Clasificación automática de documentos. Un caso práctico
Automatic Classification of Documents. A Case Study

Carlos G. Figuerola
Universidad de Salamanca, Instituto Universitario de estudios en Ciencia y Tecnología
c/ Francisco de Vitoria 6-16, 37008 Salamanca
figue@usal.es

Resumen

La clasificación de documentos consume gran cantidad de trabajo y puede llegar a ser impracticable si la cantidad de documentos es elevada. Cuando los documentos son digitales, es posible aplicar técnicas de clasificación automática. Los sistemas de clasificación automática de tipo supervisado son capaces de identificar la clase o categoría adecuada para un documento determinado, después de una fase de aprendizaje o entrenamiento, durante la cual el sistema aprende las características que definen las diferentes categorías. Se describen algunas de las técnicas más utilizadas, como los clasificadores bayesianos, así como los diferentes ajustes que pueden ser efectuados para mejorar su efectividad. Se describe una aplicación de tales técnicas en un caso real, se analizan los detalles de la implementación y se discuten los resultados.

Palabras Clave: clasificación automática de documentos, aprendizaje de máquina, clasificación documental, *naive bayes*, análisis léxico, colección de entrenamiento

Abstract

Classification of documents consumes a great amount of work and may become impractical if the number of documents is high. When documents are in digital format, one can apply automatic techniques of classification. The so called supervised automatic classification systems are able to identify the category or class to which a document must be assigned. This is achieved by means a training process, in which the system learns the key features of every class. We describe some of most used techniques, as the Bayes based classifiers, as well as the issues that we can adjust to improve their effectivity. We also describe their practical use in a real case, we analyze their implementation and results are discussed.

Keywords: automatic classification of documents, machine learning, documentary classification, naive bayes, lexical analysis, training collection

Este trabajo ha sido financiado por el Ministerio de Ciencia e Innovación de España, ref. FFI2011-27763

1 Introducción

La organización automática de objetos en general, basada en las semejanzas entre ellos, puede ayudar a manejar grandes volúmenes de tales objetos, agrupándolos en clases, categorías o clusters de objetos parecidos entre sí. Básicamente podemos distinguir dos grandes formas de abordar la organización automática: la clasificación no supervisada (o *clustering*) y la categorización o clasificación supervisada.

En cualquiera de estas formas, los objetos necesitan ser descritos a través de una lista de características que, en función de las herramientas y las técnicas que se apliquen, pueden tener valores de diversos tipos, no necesariamente numéricos. Las descripciones de los objetos deben poder construirse de manera automática, y deben ser homogéneas entre sí, de forma que sea posible calcular de forma automática la semejanza (o la distancia) entre dichas descripciones (Mitchell, 1997).

En el caso de la categorización o clasificación supervisada se parte de una serie de clases o categorías, elaboradas de antemano, manualmente, y cada objeto debe ser procesado y asignado de forma automática a la categoría que le corresponda. Para ello, los programas de ordenador elaboran modelos o patrones de cada una de las clases o categorías contempladas. Esta fase se conoce como entrenamiento y para ser llevada a cabo es preciso contar con una colección de objetos categorizados manualmente. Durante el entrenamiento, se extraen las características que definen cada una de las categorías y, dependiendo de las técnicas utilizadas, también las que determinan lo contrario, es decir, la distancia o no pertenencia a cada categoría.

Por supuesto, en aplicaciones reales muchas categorías son ambiguas y muchas de las características son comunes a varias categorías; ésta es, precisamente, la dificultad de la categorización automática.

En función de los algoritmos y técnicas que se utilicen, la categorización puede ser de diferentes formas: sistemas de filtrado o binarios, con solamente dos categorías o posibilidades de clasificación; y sistemas con multitud de categorías posibles. Otra posibilidad es la de sistemas en los que cada objeto sólo puede asignarse o pertenecer a una única categoría, y sistemas en los que un mismo objeto puede ser asignado a varias clases al mismo tiempo.

Estas técnicas pueden ser aplicadas a documentos, y de hecho se aplican con éxito en diversos ámbitos documentales (Baharudin y otros, 2010). En el desarrollo de un proyecto de investigación titulado *Indicadores de Cultura Científica y Cultura Tecnológica* (Ministerio de Ciencia e Innovación de España, ref. FFI2011-27763) fue preciso seleccionar un volumen importante de documentos (noticias de prensa) de unas características temáticas concretas, y para ello se aplicaron técnicas de clasificación automática. Este trabajo intenta describir una experiencia práctica de uso de la clasificación automática y, en lo que sigue, se organiza como sigue: en la sección siguiente se introduce la aplicación de técnicas de clasificación supervisada a la clasificación de documentos, haciéndose después una descripción de los clasificadores bayesianos, unos de los más utilizados. A continuación se revisa el software disponible capaz de aplicar estas técnicas y, después, se describe la aplicación práctica de las mismas llevada a cabo, así como los diferentes refinamientos y ajustes, y una evaluación de su efecto. Finalmente, se extraen unas conclusiones.

2 Categorización automática de documentos

La clasificación de documentos es tarea que se realiza habitualmente de forma manual de antiguo, y su finalidad fundamental es poder localizar con facilidad los documentos de unas características determinadas. La clasificación automática es aplicable a documentos en formato digital (es decir, procesables por programas de ordenador) y permite tareas diversas, como la asignación automática de palabras clave o descriptores, o códigos temáticos

(Campos Ibáñez y Romero López, 2011). También la ubicación automática de documentos dentro de ontologías, como sucede con las páginas web en la denominada web semántica o el filtrado de documentos según perfiles de interés (Sebastiani, 2002); o el filtrado de documentos considerados como ruido, como es el caso del correo electrónico basura o *spam* (Tetryakov, 2004; Androutsopoulos, 2000).

En muchas ocasiones algunas de estas tareas se han abordado mediante técnicas simples, como la detección de determinadas palabras clave en los documentos a categorizar. Sin embargo, la determinación de las palabras clave adecuadas no siempre es fácil, y, además de los sesgos -conscientes o inconscientes- que pueden producirse en esa elección, existen obstáculos como la sinonimia y la polisemia, bien conocidos desde hace tiempo en las Ciencias de la Documentación, que hacen que estos enfoques simplistas fracasen con frecuencia (Manning y otros, 2008).

La aplicación de técnicas de categorización automática a objetos consistentes en texto puede producir buenos resultados en muchas de las tareas mencionadas. El hecho de que el objeto clasificable sea texto, implica que las características observables en ellos deben ser procesadas de forma específica. Esta especificidad determina que unos algoritmos funcionan con objetos de texto mejor que otros; o es más fácil aplicarlos que otros. Por ejemplo, en la categorización de textos, las características podrían ser las palabras que conforman dichos textos, y en ese caso se debe trabajar con una dimensionalidad muy alta, puesto que el vocabulario del lenguaje natural es muy amplio (decenas e incluso centenares de miles de palabras posibles). No todos los algoritmos funcionan igual de bien con dimensiones de este tipo.

La mera obtención automática de las características (por ejemplo, las palabras) puede no ser trivial: tratamiento de determinados signos ortográficos (por ejemplo, el guión, los dígitos numéricos), normalización de palabras con el mismo contenido semántico, palabras extraordinariamente frecuentes y con poco contenido semántico, etc. Resulta obvio que el tratamiento adecuado de estas características puede determinar la calidad de los resultados.

De otro lado, es interesante mencionar aplicaciones de la categorización automática en tareas relacionadas con el texto, aunque no sean propiamente categorización de documentos. Así, la categorización automática es la base de sistemas que detectan el idioma en que está escrito un texto (Cavnar y Trenkle, 1994; Milne y otros, 2012), como también se aplica en la detección de la función morfológica de las palabras de un texto dado (Cutting y otros, 1992; Márquez y Padró, 1997); o en la identificación de la autoría de textos y aspectos relacionados (detección de plagios, falsificaciones, etc.) (Stein y otros, 2007; Pedregosa y otros, 2007).

3 Clasificadores bayesianos

Diversos algoritmos han sido planteados para conseguir efectuar una categorización o clasificación automática de textos. Uno de los más utilizados es el conocido como *Naïve Bayes* (Langley y otros, 1992; McCallum y otros, 1998). Como su propio nombre indica, está basado en la estimación de la probabilidad de que un texto o documento pertenezca a una categoría dada y, a pesar de su simpleza, produce resultados bastante buenos. Básicamente, el algoritmo estima la probabilidad de que, conocidos los elementos que describen un documento *D*, éste pertenezca a la clase o categoría *C*.

Los elementos que describen un documento pueden ser las palabras $[a_1, a_2, a_3, ..., a_n]$; no todas necesariamente, ya que es posible eliminar las demasiado frecuentes y con poco valor de discriminación; ni tampoco en su forma original, puesto que se puede aplicar algún tipo de normalización previa.

En consecuencia, si disponemos de una colección de entrenamiento, es posible estimar la probabilidad global de la categoría $(P(C))$ simplemente en base a la cantidad de ejemplos o documentos que existen de esa categoría *C* frente a otras categorías en esa colección de

entrenamiento. Del mismo modo, podemos estimar las probabilidades parciales de que documentos que contengan cada uno de los elementos $[a_1, a_2, a_3, ..., a_n]$ pertenezcan a la categoría *C* en función de las frecuencias de esas palabras en cada una de las categorías de la colección de entrenamiento.

El modelo tiene defectos como la consideración de que las palabras son independientes entre sí; pero, en contrapartida, es fácil de implementar y rápido en ser calculado, y, como se ha demostrado experimentalmente, produce resultados bastante buenos. La calidad de los resultados depende del ámbito de aplicación, pero también de las técnicas aplicadas a la hora de determinar el conjunto de palabras que describen los documentos (análisis léxico).

De igual modo, el problema de la falsa independencia de los términos puede ser paliado en parte añadiendo a la descripción de los documentos bigramas o trigramas además de las palabras. Es decir, las secuencias de dos o tres palabras más frecuentes.

Aunque se trata de un algoritmo simple, su eficacia es comparable a algoritmos mucho más sofisticados (Bouckaert, 2004). Por este motivo se utiliza ampliamente en diversas tareas, como, por ejemplo, la detección de spam en el correo electrónico (Androutsopoulos y otros, 2000) y otras.

4 Software

Existen diversos programas que permiten efectuar clasificación automática de textos. Aparte de aplicaciones *ad-hoc* para usos específicos, varios de estos programas permiten un uso genérico, permitiendo efectuar una gran variedad de ajustes (elegir algoritmos, coeficientes diversos, modalidades de análisis léxicos, etc.).

Probablemente uno de los más usados, aunque ya no en la actualidad, no es propiamente un programa, sino una librería con la que escribir programas: *The Bow Toolkit*, que data de 1996 (McCallum, 1996). La librería se ofrece en código fuente, junto con algún programa de demostración, lo que hizo que su uso y adaptación se extendiera considerablemente, sobre todo en ambientes académicos.

The Bow Toolkit se ha quedado un poco antiguo y no se actualiza desde hace años (aunque sigue estando presente en los repositorios mas frecuentados de programas). El mismo autor ha producido otro software más reciente, llamado *Mallet* (McCallum, 2002) . *Mallet* está escrito en Java y hace tiene otras funcionalidades además de clasificar documentos. Su uso está bastante difundido en la comunidad académica centrada en este tipo de temas.

Mucho más difundido en la actualidad está *Weka* (Hall y otros, 2009) (http://www.cs.waikato.ac.nz/ml/weka/). Éste es un programa dedicado a la experimentación en minería de datos, en cualquiera de sus muchos aspectos; uno de ellos es, obviamente, la clasificación automática aplicable, además de a otro tipo de cosas, a textos. *Weka* es un programa que puede ser usado por usuarios finales (académicos, especialistas en minería de datos, ...), no sólo por programadores. Tiene una interfaz gráfica como cualquier otro programa actual y es, de lejos, uno de los programas más utilizados, siempre dentro de ámbitos académicos.

El *Natural Language Toolkit* para *Python* (Bird, 2006) (http://nltk.org) es otra de las herramientas disponibles. *Python* es un lenguaje de programación popular entre administradores de sistemas informáticos, pero también en el mundo científico. En este ámbito, debido probablemente a su carácter intuitivo pero también a su potencia y al hecho de tratarse de código abierto, cuenta con numerosas aportaciones o módulos específicos para tareas relacionadas con la investigación científica. Uno de estos módulos es el *Natural Language Toolkit* que, como su propio nombre indica, contiene numerosas herramientas para el procesamiento de lenguaje natural. Estas herramientas incluyen diversos clasificadores de textos, entre ellos uno basado en *Naïve Bayes*, así como conexiones con algunos de los programas citados anteriormente (*Mallet*, *Weka*).

Pero también contiene abundantes utilidades para el preprocesamiento y preparación de los textos a clasificar; pálabras vacías, análisis de frecuencias, normalización de términos (*stemming*), extracción de bigramas, trigramas, etc.. Éste es, de hecho, el software que hemos utilizado en nuestro trabajo.

Más reciente, otro módulo para *Python* es *SciKit-Learn* (Pedregosa y otros, 2011) (http://scikit-learn.org), íntegramente dedicado al aprendizaje de máquina y que consta de una buena colección de algoritmos de clasificación, junto con utilidades específicas para el procesamiento de documentos.

5 Un caso práctico

Una de las aplicaciones de la categorización automática de textos es la selección o filtrado de éstos. En el contexto de una investigación acerca de *Indicadores de Cultura Científica y Cultura Tecnológica* (Ministerio de Ciencia e Innovación de España, ref. FFI2011-27763) llevada a cabo por un equipo multidisciplinar, se hizo preciso seleccionar las noticias y artículos de prensa relacionados con la Ciencia y la Tecnología, a fin de efectuar sobre ellos diversos análisis tanto cuantitativos como cualitativos. El elevado número de documentos de prensa disponibles hacía inviable una selección manual, por lo que se decidió aplicar un clasificador que seleccionase de forma automática las noticias o documentos que tratasen, de una u otra forma, sobre temas relacionados con la Ciencia y la Tecnología.

El conjunto de noticias a clasificar procede de varios periódicos de ámbito nacional, en sus versiones digitales. Se trata de todas las noticias que se han podido recuperar para un período comprendido entre 2002 y 2011; sin embargo, no se han considerado todos los días de estos años, sino una muestra amplia construida *ad-hoc*. En el diseño de esta muestra se ha procurado cubrir los diferentes días de la semana a lo largo de todos los meses del año, de manera que para cada diario se han considerado unos 840 días. La obtención de las noticias para cada uno de esos 840 días ha sido exhaustiva, de manera que la colección global supera las 250.000 noticias

5.1 Evaluación de la clasificación

Para medir la efectividad de la clasificación efectuada es habitual recurrir a las medidas clásicas utilizadas en Recuperación de Información: precisión, exhaustividad y algunos derivados de éstas, como la medida F (Manning y otros, 2008; Martínez Comeche, 2011; Sánchez, 2007). El uso de estas medidas requiere la adaptación del concepto de relevancia, pero, en cualquier caso, la precisión (para una clase o categoría concreta) puede obtenerse calculando la proporción de documentos clasificados correctamente respecto de el total de los que el clasificador automático ha asignado a dicha categoría. De la misma manera, la exhaustividad equivale al cociente entre el número de documentos asignados correctamente a esa categoría y la cantidad de documentos realmente pertenecientes a la categoría en cuestión.

La medida F_β trata de combinar ambos aspectos, precisón y exhaustividad. El coeficiente β permite asignar más peso a precisión o exhaustividad; se le asigna valor 1 cuando se desea considerar ambos componentes por igual.

Cuando se trabaja en multiclase, es posible calcular F para cada clase o categoría individualmente y luego obtener la media, y entonces hablamos de *macro-average* de F; o bien se puede hacer un cómputo global de falsos positivos, aciertos y cantidades reales de documentos en cada categoría y calcular precisión, exhaustividad y F de forma conjunta, y entonces se habla de *micro-average* (Lewis y otros, 1996).

5.2 El tamaño de la colección de entrenamiento

Como se ha indicado, es preciso entrenar el clasificador con un cierto número de documentos previamente etiquetados manualmente. En nuestro caso solamente se contemplan

dos categorías posibles: Ciencia y Tecnología, por un lado, y No Ciencia y Tecnología por otro.

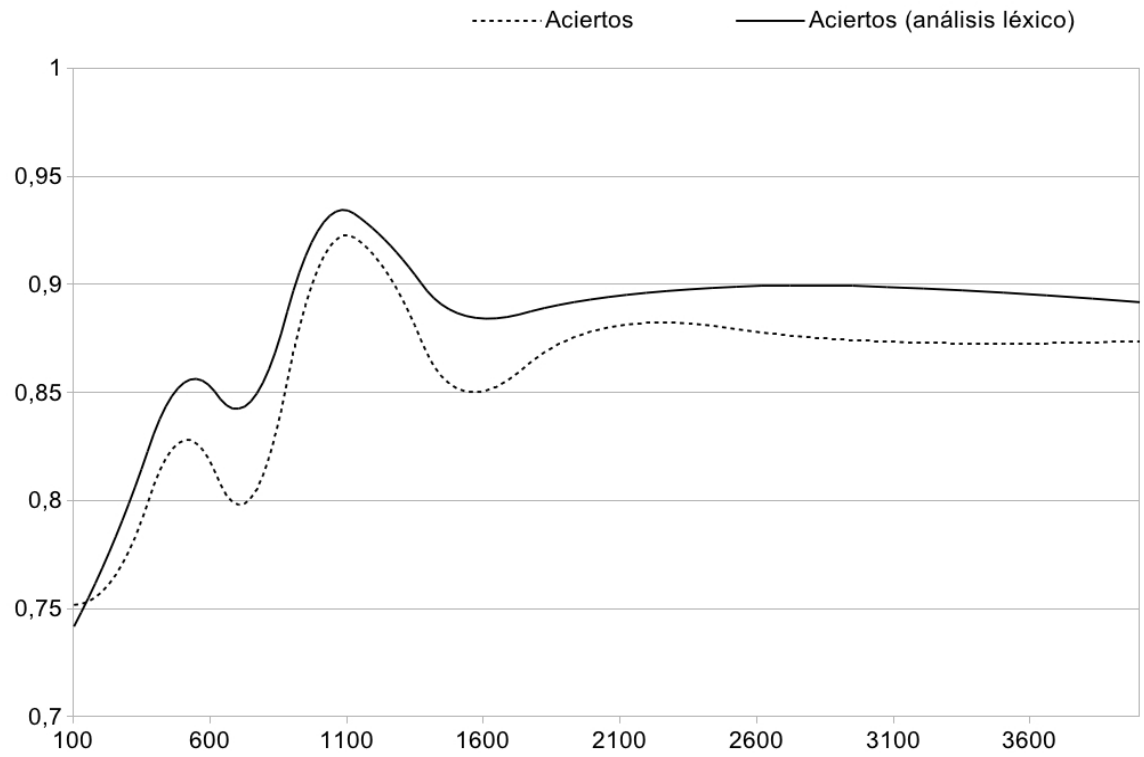

Gráfico 1: Tamaño de la colección de entrenamiento y aciertos del clasificador

El tamaño de las colecciones de entrenamiento es importante, puesto que un número insuficiente de muestras conduce a una menor precisión del clasificador; pero un número excesivamente alto produce el conocido fenómeno del *sobreentrenamiento*, perjudicial para el rendimiento del clasificador. Este aspecto fue descrito hace ya tiempo por Lewis (Lewis y otros, 1996), y ha sido objeto después de diferentes trabajos (Raina y otros, 2003; Baharudin y otros, 2010). De otro lado, obtener un número elevado de documentos clasificados manualmente es costoso en términos de tiempo de trabajo y, dependiendo de los dominios de conocimiento, también en lo que se refiere a la cualificación de las personas que han de etiquetar o comprobar los documentos (Heirmerl y otros, 2012).

El gráfico 1 muestra el porcentaje de aciertos en la selección de noticias sobre Ciencia y Tecnología, según diferentes tamaños de la colección de entrenamiento. Claramente se ve cómo según aumenta el número de documentos de entrenamiento los resultados (medidos como aciertos o verdaderos positivos) mejoran, hasta llegar a un máximo; sobrepasado el cual los resultados decrecen hasta estabilizarse. En este caso parece que el tamaño óptimo de la colección está en torno a los 1000 - 1100 documentos.

Sin embargo, el rendimiento del clasificador puede evaluarse desde distintos puntos de vista. Teniendo en cuenta que el objetivo es seleccionar noticias de un determinado contenido, podemos estar interesados en seleccionar el mayor número posible de ellas (exhaustividad); o bien en obtener una selección con poco ruido (precisión). El gráfico 2 muestra resultados para diferentes tamaños de colección de entrenamiento, desde ambos puntos de vista: precisión y exhaustividad. Nuevamente, se muestran también resultados con mejoras de análisis léxico.

Lo que ahora nos interesa de ese gráfico es que la precisión óptima se alcanza con una

cantidad relativamente pequeña de documentos de entrenamiento (entre 250 y 500 en nuestro caso), mientras que la mejor exhaustividad se logra con colecciones de entrenamiento más grandes (entre 1100 y 1200 documentos). Parece también que la exhaustividad es más sensible al tamaño de la colección de entrenamiento.

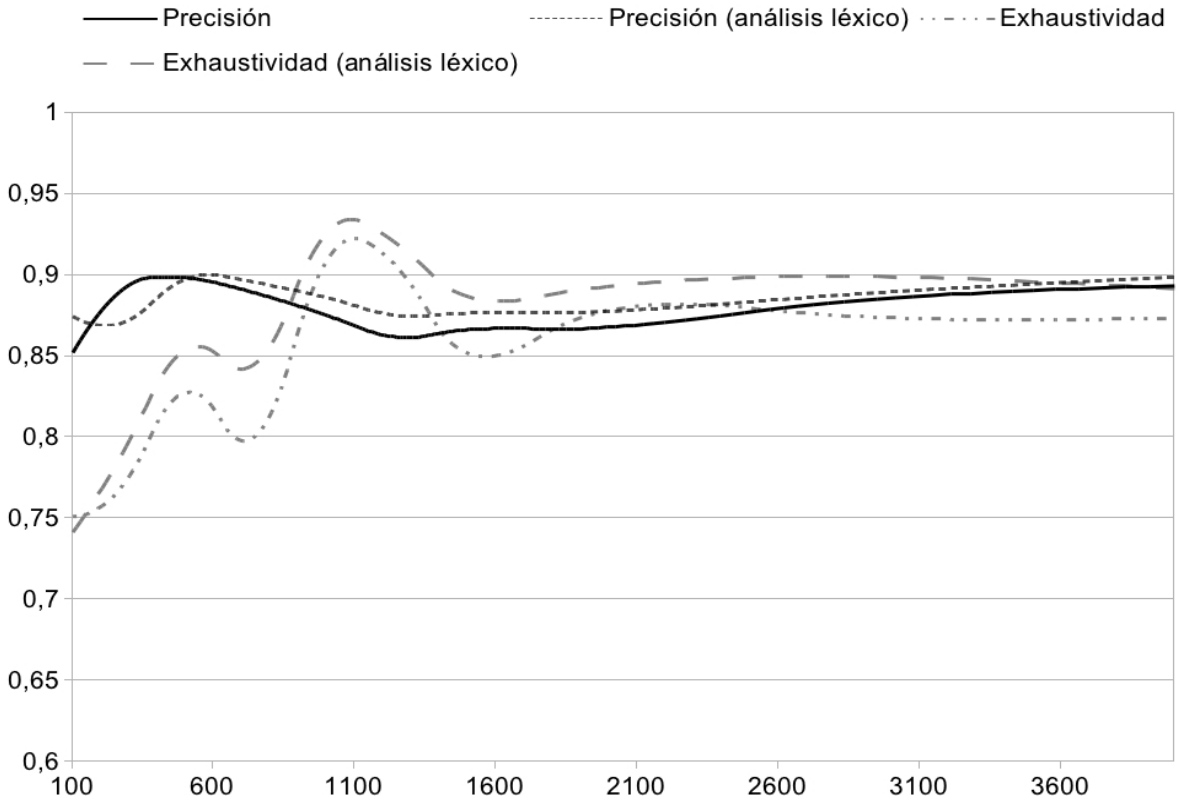

Gráfico 2: Tamaño de la colección de entrenamiento y Precisión - Exhaustividad

5.3 Análisis Léxico

Como es ya bien conocido por quienes se dedican a la clasificación automática de textos, el análisis léxico es crucial, puesto que de él se van a extraer las características que definirán cada documento y, también, cada clase o categoría. Para el tipo de documentos que estamos procesando, se han aplicado opciones simples para la extracción de palabras: eliminación de acentos, conversión a minúsculas y eliminación de cualquier símbolo distinto de letra minúscula. En consecuencia, sólo se considerará palabra una secuencia ininterrumpida de letras.

De las palabras extraídas se han eliminado las contenidas en una lista estándar de palabras vacías para el español, y las restantes se han sometido a un proceso de normalización o *stemming*. Para todo esto se han revelado de gran utilidad las herramientas del *Natural Language Toolkit*; para el caso del *stemming*, que no es precisamente trivial, el *NLTK* aplica el conocido *stemmer* de *SnowBall* (Porter, 2001) para el español. Los *stemmers* producidos por *SnowBall* son en la actualidad ampliamente utilizados; de hecho, son los que aplican la mayor parte de los motores de búsqueda, al menos los de código abierto (Cleger Tamayo y otros, 2011).

De las palabras resultantes de todo este proceso se han eliminado aquéllas de menos de cuatro caracteres.

Adicionalmente, se consideró la posibilidad de añadir a la lista de palabras que definen o

caracterizan cada documento los bigramas y trigramas derivados de cada texto (Tan y otros, 2002; Montejo Ráez y otros, 2010). Es decir, las secuencias de dos o tres palabras que aparecen juntas, después de eliminar palabras vacías y normalizar. Sin embargo, el número de bi y trigramas es excesivamente elevado, resultando buena parte de ellos irrelevantes, dado que su frecuencia es muy baja.

Por ello, es habitual aplicar algún medio de podar o reducir el número de bi y trigramas. Uno de los más utilizados es fijar un umbral de valores de tipo estadístico, como Chi2 que estime la relación entre un bi o trigrama y la categoría de la cual es ejemplo (Almeida y otros, 2011).

De manera global, parece que estas técnicas de análisis léxico mejoran los resultados del clasificador. Esto se aprecia claramente en los Gráficos 1 y 2; salvo para colecciones de entrenamiento muy pequeñas, particularmente desde el punto de vista de la precisión, los resultados son siempre mejores aplicando las técnicas de análisis léxico mejorado.

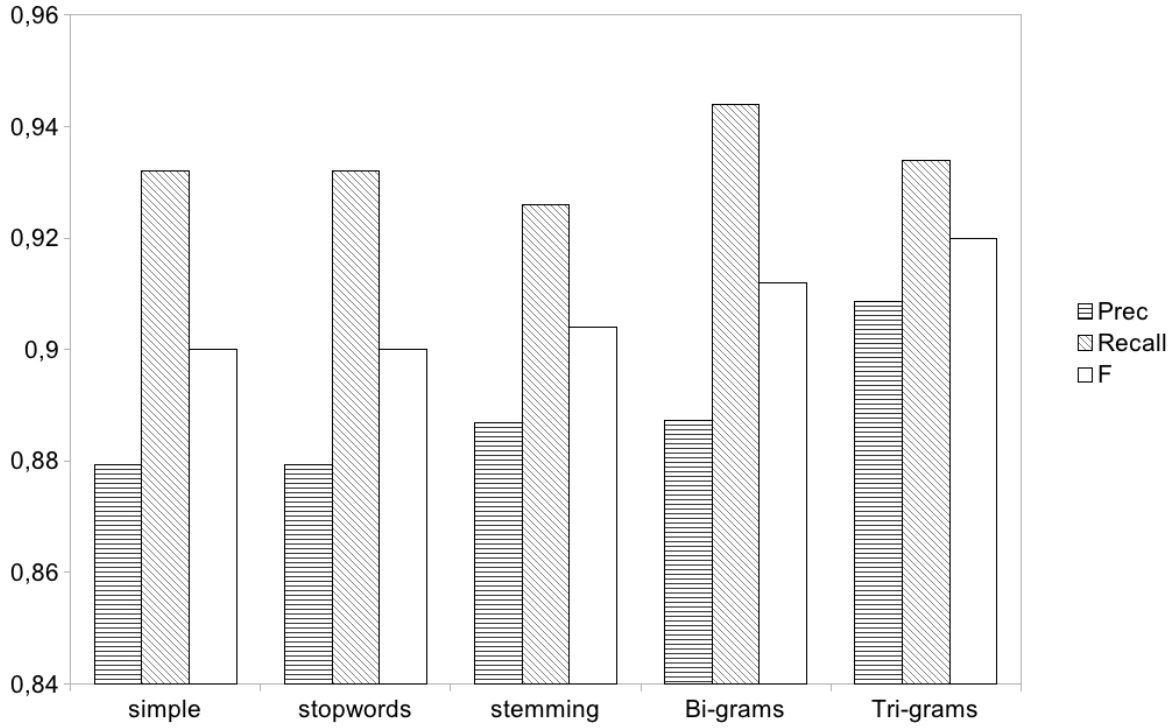

Gráfico 3: Efectos del análisis léxico

Sin embargo, puede ser interesante examinar de cerca las operaciones de análisis léxico efectuadas y la contribución de cada una a la mejora de resultados. El Gráfico 3 muestra los resultados acumulativos de aplicar diferentes técnicas de análisis léxico, siempre utilizando una colección de entrenamiento de 1000 documentos. El gráfico muestra los resultados en Precisión, Exhaustividad y F_1.

El Gráfico 3 muestra claramente que la eliminación de palabras vacías no tiene ningún efecto en la calidad de los resultados del clasificador. El *stemming* sin embargo, sí que los tiene, aunque dispares; la precisión mejora aplicando técnicas de normalización de términos, pero no la exhaustividad, que disminuye; aunque no lo suficiente como para que, desde un punto de vista más general, F_1 se vea perjudicado. Dicho en otras palabras, parece que el *stemming* tiende a producir menos falsos positivos o menos ruido, aunque a costa de no identificar algunos documentos interesantes.

La técnica que sí parece eficaz desde cualquier punto de vista es la inclusión de bigramas,

junto con los términos, como características de los documentos. Lo es con claridad desde la perspectiva de la exhaustividad, pero también, aunque en menor medida, en lo que se refiere a la precisión.

Si añadimos trigramas, los resultados aún mejoran en lo que se refiere a la precisión; no sucede lo mismo con la exhaustividad, que arroja resultados un poco por debajo que los obtenidos con bigramas, pero aún sí superiores a los que producen las otras opciones de análisis léxico.

6 Conclusiones

Hemos utilizado un clasificador automático de documentos del tipo conocido como Naive Bayes para seleccionar noticias de prensa de un determinado contenido. Los resultados obtenidos son buenos, incluso si se trabaja con una colección de entrenamiento pequeña: un 75 % de aciertos con una colección de entrenamiento de 100 documentos, y el 85 % con 500 documentos. Estos resultados mejoran notablemente entrenando el clasificador con un número mayor de documentos, hasta cierto punto, a partir del cual el sobreentrenamiento produce un efecto contrario.

Sin embargo, el efecto del número de documentos de entrenamiento es dispar desde el punto de vista de la precisión y de la exhaustividad. La precisión parece necesitar menos documentos de entrenaniento, y se mantiene más estable aunque la cantidad de esos documentos de entrenamiento aumente. La exhaustividad, sin embargo, parece necesitar más documentos para el entrenamiento, y se muestra mucho más sensible a ese factor.

De otro lado , aplicar opciones mejoradas de análisis léxico incide directamente en la obtención de resultados más satisfactorios. De tales opciones, la eliminación de palabras vacías por sí sola se muestra inútil, al menos en lo que se refiere a efectividad, pero no ocurre lo mismo con el stemming y la adición de bi y trigramas. Con el stemming mejora notablemente la precisión, aunque no así la exhaustividad; y a inclusión entre las características de los documentos de bi y trigramas es claramente beneficiosa, aunque en el caso de los trigramas la exhaustividad se resiente un poco.

En cualquier caso, los resultados pueden considerarse como muy buenos, por encima del 0.9. Así pues, el clasificador Naive Bayes produce buenos resultados, aún con pocos documentos para el entrenamiento del clasificador. Estos resultados mejoran notablemente si enriquecemos el análisis léxico de los documentos.

Bibliografía:

Almeida, T. A.; Almeida, J. ; Yamakami, A (2011). Spam filtering: how the dimensionality reduction affects the accuracy of naive bayes classifiers. *Journal of Internet Services and Applications*, vol 1(3), 183–200.

Androutsopoulos, I. ; Koutsias, J. ; Chandrinos, K. V. ; Paliouras, G. ; Spyropoulos, C. D. (2000). An evaluation of naive bayesian anti-spam filtering. arXiv preprint cs/0006013, en línea: http://arxiv.org/abs/cs/0006013 [consultado el 12/12/2012]

Baharudin, B; Lee, L. H.; Khan, K. (2010). A review of machine learning algorithms for text-documents classification. *Journal of Advances in Information Technology*, vol. 1(1), 4–20.

Bird, S. (2006). Nltk: the natural language toolkit. Proceedings *COLING/ACL on Interactive presentation sessions*, p. 69–72. Sidney, Australia: Association for Computational Linguistics.

Bouckaert, R. (2004). Naive Bayes classifiers that perform well with continuous variables, en AI 2004 : advances in artificial intelligence, p. 1089-1094, Cairns, Australia

Campos Ibáñez, L. M. de; Romero López, A. E. (2011). Clasificación documental. En: Cacheda Seijo, F.; Fernández Luna, J. M. & Huete Guadix, J. F. (Editores) *Recuperación de Información. Un enfoque práctico y multidisciplinar*. Ra-Ma, Madrid, pp. 359-392.

Cavnar, W. B.; Trenkle, J. M. (1994). N-Gram-Based Text Categorization, *Third Annual Symposium on Document Analysis and Information Retrieval*, p. 161-175, Las Vegas, Nevada.

Cleger Tamayo, S. ; Figuerola, C. G. ; Rodríguez Cano, J. C. (2011). Motores de búsqueda de Código Abierto. En: Cacheda Seijo, F.; Fernández Luna, J. M. & Huete Guadix, J. F. (Editores) *Recuperación de Información. Un enfoque práctico y multidisciplinar*. Ra-Ma, Madrid, pp. 233-260

Cutting, D. R.; Kupiec, J. ; Pedersen, O. (1992). A Practical Part-of-Speech Tagger, *ANLP 1992, 3rd Applied Natural Language Processing Conference*, p. 133-140, Trento, Italy

Hall, M.; Frank, E.; Holmes, G.; Pfahringer, B; Reutemann, P.; Witten, I. H. (2009). The WEKA data mining software: an update. *SIGKDD Explorations*, vol. 11(1), 10–18.

Heimerl, F.; Koch, S.; Bosch, H.; T. Ertl, T. (2012). Visual classifier training for text document retrieval. *Visualization and Computer Graphics, IEEE Transactions on*, vol. 18(12), 2839–2848.

Langley, P.; Iba, W.; Thompson, K. (1992). An analysis of bayesian classifiers. Proceedings *National Conference on Artificial Intelligence*, p. 223–228. San Antonio, CA: AAAI Press and MIT Press.

Lewis, D. D. ; Schapire, R. E. ; Callan, J. P. ; Papka, R. (1996). Training algorithms for linear text classifiers. En: Frei, H. P. ; Harman, D. K. ; Schüble, P. ; Wilkinson, R. (editores), *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '96*, p. 298-306. Zurich, Switzerland: ACM SIGIR.

Manning, C. D. ; Raghavan, P. ; Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge; Cambridge University Press.

Márquez, L.; Padró, L. (1997). A Flexible POS Tagger Using an Automatically Acquired Language Model, *35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics*, p. 238-245, Madrid, España: Association for Computational Linguistics.

Martínez Comeche, J. A. (2011). Evaluación de la eficacia de la recuperación. En: Cacheda Seijo, F.; Fernández Luna, J. M. & Huete Guadix, J. F. (Editores) *Recuperación de Información. Un enfoque práctico y multidisciplinar*. Ra-Ma, Madrid, pp. 125-156.

McCallum, A. K. (1996). *Bow: a toolkit for statistical language modeling, text retrieval, classification and clustering*. En línea: http://www.cs.cmu.edu/ mccallum/bow/ [consultado el 12/12/2012]

McCallum, A. ; Nigam, K. (1998) A comparison of event models for naive bayes text

classification. En: *AAAI-98 workshop on learning for text categorization*, p. 41-48. En línea: http://www.kamalnigam.com/papers/multinomial-aaaiws98.pdf [consultado el 12/12/2012]

McCallum, A. K. (2002). *Mallet: A machine learning for language toolkit*. En línea: http://mallet.cs.umass.edu [consultado el 12/12/2012]

Milne, R.; O'Keefe, R.; Trotman, A. (2012). A study in language identification, *Proceedings of the Seventeenth Australasian Document Computing Symposium*, p. 88-95., Dunedin, New Zealand: ACM-SIGIR.

Mitchell, T. (1997). *Machine Learning*, Nueva York; McGraw-Hill .

Montejo Ráez, A. ; Perea Ortega, J. M. ; Martín Valdivia, M. T. ; Ureña López, L. A. (2010). Uso de la detección de bigramas para la categorización de texto en un dominio científico, *Procesamiento de Lenguaje Natural*, vol. 44, 91-98.

Pedregosa, F. ; Varoquaux, G. ; Gramfort, A. ; Michel, V. ; Thirion, B. ; Grisel, O. ; Blondel, M. ; Stein, P. B. ; Eissen, S. M. zu; M. Potthast, M. (2007). Strategies for retrieving plagiarized documents. En *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, p. 825–826. Amsterdam, Holanda: ACM SIGIR.

Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, vol. 12, 2825–2830.

Porter, M. (2001). Snowball: A language for stemming algorithms, en línea: http://snowball.tartarus.org/texts/introduction.html [consultado el 12/12/2012]

Raina, R.; Shen, Y.; Ng, A. ; McCallum, A. (2003). Classification with hybrid generative/discriminative models. *Advances in neural information processing systems*, vol. 16, 545-552

Sánchez Jiménez, R. (2007). La documentación en el proceso de evaluación de sistemas de clasificación automática, *Documentación de Ciencias de la Información*, vol. 30, 25-44.

Sebastiani, F. (2002). Machine Learning in Automated Text Categorization, *ACM Computing Surveys*, vol. 34: 1-47.

Stein, B.; zu Eissen, S. ; Potthast, M. (2007). Strategies for retrieving plagiarized documents, *Proceedings of the 30th annual international ACM SIGIR Conference on Research and development in information retrieval*, p. 825-826 , Amsterdam Holanda: ACM – SIGIR.

Tan, C.; Wang, Y. F.; Lee, C.D. (2002). The use of bigrams to enhance text categorization. *Information Processing & Management*, vol. 38(4), 529–546.

Tretyakov, K. (2004). Machine learning techniques in spam filtering, *Data Mining Problem-oriented Seminar, MTAT*, vol. 3, 60-79 . Versión en línea: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.1.3483&rep=rep1&type=pdf

[consultado el 12/12/2012].