1, 3.3.2, 3.3.3, 3.3.4, 3.5, 5

García Cumbreras, M. A., Santiago, F. M., López, L. A. U. y Ráez, A. M. (2005). Búsqueda de respuestas multilingüe : clasificación de preguntas en español basada en aprendizaje. *Procesamiento del Lenguaje Natural*, (34):31–40. 3.2

Giozzo, A. y Strapparava, C. (2005). Domain kernels for text categorization. In *Ninth Conference on Computational Natural Language Learning (CoNLL-2005)*, pages 56–63, Ann Arbor, Michigan. 5.2.3, 5.4

Girju, R. (2003). Automatic detection of causal relations for question answering. In *Proceedings of the ACL 2003 workshop on Multilingual summarization and question answering*, pages 76–83, Morristown, NJ, USA. Association for Computational Linguistics. 6.4

Giuliano, C., Lavelli, A. y Romano, L. (2006). Exploiting shallow linguistic information for relation extraction from biomedical literature. In *11th Conference of the European Chapter of the Association for Computational Linguistics (EACL-2006)*, Trento, Italy. 5.2.3

Gliozzo, A., Giuliano, C. y Strapparava, C. (2005). Domain kernels for word sense disambiguation. In *Proceedings of the 43rd Annual Meeting of the*

Association for Computational Linguistics (ACL'05), pages 403–410, Ann Arbor, Michigan. 5

Golding, A. R. y Roth, D. (1999). A winnow-based approach to context-sensitive spelling correction. *Machine Learning*, 34(1-3):107–130. 3.4.5

Graesser, A. C., McMahen, C. L. y Johnson, B. K. (1994). *Handbook of Psycholinguistics*, chapter Question asking and answering, pages 517–538. Academic Press, San Diego, CA. 3.1

Greenwood, M. A. (2005). *Open-Domain Question Answering*. PhD thesis, Department of Computer Science, University of Sheffield, UK. (document), 2.3.1, 3.1, 3.1, 3.3, 3.3.4, 3.4.8, 5

Grishman, R. y Sundheim, B. (1996). Message understanding conference-6: a brief history. In *Proceedings of the 16th conference on Computational linguistics*, pages 466–471, Morristown, NJ, USA. Association for Computational Linguistics. 17

Grivolla, J., Jourlin, P. y Mori, R. D. (2005). Automatic classification of queries by expected retrieval performance. In *ACM SIGIR '05 Workshop: Predicting Query Difficulty - Methods and Applications*. 3.5

Groenendijk, J. y Stokhof, M. (1997). *Questions*, chapter 19, pages 1055–1124. Elsevier/MIT Press, Amsterdam/Cambridge Mass. 2.1

Hacioglu, K. y Ward, W. (2003). Question classification with support vector machines and error correcting codes. In *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 28–30, Morristown, NJ, USA. Association for Computational Linguistics. 3.2, 3.3.1, 3.3.4, 3.4.1, 5, 5.4

Hallebeek, J. (1999). El corpus paralelo. *Procesamiento del Lenguaje Natural*, (24):58–69. 3.2

Harabagiu, S., Moldovan, D., Paşca, M., Mihalcea, R., Surdeanu, M., Bunescu, R. C., Girju, R., Rus, V. y Morarescu, P. (2000). Falcon: Boosting knowledge for answer engines. In TREC-9 (2000), pages 479–488. 2.3.1, 2.5.3, 5.2.2

Hennoste, T., Gerassimenko, O., Kasterpalu, R., Koit, M., Rääbis, A., Strandson, K. y Valdisoo, M. (2005). Questions in estonian information dialogues: Form and functions. In *Text, Speech and Dialogues*, volume 3658, pages 420–427. Springer Berlin / Heidelberg. 3.2

Hermjakob, U. (2001). Parsing and question classification for question answering. In *Workshop on Open-Domain Question Answering at ACL-2001*. 2.3.1, 2.3.1, 3.1, 3.2, 3.3.3, 5.2.2

Hermjakob, U. y Mooney, R. J. (1997). Learning parse and translation decisions from examples with rich context. In *ACL-35: Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, pages 482–489, Morristown, NJ, USA. Association for Computational Linguistics. 3.3.3

Higginbotham, J. (1995). *Handbook of Contemporary Semantic Theory*, chapter The Semantics of Questions, pages 361–383. Blackwell, Oxford. 2.1

Hovy, E., Gerber, L., Hermjakob, U., Junk, M. y Lin, C.-Y. (2000). Question answering in webclopedia. 2.5.2, 3.1

Hovy, E., Gerber, L., Hermjakob, U., Lin, C.-Y. y Ravichandran, D. (2001). Toward semantics-based answer pinpointing. In *HLT '01: Proceedings of the first international conference on Human language technology research*, pages 1–7, Morristown, NJ, USA. Association for Computational Linguistics. 2.5.3

Hovy, E., Hermjakob, U. y Ravichandran, D. (2002). A question/answer typology with surface text patterns. In *Proceedings of the second international conference on Human Language Technology Research*, pages 247–251, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc. 2.5.3, 1, 3.1

Hsu, C. W., Chang, C. C. y Lin, C. J. (2003). A practical guide to support vector classification. Technical report, Taipei. 4.3

Huang, P., Bu, J., Chen, C., Qiu, G. y Zhang, L. (2007). Learning a flexible question classifier. In *ICCIT '07: Proceedings of the 2007 International Conference on Convergence Information Technology*, pages 1608–1613, Washington, DC, USA. IEEE Computer Society. 3.3.1, 3.4.8

Hull, D. A. (1999). Xerox trec-8 question answering track report. In TREC-8 (1999), pages 743–752. 2.3.1

Hutchins, W. J. y Somers, H. L. (1992). *An Introduction to Machine Translation*. Academic Press. 2.2.1

Ittycheriah, A., Franz, M., Zhu, W.-J. y Ratnaparkhi, A. (2000). Ibm's statistical question answering system. In TREC-9 (2000), pages 229–234. 3.1, 3.1, 3.2, 3.4.3, 4.4

Ittycheriah, A., Franz, M., Zhu, W.-J., Ratnaparkhi, A. y Mammone, R. J. (2001). Question answering using maximum entropy components. In *NAACL '01: Second meeting of the North American Chapter of*

the Association for Computational Linguistics on Language technologies 2001, pages 1–7, Morristown, NJ, USA. Association for Computational Linguistics. 2.5.3, 3.4.3

Jelinek, F. (1997). *Statistical methods for speech recognition*. MIT Press, Cambridge, MA, USA. 3.4.2

Jeon, J., Croft, W. B. y Lee, J. H. (2005). Finding semantically similar questions based on their answers. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 617–618, New York, NY, USA. ACM. 2.4, 3.4.2

Joachims, T. (1998). Text categorization with support vector machines: learning with many relevant features. In Nédellec, C. and Rouveirol, C., editors, *Proceedings of ECML-98, 10th European Conference on Machine Learning*, number 1398, pages 137–142, Chemnitz, DE. Springer Verlag, Heidelberg, DE. 2.4, 3.4.1, 4.5.3

Joachims, T. (1999). Making large-scale support vector machine learning practical. pages 169–184. 3.4.1

John Burger, Claire Cardie, V. C. R. G. S. H. D. I. C. J. C.-Y. L. S. M. G. M. D. M. B. O. J. P. E. R. A. S. R. S. T. S. E. V. R. W. (2003). Issues, tasks and program structures to roadmap research in question answering. Technical report, SRI International. 2.5.2

Junqua, J.-C. y Haton, J.-P. (1995). *Robustness in Automatic Speech Recognition: Fundamentals and Applications*. Kluwer Academic Publishers, Norwell, MA, USA. 2.2.1

Juola, P. (2007). Future trends in authorship attribution. In Craiger, P. and Shenoi, S., editors, *IFIP International Conference on Digital Forensics*, volume 242, pages 119–132. 2.2.1

Kando, N. (2005). Overview of the fifth ntcir workshop. In *Proceedings of NTCIR-5 Workshop*, Tokyo, Japan. 1.1

Kaszkiel, M. y Zobel, J. (1997). Passage retrieval revisited. In *SIGIR '97: Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 178–185, New York, NY, USA. ACM. 2.5.3

Kocik, K. (2004). Question classification using maximum entropy models. Master's thesis, School of Information Technologies, University of Sydney. 3.3.1, 3.3.4, 3.3.4, 3.4.3, 3.4.7, 4.5.3, 5

Krishnan, V., Das, S. y Chakrabarti, S. (2005). Enhanced answer type inference from questions using sequential models. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 315–322, Morristown, NJ, USA. Association for Computational Linguistics. 3.2, 3.3.1, 3.3.2, 3.3.4, 3.4.1, 5

Kudo, T. y Matsumoto, Y. (2001). Chunking with support vector machines. In *NAACL*. 3.4.1

Kučera, H. y Francis, W. N. (1967). *Computational Analysis of Present Day American English*, volume 19 of *American Documentation*. Brown University Press, Providence, RI. 4.2

Landauer, T. K. y Dumais, S. T. (1997). A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, 104(2):211–240. 5.1

Landauer, T. K., Foltz, P. W. y Laham, D. (1998). An introduction to latent semantic analysis. *Discourse Processes*, (25):259–284. 5.1

Lee, K.-S., Oh, J.-H., Huang, J.-X., Kim, J.-H. y Choi, K.-S. (2000). Trec-9 experiments at kaist: Qa, clir and batch filtering. In TREC-9 (2000), pages 303–316. 2.3.1

Lehnert, W. G. (1977a). Human and computational question answering. *Cognitive Science: A Multidisciplinary Journal*, 1(1):47–73. 2.5.1

Lehnert, W. G. (1977b). *The process of question answering*. PhD thesis, New Haven, CT, USA. 3.1

Lehnert, W. G. (1980). Question answering in natural language procesing. In Verlag, C. H., editor, *Natural Language Question Answering Systems*, pages 9–71. 2.5.1

Lewis, D. D. (1992). Feature selection and feature extraction for text categorization. In *HLT '91: Proceedings of the workshop on Speech and Natural Language*, pages 212–217, Morristown, NJ, USA. Association for Computational Linguistics. 4.5.3

Li, W. (2002). Question classification using language modeling. Technical report, Center for Intelligent Information Retrieval, University of Massachusetts, Amherst, MA 01003. 3.2, 3.4.2

Li, X., Huang, X. y Wu, L. (2005). Question classificaion using multiple classifiers. In *Workshop On Asian Language Resources ALR And First Symposium On Asian Language Resources Network ALRN*, pages 64–70. 3.3.1, 3.3.4, 3.4.1, 5

Li, X. y Roth, D. (2002). Learning question classifiers. In *Proceedings of the 19th international conference on Computational linguistics*, pages 1–7, Morristown, NJ, USA. Association for Computational Linguistics. 1.1, 3.1, 3.1, 3.2, 3.3.1, 3.3.2, 3.3.3, 3.3.4, 3.3.4, 3.4.5, 4.4.1, 5, 5.2.2, 5.3.1, 5.4, 6

Li, X. y Roth, D. (2005). Learning question classifiers: the role of semantic information. *Natural Language Engineering*, 12(3):229–249. 2.3, 2, 3.2, 3.3.3, 3.3.4, 3.3.4, 3.4.5, 4.4.1, 5

Lin, D. (1998). Dependency-based evaluation of minipar. In *Workshop on the Evaluation of Parsing Systems*. 3.3.3

Lin, D. y Pantel, P. (2001). Discovery of inference rules for question-answering. *Natural Language Engineering*, 7(4):343–360. 6.4

Lin, J. (1991). Divergence measures based on the shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151. 6.1.2

Lin, X.-D., Peng, H. y Liu, B. (2006). Support vector machines for text categorization in chinese question classification. In *WI '06: Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*, pages 334–337. 3.2

Magerman, D. M. (1995). Statistical decision-tree models for parsing. In *ACL*, pages 276–283. 3.4.4

Magnini, B., Negri, M., Prevete, R. y Tanev, H. (2002). Mining knowledge from repeated co-occurrences: Diogene at trec 2002. In TREC-11 (2002). 2.5.3

Magnini, B., Romagnoli, S., Vallin, A., Herrera, J., Peñas, A., Peinado, V., Verdejo, F. y de Rijke, M. (2003). Creating the disequa corpus: A test set for multilingual question answering. In *Cross-Lingual Evaluation Forum (CLEF) 2003 Workshop*, pages 311–320. 3.2

Magnini, B., Vallin, A., Ayache, C., Erbach, G., Pe nas, A., de Rijke, M., Rocha, P., Simov, K. I. y Sutcliffe, R. F. E. (2005). Overview of the clef 2004 multilingual question answering track. In Peters, C., Clough, P., Gonzalo, J., Jones, G. J. F., Kluck, M., and Magnini, B., editors, *Multilingual Information Access for Text, Speech and Images, 5th Workshop of the Cross-Language Evaluation Forum, CLEF 2004*, volume 3491 of *Lecture Notes in Computer Science*, pages 371–391. Springer. 3.2

Mahesh, K. y Nirenburg, S. (1995). A situated ontology for practical nlp. In *In Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing, International Joint Conference on Artificial Intelligence (IJCAI-95*. 2.5.2

Mani, I. (1999). *Advances in Automatic Text Summarization*. MIT Press, Cambridge, MA, USA. 2.2.1

Manning, C. D., Raghavan, P. y Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press. 2.4, 3.1

Manning, C. D. y Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA. 2.2.4, 3.4.2, 4.2, 6.1.1, 6.2.2

Maynard, D., Tablan, V., Cunningham, H., Ursu, C., Saggion, H., Bontcheva, K. y Wilks, Y. (2002). Architectural elements of language engineering robustness. *Natural Language Engineering*, 8(3):257–274. 3.3.4

Mccallum, A. y Nigam, K. (1998). A comparison of event models for naive bayes text classification. In *AAAI Workshop on Learning for Text Categorization*, pages 41–48. AAAI Press. 3.4.7

Metzler, D. y Croft, W. B. (2005). Analysis of statistical question classification for fact-based questions. *Information Retrieval*, 8(3):481–504. (document), 2, 3.1, 3.2, 3.2, 3.3.1, 3.3.2, 3.3.4, 3.3.4, 3.4.1, 5

Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill Science/Engineering/Math. 2.3.2, 3.4.7, 4.5.3

Moldovan, D., Harabagiu, S., Girju, R., Morarescu, P., Lacatusu, F., Novischi, A., Badulescu, A. y Bolohan, O. (2002). Lcc tools for question answering. In TREC-11 (2002), pages 144–155. 2.5.3

Moldovan, D., Harabagiu, S., Paşca, M., Mihalcea, R., Goodrum, R., Girju, R. y Rus, V. (1999). Lasso: A tool for surfing the answer net. In TREC-8 (1999), pages 175–183. 2.3.1, 3.3.4

Moldovan, D., Paşca, M., Harabagiu, S. y Surdeanu, M. (2003). Performance issues and error analysis in an open-domain question answering system. *ACM Transactions on Information Systems*, 21(2):133–154. 2.5.3

Moreno, L., Palomar, M., Molina, A. y Ferrández, A. (1999). *Introducción al Procesamiento del Lenguaje Natural*. Servicio de publicaciones de la Universidad de Alicante, Alicante, España. 2.2, 2.2.4

Moschitti, A. (2004). A study on convolution kernels for shallow semantic parsing. In *ACL '04: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 335, Morristown, NJ, USA. Association for Computational Linguistics. 5.2.3

Moschitti, A. y Harabagiu, S. (2004). A novel approach to focus identification in question/answering systems. In Harabagiu, S. and Lacatusu, F., editors, *HLT-NAACL 2004: Workshop on Pragmatics of Question Answering*, pages 43–51, Boston, Massachusetts, USA. Association for Computational Linguistics. 2.5.3, 3.5

Moschitti, A., Quarteroni, S., Basili, R. y Manandhar, S. (2007). Exploiting syntactic and shallow semantic kernels for question answer classification. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 776–783, Prague, Czech Republic. Association for Computational Linguistics. 3.3.1, 3.4.1, 3.6, 5.4

Murdock, V. y Croft, W. B. (2002). Task orientation in question answering. In *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 355–356, New York, NY, USA. ACM. 3.4.2

Màrquez, L. (2001). Machine learning and natural language processing. Technical Report LSI-00-45-R, LSI, Technical University of Catalonia. 3.4.5, 4.3

Nadeau, C. y Bengio, Y. (2003). Inference for the generalization error. *Machine Learning*, 52(3):239–281. 4.4.3

Navarro, B. (2006). *Metodología, construcción y explotación de corpus anotados semántica y anafóricamente*. PhD thesis, Alicante, Espa na. 3.2

Ng, H. T. y Lee, H. B. (1996). Integrating multiple knowledge sources to disambiguate word sense: an exemplar-based approach. In *Proceedings of the 34th annual meeting on Association for Computational Linguistics*, pages 40–47, Morristown, NJ, USA. Association for Computational Linguistics. 3.4.6

Nguyen, T. T., Nguyen, L. M. y Shimazu, A. (2008). Using semi-supervised learning for question classification. *Information and Media Technologies*, 3(1):112–130. 3.2, 3.3.1, 3.3.2, 3.3.4, 3.4.1, 5, 5.4

Nigam, K. y Hurst, M. (2004). Towards a robust metric of opinion. In *Proceedings of the AAAI Spring Symposium on Exploring Attitude and Affect in Text: Theories and Applications*, Standford, California. 3.4.5

Nyberg, E., Mitamura, T., Callan, J. P., Carbonell, J. G., Frederking, R. E., Collins-Thompson, K., Hiyakumoto, L., Huang, Y., Huttenhower, C., Judy, S., Ko, J., Kupsc, A., Lita, L. V., Pedro, V., Svoboda, D. y Durme, B. V. (2003). The javelin question-answering system at trec 2003: A multi-strategh approach with dynamic planning. In TREC-12 (2003). 2.3.1

Paşca, M. (2003). *Open-Domain Question Answering from Large Text Collections*. Studies in Computational Linguistics. Center for the Study of Language and Information. 2.2.1

Paşca, M. y Harabagiu, S. (2001). High performance question/answering. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 366–374, New York, NY, USA. ACM. 2.3.1, 3.1, 6, 7

Palmer, D. D. y Day, D. S. (1997). A statistical profile of the named entity task. In *Proceedings of the fifth conference on Applied natural language processing*, pages 190–193, Morristown, NJ, USA. Association for Computational Linguistics. 2.2.1

Pan, Y., Tang, Y., Lin, L. y Luo, Y. (2008). Question classification with semantic tree kernel. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 837–838, New York, NY, USA. ACM. 3.4.1, 5, 5.4

Pang, B. y Lee, L. (2008). *Opinion Mining and Sentiment Analysis*. Now Publishers Inc. 2.2.1, 3.6

Pantel, P. y Lin, D. (2002a). Discovering word senses from text. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining (KDD'02)*, pages 613–619, New York, NY, USA. ACM. 3.3.4

Pantel, P. y Lin, D. (2002b). Document clustering with committees. In *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 199–206, New York, NY, USA. ACM. 3.3.4

Pedersen, T. (2001). A decision tree of bigrams is an accurate predictor of word sense. In *NAACL '01: Second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies*, pages 1–8, Morristown, NJ, USA. Association for Computational Linguistics. 3.4.4

Peshkin, L., Pfeffer, A. y Savova, V. (2003). Bayesian nets in syntactic categorization of novel words. In *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 79–81, Morristown, NJ, USA. Association for Computational Linguistics. 3.4.7

Pinchak, C. y Bergsma, S. (2007). Automatic answer typing for how-questions. In *NAACL-HLT*, pages 516–523, Rochester, New York. Association for Computational Linguistics. 3.5

Pinchak, C. y Lin, D. (2006). A probabilistic answer type model. In *EACL 2006, 11st Conference of the European Chapter of the Association for Computational Linguistics*, pages 393–400. The Association for Computer Linguistics. 2.5.3, 1, 3.3.2, 3.3.4, 3.5, 5

Pinto, D., Branstein, M., Coleman, R., Croft, W. B., King, M., Li, W. y Wei, X. (2002). Quasm: a system for question answering using semi-structured data. In *JCDL '02: Proceedings of the 2nd ACM/IEEE-CS joint conference on Digital libraries*, pages 46–55, New York, NY, USA. ACM. 3.1, 3.1, 3.3.1, 3.3.2, 3.4.2

Platt, J. C. (1998). Fast training of support vector machines using sequential minimal optimization. In Schoelkopf, B., Burges, C., and Smola, A., editors, *Advances in Kernel Methods - Support Vector Learning*. MIT Press. 4.3

Pomerantz, J., Nicholson, S., Belanger, Y. y Lankes, R. D. (2004). The current state of digital reference: validation of a general digital reference model through a survey of digital reference services. *Information Processing and Management*, 40(2):347–363. 2.6.1

Ponte, J. M. y Croft, W. B. (1998). A language modeling approach to information retrieval. In *SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 275–281, New York, NY, USA. ACM. 3.4.2

Porter, M. F. (1997). An algorithm for suffix stripping. pages 313–316. 3.3.2

Prager, J., Chu-Carroll, J. y Czuba, K. (2002). Statistical answer-type identification in open-domain question answering. In *Proceedings of the second international conference on Human Language Technology Research*, pages 150–156, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc. 3.4.8

Prager, J., Radev, D., Brown, E., Coden, A. y Samn, V. (1999). The use of predictive annotation for question answering in trec-8. In TREC-8 (1999), pages 399–409. 2.3.1, 3.1, 3.1

Qiu, Y. y Frei, H.-P. (1993). Concept based query expansion. In *SIGIR '93: Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 160–169, New York, NY, USA. ACM. 2.5.3

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, 1(1):81–106. 3.4.4

Radev, D., Fan, W., Qi, H., Wu, H. y Grewal, A. (2002). Probabilistic question answering on the web. In *WWW '02: Proceedings of the 11th international conference on World Wide Web*, pages 408–419, New York, NY, USA. ACM. 2.3.1, 2.5.3, 3.1, 3.1, 3.2, 3.4.8

Ratnaparkhi, A. (1998). *Maximum entropy models for natural language ambiguity resolution*. PhD thesis, University of Pennsylvania, Pennsylvania, United States. 3.4.3

Ravichandran, D. y Hovy, E. (2001). Learning surface text patterns for a question answering system. In *ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 41–47, Morristown, NJ, USA. Association for Computational Linguistics. 6.4

Rennie, J. D. M. y Rifkin, R. (2001). Improving multiclass text classification with the Support Vector Machine. Technical Report AIM-2001-026, Massachusetts Insititute of Technology, Artificial Intelligence Laboratory. 3.4.1

Riloff, E. y Shepherd, J. (1997). A corpus-based approach for building semantic lexicons. In Cardie, C. and Weischedel, R., editors, *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing*, pages 117–124, Somerset, New Jersey. Association for Computational Linguistics. 6.4

Rocchio, J. J. (1971). *Relevance Feedback in Information Retrieval*, pages 313–323. Prentice Hall, Englewood, Cliffs, New Jersey. 2.5.3, 4.5.3

Roelleke, T. (2003). A frequency-based and a poisson-based definition of the probability of being informative. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 227–234, New York, NY, USA. ACM. 6.2.2

Rogati, M. y Yang, Y. (2002). High-performing feature selection for text classification. In *CIKM '02: Proceedings of the eleventh international conference on Information and knowledge management*, pages 659–661, New York, NY, USA. ACM. 4.5.3

Roger, S., Ferrández, S., Ferrández, A., Peral, J., Llopis, F., Aguilar, A. y Tomás, D. (2006). Aliqan, spanish qa system at clef-2005. In Peters, C., Gey, F. C., Gonzalo, J., Müller, H., Jones, G. J. F., Kluck, M., Magnini, B., and de Rijke, M., editors, *Accessing Multilingual Information Repositories, 6th Workshop of the Cross-Language Evaluation Forum, CLEF 2005*, volume 4022 of *Lecture Notes in Computer Science*, pages 457–466. Springer. 2.5.3

Roget, P. (1987). *Roget's Thesaurus of English Words and Phrases*. Penguin, Harmondsworth, Middlesex, England. 4.1.1

Roth, D., Cumby, C. M., Li, X., Morie, P., Nagarajan, R., Rizzolo, N., Small, K. y tau Yih, W. (2002). Question-answering via enhanced understanding of questions. In TREC-11 (2002), pages 592–601. 2.5.3

Ruthven, I. y Lalmas, M. (2003). A survey on the use of relevance feedback for information access systems. *Knowledge Engineering Review*, 18(2):95–145. 2.5.3

Salton, G. (1989). *Automatic text processing: the transformation, analysis, and retrieval of information by computer*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA. 2.4

Salton, G., Wong, A. y Yang, C. S. (1975). A vector space model for automatic indexing. *Communications ACM*, 18(11):613–620. 5

Sasaki, Y., Isozaki, H., Hirao, T., Kokuryou, K. y Maeda, E. (2002). Ntt's qa systems for ntcir qac-1. In *Working Notes of the Third NTCIR Workshop Meeting, Part IV: Question Answering Challenge (QAC1)*, pages 63–70. 2.3.1

Schlobach, S., Olsthoorn, M. y Rijke, M. D. (2004). Type checking in open-domain question answering. In *Journal of Applied Logic*, pages 398–402. IOS Press. 3.3.4, 5

Schölkopf, B. y Smola, A. J. (2001). *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA. 3.4.1, 5

Scott, S. (1999). Feature engineering for text classification. In *ICML-99, 16th International Conference on Machine Learning*, pages 379–388. Morgan Kaufmann Publishers. 3.3, 4.5.3

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47. 2.2.1, 2.4, 3.4.6, 4.4

Sekine, S., Sudo, K. y Nobata, C. (2000). Irex: Ir and ie evaluation project in japanese. In *LREC 2000: Language Resources and Evaluation Conference*, Athens, Greece. 4.1.1

Sekine, S., Sudo, K. y Nobata, C. (2002). Extended named entity hierarchy. In *LREC 2002: Language Resources and Evaluation Conference*, pages 1818–1824, Las Palmas, Spain. 3.1, 4.1.1

Shawe-Taylor, J. y Cristianini, N. (2004). *Kernel Methods for Pattern Analysis*. Cambridge University Press, New York, NY, USA. 3.4.1, 5.2.2

Singhal, A., Abney, S., Bacchiani, M., Collins, M., Hindle, D. y Pereira, F. (1999). ATT at TREC-8. In TREC-8 (1999), pages 317–330. 2.3.1, 3.1

Skowron, M. y Araki, K. (2004a). Evaluation of the new feature types for question classification with support vector machines. *IEEE International Symposium on Communications and Information Technology, 2004. ISCIT 2004*, 2:1017–1022. 3.2, 3.3.1, 3.3.2, 3.3.4, 3.4.1

Skowron, M. y Araki, K. (2004b). What can be learned from previously answered questions? a corpus-based approach to question answering. *Intelligent Information Systems. New Trends in Intelligent Information Processing and Web Mining*, pages 379–387. 2.5.3

Solorio, T., Manuel Pérez-Couti n., y Gémez, M. M., Luis Villase n.-P. y López-López, A. (2004). A language independent method for question classification. In *COLING '04: Proceedings of the 20th international conference on Computational Linguistics*, pages 1374–1380, Morristown, NJ, USA. Association for Computational Linguistics. (document), 1.1, 3.1, 3.1, 3.2, 3.3.1, 3.4.1

Solorio, T., no, M. P.-C., y Gómez, M. M., nor Pineda, L. V. y López-López, A. (2005). Question classification in spanish and portuguese. In *CICLing*, pages 612–619. 3.2

Song, F. y Croft, W. B. (1999). A general language model for information retrieval. In *CIKM '99: Proceedings of the eighth international conference on Information and knowledge management*, pages 316–321, New York, NY, USA. ACM. 3.4.2

Soubbotin, M. M. y Soubbotin, S. M. (2002). Use of patterns for detection of likely answer strings: A systematic approach. In TREC-11 (2002). 2.5.3

Stolcke, A. (2002). Srilm - An extensible language modeling toolkit. In *International Conference on Spoken Language Processing*, volume 2, pages 901–904, Denver, CO. 3.4.2

Suárez, A. y Palomar, M. (2002). A maximum entropy-based word sense disambiguation system. In *Proceedings of the 19th international conference on Computational linguistics*, pages 1–7, Morristown, NJ, USA. Association for Computational Linguistics. 3.4.3

Sundblad, H. (2007). Question classification in question answering. Master's thesis, Linköping University, Department of Computer and Information Science. 1.1, 3.2, 3.4.4, 3.4.6, 3.6, 4.4.3, 4.6, 6

Suzuki, J., Hirao, T., Sasaki, Y. y Maeda, E. (2003a). Hierarchical directed acyclic graph kernel: Methods for structured natural language data. In *ACL*, pages 32–39. 3.4.1

Suzuki, J., Taira, H., Sasaki, Y. y Maeda, E. (2003b). Question classification using hdag kernel. In *Proceedings of the ACL 2003 workshop on Multilingual summarization and question answering*, pages 61–68, Morristown, NJ, USA. Association for Computational Linguistics. 3.1, 3.1, 3.2, 3.3.1, 3.3.2, 3.3.4, 3.4.1, 3.6, 5

Taira, H. y Haruno, M. (1999). Feature selection in svm text categorization. In *AAAI '99/IAAI '99: Proceedings of the sixteenth national conference on Artificial intelligence and the eleventh Innovative applications of artificial intelligence conference innovative applications of artificial intelligence*, pages 480–486, Menlo Park, CA, USA. American Association for Artificial Intelligence. 3.4.1, 4.5.3

Tamura, A., Takamura, H. y Okumura, M. (2005). Classification of multiple-sentence questions. In *IJCNLP*, pages 426–437. 3.5

Tomás, D. y Giuliano, C. (2009). A semi-supervised approach to question classification. In *17th European Symposium on Artificial Neural Networks: Advances in Computational Intelligence and Learning*. 7.4

Tomás, D. y Vicedo, J. L. (2007a). Multiple-taxonomy question classification for category search on faceted information. In Matousek, V. and Mautner, P., editors, *Text, Speech and Dialogue, 10th International Conference, TSD 2007*, volume 4629 of *Lecture Notes in Computer Science*, pages 653–660, Pilsen, Czech Republic. Springer. 2.6.2, 3.2, 3.4.2, 7.4

Tomás, D. y Vicedo, J. L. (2007b). Re-ranking passages with lsa in a question answering system. In Peters, C., Clough, P., Gey, F. C., Karlgren, J., Magnini, B., Oard, D. W., de Rijke, M., and Stempfhuber, M., editors, *Evaluation of Multilingual and Multi-modal Information Retrieval, 7th Workshop of the Cross-Language Evaluation Forum, CLEF 2006*, volume 4730 of *Lecture Notes in Computer Science*, pages 275–279. Springer. 2.5.3, 7.4

Tomás, D., Vicedo, J. L., Bisbal, E. y Moreno, L. (2006a). Automatic feature extraction for question classification based on dissimilarity of probability distributions. In Salakoski, T., Ginter, F., Pyysalo, S., and Pahikkala, T., editors, *Advances in Natural Language Processing, 5th International Conference on NLP, FinTAL 2006*, volume 4139 of *Lecture Notes in Computer Science*, Turku, Finland. Springer. 7.4

Tomás, D., Vicedo, J. L., Saiz-Noeda, M. y Izquierdo, R. (2006b). An xml-based system for spanish question answering. In Peters, C., Gey, F. C., Gonzalo, J., Müller, H., Jones, G. J. F., Kluck, M., Magnini, B., and de Rijke, M., editors, *Accessing Multilingual Information Repositories, 6th Workshop of the Cross-Language Evaluation Forum, CLEF 2005*, volume 4022 of *Lecture Notes in Computer Science*, pages 347–350. Springer. 7.4

Tomás, D., Vicedo, J. L., Suárez, A., Bisbal, E. y Moreno, L. (2005). Una aproximación multilingüe a la clasificación de preguntas basada en aprendizaje automático. *Procesamiento del Lenguaje Natural*, (35):391–398. 3.2, 3.3.1, 3.4.1

Tomás, D., Vicedo, J. L., Suárez, A., Bisbal, E. y Moreno, L. (2006c). Desarrollo de un corpus de entrenamiento para sistemas de búsqueda de respuestas basados en aprendizaje automático. *Procesamiento del Lenguaje Natural*, (37):59–66. 7.4

Tomuro, N. (2002). Question terminology and representation for question type classification. In *COLING-02 on COMPUTERM 2002*, pages 1–7, Morristown, NJ, USA. Association for Computational Linguistics. 1

TREC-10 (2001). *Tenth Text REtrieval Conference*, volume 500-250 of *NIST Special Publication*, Gaithersburg, USA. National Institute of Standards and Technology. B

TREC-11 (2002). *Eleventh Text REtrieval Conference*, volume 500-251 of *NIST Special Publication*, Gaithersburg, USA. National Institute of Standards and Technology. B

TREC-12 (2003). *Twelfth Text REtrieval Conference*, volume 500-255 of *NIST Special Publication*, Gaithersburg, USA. National Institute of Standards and Technology. B

TREC-8 (1999). *Eighth Text REtrieval Conference*, volume 500-246 of *NIST Special Publication*, Gaithersburg, USA. National Institute of Standards and Technology. B

TREC-9 (2000). *Ninth Text REtrieval Conference*, volume 500-249 of *NIST Special Publication*, Gaithersburg, USA. National Institute of Standards and Technology. B

Tunkelang, D. (2006). Dynamic category sets: An approach for faceted search. In *Proceedings of the 29th Annual International ACM Conference on Research & Development on Information Retrieval - Faceted Search Workshop (SIGIR '06)*. 2.6.2

Vállez, M. y Pedraza-Jiménez, R. (2007). El procesamiento del lenguaje natural en la recuperación de información textual y áreas afines. *Hipertext.net*, (5). 2.2.3

Vallin, A., Magnini, B., Giampiccolo, D., Aunimo, L., Ayache, C., Osenova, P., Peñas, A., de Rijke, M., Sacaleanu, B., Santos, D. y Sutcliffe, R. (2006). Overview of the clef 2005 multilingual question answering track. In Heidelberg, S. B. ., editor, *Accessing Multilingual Information*

Repositories, volume 4022 of *Lecture Notes in Computer Science*, pages 307–331. 1.1

Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory*. Springer. 3.4.1

Verberne, S. (2006). Developing an approach for *Why*-question answering. In *Conference Companion of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2006)*, pages 39–46, Trento, Italy. 2.5.4, 3.5

Vicedo, J. L. (2003). La búsqueda de respuestas: Estado actual y perspectivas de futuro. *Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial*, 22:37–56. 2.5

Vicedo, J. L. y Ferrández, A. (2000). A semantic approach to question answering systems. In TREC-9 (2000), pages 511–516. 2.5.3

Voorhees, E. M. (1993). Using wordnet to disambiguate word senses for text retrieval. In *SIGIR '93: Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 171–180, New York, NY, USA. ACM. 7.3

Voorhees, E. M. (1999). The trec-8 question answering track report. In TREC-8 (1999), pages 77–82. 2.3.1, 4.1.1

Voorhees, E. M. (2000). Overview of the trec-9 question answering track. In TREC-9 (2000), pages 71–79. (document), 4.1

Voorhees, E. M. (2001a). Overview of the trec 2001 question answering track. In TREC-10 (2001), pages 42–51. 4.1.1

Voorhees, E. M. (2001b). The trec question answering track. *Natural Language Engineering*, 7(4):361–378. 1.1

Voorhees, E. M. (2002). Overview of the trec 2002 question answering track. In TREC-11 (2002). 4.1.1

Voorhees, E. M. (2003). Overview of the trec 2003 question answering track. In TREC-12 (2003), pages 54–78. 4.1.1

Wai, W. P. y Yongsheng, Y. (2002). A maximum entropy approach to hownet-based chinese word sense disambiguation. In *COLING-02 on SEMANET*, pages 1–7, Morristown, NJ, USA. Association for Computational Linguistics. 3.3.4

Wang, X. y He, Q. (2004). Enhancing generalization capability of svm classifiers with feature weight adjustment. In *Knowledge-Based Intelligent*

Information and Engineering Systems, 8th International Conference, KES 2004, volume 3213 of *Lecture Notes in Computer Science*, pages 1037–1043. Springer. 4.5.3

Witten, I. H., Bell, T. C. y Moffat, A. (1994). *Managing Gigabytes: Compressing and Indexing Documents and Images*. John Wiley & Sons, Inc., New York, NY, USA. 4.2

Witten, I. H. y Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, San Francisco, 2 edition. 2.3.2, 3.4.1, 4.4.2

Woods, W. A. (1970). Transition network grammars for natural language analysis. *Commun. ACM*, 13(10):591–606. 2.2.1

Woods, W. A., Bookman, L. A., Houston, A., Kuhns, R. J., Martin, P. y Green, S. (2000). Linguistic knowledge can improve information retrieval. In *Proceedings of the sixth conference on Applied natural language processing*, pages 262–267, Morristown, NJ, USA. Association for Computational Linguistics. 2.2.4

Yang, H. y Chua, T.-S. (2002). The integration of lexical knowledge and external resources for question answering. In TREC-11 (2002). 2.5.3

Yang, M., Ahuja, N. y Roth, D. (2000). View-based 3d object recognition using snow. In *Proceedings of the Fourth Asian Conference on Computer Vision (ACCV-2000)*, pages 830–835. 3.4.5

Yang, Y. (1999). An evaluation of statistical approaches to text categorization. *Information Retrieval*, 1(1-2):69–90. 3.4.4

Yang, Y. y Liu, X. (1999). A re-examination of text categorization methods. In *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 42–49, New York, NY, USA. ACM. 3.4.6

Yang, Y. y Pedersen, J. O. (1997). A comparative study on feature selection in text categorization. In *ICML '97: Proceedings of the Fourteenth International Conference on Machine Learning*, pages 412–420, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc. 4.5.3, 4.5.3

Yee, K.-P., Swearingen, K., Li, K. y Hearst, M. (2003). Faceted metadata for image search and browsing. In *CHI'03: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 401–408, New York, NY, USA. ACM. 2.6.2

Zhai, C. y Lafferty, J. (2004). A study of smoothing methods for language models applied to information retrieval. *ACM Transactions on Information Systems*, 22(2):179–214. 3.4.2

Zhang, D. y Lee, W. S. (2003). Question classification using support vector machines. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 26–32, New York, NY, USA. ACM. 3.2, 3.4.1, 3.4.4, 3.4.5, 3.4.6, 3.4.7, 3.6, 4.6, 5.4

Zhang, T. (2000). Large margin winnow methods for text categorization. In *Proceedings of the KDD-2000 Workshop on Text Mining*, pages 81–87, Boston, MA, USA. 3.4.5

Zhao, S. y Grishman, R. (2005). Extracting relations with integrated information using kernel methods. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 419–426, Morristown, NJ, USA. Association for Computational Linguistics. 5.2.3

Zhou, Z.-H. y Li, M. (2005). Tri-training: Exploiting unlabeled data using three classifiers. *IEEE Transactions on Knowledge and Data Engineering*, 17(11):1529–1541. 5.4

Zipf, G. K. (1949). *Human Behavior and the Principle of Least Effort*. Addison-Wesley (Reading MA). 19