Procesamiento del Lenguaje Natural, núm. 35 (2005), pp. 391-398 recibido 02-05-2005; aceptado 01-06-2005

Una aproximación multilingüe a la clasificación de preguntas basada en aprendizaje automático*

David Tomás, José. L. Vicedo, Armando Suárez
{dtomas,vicedo,armando}@dlsi.ua.es
Depto. de Lenguajes y Sistemas Informáticos, Universidad de Alicante (España)

Empar Bisbal y **Lidia Moreno**
{ebisbal,lmoreno}@dsic.upv.es
Depto. de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia (España)

Resumen: Este artículo presenta una aproximación multilingüe a la clasificación de preguntas basada en aprendizaje automático, empleando características de aprendizaje independientes del idioma. Esto va a permitir que el sistema sea flexible y fácilmente adaptable a nuevos idiomas. Sobre un corpus paralelo de preguntas en inglés y castellano, contrastaremos el rendimiento del sistema con tres métodos distintos de aprendizaje: Máquinas de Vectores Soporte, Máxima Entropía y aprendizaje basado en ejemplos.
Palabras clave: clasificación de preguntas, multilingüe, aprendizaje automático, Máquinas de Vectores Soporte, Máxima Entropía, aprendizaje basado en ejemplos

Abstract: This paper presents a multilingual approach to question classification based on machine learning. This approach is flexible and easy to adapt to new languages using language independent learning features. The system's performance has been compared in a parallel English-Spanish corpus of questions using three different learning methods: *Support Vector Machines*, *Maximum Entropy* and *Memory-based Learning*.
Keywords: Question Classification, multilingual, Machine Learning, Support Vector Machines, Maximum Entropy, Memory-based Learning

1. Introducción

La búsqueda de respuestas se ha convertido en uno de los principales retos en el campo del procesamiento del lenguaje natural. Plantea el desafío de obtener respuestas exactas a preguntas formuladas empleando lenguaje natural. En este sentido se diferencia de los sistemas de recuperación de información, en los que ante una petición de usuario se devuelve una serie de documentos relevantes a la misma ordenados según criterios de ponderación propios de cada sistema.

Una gran mayoría de sistemas de búsqueda de respuestas (Moldovan et al., 2002) (Soubbotin y Soubbotin, 2002) (Magnini et al., 2002) (Yang y Chua, 2002) organizan su funcionamiento en tres fases bien diferenciadas: análisis de la pregunta, selección de documentos o pasajes relevantes y extracción de la respuesta. El proceso de análisis de la pregunta es crítico debido a que condiciona notablemente el rendimiento del sistema en las fases posteriores. Una de las tareas llevadas a cabo en esta etapa es la clasificación de la pregunta, consistente en categorizar las preguntas formuladas como pertenecientes a una jerarquía de tipos predefinida por el sistema. Con esta categorización en diferentes clases semánticas se consigue imponer restricciones a las posibles respuestas devueltas por el sistema de búsqueda en la fase final de extracción. Por ejemplo, si consideramos una pregunta del tipo "¿Cuál es la ciudad más grande de Alemania?", el sistema de clasificación de preguntas podría determinar que pertenece a la clase "ciudad", de forma que al tratar de extraer la respuesta se tendrían en consideración como candidatas sólo aquellas que fueran nombres de ciudades.

En este trabajo se ha desarrollado un sistema de clasificación de preguntas multilingüe, basado en técnicas de aprendizaje automáti-

* Este trabajo ha sido subvencionado por el proyecto CICYT R2D2 (TIC2003-07158-C04) y por el Ministerio de Educación y Ciencia y el Fondo Social Europeo a través de la beca BES-2004-3935

co, empleando las mismas características de aprendizaje para todos los idiomas estudiados. Se ha comparado la eficacia del sistema de clasificación utilizando tres métodos diferentes de aprendizaje: Máquinas de Vectores Soporte, Máxima Entropía y aprendizaje basado en ejemplos. Empleando características textuales superficiales, se pretende obtener un sistema flexible que pueda ser aplicado a tareas multilingües y que se ajuste fácilmente a diferentes dominios.

El artículo presenta la siguiente estructura: la Sección 2 plantea el problema de la clasificación de preguntas y la situación actual de estos sistemas; en la Sección 3 se describen los diferentes métodos de aprendizaje automático evaluados; en la Sección 4 se describe el sistema, la jerarquía de clases empleada, el corpus sobre el que se han efectuado las pruebas y las características empleadas para el aprendizaje; la Sección 5 muestra los experimentos realizados y los resultados obtenidos; finalmente, la Sección 6 expone las conclusiones de nuestro trabajo.

2. Clasificación de preguntas

Se define la clasificación de preguntas como la tarea de asignar a una pregunta dada una clase dentro de una jerarquía previamente definida, proporcionando de esta manera una restricción semántica en la búsqueda posterior de la respuesta. La intención es que esta clasificación, junto con otras posibles restricciones impuestas por el sistema, sirva para acotar la selección de respuestas candidatas. Esta acotación se puede manifestar también a la hora de emplear diferentes estrategias de procesamiento en la búsqueda de la respuesta. Se pueden tener estrategias diferentes para cada clase identificada como, por ejemplo, el uso de patrones específicos para la extracción de respuestas de tipo definición. Esto da una idea de la importancia que tiene la clasificación correcta de las preguntas en los sistemas de búsqueda de respuestas. Un estudio realizado sobre el análisis de errores en sistemas de dominio abierto (Moldovan et al., 2003), revelaba que más de un 36 % de éstos eran directamente imputables al módulo de clasificación de la pregunta.

La mayoría de sistemas de clasificación de preguntas realizan esta tarea a partir de reglas heurísticas y patrones definidos de forma manual (Hermjakob, 2001) (Voorhees, 2000) (Voorhees, 2001). Existen dos grandes problemas con este tipo de sistemas. El primero es la cantidad de trabajo necesario para formular los patrones que capturan la clase de la pregunta, ya que existen numerosas variantes lingüísticas para expresar una misma cuestión. Por ejemplo, "¿Por qué es famosa Jane Goodall?", "¿Qué hizo famosa a Jane Goodall?", "¿Cómo consiguió la fama Jane Goodall?", son tres formulaciones distinguibles de una misma cuestión que implicarían la definición de patrones muy diversos para su captura. El segundo problema es la falta de flexibilidad y la dependencia del dominio: un cambio en el campo de aplicación o la inclusión de nuevas clases de preguntas conllevaría la revisión y posible redefinición de las heurísticas y patrones iniciales del sistema.

Los sistemas de clasificación basados en métodos de aprendizaje automático surgen con la idea de crear aplicaciones más flexibles, que se adapten a cambios en el entorno y aprendan a partir de corpus de entrenamiento. Son diversas las técnicas y estrategias empleadas para conseguir este propósito (Li y Roth, 2002) (Hacioglu y Ward, 2003). A la hora de entrenar resultan determinantes el tipo características en las que queremos que se base nuestro sistema de aprendizaje. Debido a que la cantidad de texto que aporta una pregunta es relativamente escasa (suelen ser frases cortas), algunas aproximaciones optan por usar información lingüística compleja más allá de las meras palabras, como puede ser la identificación de sintagmas, el análisis sintáctico, el análisis semántico (Li y Roth, 2002) o el etiquetado de entidades (Hacioglu y Ward, 2003). El problema del uso de estas herramientas lingüísticas es la dependencia que crean con respecto al idioma y el hecho de que no están disponibles para todos los idiomas, con el añadido de que la mayoría de ellas son muy sensibles al dominio del corpus (suelen ser herramientas entrenadas sobre determinados corpus y por tanto viciadas por éstos).

En este artículo se presenta un sistema de clasificación de preguntas para inglés y castellano basado en técnicas de aprendizaje automático. Se han empleado características de aprendizaje con bajo nivel de conocimiento del lenguaje, a fin de que el sistema pueda ser aplicado de forma flexible a diferentes idiomas y diferentes clasificaciones de preguntas. Se ha utilizado un corpus de aprendizaje bilingüe con preguntas de tipo factual y defi-

nición (Voorhees, 2003) en dominio abierto. Se han contrastado los resultados obtenidos mediante el uso de tres métodos diferentes de aprendizaje: Máxima Entropía, Máquinas de Vectores Soporte y aprendizaje basado en ejemplos.

3. Métodos de aprendizaje

Con este trabajo se pretende obtener un sistema de clasificación multilingüe basado en técnicas de aprendizaje automático que utilice las mismas características de aprendizaje para diferentes idiomas. Hemos empleado tres métodos pertenecientes a diferentes familias de algoritmos para contrastar su eficiencia: Máquinas de Vectores Soporte como método de la teoría del aprendizaje computacional, Máxima Entropía como representante de los algoritmos de aprendizaje estadístico y aprendizaje basado en ejemplos como representante de los métodos tradicionales de Inteligencia Artificial. Hemos definido un vector de características común y lo hemos aplicado a la misma tarea (entrenando y evaluando sobre los mismos corpus) con el fin de decidir cuál es el más apropiado para alcanzar nuestro objetivo.

Cualquier tarea de clasificación que se desee abordar mediante aprendizaje automático supervisado necesita, generalmente, dos corpus: uno para entrenar el modelo y otro para probarlo. En los métodos comparados en este trabajo, cada instancia del corpus de entrenamiento está formada por varios atributos[1] y una clase. El objetivo de estos métodos consiste en obtener un modelo capaz de predecir la clase de las instancias de otro corpus, de las que sólo se conocen los atributos.

3.1. Máquinas de Vectores Soporte

Las Máquinas de Vectores Soporte (*Support Vector Machines*, SVM) tratan de encontrar un hiperplano óptimo (frontera) que sea capaz de separar un conjunto de muestras binario. Para ello se extraen las muestras más cercanas a la frontera, a las que se conoce como vectores soporte. El hiperplano óptimo es aquel que maximiza el margen o distancia entre la frontera y dichos vectores soporte.

Más formalmente, sea un corpus de entrenamiento de pares (x_i, y_i) tal que $i = 1..m$ siendo m el número de muestras, donde x_i es el vector de características ($x_i \in \mathbb{R}^n$) y y_i la etiqueta que indica si la muestra x_i pertenece o no a la clase ($y_i \in \{1, -1\}^m$), los SVM (Boser, Guyon, y Vapnik, 1992) (Cortes y Vapnik, 1995) obtienen la solución al siguiente problema de optimización,

$$\begin{aligned} \min_{w,b,\xi} \quad & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}\xi_i \\ \textit{siendo} \quad & y_i(w^T x_i + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0 \end{aligned}$$

donde la función $w^T x_i + b$ representa el hiperplano buscado, C es un parámetro de compromiso entre el error cometido ξ_i y el margen, y w es un vector de pesos. Las variables ξ_i fueron introducidas para abordar problemas que no fueran linealmente separables, en los que se permite cierto error de clasificación.

Las Máquinas de Vectores Soporte fueron diseñadas originalmente para resolver problemas de clasificación binaria. Para abordar el problema de la clasificación en k clases existen dos aproximaciones básicas: uno-contra-todos (*one-versus-all*), donde se entrenan k SVM y cada uno separa una clase del resto, y uno-contra-uno (*one-against-one*), donde se han de entrenar $\frac{k(k-1)}{2}$ modelos y cada modelo discrimina entre un par de clases. Es importante notar que como uno-contra-uno trabaja 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 SVM. La complejidad de prueba de ambas estrategias es similar: uno-contra-todos necesita k evaluaciones y uno-contra-uno $k - 1$.

Para los experimentos se ha utilizado la implementación de SVM proporcionada por el conjunto de herramientas de aprendizaje automático WEKA (Witten y Frank, 2000). Esta implementación utiliza la técnica de uno-contra-uno para abordar los problemas de clasificación en k-clases. El algoritmo de optimización implementado en WEKA para el entrenamiento de los SVM es el *Sequential Optimization Algorithm* de Platt (Platt, 1999). Tras varios experimentos, se decidió que el *kernel* más apropiado (aquel que daba mejores resultados y cuyo tiempo de entrenamiento era menor) era el lineal, con el parámetro C establecido a 1. Estos parámetros son los que WEKA ofrece por defecto.

[1] En la literatura inglesa se los conoce como *features*

3.2. Máxima Entropía

El modelado con Máxima Entropía (*Maximum Entropy*, ME) (Berger, Pietra, y Pietra, 1996) permite integrar información de diversas fuentes heterogéneas para tareas de clasificación. 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 probabilidades que maximiza la entropía, es decir, se asume la máxima ignorancia no teniendo 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 la mayor probabilidad condicional en el contexto $x : cl(x) = argmax_c p(c|x)$. Cada atributo se calcula mediante una función asociada a una clase particular c' y que tiene la forma (1), donde $cp(x)$ es una característica observable en el contexto.

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

La probabilidad condicional $p(c|x)$ se define como (2), 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.

$$p(c|x) = \frac{1}{Z(x)} \prod_{i=1}^{K} \alpha_i^{f_i(x,c)} \quad (2)$$

Para llevar a cabo los experimentos hemos empleado una implementación propia de *maxent* (Suárez y Palomar, 2002), utilizando como procedimiento de estimación de coeficientes GIS (*Generalized Iterative Scaling*), sin *smoothing* ni selección de atributos.

3.3. Aprendizaje basado en ejemplos

El aprendizaje basado en ejemplos (*Memory-based Learning*, MBL) es un tipo de aprendizaje supervisado, fundamentado 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. Cada uno de estos ejemplos consiste en un vector de características de tamaño fijo y un valor identificando la clase a la que pertenece. 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 (*k-nearest neighbors*, k-NN), asignándole la categoría mayoritaria encontrada.

En este trabajo se ha utilizado el software TiMBL (Daelemans et al., 2003) para la realización de los experimentos, empleando el algoritmo de aprendizaje por defecto IB1-GT. Para determinar la similitud $\Delta(X,Y)$ entre una nueva instancia X y los ejemplos memorizados Y, se ha empleado el algoritmo k-NN con la métrica descrita por (3), donde δ es la distancia entre cada una de las características (4) y w_i el ratio de ganancia (*Gain Ratio*) que pondera cada una de las características empleadas en función de cuanta información proporcionan a la hora de determinar la correcta clasificación de la nueva instancia X.

$$\Delta(X,Y) = \sum_{i=1}^{n} w_i \delta(x_i, y_i) \quad (3)$$

$$\delta(X_i,Y_i) = \begin{cases} abs(\frac{x_i - y_i}{max_i - min_i}) & \text{si es numérico} \\ 0 & \text{si } x_i = y_i \\ 1 & \text{si } x_i \neq y_i \end{cases} \quad (4)$$

4. Descripción del sistema

El sistema desarrollado clasifica en función de la jerarquía de clases descrita por (Sekine, Sudo, y Nobata, 2002), donde se define una clasificación a varios niveles. Como se comenta en la sección siguiente, se ha modificado ligeramente dicha clasificación y se ha trabajado sólo con el primer nivel, que consta finalmente de 15 categorías: PROPER_NAME, ORGANIZATION, LOCATION, FACILITY, PRODUCT, DISEASE, EVENT, LANGUAGE, RELIGION, NATURAL_OBJECT, COLOR, TIME_TOP, NUMEX, DEFINITION y ACRONYM.

4.1. Corpus

Para la realización de este trabajo y de los experimentos que aquí se detallan se

contó con una dificultad añadida: no había corpus disponible con las características que se necesitaban. Se optó por desarrollar un corpus propio de preguntas etiquetadas con el tipo de respuesta que éstas esperaban. Para ello, utilizando como base la jerarquía de entidades descrita en (Sekine, Sudo, y Nobata, 2002), se definió una clasificación a tres niveles y se etiquetó con ella un corpus formado por las preguntas de las conferencias TREC [2] de 1999 a 2003 [3]. Para los experimentos en castellano se partió de las traducciones de las preguntas del TREC de 1999, 2000, 2001 y 2002 del Grupo de Procesamiento del Lenguaje Natural de la UNED[4]. Para obtener el mismo corpus en inglés y castellano, se tradujeron las preguntas del TREC 2003 y, con el fin de obtener una traducción uniforme, se revisaron las preguntas de los anteriores TREC. De esta forma se obtuvo un corpus paralelo de 2393 preguntas en inglés y castellano etiquetadas con la clasificación definida.

4.2. El vector de características

El vector de características es el conjunto de datos a partir de los que aprenderá nuestro sistema. Este conjunto supone una primera aproximación para contrastar el rendimiento de nuestro sistema, basado en aprendizaje, frente a los sistemas tradicionales basados en patrones, siendo susceptible de refinamiento en trabajos futuros. Hemos usado características textuales superficiales con la intención de limitar el empleo de herramientas dependientes del idioma y conseguir así un sistema fácilmente adaptable a nuevos lenguajes. A las preguntas se les aplicó un preproceso para detectar el término interrogativo (*question word*), que supone la primera característica del vector. En el caso de preguntas enunciadas de forma imperativa, esta característica no se halla presente. Para estos casos y otros muchos en los que el término interrogativo no es suficiente para determinar la clase de la pregunta formulada (preguntas con el interrogativo "qué"), se hace necesario buscar información adicional que permita definir nuevas características. Se tomaron las tres palabras siguientes al término interrogativo (o al verbo en caso de preguntas imperativas) que no fueran *palabras de parada*[5] y con ellas se construyeron el resto de características, formadas por los unigramas y bigramas resultantes de la combinación de las tres palabras: primera palabra, segunda palabra, tercera palabra, primera+segunda palabra, primera+tercera palabra y segunda+tercera palabra.

A este vector inicial se le añadió una última característica que representa el número de palabras que no sean *palabras de parada*, que tiene la pregunta a partir del término interrogativo. Esta modificación se planteó con la idea de discriminar mejor las preguntas con el interrogativo "quién". Cuando una pregunta empieza por "quién" puede pertenecer a las clases PROPER_NAME o DEFINITION. Por ejemplo, "¿Quién es el autor de 'La divina comedia'?", sería de categoría PROPER_NAME, mientras que "¿Quién es Marlon Brando?" pertenecería a DEFINITION; las preguntas DEFINITION suelen ser muy cortas, formadas únicamente por el interrogativo y uno o dos términos adicionales.

5. Experimentos realizados

Hemos llevado a cabo una serie de experimentos para determinar qué método de aprendizaje, de los citados anteriormente, es el que mejor rendimiento aporta a nuestro sistema de clasificación.

Se experimentó con dos vectores de características, uno que incluía el número de palabras de la pregunta no pertenecientes al conjunto de *palabras de parada* a partir del término interrogativo y otro que no, a los que llamaremos *caract2* y *caract1* respectivamente. Otra de las variantes experimentadas consiste en emplear los términos originales de la pregunta o, por contra, emplear las *raíces*[6] de esos términos, buscando de esta forma conseguir una mayor generalización.

Los experimentos en inglés y castellano se realizaron inicialmente sobre el corpus descrito en la sección 4.1 empleando *10-fold cross validation*. El problema que presenta este corpus es que para algunas de las clases de nuestra jerarquía (DISEASE, EVENT, LANGUAGE, RELIGION y COLOR), el número

[2] Text REtrieval Conference, http://trec.nist.gov

[3] Corpus en http://trec.nist.gov/data/qa.html

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

[5] Las *palabras de parada* o *stopwords* son palabras que se repiten mucho en una colección y que, por lo tanto, no son relevantes para dicha colección. Las preposiciones y los artículos son ejemplos clásicos.

[6] *Stemming* es un método para reducir una palabra a su raíz (también conocido como *stem* o tema).

de muestras presentes es inferior a 50. Estos casos en los que el número de instancias de la clase es muy reducido, dificultan seriamente el rendimiento de los sistemas de aprendizaje. Para subsanar este problema fusionamos estas clases en una única, llamada MIXED, realizando los mismos experimentos que hicimos sobre la jerarquía original, con la diferencia de estar trabajando esta vez sobre 10 clases.

Finalmente, realizamos un experimento adicional sobre el corpus del *Cognitive Computation Group*[7] (Li y Roth, 2002). Este corpus consta de 5452 preguntas de entrenamiento y 500 más de test (pertenecientes al TREC10) en inglés, con 6 tipos posibles de clases a asignar. Los resultados obtenidos en este experimento, nos van a permitir comparar el rendimiento de nuestro sistema frente a otro sistema basado en el uso intensivo de información lingüística. Por otra parte, nos dará una idea de la flexibilidad del sistema ante variaciones en el corpus y en la jerarquía objeto de clasificación.

5.1. Resultados

Los cuadros 1, 2 y 3 muestran los resultados obtenidos empleando SMV, ME y MBL respectivamente. QA-R2D2 hace referencia al corpus compilado por nosotros, mientras que CCG hace referencia a los experimentos realizados sobre el corpus del *Cognitive Computation Group*. La tercera columna de cada tabla indica el número de clases distintas presentes en el experimento. La cuarta columna indica el uso o no de *raíces* en el conjunto de características de aprendizaje. Las dos últimas columnas muestran la precisión promedio obtenida con los dos vectores de características descritos en la sección 4.2 sobre una cobertura del 100 % (todas las preguntas fueron catalogadas dentro de alguna clase).

En general, SVM presenta un rendimiento superior al resto de técnicas de aprendizaje, solamente superado de forma puntual en algunos experimentos por MBL. El rendimiento de SVM en la clasificación de textos es un hecho contrastado, lo que justifica su buen funcionamiento en la tarea de clasificar preguntas. Sorprenden los resultados obtenidos con ME, notablemente inferiores a los otros dos sistemas. El problema puede deberse a una necesidad de parametrizar mejor la implementación empleada del algoritmo (Suárez y Palomar, 2002), o a que las características seleccionadas para el aprendizaje no sean las más adecuadas para este caso. Independientemente de los resultados obtenidos, ME resulta desaconsejable para su uso con corpus de gran tamaño debido al coste computacional requerido para su aplicación.

En lo referente a las características de aprendizaje empleadas, el uso de *raíces* en lugar de términos completos no parece afectar de forma determinante a los resultados, fluctuando entre pequeñas ganancias y pequeñas pérdidas según el experimento tratado. La información que aporta el incluir el número de palabras de la pregunta (*caract2*) o no (*caract1*) mejora ligeramente los resultados con SVM, mientras que los empeora paralelamente en ME y MBL.

En cuanto a los corpus de trabajo, la unión de clases con pocas muestras en una única clase (experimentos con 10 clases) tampoco ofrece una mejora destacable. La clase resultante de esa unión es demasiado heterogénea, sin preguntas que ofrezcan un patrón claramente definido. La mejoría que debería aportar el hecho de utilizar menos categorías se ve contrarrestada con el ruido que introduce la nueva clase. Por contra, los experimentos realizados sobre el corpus CCG resultan más reveladores, presentando mejoras sustanciales de precisión en todos los casos. Este comportamiento era de esperar en un sistema basado en aprendizaje, dada la presencia de mayor número de muestras para el entrenamiento (5452 frente a 2393) y el menor número de clases posibles con las que etiquetar las preguntas (6 frente a 15). Si comparamos nuestros resultados (87 % en el mejor de los casos para SVM) con los obtenidos por el sistema de clasificación de preguntas de (Li y Roth, 2002) (91 %) vemos que la diferencia no resulta demasiado amplia, teniendo en cuenta que su sistema se basa en el uso de características de aprendizaje muy dependientes del idioma, empleando herramientas de análisis sintáctico y semántico.

Por último, en lo referente al idioma, los resultados son muy similares para ambos casos en todos los experimentos. Ligeras ganancias a favor de las pruebas en castellano, sugieren que la variedad estructural de las oraciones en este idioma da como resultado unas características de aprendizaje más discrimi-

[7] http://l2r.cs.uiuc.edu/~cogcomp/. El corpus se encuentra en http://l2r.cs.uiuc.edu/~cogcomp/Data/QA/QC/

Corpus	Tipo test	Nº clases	Idioma	Raíz	Precisión *caract1*	Precisión *caract2*
QA-R2D2	10-fold cv	15	Inglés	SI	79.44 %	81.11 %
				NO	79.36 %	80.82 %
			Español	SI	80.32 %	81.91 %
				NO	80.32 %	81.91 %
		10	Inglés	SI	79.31 %	80.90 %
				NO	79.19 %	80.48 %
			Español	SI	80.36 %	81.61 %
				NO	80.11 %	81.70 %
CCG	train/test	6	Inglés	SI	85.40 %	86.00 %
				NO	85.40 %	87.00 %

Cuadro 1: Resultados SVM

Corpus	Tipo test	Nº clases	Idioma	Raíz	Precisión *caract1*	Precisión *caract2*
QA-R2D2	10-fold cv	15	Inglés	SI	69.12 %	68.03 %
				NO	69.91 %	68.07 %
			Español	SI	70.25 %	69.37 %
				NO	71.88 %	69.87 %
		10	Inglés	SI	69.20 %	68.12 %
				NO	69.70 %	67.95 %
			Español	SI	70.12 %	69.41 %
				NO	71.50 %	69.95 %
CCG	train/test	6	Inglés	SI	74.40 %	72.40 %
				NO	74.60 %	71.80 %

Cuadro 2: Resultados ME

natorias para ciertas clases.

6. Conclusiones

Se ha presentado en este artículo un sistema multilingüe de clasificación de preguntas basado en el aprendizaje automático. Los experimentos realizados con diversos métodos de aprendizaje nos llevan a optar por SVM como la mejor opción para la construcción de nuestro sistema.

Las características de aprendizaje utilizadas nos han reportado unos resultados muy similares para los dos idiomas de estudio, resultando adecuadas para nuestro objetivo de crear un sistema de clasificación independiente del idioma. Comparando la precisión obtenidas por nuestro clasificador con la obtenida por otros sistemas con características mucho más ligadas al idioma, podemos asegurar que la leve pérdida de precisión compensa a la hora de obtener un sistema flexible, fácilmente adaptable a nuevos idiomas, nuevos corpus y que no depende esencialmente de herramientas que pongan en riesgo la portabilidad a otros ámbitos.

A fin de mejorar los resultados del sistema en función de los experimentos realizados, el paso más inmediato es la adquisición de un mayor número de muestras de aprendizaje y el refinamiento de las clases objetivo. Como trabajo futuro planteamos la experimentación con otros idiomas y la inclusión del sistema como módulo dentro de la fase de análisis de la pregunta de un sistema completo de búsqueda de respuestas, para evaluar así su impacto en el rendimiento de este tipo de aplicaciones.

Bibliografía

Berger, Adam L., Vincent J. Della Pietra, y Stephen A. Della Pietra. 1996. A maximum entropy approach to natural language processing. *Comput. Linguist.*, 22(1):39–71.

Boser, Bernhard E., Isabelle Guyon, y Vladimir Vapnik. 1992. A training algorithm for optimal margin classifiers. En *Computational Learing Theory*, páginas 144–152.

Cortes, Corinna y Vladimir Vapnik. 1995. Support-vector networks. *Machine Learning*, 20(3):273–297.

Corpus	Tipo test	Nº clases	Idioma	Raíz	Precisión *caract1*	Precisión *caract2*
QA-R2D2	10-fold cv	15	Inglés	SI	76.89 %	75.10 %
				NO	77.43 %	75.34 %
			Español	SI	78.48 %	78.14 %
				NO	78.77 %	77.94 %
		10	Inglés	SI	77.02 %	75.22 %
				NO	77.56 %	75.47 %
			Español	SI	78.60 %	78.27 %
				NO	78.90 %	78.06 %
CCG	train/test	6	Inglés	SI	85.80 %	84.60 %
				NO	86.80 %	85.80 %

Cuadro 3: Resultados MBL

Daelemans, Walter, Jakub Zavrel, Ko van der Sloot, y Antal van den Bosch. 2003. Timbl: Tilburg memory-based learner. Technical Report ILK 03-10, Tilburg University.

Hacioglu, Kadri y Wayne Ward. 2003. Question classification with support vector machines and error correcting codes. En *HLT-NAACL*.

Hermjakob, Ulf. 2001. Parsing and question classification for question answering. En *Proceedings of the ACL 2001 Workshop on Open-Domain Question Answering*, páginas 17–22.

Li, Xin y Dan Roth. 2002. Learning question classifiers. En *COLING*.

Magnini, Bernardo, Matteo Negri, Roberto Prevete, y Hristo Tanev. 2002. Mining knowledge from repeated co-occurrences: Diogene attrec 2002. En *TREC*.

Moldovan, Dan I., Sanda M. Harabagiu, Roxana Girju, Paul Morarescu, V. Finley Lacatusu, Adrian Novischi, Adriana Badulescu, y Orest Bolohan. 2002. Lcc tools for question answering. En *TREC*.

Moldovan, Dan I., Marius Pasca, Sanda M. Harabagiu, y Mihai Surdeanu. 2003. Performance issues and error analysis in an open-domain question answering system. *ACM Trans. Inf. Syst.*, 21(2):133–154.

Platt, John C. 1999. Fast training of support vector machines using sequential minimal optimization. páginas 185–208.

Sekine, S., K. Sudo, y C. Nobata. 2002. Extended named entity hierarchy. En *Proceedings of the Language Resource and Evaluation Conference (LREC)*.

Soubbotin, Martin M. y Sergei M. Soubbotin. 2002. Use of patterns for detection of likely answer strings: A systematic approach. En *TREC*.

Suárez, Armando y Manuel Palomar. 2002. A maximum entropy-based word sense disambiguation system. En *COLING*.

Voorhees, Ellen M. 2000. Overview of the trec-9 question answering track. En *TREC*.

Voorhees, Ellen M. 2001. Overview of trec 2001. En *TREC*.

Voorhees, Ellen M. 2003. Overview of trec 2003. En *TREC*, páginas 1–13.

Witten, Ian H. y Eibe Frank. 2000. *Data mining: practical machine learning tools and techniques with Java implementations*. Morgan Kaufmann Publishers Inc.

Yang, Hui y Tat-Seng Chua. 2002. The integration of lexical knowledge and external resources for question answering. En *TREC*.