

Kernels para la clasificación de preguntas en español y catalán

David Tomás y José L. Vicedo
Depto. de Lenguajes y Sistemas Informáticos
Universidad de Alicante
{dtomas,vicedo}@dlsi.ua.es

Resumen

Este artículo presenta una aproximación a la clasificación automática de preguntas en español y catalán. El sistema de clasificación está basado en el algoritmo SVM y en el uso de diferentes funciones kernel, empleando únicamente características textuales superficiales que permiten la obtención de un sistema fácilmente adaptable a diferentes idiomas. Se ha realizado un estudio sobre el correcto ajuste de parámetros de los kernels, la precisión de los mismos, la definición de distintos vectores de características de aprendizaje y el rendimiento en función del idioma de trabajo. Adicionalmente, se ha experimentado con el algoritmo LIBLINEAR, aplicado aquí por vez primera a la tarea de clasificación de preguntas. Con este algoritmo, así como con los kernels definidos, se han obtenido valores de precisión por encima del 80 % para los dos idiomas tratados, superando a otros algoritmos tradicionales de clasificación. Para el entrenamiento y evaluación del sistema se ha desarrollado un corpus paralelo de 2.393 preguntas en inglés, español y catalán.

1. Introducción

Los sistemas de *búsqueda de respuestas o question answering* (QA) tienen como finalidad encontrar respuestas concretas a necesidades precisas de información formuladas por los usuarios en lenguaje natural. Estos sistemas se han convertido en objeto de amplio estudio en la última década, gracias en parte a los distintos foros internacionales desarrollados en este campo: TREC¹ (Voorhees, 2001), CLEF² (Vallin et al., 2006) y NTCIR³ (Kando, 2005). Estos foros han marcado las pautas de desarrollo de estos sistemas, estableciendo los retos a superar y el marco para su evaluación.

La mayoría de sistemas de QA presentan una arquitectura común, organizando su funcionamiento en tres fases bien diferenciadas que habitualmente tienen lugar de forma secuencial (Voorhees, 1999): *análisis de la pregunta, selección de documentos o pasajes relevantes y extracción de la respuesta*. Dentro de la fase de análisis de la pregunta tiene lugar la *clasificación de preguntas*⁴. El objetivo de esta clasificación es identificar de forma automática qué se está preguntando, categorizando las preguntas en diferentes clases semánticas en función del tipo de respuesta

esperado. Por ejemplo, ante una pregunta como “¿Quién es el presidente de los Estados Unidos?”, un sistema de clasificación de preguntas detectarían que se está preguntando por una persona, mientras que para “¿Dónde está la Torre Eiffel?” identificarían que se está preguntando por un lugar. En estos ejemplos, *persona* y *lugar* representan la clase semántica de la respuesta esperada.

La clasificación de preguntas en los sistemas de QA tiene un doble propósito. En primer lugar, proporciona una restricción semántica a las respuestas esperadas, permitiendo filtrar un gran número de ellas durante la fase final de extracción. Por ejemplo, cuando se considera la pregunta “¿Cuál es la ciudad más grande de Alemania?”, detectar que se está preguntando por un *lugar* permite descartar un gran número de respuestas candidatas, manteniendo únicamente aquellas que sean nombres de localizaciones. El segundo propósito de la clasificación es proporcionar información a los procesos subsiguientes del sistema de QA para establecer la estrategia de selección de respuestas, así como las bases de conocimiento que el sistema requiera para obtener la respuesta final.

La importancia de la clasificación de preguntas en el resultado global de los sistemas de QA ha quedado patente en diversos estudios. Radev et al. (2002) detectaron que una clasificación incorrecta de la pregunta conlleva que la posibilidad del sistema de obtener una respuesta correcta sea 17 veces menor. En otro análisis, realizado

¹<http://trec.nist.org>.

²<http://clef-campaign.org>.

³<http://research.nii.ac.jp/ntcir/>.

⁴Algunos de los nombres que recibe esta tarea en la literatura anglosajona son *question classification*, *question categorization* y *answer type recognition*.

por Moldovan et al. (2003) sobre los errores ocurridos en un sistema de QA en domino abierto, se revela que más de un 35 % de éstos son directamente imputables al módulo de clasificación de la pregunta.

Al igual que sucede con el resto de tareas dentro del campo del *procesamiento del lenguaje natural* (PLN), la mayoría de sistemas de clasificación de preguntas están orientados al idioma inglés, siendo muy escasas las contribuciones para otros idiomas. En este trabajo presentamos un estudio sobre la clasificación automática de preguntas en español y catalán. Definiremos un sistema basado en corpus mediante el empleo de kernels y características textuales superficiales, dando como resultado una aproximación fácilmente adaptable a diferentes idiomas.

En el resto de este artículo, la sección 2 presenta una introducción a los métodos y funciones kernel y a su aplicación al PLN. La sección 3 describe los componentes de aprendizaje del sistema y los corpus de preguntas empleados en este trabajo. En la sección 4 se describen los experimentos llevados a cabo y los resultados obtenidos. La sección 5 muestra otras investigaciones relacionados con este trabajo. Finalmente, la sección 6 enumera las conclusiones y trabajo futuro derivado de esta investigación.

2. Métodos y funciones kernel

Los *métodos kernel* o *kernel methods* son un tipo de algoritmo de aprendizaje automático ampliamente utilizado en el campo del PLN debido a sus buenos resultados empíricos (Shawe-Taylor y Cristianini, 2004; Schölkopf y Smola, 2001). La estrategia adoptada por estos métodos consiste en dividir el problema de aprendizaje en dos partes: en primer lugar se trasladan los datos de entrada a un espacio de características adecuado, empleando seguidamente un algoritmo lineal 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 núcleo* (*kernel function*, *función kernel* o simplemente *kernel*). Una función kernel proporciona una medida de similitud entre los datos de entrada que depende exclusivamente del tipo de éstos y del dominio.

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

kernel como

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

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} .

La selección del kernel apropiado para cada tarea resulta de gran importancia, ya que es éste el que define el espacio de trabajo transformado donde se llevará a cabo el entrenamiento y la clasificación. En este trabajo vamos a estudiar el rendimiento de cuatro funciones kernel, algunas de ellas empleadas de forma habitual en diferentes tareas de clasificación dentro del campo del PLN:

- **Lineal**

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

- **Polinómico**

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

- **Radial Basis Function (RBF)**

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

- **Sigmoide**

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

Las variables γ , r y d son los parámetros de los kernels. En la sección 4.2 hablaremos más en detalle de estos parámetros.

El kernel es el único elemento específico del dominio en el sistema de clasificación, 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 *Support Vector Machines* (SVM) o *perceptrón*. En nuestros experimentos vamos a emplear SVM como algoritmo para trabajar con los kernels definidos más arriba. SVM ha sido aplicado con éxito en numerosos problemas de PLN, como el análisis sintáctico superficial (Kudo y Matsumoto, 2001) o la clasificación de textos (Joachims, 1998), demostrando su buen funcionamiento en espacios de alta dimensionalidad. En la actualidad, SVM se ha convertido en el algoritmo más popular de los empleados en clasificación de preguntas basados en corpus (Zhang y Lee, 2003). Para llevar a cabo nuestros experimentos hemos utilizado la implementación ofrecida en la librería LIBSVM (Chang y Lin, 2001).

3. Descripción del sistema

Como todo sistema de basado en corpus, nuestro clasificador de preguntas requiere la definición de una serie de elementos:

- Un conjunto de preguntas (corpus) para el entrenamiento y la evaluación del sistema.
- Una taxonomía de tipos de pregunta con la que queremos clasificar las entradas que lleguen al sistema.
- Un conjunto de características de aprendizaje extraídas del corpus 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.

El algoritmo de aprendizaje ya fue descrito en el apartado anterior. En esta sección describiremos el resto de componentes del sistema.

3.1. Corpus

Al igual que sucede con otros recursos lingüísticos en el campo de PLN, los corpus de preguntas en idiomas diferentes al inglés son muy escasos. Existen algunos corpus disponibles en diferentes idiomas, como DISEQuA⁵ (Magnini et al., 2003) (450 preguntas en holandés, italiano, español e inglés) o Multieight-04⁶ (Magnini et al., 2005) (700 preguntas en alemán, inglés, español, francés, italiano, holandés y portugués). El problema de estos conjuntos de preguntas es su reducido tamaño, que los hace poco apropiados para el entrenamiento y evaluación de sistemas basados en corpus. Por ello, en este trabajo hemos desarrollado nuestros propios corpus para español y catalán, idioma este último en el que no hay constancia de conjuntos de preguntas disponibles para este tipo de sistemas.

El corpus en español se empleó por primera vez en (Tomás et al., 2005). Para formalizar este corpus se recopilaron las preguntas de evaluación en inglés definidas para la tarea de QA de las conferencias TREC, desde 1999 (TREC-8) hasta 2003 (TREC-12).⁷ Una vez recopiladas las preguntas en inglés, se procedió a la traducción manual de las mismas. En el caso del 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. En el caso del corpus en catalán, se tradujeron íntegramente las preguntas en inglés, obteniendo finalmente un corpus paralelo de 2.393 preguntas en los tres idiomas. Este corpus, etiquetado con la taxonomía definida en el siguiente apartado, está libremente disponible para la comunidad científica.⁹

3.2. Taxonomía

Una vez recopiladas las preguntas, el siguiente paso consistió en el etiquetado manual de éstas con su correspondiente clase semántica. Las preguntas originales del TREC no presentan ningún tipo de etiquetado, ya que son los propios participantes de estas conferencias los que deciden la taxonomía de clases que más conviene a su sistema de QA. Al no existir ninguna taxonomía estándar en el campo de QA y de los sistemas de clasificación de preguntas, definimos una propia tomando como base la jerarquía extendida de entidades nombradas de Sekine (Sekine, Sudo, y Nobata, 2002). Esta jerarquía fue diseñada para cubrir aquellas entidades que, dicho de manera informal, aparecen habitualmente en textos periodísticos. El objetivo de Sekine era el de dar cobertura a entidades nombradas más específicas que las dadas habitualmente en los sistemas de extracción de información. De esta manera, 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 jerárquica era que pudiera ajustarse fácilmente a diferentes tareas según el grado de refinamiento de las entidades a detectar.

Entre los sistemas que inspiraron el diseño de esta jerarquía están aquellos empleados en la tarea de QA 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. Para este etiquetado utilizamos como base la jerarquía descrita por Sekine, centrándonos en las etiquetas que aparecen en el primer nivel. Sobre esta base se añadieron las clases *DEFINITION* y *ACRONYM*. Éstas no existían originalmente en la jerarquía de Sekine (ya que se centra en entidades) pero se han incluido para aumentar la cober-

⁵http://clef-qa.itc.it/2004/down/DISEQuA_v1.0.zip.

⁶http://clef-qa.itc.it/2005/down/corpora/multieight-04_v1.2.zip.

⁷<http://trec.nist.gov/data/qa.html>.

⁸<http://nlp.uned.es>.

⁹<http://www.dlsi.ua.es/~dtomas/resources/>.

tura de la taxonomía al ser 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* 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 figura 1 muestra la distribución de preguntas por clase. Se puede observar que para la clase *TITLE* no existe ninguna pregunta en el corpus, por lo que no se tendrá en cuenta en los experimentos, dando lugar a una taxonomía final de 15 clases.

3.3. Vector de características

Cada instancia del problema (pregunta) 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 obtenidos del propio corpus de entrenamiento:

- **Unigramas** (1-gramas). Se emplean los términos extraídos de la pregunta como componentes del vector de características.
- **Bigramas** (2-gramas). Representan todas las combinaciones de términos adyacentes en una pregunta como una secuencia de dos palabras.
- **Combinación** (1+2-gramas). Emplearemos combinaciones de unigramas y bigramas, buscando obtener una mejora con respecto a sus componentes individuales.

Vamos a utilizar una representación binaria para el vector de características, 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. Sin embargo, en la tarea de clasificación de preguntas 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.

Para obtener los n-gramas de las preguntas hemos utilizado el CMU-Cambridge Statistical

Language Modeling Toolkit,¹⁰ un conjunto de herramientas para facilitar la construcción de modelos de lenguaje. El único preproceso llevado a cabo sobre el corpus ha sido la eliminación de signos de puntuación y la sustitución de todos los caracteres en mayúsculas por su equivalente en minúsculas.

4. Evaluación

Para la evaluación de nuestro sistema hemos planteado una serie de experimentos que pretenden cubrir cuatro aspectos fundamentales de nuestro sistema de clasificación:

- Influencia de los parámetros de los kernels.
- Comparación de precisión entre kernels.
- Comparación de los vectores de características.
- Comparación de rendimiento entre idiomas.

Todos los experimentos realizados se han evaluado mediante validación cruzada equilibrada en 10 particiones (*stratified 10-fold cross-validation*). 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.

Para la verificación de los resultados hemos empleado *t-test* (Dietterich, 1998). Este test estadístico nos va a permitir saber cuándo las mejoras aportadas por unas configuraciones con respecto a otras son realmente significativas, minimizando la posibilidad de que la diferencia de precisión obtenida pueda ser fortuita. 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 clasificación de preguntas. Resulta difícil en estas condiciones certificar si las supuestas mejoras aportadas por algunas de las aproximaciones en este campo son reales o no. El grado de confianza p que vamos a considerar en estos experimentos es $p < 0,05$ 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.

4.1. Ajuste de parámetros

A la hora de llevar a cabo la clasificación mediante SVM y las funciones kernel, hay una serie de parámetros que gobiernan el proceso de entrenamiento que afectan profundamente al rendimiento final del clasificador. En este apartado

¹⁰<http://svr-www.eng.cam.ac.uk/~prc14/toolkit.html>.

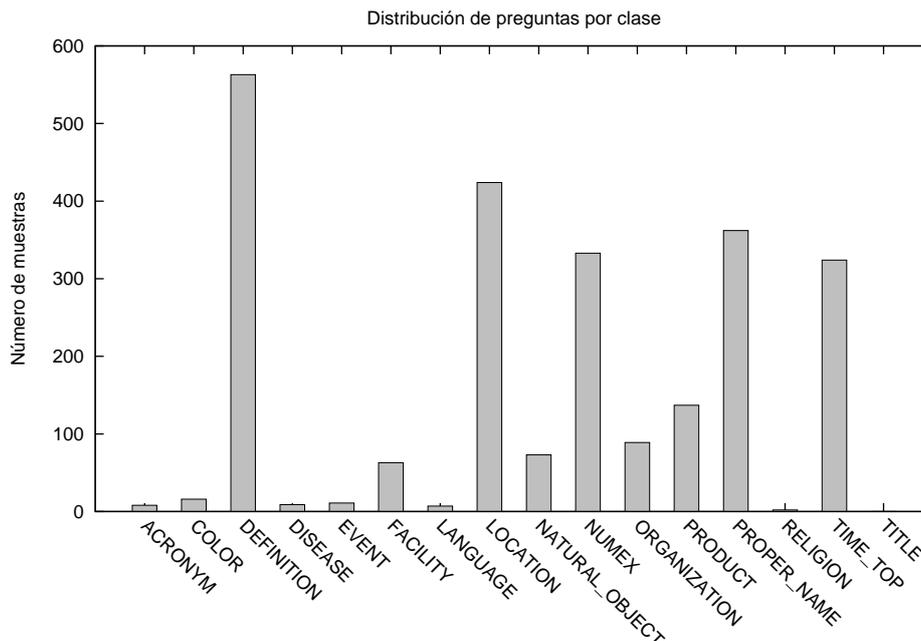


Figura 1: Número de preguntas en el corpus para cada una de las clases de la taxonomía.

queremos evaluar el efecto que tiene la correcta sintonización de estos parámetros en el rendimiento del sistema.

Cuando se emplea SVM es habitual que los conjuntos de datos con los que se trabaja no sean completamente separables mediante un hiperplano. Para estas situaciones, SVM posee un parámetro de coste C que permite la creación de márgenes suaves que permiten algunos errores de clasificación durante el entrenamiento. Un mayor valor de C implica una mayor penalización a los errores de clasificación, forzando la creación de modelos más ajustados a los datos, reduciendo su capacidad de generalización y viéndose afectados por el problema de sobreajuste (*overfitting*).

Además del parámetro C de SVM, cada uno de los kernels definidos en la sección 2, a excepción del lineal, tiene sus propios parámetros. Dichos parámetros son: γ , r y d para el kernel polinómico; γ para el RBF; y γ y r para el sigmoide.

No se conoce a priori cuál es la mejor selección de estos parámetros para un problema dado, por lo que debe llevarse a cabo algún tipo de búsqueda con el objetivo de identificar los mejores valores para la predicción de los datos de evaluación. Para realizar el ajuste de parámetros hemos llevado a cabo una búsqueda *grid* empleando validación cruzada en 5 particiones (*5-fold cross-validation*) sobre el conjunto de entrenamiento formado por los vectores de 1-gramas del corpus en español. Este tipo de validación cruzada permite evitar que haya problemas de sobreajuste al

Kernel		C	γ	r	d
Lineal	Def	1,0	-	-	-
	Opt	0,8123	-	-	-
Polinómico	Def	1,0	0,01	1	3
	Opt	73,5167	0,0769	4,4444	3
RBF	Def	1,0	0,01	-	-
	Opt	12,9960	0,0474	-	-
Sigmoide	Def	1,0	0,01	1	-
	Opt	29,8571	0,0237	0,4444	-

Tabla 1: Parámetros por defecto (Def) y optimizados (Opt) para el clasificador SVM y cada uno de los kernels estudiados.

computar los mejores parámetros sobre un conjunto de datos (Hsu, Chang, y Lin, 2003). La tabla 1 muestra los parámetros por defecto empleados de forma habitual en estos kernels (Witten y Frank, 2005; Chang y Lin, 2001) junto con los parámetros optimizados obtenidos en nuestro estudio.

Una vez determinados los mejores parámetros para SVM y para cada uno de los kernels, hemos contrastado el rendimiento de éstos con respecto a la versión con los parámetros por defecto. Hemos llevado a cabo los experimentos en español y catalán, empleando los tres vectores de características comentados en la sección 3.3: unigramas, bigramas y la combinación de ambos. La tabla 2 muestra los resultados obtenidos para los kernels, así como los resultados para dos algoritmos adicionales (Naïve Bayes y LIBLINEAR)

que estudiaremos en el siguiente apartado. A simple vista se aprecia una considerable mejoría de rendimiento con los parámetros optimizados para los kernels polinómico, RBF y sigmoide. En el caso del kernel lineal se observa también una mejora para la mayoría de vectores de aprendizaje en ambos idiomas, pero ésta no es tan evidente y debe valorarse si es estadísticamente significativa.

En la tabla 3 se muestra la comparación estadística mediante *t-test* de los resultados obtenidos con los kernels optimizados y los kernels con parámetros por defecto. Los símbolos “ \gg ” y “ $>$ ” indican que el *Kernel 1* es significativamente mejor que el *Kernel 2* con un grado de confianza $p < 0,01$ y $p < 0,05$ respectivamente. De forma equivalente, “ \ll ” y “ $<$ ” indican que el *Kernel 1* es significativamente peor que el *Kernel 2*. El símbolo “ $=$ ” indica que no hay una diferencia significativa de funcionamiento entre las dos aproximaciones.

A raíz de estos resultados, es evidente que el ajuste de parámetros en el caso de los kernels polinómico, RBF y sigmoide es absolutamente necesario. Las mejoras son evidentes y significativas en ambos idiomas y para todos los vectores de características estudiados. En el caso del kernel lineal se obtienen también mejoras en casi todos los experimentos (a excepción de 2-gramas en catalán), pero estos resultados no son estadísticamente significativos, por lo que no se puede asegurar que en este caso la correcta selección de parámetros del kernel pueda suponer una mejora real en el sistema.

4.2. Comparación entre kernels

En este apartado vamos a contrastar el rendimiento de los distintos kernels entre sí. Para ello vamos a comparar la precisión obtenida por la versión optimizada de cada uno de los kernels en los experimentos realizados en el apartado anterior. Adicionalmente, vamos a introducir dos algoritmos de aprendizaje más en esta comparativa. El primero de ellos, Naïve Bayes (NB) (Duda y Hart, 1973) es un clasificador estocástico usado de forma habitual en numerosas tareas de aprendizaje automático (Mitchell, 1997). Nos servirá como algoritmo de referencia para contrastar el rendimiento de SVM y los diferentes kernels. El segundo algoritmo que vamos a introducir es el clasificador lineal LIBLINEAR (Fan et al., 2008). Este clasificador resulta especialmente adecuado para tareas de aprendizaje en las que intervienen un gran número de instancias y características, como es el caso de nuestro problema (ver la tabla 6 para más detalles sobre el tamaño de los

vectores de aprendizaje). Destacar que no existen evaluaciones previas de este algoritmo para la tarea de clasificación de preguntas.

La tabla 2 muestra los resultados obtenidos con NB y LIBLINEAR repitiendo los experimentos realizados en el apartado anterior con los cuatro kernels. En términos generales, se aprecia un buen rendimiento de LIBLINEAR (por encima del 80 % en la mayoría de casos), mientras que los resultados con NB resultan más modestos (por debajo del 70 % en todos los casos).

La tabla 4 muestra la comparación estadística de los resultados obtenidos con los distintos kernels y los dos algoritmos adicionales planteados. La comparación de precisión entre kernels revela que no existe una diferencia significativa entre el kernel lineal, el polinómico y el sigmoide para ninguno de los idiomas y vectores de características tratados. El kernel RBF consigue una precisión equivalente al resto de kernels para los vectores de 1-gramas y 2-gramas. Sin embargo, cuando se observa su funcionamiento con el vector de 1+2-gramas, su precisión es significativamente peor ($p < 0,1$) que los otros tres kernels para ambos idiomas. La pérdida de rendimiento en este caso se achaca a un peor funcionamiento de este kernel en espacios de alta dimensionalidad (Hsu, Chang, y Lin, 2003).

Por lo que respecta a la comparación con NB, queda patente la superioridad de SVM y los distintos kernels con respecto a este algoritmo. Para los cuatro kernels, los resultados obtenidos en los dos idiomas y con todos los vectores de características definidos son significativamente mejores ($p < 0,01$) que NB. En este caso, la tarea de clasificación de preguntas pone de manifiesto el problema del algoritmo NB para trabajar en espacios con un gran número de características de aprendizaje.

Por otra parte, los resultados obtenidos con LIBLINEAR son realmente prometedores. Los valores de precisión obtenidos con este algoritmo fueron significativamente mejores ($p < 0,01$ o $p < 0,05$) que el resto de configuraciones, tanto en español como en catalán, para los tres posibles vectores de características. Esto demuestra como, en situaciones en las que el número de características de aprendizaje es grande, la proyección realizada por los kernels a un espacio de dimensionalidad mayor puede no ser necesaria, obteniendo buenos resultados con un buen clasificador lineal. Estos resultados reafirman los ya obtenidos al evaluar los kernels, donde el kernel lineal obtenía resultados comparables al resto de kernels planteados.

Kernel		Español			Catalán		
		1-gramas	2-gramas	1+2-gramas	1-gramas	2-gramas	1+2-gramas
Lineal	Def	80,92	75,25	81,25	80,66	75,97	80,84
	Opt	81,07	75,27	81,25	80,74	75,91	80,87
Polinómico	Def	65,65	53,55	72,40	66,66	57,71	72,26
	Opt	80,86	74,76	80,69	80,91	75,56	80,53
RBF	Def	62,83	45,58	67,48	65,39	53,21	67,83
	Opt	81,13	74,53	79,01	80,88	75,62	78,87
Sigmoide	Def	43,20	23,53	48,25	53,02	24,06	57,20
	Opt	81,10	74,95	80,36	81,10	76,02	80,28
NB		69,82	58,13	68,89	69,42	61,35	68,68
LIBLINEAR		82,27	77,61	83,71	82,46	78,72	82,64

Tabla 2: Precisión obtenida por cada uno de los kernels con los parámetros por defecto (Def) y los parámetros optimizados (Opt). Las dos últimas filas muestran la precisión obtenida por los algoritmos Naïve Bayes (NB) y LIBLINEAR.

Kernel 1		Kernel 2		Español			Catalán		
				1-gramas	2-gramas	1+2-gramas	1-gramas	2-gramas	1+2-gramas
Lineal	Opt	Lineal	Def	=	=	=	=	=	=
Poli.	Opt	Poli.	Def	»	»	»	»	»	»
RBF	Opt	RBF	Def	»	»	»	»	»	»
Sigmo.	Opt	Sigmo.	Def	»	»	»	»	»	»

Tabla 3: Comparación estadística entre la versión con parámetros optimizados (Opt) y con parámetros por defecto (Def) de cada uno de los kernels. El símbolo “=” indica que la diferencia no es estadísticamente significativa, mientras que “»” indica que el *Kernel 1* es significativamente mejor ($p < 0,01$) que el *Kernel 2*.

4.3. Comparación entre vectores de características

En este apartado vamos a comparar los distintos vectores de características definidos: 1-gramas, 2-gramas y 1+2-gramas. Para esta comparativa vamos a retomar los valores de precisión obtenidos en los dos apartados anteriores. La tabla 5 toma como base la precisión obtenida con el vector de 1-gramas y compara estos resultados con los obtenidos con los otros dos vectores para cada uno de los algoritmos e idiomas.

Los resultados revelan que para todos los algoritmos el vector de 1-gramas es significativamente mejor ($p < 0,01$) que el vector de 2-gramas, no existiendo diferencia con el vector de 1+2-gramas. Estos resultados reflejan que los bigramas por sí solos no resultan adecuados para la clasificación. Al ser combinados con los unigramas (1+2-gramas) se obtienen ligeras mejoras de precisión en muchos de los algoritmos tratados, pero estas mejoras no resultan ser significativas. En estos casos, el enriquecimiento de información que supone la incorporación de bigramas se ve lastrado por el aumento de atributos en el espacio de aprendizaje (ver tabla 6). La única ex-

cepción es el caso del algoritmo LIBLINEAR en español. En esta ocasión, la precisión obtenida con el vector de 1+2-gramas es significativamente mejor ($p < 0,01$) que para el resto de vectores. Se demuestra nuevamente el buen funcionamiento de este algoritmo con vectores de aprendizaje de gran tamaño.

4.4. Comparación entre idiomas

A continuación vamos a comparar el rendimiento del sistema en función del idioma de trabajo. Ya que no empleamos ningún tipo de herramienta o recurso lingüístico, a excepción de los n-gramas extraídos del corpus, este experimento nos va a permitir valorar si las características intrínsecas de cada idioma, como la flexión verbal y nominal, afectan a la tarea de clasificación de preguntas. El grado de flexión de cada lengua se va a ver reflejado en el tamaño de vocabulario del problema, tal y como se muestra en la tabla 6. Además de en español y en catalán, que poseen un grado similar de flexión, vamos a realizar el estudio comparativo con el corpus original de preguntas en inglés. Nos vamos a centrar en el kernel lineal para comprobar el rendimiento de éste en los tres idiomas propuestos.

Algoritmo 1	Algoritmo 2	Español			Catalán		
		1-gramas	2-gramas	1+2-gramas	1-gramas	2-gramas	1+2-gramas
Lineal	Polinómico	=	=	=	=	=	=
Lineal	RBF	=	>	≫	=	=	≫
Lineal	Sigmoide	=	=	=	=	=	=
Lineal	NB	≫	≫	≫	≫	≫	≫
Lineal	LIBLINEAR	<	≪	≪	≪	≪	≪
Polinómico	RBF	=	=	≫	=	=	≫
Polinómico	Sigmoide	=	=	=	=	<	=
Polinómico	NB	≫	≫	≫	≫	≫	≫
Polinómico	LIBLINEAR	≪	≪	≪	≪	≪	≪
RBF	Sigmoide	=	=	≪	=	=	≪
RBF	NB	≫	≫	≫	≫	≫	≫
RBF	LIBLINEAR	<	≪	≪	≪	≪	≪
Sigmoide	NB	≫	≫	≫	≫	≫	≫
Sigmoide	LIBLINEAR	≪	≪	≪	≪	≪	≪
NB	LIBLINEAR	≪	≪	≪	≪	≪	≪

Tabla 4: Comparación estadística de los resultados obtenidos con cada uno de los kernels y algoritmos definidos.

Algoritmo	Español		Catalán	
	2-gramas	1+2-gramas	2-gramas	1+2-gramas
Lineal	≫	=	≫	=
Polinómico	≫	=	≫	=
RBF	≫	≫	≫	≫
Sigmoide	≫	=	≫	=
NB	≫	=	≫	=
LIBLINEAR	≫	≪	≫	=

Tabla 5: Comparación estadística de los vectores de aprendizaje tomando como base la precisión obtenida con el vector de 1-gramas.

	Inglés	Español	Catalán
1-gramas	3.764	4.164	4.190
2-gramas	8.465	8.578	8.625
1+2-gramas	12.229	12.742	12.815

	Inglés	Español	Catalán
1-gramas	81,77	80,92	80,66
2-gramas	76,84	75,25	75,97
1+2-gramas	81,64	81,25	80,84

Tabla 6: Tamaño del vector de aprendizaje para cada uno de los vectores de características.

La tabla 7 muestra los resultados obtenidos. La precisión ofrecida por el clasificador en inglés es ligeramente superior al de los otros dos idiomas para los tres vectores de aprendizaje propuestos. En el caso de español y catalán, el primero obtiene resultados ligeramente favorables para 1-gramas y 1+2-gramas con respecto al segundo. Sin embargo, ninguno de estas diferencias resulta ser significativa ($p < 0,01$ y $p < 0,05$).

Podemos concluir que las características propias de cada uno de estos tres idiomas no mejoran o empeoran el rendimiento final del sistema. El aumento de vocabulario que existe en español y catalán con respecto al inglés tampoco parece

Tabla 7: Resultados de la comparación entre idiomas con el kernel lineal. No existen diferencias significativas de precisión entre los tres idiomas para ninguno de los vectores de aprendizaje.

afectar al proceso de clasificación.

5. Trabajo relacionado

Pese a los numerosos trabajos realizados dentro del área de la clasificación de preguntas, son escasas las aproximaciones dedicadas a idiomas diferentes del inglés o su aplicación a sistemas multilingües.

Dentro de los sistemas en inglés, destacan los trabajos realizados por Xin Li y Dan Roth (2002; 2005), que sirvieron para establecer la tarea de clasificación de preguntas como una tarea inde-

pendiente de los sistemas QA y evaluable por sí misma. En estos trabajos desarrollaron un clasificador jerárquico de preguntas en inglés, basado en la arquitectura de aprendizaje SNoW (Sparse Network of Winnows) (Carlson et al., 1999).

En lo que respecta al uso de kernels para esta tarea, hay numerosos sistemas que han usado SVM en su forma más básica mediante el empleo de un kernel lineal, demostrando sistemáticamente el buen funcionamiento de este algoritmo con respecto a otros en la tarea de clasificación de preguntas. Entre los sistemas que emplean esta aproximación podemos destacar los de Hacioglu y Ward (2003), Krishnan et al. (2005), Skowron y Araki (2004), Nguyen et al. (2008), Day et al. (2007), Solorio et al. (2004), Bisbal et al. (2005) y Tomás et al. (2005).

Otro kernel muy extendido en clasificación de preguntas 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 superficial codificada mediante estructuras predicativas (*Predicate-Argument Structures*). Este 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. (2003) se define un kernel denominado *Hierarchical Directed Acyclic Graph* (HDAG). 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.

Todos estos kernels han sido aplicados únicamente para inglés. En este caso, a la falta de corpus en otros idiomas se une la dependencia del sistema con respecto a los recursos utilizados, debido a que estos kernels sintáctico-semánticos requieren de herramientas de análisis lingüístico y bases de conocimiento que no existen o son difíciles de conseguir para otros idiomas.

Existen algunas aproximaciones que se han adentrado en idiomas diferentes al inglés, como por ejemplo 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, Peng, y Liu, 2006), el japonés (Suzuki et al., 2003), el portugués (Solorio et al., 2005) y el español (Ángel García Cumberras et al., 2005; Tomás et al., 2005). Ninguno de estos sistemas ha afrontado la tarea en catalán, y son escasos aquellos que afrontan la tarea en más de un idioma (Solorio et al., 2005; Bisbal et al., 2005).

6. Conclusiones y trabajo futuro

En este trabajo hemos presentado una aproximación a la clasificación de preguntas basada en kernels para español y catalán, tratando de cubrir el vacío existente en el campo de la clasificación de preguntas para idiomas distintos del inglés. La ausencia de conjuntos de preguntas en estos idiomas nos ha llevado a desarrollar nuestros propios corpus de preguntas. Estos corpus han sido puestos libremente a disposición de la comunidad científica.

El sistema propuesto se basa en la utilización de SVM con diferentes kernels: lineal, polinómico, RBF y sigmoide. Para obtener el mejor rendimiento posible de estos kernels se ha realizado un ajuste de los parámetros para su funcionamiento. El ajuste correcto de los parámetros ha demostrado una mejora significativa para los kernels polinómico, RBF y sigmoide en ambos idiomas. En el caso del kernel lineal se ha conseguido mejorar los resultados, pero esta mejora no ha resultado ser estadísticamente significativa.

Una vez determinados los mejores parámetros para SVM y los kernels, hemos comparado el funcionamiento de éstos entre sí. Los resultados revelaron un funcionamiento similar de los cuatro kernels para los vectores de 1-gramas y 2-gramas, obteniendo resultados por encima del 80% en ambos idiomas. La única diferencia significativa se da en el caso del kernel RBF y el vector 1+2-gramas, resultando significativamente peor que para el resto de kernels. Este resultado desaconseja el uso de RBF con vectores de aprendizaje de gran tamaño.

Además de la comparativa entre kernels, hemos analizado su rendimiento con respecto a otros dos algoritmos, Naïve Bayes y LIBLINEAR, este último especialmente adecuado para trabajos en espacios de alta dimensionalidad. Los resultados obtenidos con Naïve Bayes fueron estadísticamente inferiores a los obtenidos con los cuatro kernels empleados, demostrando el buen funcionamiento de éstos en espacios de alta di-

mensionalidad. Por otra parte, resulta destacable el rendimiento obtenido por LIBLINEAR, que superó a los cuatro kernels definidos en ambos idiomas. Este algoritmo no había sido aplicado con anterioridad a la tarea de clasificación de preguntas.

Por otra parte, hemos experimentado con diversos vectores de características. Los resultados revelan que al trabajar con kernels, el uso de 1-gramas mejora significativamente al de 2-gramas, y en ningún caso obtiene resultados significativamente inferiores a los obtenidos con la combinación de ambos (1+2-gramas). Sólo en el caso del algoritmo LIBLINEAR se obtiene una mejora significativa al emplear un vector de gran tamaño formado por 1+2-gramas.

Por último hemos realizado una comparación de rendimiento del clasificador entre idiomas, añadiendo en este caso el corpus en inglés a nuestro conjunto de experimentos. Los resultados obtenidos revelan que un kernel lineal aplicado sobre un corpus paralelo en inglés, español y catalán, obtiene resultados muy similares para los tres idiomas. En los experimentos realizados existe una aparente correlación entre el tamaño del vocabulario y el rendimiento del sistema, siendo éste ligeramente mejor en inglés que en español y en catalán. Sin embargo, ninguno de los resultados obtenidos para inglés es significativamente mejor que para el resto de idiomas. La aproximación basada en kernels y características textuales superficiales resulta, por tanto, adecuada para su aplicación en entornos multilingües.

Con el fin de obtener sistemas de alto rendimiento para español y catalán, se plantea como trabajo futuro el enriquecimiento del vector de características empleando información lingüística más profunda (usando herramientas como FreeLing (Atserias et al., 2006), que ofrece numerosas aplicaciones lingüísticas en los dos idiomas citados). Los resultados obtenidos con LIBLINEAR hacen de este algoritmo una apuesta destacable de cara al trabajo futuro con estos vectores enriquecidos.

Bibliografía

Ángel García Cumberras, Miguel, Fernando Martínez Santiago, Luis Alfonso Ureña López, y Arturo Montejó Ráez. 2005. Búsqueda de respuestas multilingüe : clasificación de preguntas en español basada en aprendizaje. *Procesamiento del Lenguaje Natural*, (34):31–40, March. <http://www.sepln.org/revistaSEPLN/revista/34/03.pdf>.

Atserias, Jordi, Bernardino Casas, Elisabet Comelles, Meritxell González, Lluís Padró, y

Muntsa Padró. 2006. Freeling 1.3: Syntactic and semantic services in an open-source nlp library. En *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC'06)*, páginas 48–55. <http://www.lsi.upc.edu/~nlp/papers/atserias06.pdf>.

Aunimo, Lili y Reeta Kuuskoski. 2005. Reformulations of finnish questions for question answering. En *Proceedings of the 15th NODALIDA conference*, páginas 12–21. <http://phon.joensuu.fi/lingjoy/01/aunimoF.pdf>.

Bisbal, Empar, David Tomás, Lidia Moreno, José L. Vicedo, y Armando Suárez. 2005. A multilingual svm-based question classification system. En Alexander F. Gelbukh Alvaro de Albornoz, y Hugo Terashima-Marín, editores, *MICAI 2005: Advances in Artificial Intelligence, 4th Mexican International Conference on Artificial Intelligence*, volumen 3789 de *Lecture Notes in Computer Science*, páginas 806–815. Springer, November. <http://www.springerlink.com/index/75jr3067j3472680.pdf>.

Carlson, Andrew, Chad Cumby, Jeff Rosen, y Dan Roth. 1999. The snow learning architecture. Informe Técnico UIUCDCS-R-99-2101, UIUC Computer Science Department, May.

Chang, Chih Chung y Chih Jen Lin, 2001. *LIBSVM: a library for support vector machines*. <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.

Collins, Michael y Nigel Duffy. 2001. Convolution kernels for natural language. En *Advances in Neural Information Processing Systems (NIPS14)*, páginas 625–632. MIT Press. <http://12r.cs.uiuc.edu/~danr/Teaching/CS546-09/Papers/Collins-kernels.pdf>.

Day, Min-Yuh, Cheng-Wei Lee, Shih-Hung Wu, Chorng-Shyong Ong, y Wen-Lian Hsu. 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*, páginas 620–625, October.

Day, Min-Yuh, Chorng-Shyong Ong, y Wen-Lian Hsu. 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*, páginas 203–208, August.
- Dietterich, Thomas G. 1998. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10:1895–1923. <http://web.engr.oregonstate.edu/~tgd/publications/nc-stats.ps.gz>.
- Duda, Richard O. y Peter E. Hart, 1973. *Pattern Classification and Scene Analysis*, páginas 98–105. John Wiley and Sons.
- Fan, Rong-En, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, y Chih-Jen Lin. 2008. Liblinear: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874. <http://www.csie.ntu.edu.tw/~cjlin/papers/liblinear.pdf>.
- Feiguina, Olga y Balázs Kégl. 2005. Learning to classify questions. En *CLINE 05: 3rd Computational Linguistics in the North-East Workshop*, August. http://www.crtl.ca/cline05/cline05_papers/FeiguinaKegl.pdf.
- Fleiss, Joseph L. 1971. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382.
- Hacioglu, Kadri y Wayne Ward. 2003. Question classification with support vector machines and error correcting codes. En *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, páginas 28–30, Morristown, NJ, USA. Association for Computational Linguistics. <http://www.aclweb.org/anthology/N/N03/N03-2010.pdf>.
- Hennoste, Tiit, Olga Gerassimenko, Riina Kasterpalu, Mare Koit, Andriela Rääbis, Krista Strandson, y Maret Valdisoo. 2005. Questions in estonian information dialogues: Form and functions. En *Text, Speech and Dialogues*, volumen 3658, páginas 420–427. Springer Berlin / Heidelberg. <http://www.springerlink.com/index/6C2298L0XC04T08B.pdf>.
- Hsu, Chih Wei, Chih Chung Chang, y Chih Jen Lin. 2003. A practical guide to support vector classification. Informe técnico, Taipei. <http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf>.
- Joachims, Thorsten. 1998. Text categorization with support vector machines: learning with many relevant features. En Claire Nédellec y Céline Rouveirol, editores, *Proceedings of ECML-98, 10th European Conference on Machine Learning*, numero 1398, páginas 137–142, Chemnitz, DE. Springer Verlag, Heidelberg, DE. http://www.joachims.org/publications/joachims_98a.ps.gz.
- Kando, Noriko. 2005. Overview of the fifth ntcir workshop. En *Proceedings of NTCIR-5 Workshop*, Tokyo, Japan. <http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings5/data/NTCIR5-0V-KandoN.pdf>.
- Krishnan, Vijay, Sujatha Das, y Soumen Chakrabarti. 2005. Enhanced answer type inference from questions using sequential models. En *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, páginas 315–322, Morristown, NJ, USA. Association for Computational Linguistics. <http://www.cse.iitb.ac.in/~soumen/doc/emnlp2005/382.pdf>.
- Kudo, Taku y Yuji Matsumoto. 2001. Chunking with support vector machines. En *NAACL*. <http://www.aclweb.org/anthology/N/N01/N01-1025.pdf>.
- Li, Xin y Dan Roth. 2002. Learning question classifiers. En *Proceedings of the 19th international conference on Computational linguistics*, páginas 1–7, Morristown, NJ, USA. Association for Computational Linguistics. <http://www.aclweb.org/anthology/C/C02/C02-1150.pdf>.
- Li, Xin y Dan Roth. 2005. Learning question classifiers: the role of semantic information. *Natural Language Engineering*, 12(3):229–249. <http://12r.cs.uiuc.edu/~danr/Papers/LiRo05a.pdf>.
- Lin, Xu-Dong, Hong Peng, y Bo Liu. 2006. Support vector machines for text categorization in chinese question classification. En *WI '06: Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*, páginas 334–337.
- Magnini, Bernardo, Simone Romagnoli, Alessandro Vallin, Jesús Herrera, Anselmo Peñas, Víctor Peinado, Felisa Verdejo, y Maarten de Rijke. 2003. Creating the disequa corpus: A test set for multilingual question answering. En *Cross-Lingual Evaluation Forum (CLEF) 2003 Workshop*, páginas 311–320. <http://www.springerlink.com/index/6135q8c17e864nmn.pdf>.
- Magnini, Bernardo, Alessandro Vallin, Christelle Ayache, Gregor Erbach, Anselmo Peñas,

- Maarten de Rijke, Paulo Rocha, Kiril Ivanov Simov, y Richard F. E. Sutcliffe. 2005. Overview of the clef 2004 multilingual question answering track. En Carol Peters Paul Clough Julio Gonzalo Gareth J. F. Jones Michael Kluck, y Bernardo Magnini, editores, *Multilingual Information Access for Text, Speech and Images, 5th Workshop of the Cross-Language Evaluation Forum, CLEF 2004*, volumen 3491 de *Lecture Notes in Computer Science*, páginas 371–391. Springer. <http://www.springerlink.com/index/ebtpv2e71eg4txbu.pdf>.
- Mitchell, Tom M. 1997. *Machine Learning*. McGraw-Hill Science/Engineering/Math, March.
- Moldovan, Dan, Marius Paşca, Sanda Harabagiu, y Mihai Surdeanu. 2003. Performance issues and error analysis in an open-domain question answering system. *ACM Transactions on Information Systems*, 21(2):133–154.
- Moschitti, Alessandro, Silvia Quarteroni, Roberto Basili, y Suresh Manandhar. 2007. Exploiting syntactic and shallow semantic kernels for question answer classification. En *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, páginas 776–783, Prague, Czech Republic, June. Association for Computational Linguistics. <http://www.ist-luna.eu/pdf/ACL07.pdf>.
- Nguyen, Tri Thanh, Le Minh Nguyen, y Akira Shimazu. 2008. Using semi-supervised learning for question classification. *Information and Media Technologies*, 3(1):112–130. <http://www.springerlink.com/index/y85h00v3825r4081.pdf>.
- Pan, Yan, Yong Tang, Luxin Lin, y Yemin Luo. 2008. Question classification with semantic tree kernel. En *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, páginas 837–838, New York, NY, USA. ACM.
- Radev, Dragomir, Weiguo Fan, Hong Qi, Harris Wu, y Amardeep Grewal. 2002. Probabilistic question answering on the web. En *WWW '02: Proceedings of the 11th international conference on World Wide Web*, páginas 408–419, New York, NY, USA. ACM. <http://filebox.vt.edu/users/wfan/paper/www/www.pdf>.
- Schölkopf, Bernhard y Alexander J. Smola. 2001. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA.
- Sekine, Satoshi, Kiyoshi Sudo, y Chikashi Nobata. 2002. Extended named entity hierarchy. En *LREC 2002: Language Resources and Evaluation Conference*, páginas 1818–1824, Las Palmas, Spain. <http://nlp.cs.nyu.edu/pubs/papers/sekine-lrec02.pdf>.
- Shawe-Taylor, John y Nello Cristianini. 2004. *Kernel Methods for Pattern Analysis*. Cambridge University Press, New York, NY, USA, June.
- Skowron, Marcin y Kenji Araki. 2004. 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, October.
- Solorio, Tamar, no Manuel Pérez-Couti Manuel Montes y Gémez, nor-Pineda Luis Villase y Aurelio López-López. 2004. A language independent method for question classification. En *COLING '04: Proceedings of the 20th international conference on Computational Linguistics*, páginas 1374–1380, Morristown, NJ, USA. Association for Computational Linguistics. <http://www.aclweb.org/anthology/C/C04/C04-1201.pdf>.
- Solorio, Tamar, Manuel Pérez-Couti no, Manuel Montes y Gómez, Luis Villase nor Pineda, y Aurelio López-López. 2005. Question classification in spanish and portuguese. En *CICLing*, páginas 612–619. <http://www.springerlink.com/index/46d3nw2qpe7tpx3f.pdf>.
- Sundblad, Håkan. 2007. Question classification in question answering. Master's thesis, Linköping University, Department of Computer and Information Science. <http://liu.diva-portal.org/smash/get/diva2:23705/FULLTEXT01>.
- Suzuki, Jun, Hirotohi Taira, Yutaka Sasaki, y Eisaku Maeda. 2003. Question classification using hdag kernel. En *Proceedings of the ACL 2003 workshop on Multilingual summarization and question answering*, páginas 61–68, Morristown, NJ, USA. Association for Computational Linguistics. <http://www.aclweb.org/anthology/W/W03/W03-1208.pdf>.
- Tomás, David, José L. Vicedo, Armando Suárez, Empar Bisbal, y Lidia Moreno. 2005. Una aproximación multilingüe a la clasificación de preguntas basada en aprendizaje automático. *Procesamiento del Lenguaje Natural*, (35):391–398. <http://www.sepln.org/revistaSEPLN/revista/35/48.pdf>.

- Vallin, Alessandro, Bernardo Magnini, Danilo Giampiccolo, Lili Aunimo, Christelle Aya-che, Petya Osenova, Anselmo Peñas, Maarten de Rijke, Bogdan Sacaleanu, Diana Santos, y Richard Sutcliffe. 2006. Overview of the clef 2005 multilingual question answering track. En Springer Berlin / Heidelberg, editor, *Accessing Multilingual Information Repositories*, volumen 4022 de *Lecture Notes in Computer Science*, páginas 307–331. <http://www.springerlink.com/index/dm61h684k55150p2.pdf>.
- Voorhees, Ellen M. 1999. The trec-8 question answering track report. En *Eighth Text REtrieval Conference*, volumen 500-246 de *NIST Special Publication*, páginas 77–82, Gaithersburg, USA, November. National Institute of Standards and Technology. http://trec.nist.gov/pubs/trec8/papers/qa_report.ps.
- Voorhees, Ellen M. 2001. The trec question answering track. *Natural Language Engineering*, 7(4):361–378.
- Witten, Ian H. y Eibe Frank. 2005. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, San Francisco, 2 edición.
- Zhang, Dell y Wee Sun Lee. 2003. Question classification using support vector machines. En *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, páginas 26–32, New York, NY, USA. ACM. <http://www.comp.nus.edu.sg/~leews/publications/p31189-zhang.pdf>.