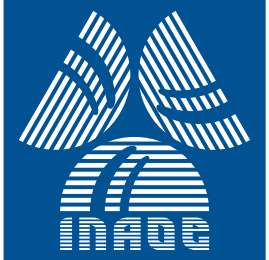

Clasificación Automática de Textos usando Reducción de Clases basada en Prototipos

por

Juan de Dios Álvarez Romero

Tesis sometida como requisito parcial para obtener el grado de

Maestro en Ciencias en el Área de Ciencias Computacionales

en el

Instituto Nacional de Astrofísica, Óptica y Electrónica
INAOE

Supervisada por:

Dr. Manuel Montes y Gómez, INAOE
Dr. Luis Villaseñor Pineda, INAOE

Tonantzintla, Puebla
Enero 2009

Dedico este trabajo

a mis padres con mucho cariño

Agradecimientos

A mis asesores Dr. Manuel Montes y Gómez y Dr. Luis Villaseñor Pineda un especial agradecimiento por su apoyo, ayuda y orientación no sólo a lo largo de la elaboración de la tesis sino durante toda la maestría.

A mis sinodales y revisores del presente documento, Dr. Aurelio López López, Dr. Ariel Carrasco y Dr. Francisco Martínez por sus comentarios y observaciones

Al INAOE y en particular a la Coordinación de Ciencias Computacionales, por todas las facilidades proporcionadas durante mi estancia académica.

A CONACYT por el apoyo económico a través de la beca recibida durante el posgrado.

A mis padres, Milagros Romero y Arturo Alvarez, y a mi hermana Milagros por todo su cariño y ánimos de seguir siempre adelante.

A mi madrina Margarita Alvarez que siempre me ha apoyado.

Resumen

La facilidad con que se producen hoy en día documentos electrónicos, tiene por consecuencia la enorme cantidad de datos existentes en Internet, bibliotecas digitales, correos electrónicos, entre otros. Toda esta información es difícil de manejar si no existen mecanismos de acceso, organización y extracción de la misma. En este sentido, la *Clasificación Automática de Textos* juega un papel muy importante al ordenar los documentos que se van generando, su objetivo es asignar una clase a un documento nuevo, de una lista de clases previamente definidas. Esta tarea se vuelve más complicada a medida que aumenta el número de clases, es por eso, que muchos de los clasificadores tratan los problemas multi-clase como varios problemas binarios. En el presente trabajo se estudia el desempeño que pueden alcanzar los clasificadores más usados en Clasificación de Textos (*i.e.* Naive Bayes y SVM), si se reduce el problema inicial multi-clase a un problema donde el clasificador sólo tenga que distinguir entre dos clases, es decir un problema binario. Para la reducción se propone un esquema de *prototipo* para representar a las clases, que a diferencia de otros esquemas, asigna un peso a cada atributo de acuerdo a la importancia que este tiene para cada clase. Además, se propone una medida de similitud que se base en la intersección pesada de atributos. Los experimentos realizados con este método, arrojan resultados que muestran una mejor exactitud o en el peor de los casos, de igual desempeño frente al método tradicional.

Abstract

Currently, there exist a lot of available information on the Web, digital libraries, e-mails and databases. In order to take advantage of all of it, they are necessary more efficient mechanism for information retrieval and organization. In particular, Text Categorization plays a very important roll on the arrangement of documents, since its goal is to assign a class, taken from a list previously defined categories, to each new given document. Evidently, it is expected that as the number of classes increased, the categorization task will be more complicated. As a consequence of this fact, most current classification methods tackle multi-class problems by using a combination of several binary classifiers. In this work, we study the performance that can be achieved by traditional categorization methods (i.e. Naive Bayes y SVM) when a multi-class task is reduced to a single binary problem for each document that needs to be classified. Mainly, this document proposed a new prototype scheme to represent each class in order to do the class reduction, and a new weighting scheme to evaluate the importance of terms to each class. Furthermore, we propose a similarity measure based on the intersection of the weighted terms. The experimental results show that the proposed method outperformed traditional approaches in most of the cases, and that in the rest of them, it obtained the same accuracy.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

Durante la última década, la producción de documentos en formato digital ha ido en aumento debido, en gran parte, a la accesibilidad, cada vez más baratos de *hardware* y *software* para generar datos digitales, (computadoras personales y procesadores de textos), para digitalizar datos generados por medios no digitales (escáners y *software* de reconocimiento óptico de caracteres *OCR*) teniendo por consecuencia que en la actualidad exista un volumen muy grande de textos disponibles en linea a través de internet: periódicos electrónicos, bibliotecas digitales, correos electrónicos, bases de datos en corporaciones, registros médicos de pacientes y muchos más.

Para aprovechar eficientemente la información contenida en estos documentos, se han hecho estudios en distintas líneas de investigación como en Minería de Texto [*Berry*, 2003], Aprendizaje Automático [*Sebastiani*, 2002], Recuperación de Información [*Henzinger*, 2004], Búsqueda de Respuestas [*Montes et˜al.*, 2008] entre muchas otras. Esto con el fin de invertir el menor tiempo posible, en buscar la información deseada de manera que el resultado de la búsqueda traiga consigo los documentos (o fragmentos de ellos) con la información más relevante a lo

que buscamos, evitando aquellos documentos que no contienen o contienen poca información relevante. Sin duda una buena organización de los documentos es de gran ayuda a la hora de buscar información, de esta tarea se encarga la Clasificación Automática de Textos [*Sebastiani*, 2006]. Esta área aporta los métodos necesarios para ordenar los documentos por clases, donde las clases pueden ser idiomas, autores, temas, opiniones o estilos de escritura por mencionar algunas.

1.1. Clasificación de Textos

La Clasificación de Textos es la tarea de asignar una clase a un documento nuevo (documento nunca antes visto), la clase es tomada de una lista de clases previamente definidas. Un documento puede pertenecer a una sola clase, varias clases o bien no pertenecer a ninguna clase [*Joachims*, 1998]. Usando Aprendizaje Automático el objetivo es entrenar clasificadores con ejemplos de documentos (documentos previamente clasificados) para que puedan realizar la tarea de asignar la clase automáticamente a documentos nuevos.

Investigadores de las áreas de Recuperación de Información y de Aprendizaje Automático han fijado su atención en la Clasificación de Textos y es debido a eso, que muchas de las herramientas usadas en esta tarea provienen de esas dos áreas. Los documentos usados en clasificación son *indexados* usando las mismas técnicas que en Recuperación de Información, más aún, los documentos son comparados y la similitud entre ellos es medida usando técnicas originalmente desarrolladas en Recuperación de Información. Para investigadores del área de Aprendizaje Automático, la Clasificación de Textos es una tarea con la pueden comparar sus técnicas y metodologías ya que las aplicaciones de Clasificación de Texto usan espacios de atributos de alta dimensionalidad y la cantidad de datos que se maneja

es muy grande.

Para desarrolladores en la industria, la Clasificación de Textos es importante por la cantidad enorme de documentos que se necesita procesar y clasificar, de manera correcta y eficiente. Pero aún más importante, es el hecho de que las técnicas de Clasificación de Textos han alcanzado niveles de exactitud comparables al desempeño de profesionales entrenados [*Sebastiani*, 2002], con la gran diferencia que el tiempo invertido por el sistema de clasificación automático es mucho menor que el de una persona experta.

Las técnicas de Clasificación de Textos son usadas en una variedad de tareas, pero en general se puede hablar de dos tipos: la clasificación temática, para distinguir entre grupos de noticias o la clasificación de patentes de cuerdo a cierta taxonomía o la detección de patentes existentes y la clasificación no temática para la atribución de autoría, la clasificación de textos de opinión, etc. [*Sebastiani*, 2005a]. En cada caso, la meta de la clasificación es asignar automáticamente la clase apropiada a cada documento.

En general, para que un clasificador pueda clasificar correctamente nuevos documentos, es necesario entrenarlo con algunos documentos pre-clasificados de cada clase, de esa manera el clasificador podrá, a partir del modelo aprendido con los documentos pre-clasificados, generalizarlo para documentos nuevos. Se han estudiado varios tipos de clasificadores en el área de Aprendizaje Automático y algunos de ellos pueden ser usados para la Clasificación de Textos, entre estos podemos mencionar a: los probabilísticos, los basados en *Kernels* y los basados en prototipos, estos últimos provienen de trabajos hechos en Recuperación de Información. Un punto importante que se debe cubrir es que el clasificador debe ser eficiente, independiente del dominio de los documentos a ser clasificados.

La evaluación del desempeño puede ser medida en la eficiencia de entrenamien-

to (qué tanto tiempo toma aprender de los datos de entrenamiento), la eficiencia de clasificación (qué tiempo toma clasificar un documento) y la efectividad (promedio de documentos correctamente clasificados). Para hacer esta evaluación, un conjunto de documentos pre-clasificados se divide en conjunto de entrenamiento y de prueba, después estos últimos son usados para entrenar y evaluar respectivamente al clasificador.

Mucho del trabajo orientado a mejorar los clasificadores se enfoca en resolver alguno de los siguientes aspectos: tener menos errores al clasificar los documentos, ser más rápidos en el proceso de aprendizaje, o bien, requerir un conjunto de datos etiquetados más pequeño para entrenarse. En particular, el presente trabajo busca reducir los errores de clasificación.

1.2. Descripción del Problema

En el proceso de Clasificación de Textos, se eligen atributos (generalmente palabras), que representen de la mejor manera a cada clase, de modo tal que se distinga del resto de las clases. Aún así, puede suceder que documentos de una clase sean similares a documentos de otra clase a la que no pertenecen, debido a que comparten vocabulario. Un ejemplo que ilustra esto es la clasificación de noticias de desastres naturales. Si se tiene desastres naturales como *Huracanes* e *Inundaciones*, palabras como lluvia, viento, agua, inundación, emergencia y DN-3 por mencionar algunas, están en los vocabularios de ambos desastres por lo que documentos que contengan estas palabras comunes provocan que la elección entre Huracán e Inundación sea de mayor complejidad. Así pues, podemos ver que elegir una clase de entre los dos tipos de noticias puede llegar a ser confuso y más aún si tomamos en cuenta que la clasificación no sólo se hace entre estas clases

"conflictivas", sino con otras más, esto complica aún más la elección de la clase correcta.

Una posible solución sería aislar el problema de las clases "conflictivas" del resto de las clases, enfocando los esfuerzos del clasificador a solo distinguir entre las clases con mayor similitud.

En el presente trabajo, se propone un método en dos etapas que mejora la clasificación en exactitud. El método busca llevar la clasificación original a una clasificación reducida en clases, donde sólo se tenga que distinguir entre las clases que son más similares al documento, clases "conflictivas", aprovechando de mejor manera los atributos discriminativos de las clases. En una primera instancia, dos de las clases más similares a cada nuevo documento son seleccionadas, y como segunda etapa, se usan técnicas tradicionales de clasificación para elegir la clase de entre estas dos opciones.

1.3. Objetivo de la Tesis

Objetivo General

Proponer un método de clasificación automática de textos aplicando una reducción de clases basada en prototipos.

Objetivos Específicos

- Definir un conjunto de criterios para el cálculo del prototipo representativo de cada clase.

- Proponer un método de reducción de clases basada en prototipos.

- Evaluar la reducción de clases en la clasificación de textos tanto temática como no-temática.

1.4. Estructura de la Tesis

El presente documento está organizado de la siguiente manera:

En el Capítulo 2 se dan las definiciones de algunos conceptos básicos que se utilizan a lo largo del texto, citando paralelamente el trabajo previo relacionado con la Clasificación de Textos, la representación de los documentos, métodos de clasificación, medidas de evaluación y el corpora usado.

El Capítulo 3 describe el método propuesto, la manera de llevarlo a cabo, las herramientas y los datos que se requirieron.

En el Capítulo 4 se presentan los resultados obtenidos con el método de dos etapas propuesto y se compara con el método tradicional existente.

Finalmente, el Capítulo 5 resume las conclusiones del trabajo de tesis y presenta ideas para trabajo futuro.

Capítulo 2

Antecedentes

En este capítulo se presentan las definiciones básicas utilizadas durante esta tesis y se hace referencia a trabajos previos de áreas relacionadas con la Clasificación de Textos. Además, se describen técnicas de indexado para los términos de los documentos y algunos de los métodos más ampliamente usados para la clasificación, así como la manera en que son usualmente evaluados estos métodos.

2.1. Aprendizaje Automático

Se dice que ocurre Aprendizaje Automático en un programa que puede modificar aspectos de sí mismo, de tal modo que en una ejecución subsecuente con la misma entrada, se produce un resultado mejor. Dicho de otro modo, un programa se considera que ha "aprendido" a realizar una tarea T, si después de obtener una experiencia E el programa se desempeña mejor de acuerdo a una medida o métrica de calidad P. [*Mitchell*, 1997; *Witten and Frank*, 2005]. Uno de los tipos de Aprendizaje Automático es el *Aprendizaje Supervisado* donde al algoritmo de aprendizaje se le proporciona un conjunto de entrada con las correspondientes

salidas correctas, el algoritmo "aprende" comparando su salida con la correcta, con esto sabe el error, luego entonces, se modifica para corregir. Un ejemplo de este tipo de aprendizaje es la Clasificación de Textos.

2.2. Clasificación de Textos

La meta principal de la Clasificación de Textos, es crear modelos que, dado un conjunto de documentos de entrenamiento, con clases conocidas y un nuevo documento, puedan predecir la clase de dicho documento. La Clasificación de Textos Puede ser formalizada como la tarea de aproximar una *función objetivo* desconocida $\Phi : \mathcal{D} \times \mathcal{C} \rightarrow \{T, F\}$ (que describe cómo deben ser clasificados los documentos, de acuerdo a un experto) por medio de una función $\hat{\Phi} : \mathcal{D} \times \mathcal{C} \rightarrow \{T, F\}$ llamada el *clasificador*, donde $\mathcal{C} = \{c_1, \ldots, c_{|\mathcal{C}|}\}$ es un conjunto predefinido de clases y $\mathcal{D}$ es un conjunto (posiblemente infinito) de documentos. Si $\Phi(d_j, c_i) = T$, entonces d_j se llama *ejemplo positivo* de c_i, por el contrario si $\Phi(d_j, c_i) = F$ se llama *ejemplo negativo* de c_i. Dependiendo de la aplicación, la Clasificación de Textos puede ser una tarea de una sola etiqueta (*i.e.* exactamente un $c_i \in \mathcal{C}$ debe ser asignada a cada $d_j \in \mathcal{D}$), o una tarea de multi-etiqueta (*i.e.* cualquier número $0 \leq n_j \leq |\mathcal{C}|$ de clases pueden ser asignadas al documento $d_j \in \mathcal{D}$).

Un clasificador de textos para c_i se genera automáticamente mediante un proceso inductivo (el *aprendiz*) el cual, observando las características de un conjunto de documentos pre-clasificados en c_i o $\bar{c_i}$, obtiene las características que debe tener un documento nuevo para pertenecer a c_i. Para construir clasificadores para $\mathcal{C}$, se necesita un conjunto Ω de documentos tales que el valor de $\Phi(d_j, c_i)$ sea conocido para cada $(d_j, c_i) \in \Omega \times \mathcal{C}$ [*Sebastiani*, 2005b].

La construcción de un clasificador inicia con la clasificación manual de un

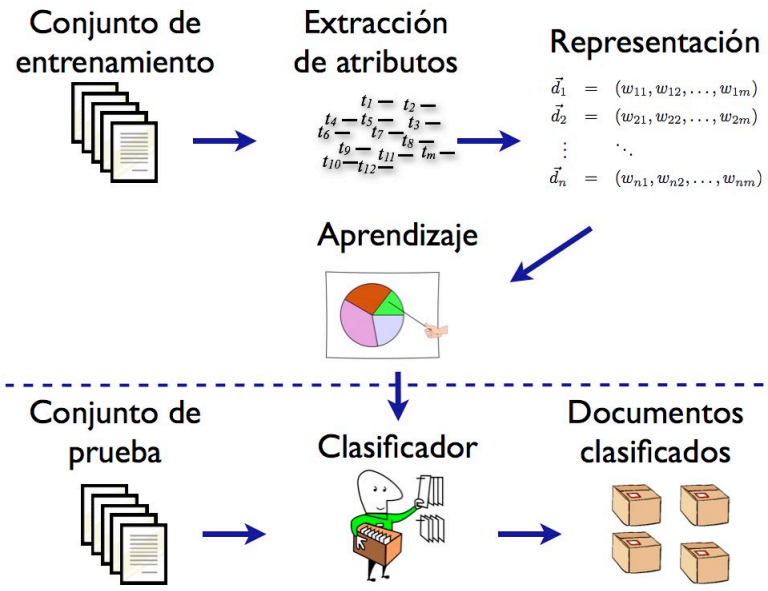

Figura 2.1: Esquema de construcción y prueba de un clasificador.

conjunto de documentos llamado *conjunto de entrenamiento*. Se seleccionan los atributos, en seguida, se elige una representación de los documentos para después aplicar alguno de los distintos algoritmos de clasificación. Con esto, el clasificador queda entrenado, el resto es probar su desempeño con un *conjunto de prueba*; un conjunto de documentos nuevos que nunca se hayan visto antes. Un esquema del proceso de clasificación se puede ver en la Figura 2.1.

2.2.1. Representación de los Documentos

El contenido de un documento como tal es el conjunto de palabras. Para evitar construir un sistema sofisticado que trate directamente con las palabras, se lleva este contenido a una representación compacta que el clasificador pueda interpretar fácilmente. Las técnicas de indexación de documentos en Clasificación de Textos

son tomadas de las usadas en Recuperación de Información.

Entre los modelos de representación, usualmente basados en análisis estadístico, el más utilizado es el *modelo vectorial*. Un documento es representado como un vector n-dimensional, de términos indexados o de palabras clave [*Yang and Liu*, 1999; *Sebastiani*, 2002]. De esta manera, los documentos quedan representados como un vector $\mathbf{d} = (w_1, w_2, \ldots, w_n)$, donde cada término indexado corresponde a una palabra en el texto y tiene un peso asociado a él, que refleja la importancia del término para el documento y/o para la colección completa de documentos (ver Sección 2.2.3).

Por razones de eficiencia, algunos métodos usados en Clasificación de Textos (incluyendo los usados en este trabajo) hacen simplificaciones que pueden no estar completamente justificadas, pero que son experimentalmente válidas. Algunas de estas simplificaciones son:

- Ignorar la estructura del texto. No se intenta "comprender" completamente un documento usando análisis semántico.

- Asumir que los pesos para los términos indexados son mutuamente independientes. Aunque esta simplificación permite un tratamiento más sencillo de los documentos, los pesos usualmente no son independientes, el hecho que un término aparezca en un texto puede incrementar la probabilidad de encontrar otro término que esté usualmente relacionado con él.

- Ignorar el orden de las palabras. En esta "bolsa de palabras" todo texto que sea permutación de las mismas palabras es considerado igual. Esta simplificación no está siempre justificada, pero es necesaria por razones de eficiencia.

- Omitir palabras frecuentes que no contienen información semántica. A éstas palabras se les conoce con el nombre de "palabras vacías", ejemplos de ellas son las preposiciones, conjunciones y artículos.

- También se omiten las palabras que se presenten sólo una vez en la toda la colección. Algunas de estas palabras son debidas a errores ortográficos del texto. El resto, aún y cuando estén bien escritas, aportan muy poca o nula información comparada con la enorme cantidad de palabras en los documentos.

2.2.2. Reducción de Dimensionalidad

Al hacer la representación vectorial de los documentos se presenta un problema de alta dimensionalidad, debido a la cantidad tan grande de palabras (atributos) que aparecen en los documentos. Para facilitar el trabajo con estos datos es necesario hacer una reducción del conjunto de atributos [*Kim et˜al.*, 2005; *Yang and Pedersen*, 1997]. Se selecciona un subconjunto de características, de modo tal, que los atributos seleccionados tengan mayor valor de discriminación entre los documentos de las clases. Existen diversas técnicas para reducir el número de atributos, entre éstas se encuentra la *ganancia de información, IG* (por sus siglas en inglés *information gain*) [*Lee and Lee*, 2006] medida basada en la entropía de un sistema, es decir, en el grado de desorden de un sistema [*Shannon and Weaver*, 1963]. *IG* indica cuánto se reduce la entropía de todo el sistema si se conoce el valor de un atributo determinado. Con esto se puede saber cómo está relacionado ese atributo con respecto al sistema completo, en otras palabras, cuánta información aporta dicho atributo al sistema.

Para un conjunto $C = \{c_1, c_2, \ldots, c_m\}$ de clases posibles, la ganancia de infor-

mación del atributo o término t se define como:

$$IG(t) = -\sum_{i=1}^{m} P(c_i)\log(P(c_i)) + P(t)\sum_{i=1}^{m} P(c_i|t)\log(P(c_i|t)) + P(\bar{t})\sum_{i=1}^{m} P(c_i|\bar{t})\log(P(c_i|\bar{t}))$$

donde $P(c_i)$ es la probabilidad de la clase c_i, $P(t)$ es la probabilidad de seleccionar un documento que contenga el término t, $P(c_i|t)$ es la probabilidad condicional de que pertenezca a la clase c_i dado el documento con el término t; $P(\bar{t})$ es la probabilidad de seleccionar un documento que no contiene el término t y por último, $P(c_i|\bar{t})$ es la probabilidad condicional de que un documento pertenezca a la categoría c_i dado que el documento no contiene el término t.

Para la selección de atributos o términos se fija un umbral y el conjunto de atributos con IG menor que el umbral son eliminados.

2.2.3. Peso de los Términos

En la representación vectorial, a cada palabra se le asigna un peso, así w_i es el peso de la i-ésima palabra o término del documento. Se tienen distintos esquemas de pesado que se seleccionan de acuerdo la importancia que se le quiera dar a cada palabra con relación al resto de ellas.

- Booleano: 1's y 0's de acuerdo a si aparece o no el término en el documento.

$$w_i = \begin{cases} 1 & \text{si el término } i\text{-ésimo aparece en el documento} \\ 0 & \text{en caso contrario} \end{cases}$$

- Frecuencia del término (o tf): número de veces que aparece el término en el documento. De acuerdo a esto entre más veces aparezca un término en el documento más importante será.

$$w_i = tf_i$$

- $tf \cdot idf$: combina la frecuencia del término en el documento con la frecuencia de éste en el resto de los documentos de la colección [*Salton and Buckley*, 1987; *Debole and Sebastiani*, 2003].

$$w_i = tf_i \cdot \log\left(\frac{N}{n_i}\right)$$

donde N es el tamaño de la colección, es decir, el número total de documentos y n_i es el número de documentos en los que aparece el término i-ésimo.

2.3. Métodos de clasificación

A continuación, hablemos del proceso de aprendizaje automático para la clasificación de textos en lenguaje natural.

2.3.1. Vecinos más Cercanos

También conocido como k-NN del inglés *k-Nearest Neighbors* [*Cover and Hart*, 1967; *Masand et˜al.*, 1992]. La idea de este método es comparar el documento con los k vecinos más cercanos (o más similares), determinando así la clase del documento con la clase mayoritaria de los k vecinos. El método vectorial se puede ver como un k-NN con $k = 1$. Las medidas de similitud usadas por este método se explican en la Sección 2.4.

2.3.2. Naive Bayes

El método bayesiano o probabilístico ha sido ampliamente usado para la clasificación de textos [*Lewis*, 1998]. Este método usa la probabilidad conjunta de las palabras y las clases para estimar la probabilidad $P(c_i|\mathbf{d}_j)$ de cada clase dado un documento. Si se tiene un conjunto de documentos $\mathcal{D} = \{\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_m\}$ asociado a las clases predefinidas $\mathcal{C} = \{c_1, c_2, \ldots, c_n\}$, cada documento es representado por un vector $\mathbf{d}_j = (w_{1j}, w_{2j}, \ldots, w_{|\mathcal{T}|j})$ donde $\mathcal{T}$ es el conjunto de términos que pertenecen a c_i; el método bayesiano estima la probabilidad *a posteriori* de cada clase c_i dado el documento $\mathbf{d}_j$.

$$P(c_i|\mathbf{d}_j) = \frac{P(c_i)P(\mathbf{d}_j|c_i)}{P(\mathbf{d}_j)} \tag{2.1}$$

En la fórmula (2.1), $P(\mathbf{d}_j)$ es la probabilidad que se elija aleatoriamente el documento $\mathbf{d}_j$, esta probabilidad es independiente de las clases por lo que se puede omitir; y $P(c_i)$ es la probabilidad de que el documento elegido pertenezca a la clase c_i. Debido a que el número de posibles documentos $\mathbf{d}_j$ es muy grande, se vuelve complicado el cálculo de $P(\mathbf{d}_j|c_i)$. Para simplificar el cálculo de $P(\mathbf{d}_j|c_i)$, es común asumir que la probabilidad de una palabra o término dado es independiente de los otros términos que aparecen en el mismo documento. Aunque a primera vista esto puede ser visto como una simplificación exagerada, Naive Bayes presenta resultados comparables con los obtenidos por métodos más elaborados [*Yang and Liu*, 1999]. Usando esta simplificación es posible determinar $P(\mathbf{d}_j|c_i)$ como el producto de probabilidades de cada término que aparece en el documento.

$$P(\mathbf{d}_j|c_i) = \prod_{t=1}^{|\mathcal{T}|} P(w_{tj}|c_i) \tag{2.2}$$

De las fórmulas (2.1) y (2.2) tenemos que la probabilidad de que el documento $\mathbf{d}_j$ elegido aleatoriamente pertenezca a la clase c_i es:

$$P(c_i|\mathbf{d}_j) = P(c_i) \prod_{t=1}^{|\mathcal{T}|} P(w_{tj}|c_i) \tag{2.3}$$

con $P(c_i)$ calculado como:

$$P(c_i) = \frac{N_{c_i}}{N}$$

donde N_{c_i} es el número de documentos de la clase c_i y N es el total de documentos en el conjunto de entrenamiento. Por su parte $P(w_{tj}|c_i)$ usualmente se calcula como

$$P(w_{tj}|c_i) = \frac{1 + count(w_{tj}, c_i)}{N_{c_i} + |\mathcal{T}|}$$

donde $count(w_{tj}, c_i)$ es el número de veces que el término w_{tj} aparece en los documentos de la clase c_i. Para resolver el problema de probabilidad 0 se usa una estimación de Laplace (*Add-One Smoothing*) [*Jurafsky and Martin*, 2000].

De esta manera a $\mathbf{d}_j$ se le asigna la clase c_i donde $P(c_i|\mathbf{d}_j)$ es máxima en la igualdad (2.3) [*Sebastiani*, 2002]. El método Naive Bayes es muy popular en el área de Clasificación de Textos, y muchos investigadores han publicado resultados basados en él [*Lewis and Ringuette*, 1994; *Joachims*, 1999; *Koller and Sahami*, 1997; *Fuhr and Pfeifer*, 1994].

2.3.3. Máquinas de Vectores de Soporte

Conocido también como SVM por sus siglas en inglés, fue presentado por Vapnik [*Vapnik*, 1995; *Cortes and Vapnik*, 1995] y fue aplicado por primera vez a la Clasificación de Textos por Joachims [*Joachims*, 1998; *Burges*, 1998; *Cristianini and Shawe-Taylor*, 2000].

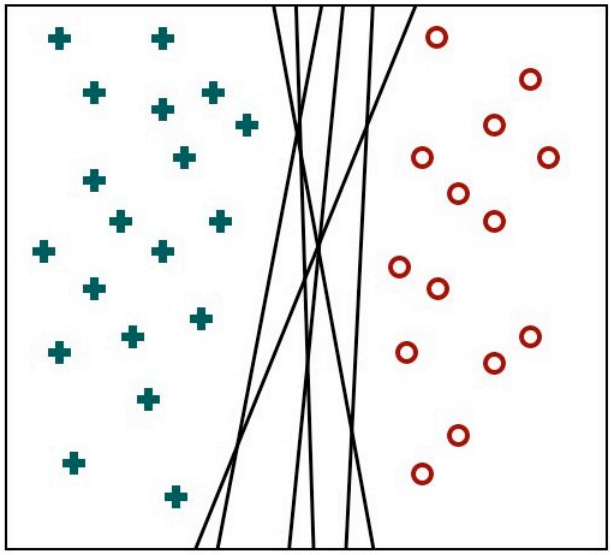

Figura 2.2: Posibles hiperplanos que separan las clases.

Esta técnica tiene raíces en la teoría de aprendizaje estadístico. Mapea los documentos en un espacio de atributos de alta dimensionalidad e intenta aprender hiperplanos de margen máximo entre dos clases de documentos. Además representa los límites de decisión usando un subconjunto de ejemplos de entrenamiento, conocidos como *vectores de soporte*.

La Figura 2.2 muestra una gráfica de un conjunto de ejemplos de entrenamiento que pertenecen a dos diferentes clases representadas por cruces y círculos. Los datos son linealmente separables, es decir, podemos encontrar un hiperplano tal, que todos las cruces estén de un lado del hiperplano y todos los círculos queden en el otro lado. Sin embargo, hay una infinidad de posibles hiperplanos. El hecho de que estos hiperplanos no tengan ningún error al separar los ejemplos de entrenamiento, no garantiza que con nuevos documentos suceda lo mismo.

La Figura 2.3, muestra dos hiperplanos y sus márgenes de riesgo de error. Entre mayores son los márgenes, menor será el riesgo de que un documento nuevo sea

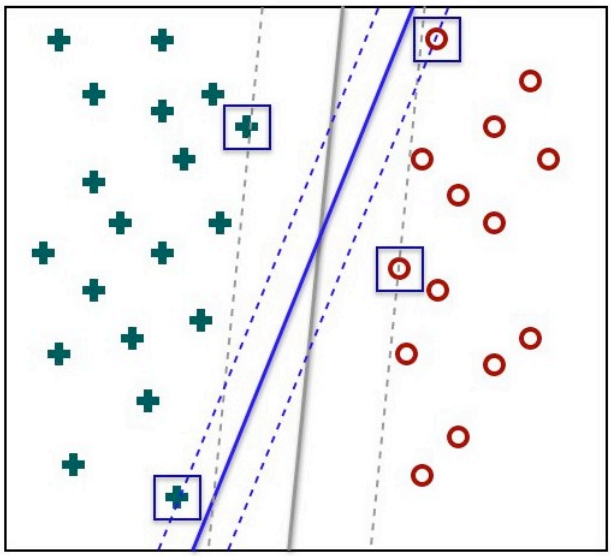

Figura 2.3: Un par de hiperplanos y sus márgenes de riesgo de error.

clasificado de manera errónea. En esta figura los cuadros indican los ejemplos que son tomados como vectores de soporte. Esto es para el caso en que los conjuntos son linealmente separables.

Para conjuntos de documentos que no son linealmente separables, SVM usa *funciones de convolución* o *Kernels*. Estos kernels transforman el espacio de atributos iniciales en otro, donde los documentos transformados son linealmente separables. El método SVM puede encontrar el hiperplano que separa los documentos con el valor máximo. Para una descripción detallada del SVM consulte [*Joachims*, 2002].

Esta idea puede ser generalizada fácilmente para colecciones con más de dos clases de documentos, la idea es dividir el problema multi-clase y convertirlo en varios problemas binarios. Generalmente es usado en *uno contra uno* o *uno contra todos* [*Scholköpf and Smola*, 2003]. Para uno contra uno, si se tienen q clases, se construye $q(q-1)/2$ clasificadores usando los documentos de cada combinación

de dos clases distintas. Para determinar la clase del documento nuevo se usa una estrategia de voto. En uno contra todos, se construyen q clasificadores para cada clase, usando los ejemplos de una clase y mezclando todas las demás clases. En este segundo caso, el clasificador produce una función que da un valor relativamente mayor a una de las dos clases, al documento nuevo se le asigna la clase que obtuvo el valor más alto. En uno contra uno se construyen más clasificadores pero cada clasificador tiene menos ejemplos de entrenamiento. La clasificación uno contra uno ha mostrado ser mejor en la práctica [*Hsu and Lin*, 2002].

Se han hecho amplias comparaciones reportando que de los clasificadores disponibles actualmente, el método SVM es el que mejores resultados obtiene en la mayoría de los casos [*Joachims*, 1998; *Yang and Liu*, 1999; *Liu et˜al.*, 2005; *Cardoso-Cachopo and Oliveira*, 2006].

2.3.4. Métodos Basados en Prototipos

Este método busca un documento representante de cada clase. El problema es elegir cuál de los documentos de la clase tomar como representante o, si es necesario, crear un documento "virtual" que represente de mejor manera a la clase, a este documento representante de la clase se le conoce como *prototipo*.

La clasificación se lleva a cabo comparando el documento nuevo con cada prototipo y asignando la clase del prototipo más similar. Las medidas usadas para encontrar la similitud entre el documento y el prototipo se describen en la Sección 2.4.

Dado un conjunto de documentos $\mathcal{D} = \{\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_m\}$ asociado a la clase C, el prototipo $\mathbf{p}$ de la clase se puede calcular de las siguientes maneras:

Suma

Sumando cada documento de la clase [*Chuang et˜al.*, 2000].

$$\mathbf{p}_{suma} = \sum_{\mathbf{d}_k \in C} \mathbf{d}_k$$

Promedio

Cada clase es representada por el promedio de sus documentos. La suma de todos los documentos de la clase (ejemplos positivos) se divide entre el número de documentos de la clase. [*Han and Karypis*, 2000; *Shankar and Karypis*, 2000].

$$\mathbf{p}_{prom} = \frac{1}{|C|} \sum_{\mathbf{d}_k \in C} \mathbf{d}_k$$

Suma Normalizada

Se calcula una suma normalizada. La suma de todos los documentos de la clase (ejemplos positivos) entre la norma euclidiana de la suma [*Lertnattee and Theeramunkong*, 2004; *Tan et˜al.*, 2005b].

$$\mathbf{p} = \frac{1}{||\mathbf{p}_{suma}||} \sum_{\mathbf{d}_k \in C} \mathbf{d}_k$$

Basado en la formula del Rocchio

Usando la fórmula de Rocchio se calcula el prototipo de la clase haciendo una suma ponderada. La suma de todos los documentos que pertenecen a la clase (ejemplos positivos) menos la suma ponderada de todos los no pertenecientes a ella (ejemplos negativos). Para cada clase, su prototipo se define como:

$$\mathbf{p} = \beta \sum_{\mathbf{d}_k \in C} \mathbf{d}_k - \gamma \sum_{\mathbf{d}_k \notin C} \mathbf{d}_k$$

el valor de β se fija mayor al de γ para darle más peso a la suma de los ejemplos positivos. Los valores usuales son $\beta = 16$ y $\gamma = 4$ [*Hull*, 1994; *Joachims*, 1997; *Chuang et˜al.*, 2000].

2.4. Medidas de Similitud

La similitud entre dos objetos es una cantidad numérica del grado de semejanza de éstos, consecuentemente la similitud es *mayor* entre mayor es la semejanza de los objetos [*Tan et˜al.*, 2005a]. Las similitudes generalmente no son negativas y a menudo se definen entre 0 (no similares) y 1 (similares).

Si queremos comparar dos vectores $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ y $\mathbf{y} = (y_1, y_2, \ldots, y_n)$ que constan de valores binarios, tenemos las siguientes frecuencias:

f_{00} = número de atributos tal que $x_i = 0$ y $y_i = 0$
f_{01} = número de atributos tal que $x_i = 0$ y $y_i = 1$
f_{10} = número de atributos tal que $x_i = 1$ y $y_i = 0$
f_{11} = número de atributos tal que $x_i = 1$ y $y_i = 1$

2.4.1. Coeficiente de Empalme Simple

Conocido también como SMC del inglés *Simple Matching Coefficient*, se define como:

$$SMC = \frac{f_{00} + f_{11}}{f_{01} + f_{10} + f_{00} + f_{11}}$$

el cual representa al número de atributos que se empalman divididos entre el total de atributos.

2.4.2. Coeficiente Jaccard

A diferencia de SMC el coeficiente Jaccard sólo considera los elementos de intersección, omitiendo los casos donde el atributo es 0 en ambos vectores. Así el coeficiente de Jaccard es calculado como:

$$J = \frac{f_{11}}{f_{01} + f_{10} + f_{11}}$$

es decir, el número de atributos en común (intersección) entre el total de atributos que aparecen en $\mathbf{x}$ o $\mathbf{y}$. De esta medida de similitud existe una variante conocida como *Coeficiente Dice*, en la que se da una mayor importancia a los atributos de la intersección.

Coeficiente Dice

$$Dice = \frac{2f_{11}}{f_{01} + f_{10} + 2f_{11}}$$

2.4.3. Coseno

Es una de las medidas más usadas en la similitud de documentos, en la cual se omiten los empalmes 0-0 entre los vectores, es decir, se descartan atributos que

tengan peso = 0 en los vectores comparados, al igual que sucede con el coeficiente de Jaccard, sin embargo puede manejar vectores de valores no binarios. Así, la similitud entre dos vectores se calcula como:

$$sim(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}}{||\mathbf{x}|| \times ||\mathbf{y}||}$$

Por otra parte, la *disimilitud* entre dos objetos es la cantidad numérica del grado de diferencia de los objetos. La disimilitud es *menor* entre *mayor* similitud existe entre los documentos. Frecuentemente el término distancia es usado como sinónimo de disimilitud. Las disimilitudes frecuentemente tienen su valor en el intervalo $[0, 1]$, aunque también es común el rango $[0, \infty)$

2.4.4. Manhattan

Distancia de bloques o calles, la idea intuitiva es contar las calles para ir de un punto a otro en una ciudad.

$$d(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} |x_i - y_i|$$

2.4.5. Euclidiana

La distancia Euclidiana entre dos vectores de n dimensiones está dada por:

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

Clase	Predicción **A**	Predicción **B**
A	a	b
B	c	d

Tabla 2.1: Predicciones de un sistema de clasificación.

Minkowski

Generalizando la distancia Euclidiana se tiene

$$d(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^{n} |x_i - y_i|^r \right)^{\frac{1}{r}}$$

para $r = 1$ se tiene la distancia Manhattan y para $r = 2$ la distancia Euclidiana.

2.5. Medidas de evaluación

La tarea del sistema de clasificación de textos consiste en asociar un documento nuevo con la clase a la que pertenece, de entre una lista de clases previamente definidas. Sin embargo no es suficiente que la clasificación se lleve a cabo en un tiempo razonable, la clasificación también debe ser correcta, es decir, que el documento realmente pertenezca a la clase que se le asigna. Medidas como *Precisión* (π), la *Cobertura* o *Recuerdo* (ρ) y F_β han sido ampliamente usadas para comparar el desempeño de los métodos de Clasificación de Textos [*Sebastiani*, 2002].

Consideremos el problema de la clasificación binaria, clasificación de documentos en la clase **A** y la clase **B**. En la Tabla 2.1 se muestran los resultados de la predicción que hace el sistema de clasificación

De los $a + b$ documentos que son de la clase **A**, el sistema predice que a

documentos pertenecen a la clase **A** y que los b restantes pertenecen a la clase **B**. Por otro lado la clase **B** cuenta con $c + d$ documentos de los cuales, el sistema predijo que c documentos a la clase **A** y d como pertenecientes a la clase **B**. Con lo anterior se tiene que, el número de documentos correctamente clasificados es $a + d$, y por tanto, $c + b$ es el número de documentos que fueron clasificados incorrectamente. La manera de presentar los datos en la Tabla 2.1, se conoce como *matriz de confusión*, donde a recibe el nombre de Verdaderos Positivos (VP), b Falsos Negativos (FN) y c son los Falsos Positivos (FP).

Precisión (π)

La Precisión es la proporción del número de documentos correctamente clasificados, entre todos los que se predijo que pertenecían a la clase. La Precisión se expresa como:

$$\pi = \frac{a}{a + c}$$

Recuerdo (ρ)

El Recuerdo indica cuántos documentos fueron clasificados correctamente de entre todos los documentos de la clase. El Recuerdo viene dado por la expresión:

$$\rho = \frac{a}{a + b}$$

F-measure (F_β)

Es una medida que engloba en una sola el Recuerdo y la Precisión descrita por:

$$F_\beta = \frac{(1 + \beta^2)\,\pi\,\rho}{\beta^2\,\pi + \rho} \tag{2.4}$$

	Micropromedio	Macropromedio
Precisión (π)	$\pi = \frac{\sum_{i=1}^{\|\mathcal{C}\|} VP_i}{\sum_{i=1}^{\|\mathcal{C}\|} VP_i + FP_i}$	$\pi = \frac{\sum_{i=1}^{\|\mathcal{C}\|} \pi_i}{\|\mathcal{C}\|} = \frac{\sum_{i=1}^{\|\mathcal{C}\|} \frac{VP_i}{VP_i + FP_i}}{\|\mathcal{C}\|}$
Recuerdo (ρ)	$\rho = \frac{\sum_{i=1}^{\|\mathcal{C}\|} VP_i}{\sum_{i=1}^{\|\mathcal{C}\|} VP_i + FN_i}$	$\rho = \frac{\sum_{i=1}^{\|\mathcal{C}\|} \rho_i}{\|\mathcal{C}\|} = \frac{\sum_{i=1}^{\|\mathcal{C}\|} \frac{VP_i}{VP_i + FN_i}}{\|\mathcal{C}\|}$

Tabla 2.2: Promedios de Precisión y Recuerdo; VP_i, VN_i, FP_i y FN_i son los Verdaderos Positivos, Verdaderos Negativos, Falsos Positivos y Falsos Negativos para la clase $c_i \in \mathcal{C}$ respectivamente.

donde β es el parámetro que controla la importancia relativa entre las dos medidas. Es común usar el valor $\beta = 1$ en la que se da igual importancia a ambos valores.

Micro y Macro Promedios

Las medidas anteriores pueden ser calculadas sobre la colección completa, a esto se le llama *micro promedio*, o bien, para cada clase y enseguida promediar entre las clases, lo que se conoce como *macro promedio*.

En el micro promedio, cada documento vale lo mismo para el promedio y las clases pequeñas tienen poco impacto en el resultado final. En el macro promedio, primero se determina el promedio para cada clase y después cada clase vale lo mismo para el promedio final (ver Tabla 2.2).

La diferencia es particularmente importante cuando la colección es desbalanceada, esto es, cuando hay clases con mucha diferencia en el número de documentos.

Exactitud

Por último, se define la *Exactitud* como el porcentaje de documentos correctamente clasificados. Esta medida es generalmente usada para evaluar tareas de una sola etiqueta [*Nigam et˜al.*, 2000; *Han and Karypis*, 2000; *Chuang et˜al.*, 2000; *Han et˜al.*, 2001; *Lertnattee and Theeramunkong*, 2004; *Sebastiani*, 2005a]. Usualmente la Exactitud es representada como valor real entre 0 y 1, y es dado por la expresión:

$$Exactitud = \frac{\sum_{i=1}^{|\mathcal{C}|} VP_i}{\sum_{i=1}^{|\mathcal{C}|} VP_i + FP_i}$$

con VP_i y FP_i son los Verdaderos Positivos y Falsos Positivos para la clase $c_i \in \mathcal{C}$ respectivamente, es decir, el número de documentos clasificados correctamente del total de documentos.

Esta medida tiene un problema que a menudo se hace referencia. Si las colecciones de datos están desbalanceadas, es posible encontrar fácilmente un buen clasificador que prediga la clase más frecuente en los datos de entrenamiento. Es posible evitar el problema proveyendo junto a cada conjunto de datos, el resultado que se podría alcanzar con este clasificador "simple" y evaluando otros clasificadores teniendo este valor como punto de referencia.

2.5.1. Validación Cruzada

Para conjuntos de datos donde existe una partición estándar de entrenamiento y otra de prueba, es suficiente con comparar los resultados obtenidos con clasificadores distintos usando las medidas de evaluación anteriores. Si el conjunto no cuenta con esta partición y se hace una partición particular, puede que el clasi-

ficador salga mal evaluado debido a diversos factores como: que los documentos de entrenamiento no fueron lo suficiente representativos de las clases o que fueran demasiado pocos por mencionar algunos. Puede inclusive que suceda lo contrario, que el clasificador obtenga muy buenos resultados lo cual podría no ser cierto.

Para evitar este problema, existen varios métodos de validación, uno de los más usados es *k-fold cross-validation* o *Validación Cruzada de k-pliegues*. El método consiste en partir el conjunto en k partes iguales, se selecciona una de ellas para prueba y las $k-1$ restantes, para entrenamiento. Este proceso se repite k veces de modo que cada partición sea usada exactamente una vez como prueba. Al final se hace un promedio de los resultados con cada partición. Es común usar un valor de $k = 10$ para dividir el conjunto original (*10 fold cross validation*).

2.6. Corpora de Evaluación

Estas son colecciones de documentos pre-clasificados y que son fundamentales para el desarrollo y evaluación de los sistemas de Clasificación de Textos. Son necesarios para entrenar el sistema y después para probar que tan bien se comporta cuando se da un nuevo documento a clasificar. En la primera fase (la fase de entrenamiento), algunos de estos documentos llamados los documentos de entrenamiento, son usados para entrenar el sistema de Clasificación de Textos permitiendo aprender un modelo de los datos. En la segunda fase (la de prueba), el resto de los documentos, los documentos conocidos como de prueba, son usados para probar el sistema.

Para poder hacer una comparación justa entre varios sistemas de Clasificación de Textos es deseable que se prueben en configuraciones equivalentes. Con esta idea en mente, muchas de las colecciones son creadas y hechas públicas, gene-

ralmente con una división estándar de entrenamiento/prueba. Así, los resultados obtenidos por diferentes sistemas pueden ser comparados correctamente.

Algunas de las colecciones públicas disponibles son más usadas que otras. En el campo de la Clasificación de Textos, la colección de noticias Reuters-21578 se encuentra entre las más usadas.

Capítulo 3

Método propuesto

El siguiente capítulo describe las dos etapas del método de Clasificación de Textos propuesto en esta tesis. Se mencionan los clasificadores y medidas de similitud usados.

3.1. Método en dos Etapas

La tarea de clasificación de textos se vuelve más compleja a medida que aumenta el número de clases predefinidas el las que un documento nuevo puede ser clasificado. Sumado a lo anterior, puede presentarse similitud entre clases (debido a que tienen vocabulario en común), con lo que, la asignación de la clase al documento nuevo se dificulta todavía más. Una solución a este problema pudiera ser reducir el número de clases, es más sencillo elegir la clase a la que pertenece un documento de entre un número reducido de clases. Con esto, el clasificador podría enfocarse con mayor facilidad a resolver un problema con clases semejante si esto llegara a presentarse.

Por lo anterior, se propone un método en dos etapas, la primera de ellas busca

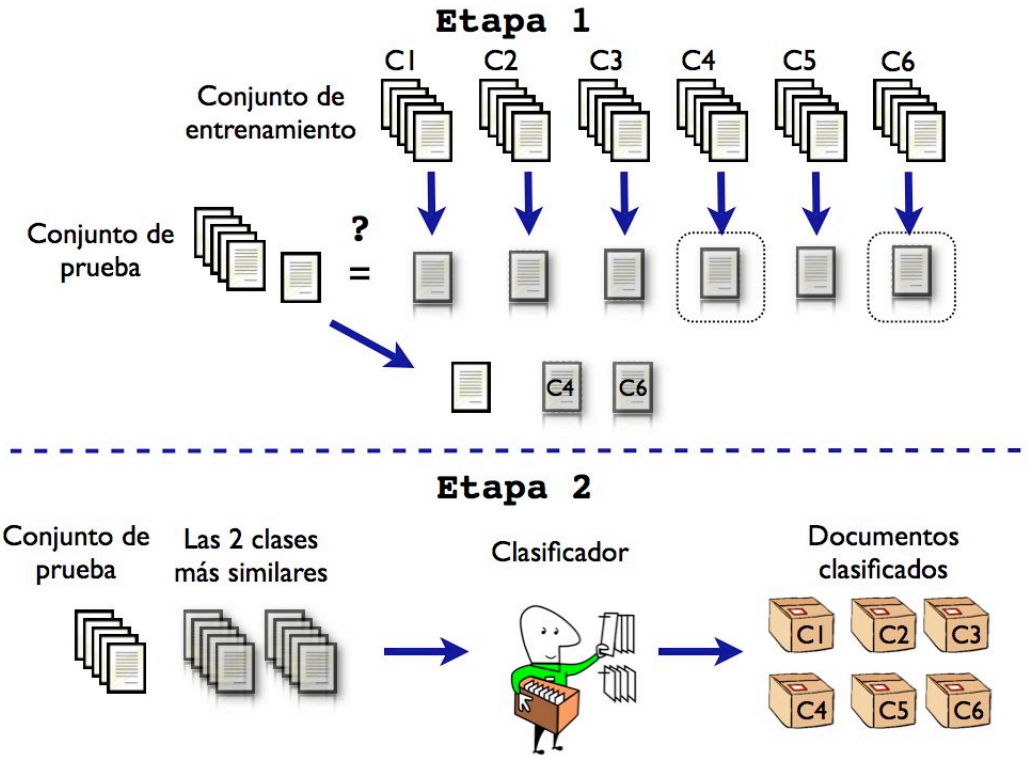

Figura 3.1: Esquema del método propuesto de Clasificación de Textos en dos etapas. **Etapa 1**: Reducción de clases, selección de las dos clases más similares al documento. **Etapa 2**: Clasificación del documento, elección de la clase correcta de entre las dos clases más similares.

reducir el número de clases iniciales seleccionando las dos clases más similares a cada documento. La segunda etapa clasifica el documento asignado una de las dos clases resultantes de la primera etapa. La Figura 3.1 muestra el esquema del proceso de clasificación propuesto.

Primera Etapa

El objetivo de esta primera etapa es reducir las clases. Se eligen sólo dos clases debido a que muchos de los clasificadores tratan internamente los problemas multi-clases como varios problemas de clasificación binaria, en particular el clasificador SVM el que mejor desempeño ha tenido en la tarea de Clasificación de Textos y que se usa en este trabajo.

Es necesario definir un esquema de comparación para elegir las clases similares, se podría pensar en comparar cada documento nuevo con todos los documentos de entrenamiento (un esquema de vecinos más cercanos) y elegir las clases de los dos documentos más similares. El inconveniente es que el conjunto de documentos de entrenamiento puede llegar a ser muy grande, lo que provocaría que el costo en tiempo dedicado a comparaciones sea igualmente grande. Usar las probabilidades del clasificador Naive Bayes para cada clase dado el documento nuevo (como se explica en la Sección 2.3.2) resulta a primera vista una buena opción para medir la similitud del documento con las clases, el problema es que en la práctica y después de hacer un par de pruebas, Naive Bayes da probabilidad de 1 en la gran mayoría de los casos, no dejando opción a elegir una clase con menor probabilidad.

Una alternativa más es buscar una representación de las clases para comparar cada documento nuevo sólo con la representación y no con todo el conjunto de documentos de la clase. A esta representación se le conoce como *prototipo* y de

éstos existen varias maneras de obtenerlos explicadas en la Sección 2.3.4.

El prototipo basado en la Suma Normalizada descrito en la Sección 2.3.4 es el mejor de acuerdo a los resultados mostrados por Cardoso [*Cardoso-Cachopo and Oliveira*, 2006]. Para construir los prototipos de las clases, es necesario dar pesos a los términos y con esto generar la representación vectorial de los documentos. Una de la formas más usadas de tomar el peso del término es $tf \cdot idf$, definido en la Sección 2.2.3 [*Cardoso-Cachopo and Oliveira*, 2006].

Lo siguiente es comparar el documento a clasificar con los prototipos de las clases. La medida de similitud generalmente usada para este tipo de prototipo es la del Coseno descrita en la Sección 2.4.3. De esta manera los dos valores más grandes nos darán las clases más similares.

3.1.1. Esquema basado en Prototipos

El peso de un término calculado como $tf \cdot idf$ y usado para la generación del prototipo de Suma Normalizada, es una combinación de pesos respecto al documento (tf) y respecto a la colección completa de documentos (idf). Este peso y por lo tanto el prototipo, no reflejan la importancia que pudiera tener el término para cada clase. Un caso extremo sería que, en una colección se tuviera un sólo documento de una clase en el que ocurre muchas veces una palabra que pudiera ser característica para la clase, pero este esquema no le da esa importancia debido a que aparece esta misma palabra una vez en cada documento del resto de las clases.

Por esta razón, se propone un esquema de prototipo en el cual, el peso del término indique la importancia que tiene éste para cada clase, y al mismo tiempo sea discriminante para las demás clases. Dicho de otra manera, un término

tendrá mayor valor para una clase, cuantas más veces aparezca en la clase y menos veces aparezca en el resto de las clases. Teniendo en cuenta lo anterior, el peso del término será distinto para distintas clases y no para distintos documentos. El nuevo peso refleja la importancia por documento respecto a la clase y el conjunto de entrenamiento, a diferencia del anterior $tf \cdot idf$, donde se refleja la importancia del término respecto al documento y al conjunto de entrenamiento. El peso de la palabra i-ésima wi_{clase}, respecto a la *clase* se calcula como

$$w_{i_{clase}} = df_i \cdot \log\left(\frac{N_{clases}}{n_{i_{clases}}}\right) \tag{3.1}$$

donde df_i es el número de documentos en la clase en los que el término i-ésimo aparece, este valor es normalizado entre el total de documentos en la clase; N_{clases} es el total de clases; y $n_{i_{clases}}$ es el número de clases que tienen documentos con el i-ésimo término.

De la ecuación (3.1), se puede observar que una palabra no tiene valor si aparece en todas las clases, o si la clase a analizar no la contiene. Por esto, tendremos para cada clase distinto número de términos como atributos característicos de la misma. Podemos considerar a estos atributos como una "bolsa de palabras" o "bolsa de atributos", estos atributos son los que ayudan a distinguir entre los documentos de las clases y que, a su vez, representan de mejor manera a la clase. De este modo nuestro prototipo de la clase es esta *Bolsa de Atributos*.

3.1.2. Medida de Similitud

Es necesario ahora, con este nuevo esquema de prototipo, definir una manera de comparar este prototipo con cada documento. Se puede considerar como con-

juntos de atributos donde se quiere ver qué tanto comparten esos atributos los documentos con la clase, y no sólo eso, sino también saber si los atributos que comparten son importantes o no, lo que se puede saber con el peso que hemos descrito. Llamamos *Intersección Pesada* a esta forma de comparar documentos con las clases y lo definimos como:

$$sim(\mathbf{d}, \mathbf{p}) = \sum_{i \in \mathbf{d}} w_{i_{doc}} \cdot w_{i_{clase}} \qquad (3.2)$$
$$\text{donde} \qquad w_{i_{doc}} = tf_i$$

donde **d** es el documento que se quiere comparar, **p** es el prototipo de la case (la bolsa de atributos). $w_{i_{doc}}$, representa el peso del término i-ésimo en el documento, que en este caso se lo da la frecuencia del término.

Después de haber definido la representación de las clases y la manera de cómo medir su similitud con un nuevo documento, se puede proceder a elegir las dos clases más similares al documento y continuar con la segunda etapa. Cabe mencionar que no existe ningún umbral de similitud para seleccionar las clases más similares, simplemente se ordenan de la clase más similar a la menos similar y se toman las dos primeras, puede suceder que se de empate en cuanto a los valores de similitud, en ese caso se toma aleatoriamente una de ellas.

Segunda Etapa

En esta etapa se tiene ya un problema reducido, para cada documento nuevo se tienen dos clases de las cuales, un clasificador deberá elegir una de ellas como la correcta para el documento. Los clasificadores empleados para esta tarea fueron: Naive Bayes, descrito en la Sección 2.3.2 y SVM, brevemente explicado en la

Sección 2.3.3. Se usaron las implementaciones de éstos en la herramienta *Weka* de minería de datos [*Witten and Frank*, 2005].

Es interesante que en esta segunda etapa no sólo se pueden utilizar clasificadores tradicionales sino también el mismo esquema de prototipos propuesto basado en Bolsa de Atributos y usando la medida de similitud de Intersección Pesada encontrar la clase más similar de entre las dos restantes. En esta segunda etapa los pesos para cada atributo serían distintos a los de la primera etapa ya que palabras que ocurrían en la mayoría de las clases y por eso su valor era menor, ahora puede que sólo aparezcan en una de las dos clases.

Capítulo 4

Experimentos y Resultados

En este capítulo se muestran los experimentos y sus resultados, llevados a cabo, con el objeto de evaluar el método propuesto. Se describe el corpora usado, el desarrollo, los resultados y discusión de los mismos.

4.1. Corpora usado

Reuters-21578

Esta colección y sus variantes, se han tomado como una comparación estándar para la clasificación de textos en al menos los últimos 10 años. *Reuters-21578* es un conjunto de $21,578$ textos de noticias del año 1987 de la agencia de noticias Reuters. El conjunto está clasificado en 135 clases temáticas referentes en su mayoría a negocios y economía.

Todos los documentos contenidos en el Reuters-21578 fueron clasificados manualmente por personal de Reuters Ltd y Carnegie Group, Inc. (CGI). Posteriormente David D. Lewis y colaboradores dieron formato a los documentos y crearon

la división del conjunto en conjunto de entrenamiento y conjunto de prueba, generando así la *Distribución 1.0 de Reuters-21578* [*Lewis*, 1991].

Esta colección es desbalanceada, es decir, contiene documentos distribuidos en cantidades muy disparejas en las distintas clases. Para tener conjuntos estándar con los cuales poder comparar los resultados obtenidos por diferentes experimentos, se han formado subconjuntos de la colección original, de los cuales podemos mencionar a *R(10)*, que contiene las 10 clases de mayor frecuencia en la colección, es decir, estas clases son las que tienen más documentos; otro subconjunto es *R(90)*, que busca las clases que al menos tengan un documento de prueba y uno de entrenamiento, y *R(115)* donde se pide que al menos tengan un documento de entrenamiento. [*Sebastiani*, 2002; *Debole and Sebastiani*, 2005].

Aún así las colecciones anteriores tienen documentos que pertenecen a más de una clase, ya que el presente trabajo se enfoca a la tarea de Clasificación de Textos con una sola clase, se necesita tener un corpus con documentos etiquetados para tal propósito. Partiendo de los conjuntos R(10) y R(90), se consideran todos los documentos que estén etiquetados con una sola clase. El resultado son conjuntos con 8 y 52 clases respectivamente, siguiendo la convención de Sebastiani [*Sebastiani*, 2002] llamamos a estos conjuntos *R(8)* y *R(52)*. Ambos conjuntos han sido utilizados anteriormente [*Cardoso-Cachopo*, 2007] y se toman como referencia para comparar resultados.

La composición de los conjuntos, sus clases, el número de documentos por clase y la división tanto para entrenamiento como para prueba, se muestran en la Tabla 4.1 para R(8) y en la Tabla 4.2 para R(52).

Ambos conjuntos tienen desbalance en el número de documentos para entrenar por clase, con lo que es de esperarse que el clasificador trabaje de mejor manera con las clases que tienen mayor número de documentos, ya que tienen muchos

R(8)			
Clases	Entrena	Prueba	Total
acq	1596	696	2292
crude	253	121	374
earn	2840	1083	3923
grain	41	10	51
interest	190	81	271
money-fx	206	87	293
ship	108	36	144
trade	251	75	326
Total	**5485**	**2189**	**7674**

Tabla 4.1: Composición del conjunto R(8)

ejemplos de donde "aprender" las características de la clase. Por la misma razón se puede esperar que el desempeño con las clases con menos documentos sea mucho menor.

20 Newsgroups

20 Newsgroups es una colección de aproximadamente 20,000 documentos organizados en 20 grupos de noticias (similares a los grupos de discusión) [*Lang*, 1995]. Se usa la versión *bydate* que es también usada por Cardoso [*Cardoso-Cachopo and Oliveira*, 2006] donde se omiten los documentos duplicados. La composición del conjunto se detalla en la Tabla 4.3.

R(52)							
Clases	Entrena	Prueba	Total	Clases	Entrena	Prueba	Total
acq	1596	696	2292	jobs	37	12	49
alum	31	19	50	lead	4	4	8
bop	22	9	31	lei	11	3	14
carcass	6	5	11	livestock	13	5	18
cocoa	46	15	61	lumber	7	4	11
coffee	90	22	112	meal-feed	6	1	7
copper	31	13	44	money-fx	206	87	293
cotton	15	9	24	money-supply	123	28	151
cpi	54	17	71	nat-gas	24	12	36
cpu	3	1	4	nickel	3	1	4
crude	253	121	374	orange	13	9	22
dlr	3	3	6	pet-chem	13	6	19
earn	2840	1083	3923	platinum	1	2	3
fuel	4	7	11	potato	2	3	5
gas	10	8	18	reserves	37	12	49
gnp	58	15	73	retail	19	1	20
gold	70	20	90	rubber	31	9	40
grain	41	10	51	ship	108	36	144
heat	6	4	10	strategic-metal	9	6	15
housing	15	2	17	sugar	97	25	122
income	7	4	11	tea	2	3	5
instal-debt	5	1	6	tin	17	10	27
interest	190	81	271	trade	251	75	326
ipi	33	11	44	veg-oil	19	11	30
iron-steel	26	12	38	wpi	14	9	23
jet	2	1	3	zinc	8	5	13
				Total	**6532**	**2568**	**9100**

Tabla 4.2: Composición del conjunto R(52)

20newsgroups			
Clases	Entrena	Prueba	Total
alt.atheism	480	319	799
comp.graphics	584	389	973
comp.os.ms-windows.misc	572	394	966
comp.sys.ibm.pc.hardware	590	392	982
comp.sys.mac.hardware	578	385	963
comp.windows.x	593	392	985
misc.forsale	585	390	975
rec.autos	594	395	989
rec.motorcycles	598	398	996
rec.sport.baseball	597	397	994
rec.sport.hockey	600	399	999
sci.crypt	595	396	991
sci.electronics	591	393	984
sci.med	594	396	990
sci.space	593	394	987
soc.religion.christian	598	398	996
talk.politics.guns	545	364	909
talk.politics.mideast	564	376	940
talk.politics.misc	465	310	775
talk.religion.misc	377	251	628
Total	**11293**	**7528**	**18821**

Tabla 4.3: Composición del conjunto 20newsgroups

WebKB			
Clases	Entrena	Prueba	Total
course	620	310	930
faculty	750	374	1124
project	336	168	504
student	1097	544	1641
Total	**2803**	**1396**	**4199**

Tabla 4.4: Composición del conjunto R(8)

WebKB

Esta colección contiene páginas web recolectadas de los departamentos de ciencias computacioneslas de 4 universidades en Enero de 1997 como parte del proyecto *World Wide Knowledge Base* a cargo de *Text Learning Group* de la universidad de Carnegie Mellon. El conjunto original cuenta con $8,282$ páginas web clasificadas manualmente en 7 clases. Para los experimentos realizados en esta tesis se usa un subconjunto del original, mismo que es usado por Cardoso [*Cardoso-Cachopo and Oliveira*, 2006] el cual ya cuenta con una partición de entrenamiento y prueba. La composición del corpus se muestra en la Tabla 4.4.

Desastres Naturales en México

La mayor parte del trabajo hecho en Clasificación de Texto ha usado corpus en inglés, con el objeto de realizar estudios con copora en el idioma español el grupo del Laboratorio de Tecnologías del Lenguaje en el INAOE se ha dado a la tarea de recopilar colecciones en español. Una de ellas *Noticias de Desastres Naturales en México* [*Téllez*, 2005], es un conjunto de noticias de desastres naturales publicadas

Desastres Naturales	
Clases	Noticias
Forestal	92
Huracán	76
Inundación	87
Sequía	41
Sismo	143
Total	**353**

Tabla 4.5: Composición del conjunto Desastres Naturales en México

en periódicos de Internet [*López et˜al.*, 2003]. Es una colección pequeña formada por 5 clases, aún y cuando los documentos entre clase y clase varían, no existe un desbalance tan grande como en R(8) y R(52). Debido a que no existe una división estándar de la colección para documentos de entrenamiento y documentos de prueba en la evaluación de los clasificadores se usa Validación Cruzada con partición de 10 (ver Sección 2.5.1). La Tabla 4.5 muestra la composición de esta colección.

Poemas de Escritores Mexicanos

Adicionalmente a los anteriores y con el fin de estudiar la clasificación no-temátca, se tiene el corpus de *Poemas de Escritores Mexicanos*. La principal tarea en la atribución de autoría es identificar las características de cada clase (en este caso autor), las cuales deben ser invariantes y ayudar a discriminar a un autor de otros. En contraste con las tareas de clasificación por contenido (temáticas), en este caso aún no es claro cómo determinar el conjunto de características que

Poemas	
Clases	Poemas
Efraín Huerta	48
Jaime Sabines	80
Octavio Paz	75
Rosario Castellanos	80
Rubén Bonifaz	70
Total	**439**

Tabla 4.6: Composición del conjunto Poemas de Escritores Mexicanos

deben ser utilizadas para identificar un autor [*Coyotl*, 2007]. En el proceso de descubrir estas características de cada autor se usan las palabras para hacer esta caracterización.

Esta colección es un conjunto de poemas de escritores mexicanos, construido también por el grupo del Laboratorio de Tecnologías del Lenguaje del INAOE para tratar el problema de clasificar textos por atribución de autoría. Al igual que para el corpus de desastres naturales, se usa validación cruzada para la evaluación. La Tabla 4.6 muestra el contenido de este corpus.

4.2. Datos de Comparación

Se construye un punto de comparación o *baseline* con los clasificadores Naive Bayes y SVM. Debido a que la cantidad de atributos (palabras) puede llegar a ser muy grande ($\approx 45,000$ en el caso de 20 Newsgroups) se aplica IG (ganancia de información) > 0 para reducirlos ($\approx 5,000$) y hacer el manejo de los datos más sencillo. El proceso de creación del baseline es de la manera tradicional, proceso descrito en la Sección 2.2. La Tabla 4.7 muestra el comportamiento del clasificador

Exactitud en Datos de Comparación		
Corpus	**Naive Bayes**	**SVM**
R(8)	0.85	0.92
R(52)	0.81	0.86
20newsgroups	0.55	0.68
webkb	0.63	0.84
Desastres	0.90	0.92
Poemas	0.59	0.60

Tabla 4.7: Baseline: Valores de Exactitud para cada corpus.

en exactitud para cada corpus usado.

Como era de esperarse, el desempeño en el baseline del clasificador SVM es superior al de Naive Bayes en todos los casos. Otra cosa de esperarse es que el clasificador tiene mejor desempeño para las clases con más documentos de entrenamiento como se puede ver en las precisiones y recuerdos para *acq* y *earn* para R(8) (Tabla 4.10). El caso contrario sucede para las clases con menor número de documentos para entrenar al clasificador, donde la precosión y el recuerdo llegan a ser nulos como *platinum*, *potato* y *tea* entre otros para el caso de R(52) (Tabla 4.12).

El corpus de Desastres Naturales muestra una exactitud con los métodos tradicionales muy alta, de al menos 0.90 (Tabla 4.20). Por su parte los conjuntos 20 Newsgroups y el de Poemas obtienen exactitudes de al rededor de 0.60 (Tablas 4.16 y 4.22). En teoría conseguir mejorar una exactitud de 0.90 es más complicado que mejorar una de 0.60.

En el detalle del baseline para el corpus de Poemas presentado en la Tabla 4.22 se puede decir que no hay diferencia entre usar un clasificador u otro, ya que

presentan una exactitud similar con ambos clasificadores.

4.3. Método Propuesto

Los experimentos siguientes pretenden validar el método propuesto, los resultados se comparán con el baseline.

Primera Etapa: Reducción de Clases

El objetivo en esta etapa, como ya se mencionó en el capítulo anterior, es reducir el problema inicial multi-clase a un problema de clasificación binaria; en particular se buscan las dos clases más similares. Para ello, se representa a cada clase con la bolsa de atributos obtenida de los valores $tf \cdot idf$ (ver Fórmula 3.1), y para la comparación se usa la intersección pesada (ver Fórmula 3.2).

Como dato de referencia, se mide sí entre esas dos clases seleccionadas se encuentra la clase correcta de cada documento, esto es, puede suceder que al seleccionar las dos clases más similares se haya perdido la clase del documento. Esta medida nos da una idea de hasta donde es posible llegar en exactitud en el paso siguiente. Los resultados aparecen en la Tabla 4.8.

Segunda Etapa: Clasificación

En esta etapa, por cada documento de prueba existe un clasificador binario, que puede o no ser distinto al de otro documento. Este clasificador binario elige de entre las dos posibles opciones obtenidas del proceso de reducción de clases.

La Tabla 4.9 muestra los desempeños de los clasificadores Naive Bayes y SVM

Máxima Exactitud	
Corpus	Exactitud
R(8)	0.97
R(52)	0.86
20newsgroups	0.90
webkb	0.88
Desastres	0.99
Poemas	0.80

Tabla 4.8: Máximo valor que puede ser alcanzado en la etapa de clasificación, después de reducir las clases a dos.

Exactitud				
	Naive Bayes		**SVM**	
Corpus	Baseline	2 Etapas	Baseline	2 Etapas
R(8)	0.85	**0.90**	0.92	**0.94**
R(52)	0.81	**0.85**	**0.86**	**0.86**
20newsgroups	0.55	**0.71**	0.68	**0.76**
webkb	0.63	**0.70**	**0.84**	**0.84**
Desastres	0.90	**0.92**	0.92	**0.95**
Poemas	**0.59**	**0.59**	0.60	**0.65**

Tabla 4.9: Desempeño de los clasificadores con el método propuesto comparado con el Baseline

al realizar esta selección. En esta tabla se aprecia claramente que en todos los casos, el método en dos etapas es superior en exactitud, exceptuando 3 casos donde la exactitud es la misma comparado con aplicar directamente un clasificador como Naive Bayes o SVM. El valor de exactitud nos indica que se superó el número de documentos correctamente clasificados del total. Aún y cuando el valor del baseline es alto para la tarea de clasificación temática en Desastres Naturales, el método propuesto logra aumentar este valor.

Para el caso de Poemas, el método logra hacer una caracterización de los autores, si bien no es la ideal, es considerable ya que se está tomando sólo las palabras para hacer la distinción entre los autores y se logra una mejor exactitud que con el método tradicional.

La mejora más grande en exactitud ocurrió para el corpus de 20 Newsgroups, utilizando Naive Bayes para clasificar en la segunda etapa se logra tener una mejor exactitud que inclusive la lograda en el baseline de SVM. R(52) si bien no mejora la exactitud con SVM aumenta el recuerdo sin perder mucha precisión (Tabla 4.15), ademas de llegar al límite de clasificación posible de alcanzar en la segunda etapa segun lo visto en la Tabla 4.8.

Naive Bayes			
Clases	π	ρ	F_1
acq	0.96	0.79	0.87
crude	0.73	0.71	0.72
earn	0.87	0.96	0.91
grain	0.25	0.50	0.33
interest	0.69	0.45	0.55
money-fx	0.67	0.63	0.65
ship	0.45	0.63	0.52
trade	0.68	0.85	0.75
Macro Promedio	0.66	0.69	0.66
Exactitud 0.85			

SVM			
Clases	π	ρ	F_1
acq	0.89	0.96	0.92
crude	0.91	0.85	0.88
earn	0.97	0.99	0.98
grain	0.57	0.40	0.47
interest	0.83	0.62	0.71
money-fx	0.78	0.50	0.61
ship	0.73	0.52	0.61
trade	0.86	0.80	0.83
Macro Promedio	0.82	0.70	0.75
Exactitud 0.92			

Tabla 4.10: **Baseline**: Exactitudes para R(8)

Naive Bayes			
Clases	π	ρ	F_1
acq	0.97	0.89	0.93
crude	0.69	0.74	0.72
earn	0.95	0.96	0.96
grain	0.36	0.40	0.38
interest	0.76	0.71	0.73
money-fx	0.69	0.82	0.75
ship	0.59	0.72	0.65
trade	0.70	0.90	0.79
Macro Promedio	0.71	0.77	0.74
Exactitud 0.90			

SVM			
Clases	π	ρ	F_1
acq	0.97	0.95	0.96
crude	0.82	0.86	0.84
earn	0.98	0.98	0.98
grain	0.50	0.50	0.50
interest	0.76	0.79	0.77
money-fx	0.74	0.67	0.71
ship	0.72	0.50	0.59
trade	0.68	0.92	0.78
Macro Promedio	0.77	0.77	0.76
Exactitud 0.94			

Tabla 4.11: **Método Propuesto**: Exactitudes para R(8).

Naive Bayes							
Clases	π	ρ	F_1	Clases	π	ρ	F_1
acq	0.96	0.82	0.88	jobs	0.91	0.91	0.91
alum	0.44	0.42	0.43	lead	0.00	0.00	0.00
bop	0.35	0.77	0.48	lei	0.60	1.00	0.75
carcass	0.00	0.00	0.00	livestock	0.25	0.40	0.30
cocoa	0.61	0.53	0.57	lumber	0.00	0.00	0.00
coffee	0.76	0.86	0.80	meal-feed	0.00	0.00	0.00
copper	0.61	0.61	0.61	money-fx	0.68	0.58	0.63
cotton	0.60	0.66	0.63	money-supply	0.68	0.60	0.64
cpi	0.71	0.58	0.64	nat-gas	0.25	0.66	0.36
cpu	0.33	1.00	0.50	nickel	0.00	0.00	0.00
crude	0.74	0.59	0.66	orange	0.42	0.66	0.52
dlr	0.04	0.33	0.07	pet-chem	0.12	0.16	0.14
earn	0.98	0.96	0.97	platinum	0.00	0.00	0.00
fuel	1.00	0.28	0.44	potato	0.33	0.66	0.44
gas	0.11	0.12	0.11	reserves	0.66	0.50	0.57
gnp	0.46	0.86	0.60	retail	1.00	1.00	1.00
gold	0.70	0.85	0.77	rubber	0.88	0.88	0.88
grain	0.26	0.40	0.32	ship	0.56	0.58	0.57
heat	0.10	0.75	0.18	strategic-metal	0.00	0.00	0.00
housing	0.13	1.00	0.23	sugar	0.76	0.80	0.78
income	0.75	0.75	0.75	tea	0.00	0.00	0.00
instal-debt	1.00	1.00	1.00	tin	0.38	0.50	0.43
interest	0.78	0.49	0.60	trade	0.64	0.70	0.67
ipi	0.83	0.45	0.58	veg-oil	0.60	0.81	0.69
iron-steel	0.53	0.66	0.59	wpi	0.71	0.55	0.62
jet	0.00	0.00	0.00	zinc	0.50	0.20	0.28
				Macro Promedio	0.47	0.53	0.47
Exactitud 0.81							

Tabla 4.12: **Baseline Bayes**: Exactitudes para R(52)

SVM								
Clases	π	ρ	F_1		Clases	π	ρ	F_1
acq	0.78	0.96	0.864		jobs	1.00	0.75	0.85
alum	0.85	0.31	0.462		lead	0.00	0.00	0.00
bop	1.00	0.55	0.714		lei	1.00	0.66	0.80
carcass	0.00	0.00	0.000		livestock	1.00	0.60	0.75
cocoa	1.00	0.53	0.696		lumber	0.00	0.00	0.00
coffee	0.95	0.86	0.905		meal-feed	0.00	0.00	0.00
copper	0.90	0.69	0.783		money-fx	0.73	0.50	0.59
cotton	0.80	0.44	0.571		money-supply	0.71	0.71	0.71
cpi	0.70	0.70	0.706		nat-gas	0.75	0.50	0.60
cpu	0.00	0.00	0.000		nickel	0.00	0.00	0.00
crude	0.88	0.81	0.845		orange	1.00	0.55	0.71
dlr	0.00	0.00	0.000		pet-chem	0.00	0.00	0.00
earn	0.96	0.99	0.976		platinum	0.00	0.00	0.00
fuel	0.00	0.00	0.000		potato	0.00	0.00	0.00
gas	1.00	0.12	0.222		reserves	0.87	0.58	0.70
gnp	0.85	0.80	0.828		retail	1.00	1.00	1.00
gold	0.88	0.75	0.811		rubber	1.00	0.77	0.87
grain	0.33	0.40	0.364		ship	0.75	0.61	0.67
heat	0.00	0.00	0.000		strategic-metal	0.00	0.00	0.00
housing	1.00	0.50	0.667		sugar	0.85	0.72	0.78
income	0.00	0.00	0.000		tea	0.00	0.00	0.00
instal-debt	0.00	0.00	0.000		tin	1.00	0.80	0.88
interest	0.66	0.53	0.589		trade	0.76	0.88	0.82
ipi	0.72	0.72	0.727		veg-oil	1.00	0.81	0.90
iron-steel	0.85	0.50	0.632		wpi	1.00	0.33	0.50
jet	0.00	0.00	0.000		zinc	1.00	0.40	0.57
					Macro Promedio	0.58	0.43	0.48
Exactitud 0.86								

Tabla 4.13: **Baseline SVM**: Exactitudes para R(52)

Naive Bayes							
Clases	π	ρ	F_1	Clases	π	ρ	F_1
acq	0.97	0.86	0.91	jobs	1.00	0.91	0.95
alum	0.88	0.78	0.83	lead	0.00	0.00	0.00
bop	0.35	0.77	0.48	lei	1.00	1.00	1.00
carcass	0.00	0.00	0.00	livestock	0.15	0.40	0.22
cocoa	0.68	0.86	0.76	lumber	0.66	0.50	0.57
coffee	0.71	0.90	0.80	meal-feed	0.00	0.00	0.00
copper	0.80	0.92	0.85	money-fx	0.71	0.74	0.73
cotton	0.75	1.00	0.85	money-supply	0.79	0.82	0.80
cpi	0.85	0.70	0.77	nat-gas	0.39	0.91	0.55
cpu	0.14	1.00	0.25	nickel	0.00	0.00	0.00
crude	0.81	0.61	0.69	orange	1.00	0.88	0.94
dlr	0.11	0.33	0.16	pet-chem	0.66	0.66	0.66
earn	0.98	0.95	0.97	platinum	0.00	0.00	0.00
fuel	0.08	0.28	0.12	potato	1.00	0.66	0.80
gas	0.40	0.25	0.30	reserves	0.52	0.83	0.64
gnp	0.33	0.86	0.48	retail	0.25	1.00	0.40
gold	0.70	0.95	0.80	rubber	0.50	0.77	0.60
grain	0.55	0.50	0.52	ship	0.79	0.75	0.77
heat	0.10	0.75	0.18	strategic-metal	0.00	0.00	0.00
housing	0.50	1.00	0.66	sugar	0.94	0.72	0.81
income	0.80	1.00	0.88	tea	0.00	0.00	0.00
instal-debt	0.05	1.00	0.09	tin	0.53	0.70	0.60
interest	0.87	0.49	0.63	trade	0.67	0.82	0.74
ipi	0.63	0.63	0.63	veg-oil	0.72	0.72	0.72
iron-steel	0.76	0.83	0.80	wpi	1.00	0.66	0.80
jet	0.00	0.00	0.00	zinc	0.33	0.20	0.25
				Macro Promedio	0.53	0.63	0.54
Exactitud 0.85							

Tabla 4.14: **Método Propuesto Bayes**: Exactitudes para R(52)

SVM							
Clases	π	ρ	F_1	Clases	π	ρ	F_1
acq	0.97	0.89	0.93	jobs	1.00	0.91	0.95
alum	0.91	0.57	0.71	lead	0.20	0.25	0.22
bop	0.61	0.88	0.72	lei	1.00	0.66	0.80
carcass	0.40	0.40	0.40	livestock	0.22	0.40	0.28
cocoa	0.50	0.73	0.59	lumber	1.00	0.25	0.40
coffee	0.69	0.90	0.78	meal-feed	0.00	0.00	0.00
copper	0.84	0.84	0.84	money-fx	0.77	0.74	0.76
cotton	0.88	0.88	0.88	money-supply	0.70	0.85	0.77
cpi	0.88	0.88	0.88	nat-gas	0.42	0.66	0.51
cpu	0.00	0.00	0.00	nickel	0.00	0.00	0.00
crude	0.81	0.68	0.74	orange	1.00	0.77	0.87
dlr	0.12	0.33	0.18	pet-chem	0.80	0.66	0.72
earn	0.99	0.95	0.97	platinum	0.00	0.00	0.00
fuel	0.11	0.42	0.18	potato	1.00	1.00	1.00
gas	0.75	0.37	0.50	reserves	0.50	0.83	0.62
gnp	0.44	1.00	0.61	retail	0.25	1.00	0.40
gold	0.70	0.85	0.77	rubber	0.33	0.44	0.38
grain	0.57	0.80	0.66	ship	0.85	0.66	0.75
heat	0.11	0.50	0.19	strategic-metal	0.00	0.00	0.00
housing	0.50	1.00	0.66	sugar	0.84	0.84	0.84
income	0.75	0.75	0.75	tea	0.00	0.00	0.00
instal-debt	0.00	0.00	0.00	tin	0.60	0.60	0.60
interest	0.83	0.60	0.70	trade	0.69	0.92	0.78
ipi	0.60	0.81	0.69	veg-oil	0.87	0.63	0.73
iron-steel	0.76	0.83	0.80	wpi	1.00	0.77	0.87
jet	0.00	0.00	0.00	zinc	0.60	0.60	0.60
				Macro Promedio	0.56	0.60	0.56
Exactitud 0.86							

Tabla 4.15: **Método Propuesto SVM**: Exactitudes para R(52)

Naive Bayes			
Clases	π	ρ	F_1
alt.atheism	0.61	0.58	0.59
comp.graphics	0.46	0.26	0.33
comp.os.ms-windows.misc	0.42	0.53	0.47
comp.sys.ibm.pc.hardware	0.61	0.39	0.48
comp.sys.mac.hardware	0.60	0.47	0.53
comp.windows.x	0.68	0.38	0.49
misc.forsale	0.18	0.64	0.29
rec.autos	0.62	0.58	0.60
rec.motorcycles	0.57	0.76	0.65
rec.sport.baseball	0.75	0.72	0.73
rec.sport.hockey	0.90	0.70	0.79
sci.crypt	0.77	0.62	0.69
sci.electronics	0.44	0.40	0.42
sci.med	0.70	0.42	0.53
sci.space	0.72	0.56	0.63
soc.religion.christian	0.73	0.59	0.65
talk.politics.guns	0.60	0.67	0.63
talk.politics.mideast	0.85	0.66	0.74
talk.politics.misc	0.56	0.49	0.52
talk.religion.misc	0.36	0.41	0.38
Macro Promedio	0.61	0.54	0.56
Exactitud 0.55			

SVM			
Clases	π	ρ	F_1
alt.atheism	0.70	0.63	0.67
comp.graphics	0.22	0.76	0.34
comp.os.ms-windows.misc	0.78	0.65	0.71
comp.sys.ibm.pc.hardware	0.58	0.66	0.62
comp.sys.mac.hardware	0.80	0.61	0.69
comp.windows.x	0.67	0.54	0.60
misc.forsale	0.74	0.81	0.77
rec.autos	0.83	0.69	0.75
rec.motorcycles	0.93	0.79	0.85
rec.sport.baseball	0.93	0.78	0.85
rec.sport.hockey	0.94	0.78	0.86
sci.crypt	0.90	0.73	0.81
sci.electronics	0.52	0.57	0.55
sci.med	0.81	0.53	0.64
sci.space	0.86	0.67	0.75
soc.religion.christian	0.84	0.78	0.81
talk.politics.guns	0.68	0.75	0.71
talk.politics.mideast	0.96	0.65	0.77
talk.politics.misc	0.72	0.50	0.59
talk.religion.misc	0.65	0.53	0.58
Macro Promedio	0.75	0.67	0.70
Exactitud 0.68			

Tabla 4.16: **Baseline**: Exactitudes para 20 Newsgroups.

Naive Bayes			
Clases	π	ρ	F_1
alt.atheism	0.66	0.64	0.65
comp.graphics	0.64	0.60	0.62
comp.os.ms-windows.misc	0.58	0.71	0.64
comp.sys.ibm.pc.hardware	0.53	0.61	0.57
comp.sys.mac.hardware	0.74	0.72	0.73
comp.windows.x	0.70	0.52	0.60
misc.forsale	0.82	0.62	0.70
rec.autos	0.81	0.75	0.78
rec.motorcycles	0.82	0.83	0.83
rec.sport.baseball	0.77	0.87	0.82
rec.sport.hockey	0.85	0.78	0.81
sci.crypt	0.69	0.78	0.73
sci.electronics	0.76	0.52	0.62
sci.med	0.88	0.78	0.82
sci.space	0.79	0.78	0.79
soc.religion.christian	0.70	0.76	0.73
talk.politics.guns	0.57	0.86	0.69
talk.politics.mideast	0.73	0.79	0.76
talk.politics.misc	0.59	0.57	0.58
talk.religion.misc	0.55	0.56	0.56
Macro Promedio	0.71	0.70	0.70
Exactitud 0.71			

SVM			
Clases	π	ρ	F_1
alt.atheism	0.68	0.64	0.66
comp.graphics	0.71	0.67	0.69
comp.os.ms-windows.misc	0.63	0.66	0.65
comp.sys.ibm.pc.hardware	0.53	0.69	0.60
comp.sys.mac.hardware	0.78	0.67	0.72
comp.windows.x	0.72	0.69	0.71
misc.forsale	0.87	0.77	0.82
rec.autos	0.88	0.85	0.87
rec.motorcycles	0.89	0.89	0.89
rec.sport.baseball	0.83	0.90	0.86
rec.sport.hockey	0.83	0.84	0.84
sci.crypt	0.74	0.87	0.80
sci.electronics	0.81	0.56	0.67
sci.med	0.90	0.79	0.84
sci.space	0.83	0.82	0.83
soc.religion.christian	0.76	0.92	0.83
talk.politics.guns	0.65	0.85	0.74
talk.politics.mideast	0.80	0.80	0.80
talk.politics.misc	0.68	0.53	0.59
talk.religion.misc	0.69	0.61	0.65
Macro Promedio	0.76	0.75	0.75
Exactitud 0.76			

Tabla 4.17: **Método Propuesto**: Exactitudes para 20 Newsgroups.

Naive Bayes			
Clases	π	ρ	F_1
course	0.86	0.68	0.76
faculty	0.55	0.56	0.55
project	0.43	0.42	0.43
student	0.64	0.71	0.67
Macro Promedio	0.62	0.59	0.60
Exactitud 0.63			

SVM			
Clases	π	ρ	F_1
course	0.94	0.86	0.89
faculty	0.84	0.83	0.84
project	0.69	0.70	0.70
student	0.84	0.88	0.86
Macro Promedio	0.83	0.82	0.82
Exactitud 0.84			

Tabla 4.18: **Baseline**: Exactitudes para WebKB.

Naive Bayes			
Clases	π	ρ	F_1
course	0.79	0.76	0.77
faculty	0.57	0.74	0.64
project	0.58	0.48	0.53
student	0.82	0.71	0.76
Macro Promedio	0.69	0.67	0.68
Exactitud 0.70			

SVM			
Clases	π	ρ	F_1
course	0.85	0.90	0.87
faculty	0.80	0.86	0.83
project	0.73	0.65	0.69
student	0.90	0.86	0.88
Macro Promedio	0.82	0.82	0.82
Exactitud 0.84			

Tabla 4.19: **Método Propuesto**: Exactitudes para WebKB.

Naive Bayes			
Clases	π	ρ	F_1
Forestal	0.96	0.96	0.96
Huracan	0.83	0.92	0.87
Inundacion	0.81	0.82	0.82
Sequia	0.93	0.75	0.83
Sismo	0.95	0.94	0.94
Macro Promedio	0.90	0.88	0.89
Exactitud 0.90			

SVM			
Clases	π	ρ	F_1
Forestal	0.93	0.95	0.94
Huracan	0.96	0.78	0.87
Inundacion	0.84	0.93	0.88
Sequia	1.00	0.75	0.86
Sismo	0.91	0.99	0.95
Macro Promedio	0.93	0.88	0.90
Exactitud 0.92			

Tabla 4.20: **Baseline**: Exactitudes para Desastres Naturales.

Naive Bayes			
Clases	π	ρ	F_1
Forestal	0.96	0.97	0.97
Huracan	0.82	0.93	0.87
Inundacion	0.90	0.85	0.87
Sequia	0.83	0.87	0.85
Sismo	1.00	0.94	0.97
Macro Promedio	0.90	0.91	0.91
Exactitud 0.92			

SVM			
Clases	π	ρ	F_1
Forestal	0.92	1.00	0.96
Huracan	0.97	0.86	0.91
Inundacion	0.91	0.96	0.93
Sequia	0.97	0.82	0.89
Sismo	0.97	0.99	0.98
Macro Promedio	0.95	0.93	0.94
Exactitud 0.95			

Tabla 4.21: **Método Propuesto**: Exactitudes para Desastres Naturales.

Naive Bayes			
Clases	π	ρ	F_1
Efraín Huerta	0.53	0.66	0.59
Jaime Sabines	0.66	0.52	0.58
Octavio Paz	0.72	0.53	0.61
Rosario Castellanos	0.52	0.70	0.59
Rubén Bonifaz	0.58	0.57	0.58
Macro Promedio	0.60	0.59	0.59
Exactitud 0.59			

SVM			
Clases	π	ρ	F_1
Efraín Huerta	0.76	0.47	0.59
Jaime Sabines	0.48	0.73	0.58
Octavio Paz	0.70	0.56	0.62
Rosario Castellanos	0.55	0.70	0.61
Rubén Bonifaz	0.75	0.42	0.54
Macro Promedio	0.65	0.58	0.59
Exactitud 0.60			

Tabla 4.22: **Baseline**: Exactitudes para Poemas.

Naive Bayes			
Clases	π	ρ	F_1
Efraín Huerta	0.36	0.66	0.47
Jaime Sabines	0.77	0.58	0.66
Octavio Paz	0.61	0.66	0.63
Rosario Castellanos	0.70	0.41	0.52
Rubén Bonifaz	0.60	0.64	0.62
Macro Promedio	0.60	0.59	0.58
Exactitud 0.59			

SVM			
Clases	π	ρ	F_1
Efraín Huerta	0.39	0.60	0.47
Jaime Sabines	0.74	0.72	0.73
Octavio Paz	0.63	0.84	0.72
Rosario Castellanos	0.80	0.46	0.58
Rubén Bonifaz	0.75	0.61	0.67
Macro Promedio	0.66	0.64	0.64
Exactitud 0.65			

Tabla 4.23: **Método Propuesto**: Exactitudes para Poemas.

Capítulo 5

Conclusiones

Se presentó un método de clasificación automática de textos de dos etapas. Una primera etapa busca reducir el problema inicial de clasificación multi-clase a un problema de clasificación binaria. Este problema reducido permitió al clasificador en una segunda etapa distinguir de una manera más sencilla, las características que distinguen entre si a las clases, de esta manera entrenarse para lograr una mejor exactitud al clasificar nuevos documentos. El proceso de reducción usa un método basado en prototipos para encontrar las dos clases más similares al documento.

Concretamente, como resultado de los experimentos realizados, el presente trabajo de tesis aporta:

- Un método de Clasificación Automática de Textos usando reducción de clases basada en prototipos. Este método resultó tener mejor desempeño en la gran mayoría de los casos y para el resto de casos un desempeño igual que el de aplicar la clasificación de la manera tradicional.

- Un procedimiento para el cálculo de prototipos de las clases, que a diferencia de otros métodos, da un valor a las palabra de acuerdo al valor que ésta tenga

para cada clase.

- Una medida de similitud basada en una intersección pesada de palabras, que junto con el cálculo propuesto de prototipo llegan a tener desempeños comparables al mejor método de clasificación basado en prototipos.

5.1. Trabajo Futuro

En el método propuesto, se eligieron solamente las dos clases más similares al documento, queda como trabajo futuro una configuración donde se fije un umbral de similitud y en base a éste se definan con cuántas clases quedarnos, para posteriormente hacer la clasificación.

Por otro lado, se propone usar más de un prototipo por clase en vez de representar a la clase por un sólo prototipo, donde la fase de reducción se vea como un problema de vecinos más cercanos con los prototipos.

Además, también se desea comprobar el comportamiento del método con otro tipo de atributos diferentes a la bolsa de palabras en la etapa de clasificación, como por ejemplo n-gramas o secuencias frecuentes maximales.

Por último, también es de interés experimentar con problemas involucrando decenas de clases, así como en el otro extremo, probar su desempeño usando a la intersección pesada como método de clasificación binaria.

Referencias

Berry, M. W., *Survey of Text Mining: Clustering, Classification, and Retrieval*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2003.

Burges, C. J. C., A tutorial on support vector machines for pattern recognition, *Data Mining and Knowledge Discovery*, *2*(2), 121–167, 1998.

Cardoso-Cachopo, A., Improving methods for single-label text categorization, Ph.D. thesis, ITS Universidade Técnica de Lisboa, Lisboa, Portugal, 2007.

Cardoso-Cachopo, A., and A. L. Oliveira, Empirical evaluation of centroid-based models for single-label text categorization, *Tech. Rep. 7/2006*, INESC-ID, Lisboa, Portugal, 2006.

Chuang, W. T., A. Tiyyagura, J. Yang, and G. Giuffrida, A fast algorithm for hierarchical text classification, in *Lecture Notes In Computer Science, Proceedings of the Second International Conference on Data Warehousing and Knowledge Discovery*, vol. 1874, pp. 409–418, Springer-Verlag, London, UK, 2000.

Cortes, C., and V. Vapnik, Support-vector networks, *Machine Learning*, *20*(3), 273–297, 1995.

Cover, T., and P. Hart, Nearest neighbor pattern classification, *IEEE Transactions on Information Theory*, *13*(1), 21–27, 1967.

Coyotl, R., Clasificación automática de textos considerando el estilo de redacción, Master's thesis, Ciencias Computacionales, Instituto Nacional de Astrofísica Óptica y Electrónica, Puebla, México, 2007.

Cristianini, N., and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*, Cambridge University Press, 2000.

Debole, F., and F. Sebastiani, Supervised term weighting for automated text categorization, in *Proceedings of the 2003 ACM Symposium on Applied Computing*, pp. 784–788, ACM, New York, US, 2003.

Debole, F., and F. Sebastiani, An analysis of the relative hardness of reuters-21578 subsets, *Journal of the American Society for Information Science and Technology*, *56*(6), 584–596, 2005.

Fuhr, N., and U. Pfeifer, Probabilistic information retrieval as a combination of abstraction, inductive learning and probabilistic assumptions, *ACM Transactions on Information Systems*, *12*(1), 92–115, 1994.

Han, E.-H., and G. Karypis, Centroid-based document classification: Analysis and experimental results, in *Lecture Notes In Computer Science, Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery*, vol. 1910, pp. 424–431, Springer-Verlag, London, UK, 2000.

Han, E.-H., G. Karypis, and V. Kumar, Text categorization using weight adjusted k -nearest neighbor classification, in *Lecture Notes in Computer Science Pro-*

ceedings of the 5th Pacific-Asia Conference on Knowledge Discovery and Data Mining, vol. 2035, pp. 53–65, Springer-Verlag, London, UK, 2001.

Henzinger, M., The past, present and future of web information retrieval, in *Proceedings of the 23th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pp. 46–46, ACM, New York, NY, USA, 2004.

Hsu, C.-W., and C.-J. Lin, A comparison of methods for multi-class support vector machines, *IEEE Transactions on Neural Networks*, *13*(2), 415–425, 2002.

Hull, D., Improving text retrieval for the routing problem using latent semantic indexing, in *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 282–291, Springer-Verlag New York, Inc., New York, NY, USA, 1994.

Joachims, T., A probabilistic analysis of the rocchio algorithm with tfidf for text categorization, in *Proceedings of the 14th International Conference on Machine Learning*, pp. 143–151, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997.

Joachims, T., Text categorization with support vector machines: Learning with many relevant features, in *Lecture Notes In Computer Science, Proceedings 10th European Conference on Machine Learning*, 1398, pp. 137–142, Springer-Verlag, London, UK, 1998.

Joachims, T., Transductive inference for text classification using support vector machines, in *Proceedings of the 16th International Conference on Machine Learning*, pp. 200–209, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999.

Joachims, T., *Learning to Classify Text Using Support Vector Machines: Methods, Theory and Algorithms*, 224 pp., Kluwer Academic Publishers, Norwell, MA, USA, 2002.

Jurafsky, D., and J. H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*, 1 ed., 934 pp., Prentice Hall, 2000.

Kim, H., P. Howland, and H. Park, Dimension reduction in text classification with support vector machines, *The Journal of Machine Learning Research*, *6*, 37–53, 2005.

Koller, D., and M. Sahami, Hierarchically classifying documents using very few words, in *Proceedings of the 14th International Conference on Machine Learning*, pp. 170–178, Morgan Kaufmann Publishers, San Francisco, US, 1997.

Lang, K., Newsweeder: Learning to filter netnews, in *Proceedings of the Twelfth International Conference on Machine Learning*, pp. 331–339, 1995.

Lee, C., and G. G. Lee, Information gain and divergence-based feature selection for machine learning-based text categorization, *Information Processing and Management: an International Journal*, *42*(1), 155–165, 2006.

Lertnattee, V., and T. Theeramunkong, Effect of term distributions on centroid-based text categorization, *Information Sciences—Informatics and Computer Science: An International Journal*, *158*(1), 89–115, 2004.

Lewis, D. D., Evaluating text categorization, in *Human Language Technology Conference, Proceedings of the workshop on Speech and Natural Language*, pp. 312–318, Association for Computational Linguistics, Morristown, NJ, USA, 1991.

Lewis, D. D., Naive (bayes) at forty: The independence assumption in information retrieval, in *Lecture Notes In Computer Science, Proceedings of the 10th European Conference on Machine Learning*, pp. 4–15, Springer-Verlag, London, UK, 1998.

Lewis, D. D., and M. Ringuette, A comparison of two learning algorithms for text categorization, in *Proceedings of 3rd Annual Symposium on Document Analysis and Information Retrieval*, pp. 81–93, Las Vegas, US, 1994.

Liu, T.-Y., Y. Yang, H. Wan, Q. Zhou, B. Gao, H.-J. Zeng, Z. Chen, and W.-Y. Ma, An experimental study on large-scale web categorization, in *In Posters Proceedings of the 14th International World Wide Web Conference*, pp. 1106–1107, ACM, New York, NY, USA, 2005.

López, A., M. Montes, L. Villaseñor, M. Pérez, and A. Téllez, Gestión automática de información de desastres naturales en méxico, IX Jornadas Iberoamericanas de Informática, 2003.

Masand, B., G. Linoff, and D. Waltz, Classifying news stories using memory based reasoning, in *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 59–65, ACM, New York, NY, USA, 1992.

Mitchell, T. M., *Machine Learning*, 432 pp., McGraw-Hill, New York, 1997.

Montes, M., L. Villaseñor, and A. López, Mexican experience in spanish question answering, *Computación y Sistemas*, *12*(1), 2008.

Nigam, K., A. K. McCallum, S. Thrun, and T. M. Mitchell, Text classification

from labeled and unlabeled documents using EM, *Machine Learning*, *39*(2/3), 103–134, 2000.

Porter, M. F., An algorithm for suffix stripping, in *Readings in information retrieval*, pp. 313–316, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997.

Salton, G., and C. Buckley, Term weighting approaches in automatic text retrieval, *Tech. Rep. TR87-881*, Cornell University, Ithaca, NY, USA, 1987.

Scholköpf, B., and A. J. Smola, A short introduction to learning with kernels, *Lecture Notes In Artificial Intelligence, Advanced Lectures on Machine Learning*, pp. 41–64, 2003.

Sebastiani, F., Machine learning in automated text categorization, *ACM Computing Surveys*, *34*(1), 1–47, 2002.

Sebastiani, F., Text categorization, in *The Encyclopedia of Database Technologies and Applications*, pp. 683–687, Idea Group Publishing, Hershey, US, 2005a.

Sebastiani, F., Text categorization, in *Text Mining and its Applications to Intelligence, CRM and Knowledge Management*, pp. 109–129, WIT Press, Southampton, UK, 2005b.

Sebastiani, F., Classification of text, automatic, in *The Encyclopedia of Language and Linguistics*, vol. 2, second ed., pp. 457–463, Elsevier Science Publishers, Amsterdam, NL, 2006.

Shankar, S., and G. Karypis, Weight adjustment schemes for a centroid based classifier, *Tech. Rep. TR00-035*, Department of Computer Science, University of Minnesota, Minneapolis, Minnesota, 2000.

Shannon, C. E., and W. Weaver, *A Mathematical Theory of Communication*, University of Illinois Press, Champaign, IL, USA, 1963.

Tan, P.-N., M. Steinbach, and V. Kumar, *Introduction to Data Mining*, 1 ed., Addison Wesley, 2005a.

Tan, S., X. Cheng, M. M. Ghanem, B. Wang, and H. Xu, A novel refinement approach for text categorization, in *Proceedings of the 14th ACM International Conference on Information and Mnowledge Management*, pp. 469–476, ACM, New York, NY, USA, 2005b.

Téllez, A., Extracción de información con algoritmos de clasificación, Master's thesis, Ciencias Computacionales, Instituto Nacional de Astrofísica Óptica y Electrónica, Puebla, México, 2005.

Vapnik, V. N., *The Nature of Statistical Learning Theory*, Springer-Verlag New York, Inc., New York, NY, USA, 1995.

Witten, I. H., and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition (Morgan Kaufmann Series in Data Management Systems)*, Morgan Kaufmann, 2005.

Yang, Y., and X. Liu, A re-examination of text categorization methods, in *Proceedings of the 22nd annual international ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 42–49, ACM, New York, NY, USA, 1999.

Yang, Y., and J. O. Pedersen, A comparative study on feature selection in text categorization, in *Proceedings of the 14th International Conference on Machine*

Learnin, pp. 412–420, Morgan Kaufmann Publishers Inc., San Francisco, US, 1997.

Apéndice A

Intersección Pesada como Método de Clasificación

En el método propuesto se usa el prototipo calculado como la Intersección Pesada para la reducción de clases, pero es interesante ver el desempeño que tiene si se usa como un clasificador. La clasificación de un documento nuevo se hace en base a la similitud con los prototipos de las clases y la clase se asigna de acuerdo a la clase del prototipo con la similitud mayor. Los resultados de este experimento se muestran en la Tabla A.1.

En el caso de la clasificación por autoría, la Intersección Pesada como método

Exactitud					
Corpus	Bayes Baseline	SVM Baseline	Bayes 2 Etapas	SVM 2 Etapas	Intersección Pesada
R(8)	0.85	0.92	0.91	**0.94**	0.93
Desastres	0.90	0.92	0.92	0.95	**0.97**
Poemas	0.59	0.60	0.60	**0.65**	0.60

Tabla A.1: Desempeño del esquema de Intersección Pesada como clasificador.

de clasificación tiene un desempeño igual al obtenido con el baseline. Por lo que respecta a la clasificación temática, la Intersección Pesada supera al método en dos etapas usando Naive Bayes, pero en el caso de la clasificación en dos etapas con SVM, la Intersección Pesada en el corpus de Desastres Naturales puede considerar que es comparable con este método.

Sin embargo, esta comparación no es del todo "justa", ya que la Intersección Pesada se trata de un método basado en prototipos, lo ideal será compararlo con métodos de este tipo. Según el estudio hecho por Cardoso [*Cardoso-Cachopo and Oliveira*, 2006], el mejor método de clasificación basado en prototipos (centroides) es la Suma Normalizada por lo que una comparación con ese método es más adecuada.

La Tabla A.2 muestra que ambos métodos tienen un comportamiento muy similar en exactitud, del orden de 0.01 de diferencia entre ambos clasificadores, esto para la clasificación temática, que son R(8) y Desastres Naturales. Para el caso de Poemas el esquema propuesto resulta por debajo del desempeño de la Suma Normalizada, esto se debe a que nuestro método por la manera en que son seleccionados los atributos de cada clase, produce una reducción del vocabulario parte clave, ya que, para la clasificación no temática en particular en cuestión de estilo es necesario mayor número de atributos (palabras) para hacer la distinción entre las clases.

Una prueba más es realizada para observar el desempeño de la Intersección Pesada frente a la Suma Normalizada. En este caso, se usa la Suma Normalizada para hacer la reducción de clases en la primera etapa del método propuesto. Como clasificador se usa SVM por ser el que mejor exactitud ha tenido y como corpus de prueba R(8).

La Tabla A.3 muestra el desempeño en comparación con el obtenido con la In-

Exactitud			
Corpus	SVM 2 Etapas	Intersección Pesada	Suma Normalizada
R(8)	**0.95**	0.93	0.94
Desastres	0.95	**0.97**	0.96
Poemas	0.65	0.60	**0.69**

Tabla A.2: Desempeño de los métodos basados en prototipos.

R(8)		
Corpus	Intersección Pesada	Suma Normalizada
Exactitud	0.95	0.95

Tabla A.3: Comparación de los métodos basados en prototipos para la reducción de clases usando SVM como clasificador en R(8)

tersección Pesada frente a la Suma Normalizada. Se presenta un comportamiento igual entre ambos métodos de prototipos.

Aún y cuando se dió un caso de que el método por Intersección Pesada (usado como clasificador), fue superior (en 0.02 de exactitud) en clasificación temática al método propuesto de reducción de clases, una ventaja de éste último sobre los métodos basados en prototipos es que, se pueden hacer uso de atributos adicionales a la bolsa de palabras en la segunda etapa (por ejemplo secuencias frecuentes maximales), donde el problema tiene solo dos clases y por lo tanto es más sencillo de obtener y manejar esta información. Si los métodos basados en prototipos quisieran usar los mismos atributos sería más difícil extraerlos del problema inicial multi-clase donde el volumen de datos es mucho mayor.

Apéndice B

Experimentos y Resultados Adicionales

Adicionalmente a los experimentos reportados en la tesis, se hicieron pruebas con dos variaciones de tres conjuntos de datos: R(8), R(52) y 20Newsgroups. En la primera variación, llamada *no-short*, se eliminan todas las palabras de longitud menor a 3 caracteres; la segunda de las variaciones, *stemmed* se busca representar distintas formas morfológicas con un solo atributo en común, para lograr esta representación se usa el algoritmo de Martin Porter [*Porter*, 1997]. Los conjuntos de datos utilizados en esta sección fueron construidos por Cardoso [*Cardoso-Cachopo*, 2007].

Naive Bayes			
Clases	π	ρ	F_1
acq	0.96	0.77	0.85
crude	0.72	0.71	0.72
earn	0.86	0.96	0.91
grain	0.25	0.50	0.33
interest	0.68	0.45	0.54
money-fx	0.67	0.63	0.65
ship	0.45	0.63	0.52
trade	0.67	0.86	0.76
Macro Promedio	0.66	0.69	0.66
Exactitud 0.84			

SVM			
Clases	π	ρ	F_1
acq	0.89	0.96	0.93
crude	0.90	0.84	0.87
earn	0.97	0.99	0.98
grain	0.57	0.40	0.47
interest	0.85	0.63	0.72
money-fx	0.77	0.50	0.61
ship	0.67	0.52	0.59
trade	0.87	0.80	0.83
Macro Promedio	0.81	0.70	0.75
Exactitud 0.92			

Tabla B.1: **Baseline**: Exactitudes para R(8) no-short.

Naive Bayes			
Clases	π	ρ	F_1
acq	0.97	0.88	0.92
crude	0.69	0.74	0.71
earn	0.95	0.96	0.96
grain	0.40	0.40	0.40
interest	0.75	0.71	0.73
money-fx	0.67	0.82	0.74
ship	0.56	0.72	0.63
trade	0.70	0.90	0.79
Macro Promedio	0.71	0.77	0.73
Exactitud 0.90			

SVM			
Clases	π	ρ	F_1
acq	0.97	0.95	0.96
crude	0.84	0.86	0.85
earn	0.98	0.98	0.98
grain	0.63	0.70	0.66
interest	0.75	0.77	0.76
money-fx	0.70	0.64	0.67
ship	0.73	0.52	0.61
trade	0.67	0.92	0.77
Macro Promedio	0.78	0.79	0.78
Exactitud 0.93			

Tabla B.2: **Método Propuesto**: Exactitudes para R(8) no-short.

Naive Bayes			
Clases	π	ρ	F_1
acq	0.95	0.76	0.84
crude	0.70	0.74	0.72
earn	0.86	0.95	0.90
grain	0.22	0.50	0.31
interest	0.75	0.37	0.49
money-fx	0.58	0.65	0.62
ship	0.47	0.66	0.55
trade	0.61	0.85	0.71
Macro Promedio	0.64	0.68	0.64
Exactitud 0.83			

SVM			
Clases	π	ρ	F_1
acq	0.89	0.97	0.93
crude	0.95	0.83	0.89
earn	0.97	0.99	0.98
grain	0.66	0.40	0.50
interest	0.83	0.54	0.65
money-fx	0.77	0.51	0.62
ship	0.80	0.66	0.72
trade	0.84	0.85	0.84
Macro Promedio	0.84	0.72	0.77
Exactitud 0.92			

Tabla B.3: **Baseline**: Exactitudes para R(8) stemmed.

Naive Bayes			
Clases	π	ρ	F_1
acq	0.96	0.86	0.91
crude	0.66	0.76	0.71
earn	0.95	0.95	0.95
grain	0.42	0.60	0.50
interest	0.77	0.63	0.69
money-fx	0.64	0.80	0.71
ship	0.54	0.72	0.61
trade	0.60	0.82	0.69
Macro Promedio	0.69	0.77	0.72
Exactitud 0.88			

SVM			
Clases	π	ρ	F_1
acq	0.97	0.93	0.95
crude	0.77	0.86	0.81
earn	0.98	0.98	0.98
grain	0.57	0.80	0.66
interest	0.74	0.69	0.71
money-fx	0.73	0.75	0.74
ship	0.69	0.55	0.61
trade	0.67	0.89	0.77
Macro Promedio	0.76	0.80	0.78
Exactitud 0.92			

Tabla B.4: **Método Propuesto**: Exactitudes para R(8) stemmed.

Naive Bayes							
Clases	π	ρ	F_1	Clases	π	ρ	F_1
acq	0.96	0.82	0.88	jobs	0.91	0.91	0.91
alum	0.50	0.36	0.42	lead	0.00	0.00	0.00
bop	0.33	0.77	0.46	lei	0.60	1.00	0.75
carcass	0.00	0.00	0.00	livestock	0.28	0.40	0.33
cocoa	0.66	0.53	0.59	lumber	0.00	0.00	0.00
coffee	0.79	0.86	0.82	meal-feed	0.00	0.00	0.00
copper	0.57	0.61	0.59	money-fx	0.68	0.57	0.62
cotton	0.60	0.66	0.63	money-supply	0.68	0.60	0.64
cpi	0.71	0.58	0.64	nat-gas	0.26	0.66	0.38
cpu	0.33	1.00	0.50	nickel	0.00	0.00	0.00
crude	0.74	0.59	0.66	orange	0.50	0.66	0.57
dlr	0.04	0.33	0.08	pet-chem	0.10	0.16	0.12
earn	0.97	0.95	0.96	platinum	0.00	0.00	0.00
fuel	1.00	0.28	0.44	potato	0.33	0.66	0.44
gas	0.11	0.12	0.11	reserves	0.60	0.50	0.54
gnp	0.50	0.86	0.63	retail	1.00	1.00	1.00
gold	0.70	0.85	0.77	rubber	0.88	0.88	0.88
grain	0.25	0.40	0.30	ship	0.59	0.61	0.60
heat	0.10	0.75	0.18	strategic-metal	0.00	0.00	0.00
housing	0.15	1.00	0.26	sugar	0.73	0.76	0.74
income	0.60	0.75	0.66	tea	0.00	0.00	0.00
instal-debt	1.00	1.00	1.00	tin	0.42	0.60	0.50
interest	0.78	0.50	0.61	trade	0.64	0.73	0.68
ipi	0.83	0.45	0.58	veg-oil	0.69	0.81	0.75
iron-steel	0.42	0.66	0.51	wpi	0.71	0.55	0.62
jet	0.00	0.00	0.00	zinc	0.50	0.20	0.28
				Macro Promedio	0.47	0.54	0.47
Exactitud 0.81							

Tabla B.5: **Baseline Bayes**: Exactitudes para R(52) no-short

SVM							
Clases	π	ρ	F_1	Clases	π	ρ	F_1
acq	0.78	0.96	0.86	jobs	1.00	0.75	0.85
alum	1.00	0.31	0.48	lead	0.00	0.00	0.00
bop	1.00	0.55	0.71	lei	1.00	0.66	0.80
carcass	0.00	0.00	0.00	livestock	1.00	0.60	0.75
cocoa	1.00	0.53	0.69	lumber	0.00	0.00	0.00
coffee	0.95	0.86	0.90	meal-feed	0.00	0.00	0.00
copper	0.90	0.69	0.78	money-fx	0.72	0.50	0.59
cotton	0.75	0.33	0.46	money-supply	0.71	0.71	0.71
cpi	0.68	0.64	0.66	nat-gas	0.75	0.50	0.60
cpu	0.00	0.00	0.00	nickel	0.00	0.00	0.00
crude	0.88	0.81	0.84	orange	1.00	0.55	0.71
dlr	0.00	0.00	0.00	pet-chem	0.00	0.00	0.00
earn	0.96	0.98	0.97	platinum	0.00	0.00	0.00
fuel	0.00	0.00	0.00	potato	0.00	0.00	0.00
gas	1.00	0.12	0.22	reserves	0.87	0.58	0.70
gnp	0.80	0.80	0.80	retail	1.00	1.00	1.00
gold	0.88	0.75	0.81	rubber	1.00	0.77	0.87
grain	0.33	0.40	0.36	ship	0.75	0.61	0.67
heat	0.00	0.00	0.00	strategic-metal	0.00	0.00	0.00
housing	1.00	0.50	0.66	sugar	0.81	0.72	0.76
income	0.00	0.00	0.00	tea	0.00	0.00	0.00
instal-debt	0.00	0.00	0.00	tin	1.00	0.80	0.88
interest	0.67	0.53	0.59	trade	0.76	0.88	0.82
ipi	0.72	0.72	0.72	veg-oil	1.00	0.81	0.90
iron-steel	0.85	0.50	0.63	wpi	1.00	0.33	0.50
jet	0.00	0.00	0.00	zinc	1.00	0.40	0.57
				Macro Promedio	0.58	0.42	0.48
Exactitud 0.86							

Tabla B.6: **Baseline SVM**: Exactitudes para R(52) no-short

Naive Bayes							
Clases	π	ρ	F_1	Clases	π	ρ	F_1
acq	0.97	0.86	0.91	jobs	1.00	0.91	0.95
alum	0.88	0.78	0.83	lead	0.37	0.75	0.50
bop	0.31	0.77	0.45	lei	1.00	1.00	1.00
carcass	0.00	0.00	0.00	livestock	0.15	0.40	0.22
cocoa	0.72	0.86	0.78	lumber	0.66	0.50	0.57
coffee	0.69	0.90	0.78	meal-feed	0.00	0.00	0.00
copper	0.80	0.92	0.85	money-fx	0.69	0.74	0.72
cotton	0.75	1.00	0.85	money-supply	0.79	0.82	0.80
cpi	0.85	0.70	0.77	nat-gas	0.42	0.91	0.57
cpu	0.14	1.00	0.25	nickel	0.00	0.00	0.00
crude	0.81	0.60	0.69	orange	1.00	0.88	0.94
dlr	0.11	0.33	0.16	pet-chem	0.66	0.66	0.66
earn	0.98	0.94	0.96	platinum	0.00	0.00	0.00
fuel	0.08	0.28	0.12	potato	1.00	0.66	0.80
gas	0.42	0.37	0.40	reserves	0.37	0.75	0.50
gnp	0.33	0.86	0.48	retail	0.25	1.00	0.40
gold	0.70	0.95	0.80	rubber	0.50	0.77	0.60
grain	0.60	0.60	0.60	ship	0.79	0.75	0.77
heat	0.12	0.75	0.21	strategic-metal	0.00	0.00	0.00
housing	0.50	1.00	0.66	sugar	0.94	0.72	0.81
income	0.80	1.00	0.88	tea	0.00	0.00	0.00
instal-debt	0.05	1.00	0.09	tin	0.57	0.80	0.66
interest	0.87	0.49	0.63	trade	0.67	0.81	0.73
ipi	0.63	0.63	0.63	veg-oil	0.72	0.72	0.72
iron-steel	0.76	0.83	0.80	wpi	1.00	0.66	0.80
jet	0.00	0.00	0.00	zinc	0.33	0.20	0.25
				Macro Promedio	0.53	0.65	0.55
Exactitud 0.84							

Tabla B.7: **Método Propuesto Bayes**: Exactitudes para R(52) no-short

SVM							
Clases	π	ρ	F_1	Clases	π	ρ	F_1
acq	0.97	0.89	0.93	jobs	1.00	0.91	0.95
alum	0.91	0.57	0.71	lead	0.50	1.00	0.66
bop	0.57	0.88	0.69	lei	1.00	0.66	0.80
carcass	0.40	0.40	0.40	livestock	0.22	0.40	0.28
cocoa	0.50	0.73	0.59	lumber	1.00	0.25	0.40
coffee	0.66	0.90	0.76	meal-feed	0.00	0.00	0.00
copper	0.84	0.84	0.84	money-fx	0.76	0.74	0.75
cotton	0.88	0.88	0.88	money-supply	0.72	0.85	0.78
cpi	0.88	0.88	0.88	nat-gas	0.44	0.66	0.53
cpu	0.00	0.00	0.00	nickel	0.00	0.00	0.00
crude	0.81	0.67	0.73	orange	1.00	0.77	0.87
dlr	0.12	0.33	0.18	pet-chem	0.80	0.66	0.72
earn	0.99	0.95	0.97	platinum	0.00	0.00	0.00
fuel	0.11	0.42	0.18	potato	1.00	0.66	0.80
gas	0.71	0.62	0.66	reserves	0.40	0.83	0.54
gnp	0.40	0.86	0.55	retail	0.25	1.00	0.40
gold	0.70	0.85	0.77	rubber	0.33	0.44	0.38
grain	0.56	0.90	0.69	ship	0.85	0.66	0.75
heat	0.15	0.50	0.23	strategic-metal	0.00	0.00	0.00
housing	0.50	1.00	0.66	sugar	0.84	0.84	0.84
income	0.75	0.75	0.75	tea	0.00	0.00	0.00
instal-debt	0.00	0.00	0.00	tin	0.60	0.60	0.60
interest	0.83	0.60	0.70	trade	0.67	0.92	0.78
ipi	0.60	0.81	0.69	veg-oil	0.87	0.63	0.73
iron-steel	0.76	0.83	0.80	wpi	1.00	0.77	0.87
jet	0.00	0.00	0.00	zinc	0.75	0.60	0.66
				Macro Promedio	0.57	0.61	0.56
Exactitud 0.86							

Tabla B.8: **Método Propuesto SVM**: Exactitudes para R(52) no-short

Naive Bayes							
Clases	π	ρ	F_1	Clases	π	ρ	F_1
acq	0.96	0.79	0.87	jobs	1.00	0.91	0.95
alum	0.57	0.42	0.48	lead	0.75	0.75	0.75
bop	0.30	0.66	0.41	lei	1.00	1.00	1.00
carcass	0.00	0.00	0.00	livestock	0.50	0.60	0.54
cocoa	0.64	0.60	0.62	lumber	0.00	0.00	0.00
coffee	0.76	0.86	0.80	meal-feed	0.00	0.00	0.00
copper	0.47	0.61	0.53	money-fx	0.69	0.59	0.64
cotton	0.58	0.77	0.66	money-supply	0.68	0.53	0.60
cpi	0.71	0.58	0.64	nat-gas	0.28	0.66	0.40
cpu	0.33	1.00	0.50	nickel	0.00	0.00	0.00
crude	0.73	0.56	0.63	orange	0.63	0.77	0.70
dlr	0.16	0.66	0.26	pet-chem	0.00	0.00	0.00
earn	0.97	0.95	0.96	platinum	0.00	0.00	0.00
fuel	1.00	0.14	0.25	potato	0.50	0.33	0.40
gas	0.25	0.12	0.16	reserves	0.62	0.41	0.50
gnp	0.58	0.93	0.71	retail	0.25	1.00	0.40
gold	0.63	0.70	0.66	rubber	0.72	0.88	0.80
grain	0.23	0.30	0.26	ship	0.62	0.63	0.63
heat	0.08	0.75	0.15	strategic-metal	0.00	0.00	0.00
housing	0.25	1.00	0.40	sugar	0.77	0.68	0.72
income	0.66	0.50	0.57	tea	0.00	0.00	0.00
instal-debt	1.00	1.00	1.00	tin	0.63	0.70	0.66
interest	0.76	0.44	0.56	trade	0.62	0.72	0.66
ipi	0.62	0.45	0.52	veg-oil	0.58	0.63	0.60
iron-steel	0.38	0.58	0.46	wpi	1.00	0.11	0.20
jet	0.00	0.00	0.00	zinc	0.25	0.20	0.22
				Macro Promedio	0.49	0.53	0.47
Exactitud 0.79							

Tabla B.9: **Baseline Bayes**: Exactitudes para R(52) stemmed

SVM							
Clases	π	ρ	F_1	Clases	π	ρ	F_1
acq	0.79	0.97	0.87	jobs	1.00	0.75	0.85
alum	0.85	0.31	0.46	lead	1.00	0.25	0.40
bop	0.00	0.00	0.00	lei	0.66	0.66	0.66
carcass	0.00	0.00	0.00	livestock	1.00	0.60	0.75
cocoa	1.00	0.53	0.69	lumber	0.00	0.00	0.00
coffee	0.95	0.86	0.90	meal-feed	0.00	0.00	0.00
copper	1.00	0.76	0.87	money-fx	0.78	0.58	0.67
cotton	0.66	0.22	0.33	money-supply	0.72	0.57	0.64
cpi	0.63	0.70	0.66	nat-gas	0.87	0.58	0.70
cpu	0.00	0.00	0.00	nickel	0.00	0.00	0.00
crude	0.87	0.84	0.85	orange	1.00	0.55	0.71
dlr	0.00	0.00	0.00	pet-chem	0.00	0.00	0.00
earn	0.96	0.99	0.97	platinum	0.00	0.00	0.00
fuel	0.00	0.00	0.00	potato	0.00	0.00	0.00
gas	1.00	0.25	0.40	reserves	1.00	0.66	0.80
gnp	0.76	0.86	0.81	retail	0.50	1.00	0.66
gold	0.94	0.80	0.86	rubber	1.00	0.77	0.87
grain	0.55	0.50	0.52	ship	0.88	0.61	0.72
heat	0.00	0.00	0.00	strategic-metal	0.00	0.00	0.00
housing	1.00	0.50	0.66	sugar	0.81	0.72	0.76
income	0.00	0.00	0.00	tea	0.00	0.00	0.00
instal-debt	0.00	0.00	0.00	tin	1.00	0.70	0.82
interest	0.65	0.48	0.55	trade	0.70	0.85	0.77
ipi	0.72	0.72	0.72	veg-oil	1.00	0.72	0.84
iron-steel	0.71	0.41	0.52	wpi	1.00	0.11	0.20
jet	0.00	0.00	0.00	zinc	1.00	0.40	0.57
				Macro Promedio	0.57	0.42	0.46
Exactitud 0.86							

Tabla B.10: **Baseline SVM**: Exactitudes para R(52) stemmed

Naive Bayes							
Clases	π	ρ	F_1	Clases	π	ρ	F_1
acq	0.96	0.84	0.90	jobs	1.00	0.83	0.90
alum	0.93	0.73	0.82	lead	0.00	0.00	0.00
bop	0.31	0.77	0.45	lei	0.75	1.00	0.85
carcass	0.00	0.00	0.00	livestock	0.18	0.40	0.25
cocoa	0.61	0.86	0.72	lumber	0.00	0.00	0.00
coffee	0.69	0.90	0.78	meal-feed	0.00	0.00	0.00
copper	1.00	0.92	0.96	money-fx	0.67	0.72	0.70
cotton	0.63	0.77	0.70	money-supply	0.91	0.75	0.82
cpi	0.85	0.70	0.77	nat-gas	0.40	0.91	0.56
cpu	0.16	1.00	0.28	nickel	0.10	1.00	0.18
crude	0.84	0.73	0.78	orange	0.80	0.88	0.84
dlr	0.12	0.33	0.18	pet-chem	0.50	0.50	0.50
earn	0.99	0.93	0.96	platinum	0.00	0.00	0.00
fuel	0.08	0.28	0.12	potato	1.00	0.66	0.80
gas	0.38	0.62	0.47	reserves	0.39	0.75	0.51
gnp	0.30	0.86	0.44	retail	0.25	1.00	0.40
gold	0.73	0.95	0.82	rubber	0.38	0.77	0.51
grain	0.60	0.60	0.60	ship	0.86	0.72	0.78
heat	0.21	0.75	0.33	strategic-metal	0.00	0.00	0.00
housing	0.28	1.00	0.44	sugar	0.95	0.84	0.89
income	1.00	0.75	0.85	tea	0.00	0.00	0.00
instal-debt	0.05	1.00	0.10	tin	0.43	0.70	0.53
interest	0.90	0.49	0.64	trade	0.59	0.82	0.69
ipi	0.85	0.54	0.66	veg-oil	0.37	0.54	0.44
iron-steel	0.69	0.75	0.72	wpi	1.00	0.66	0.80
jet	0.00	0.00	0.00	zinc	1.00	0.80	0.88
				Macro Promedio	0.51	0.64	0.52
Exactitud 0.84							

Tabla B.11: **Método Propuesto Bayes**: Exactitudes para R(52) stemmed

SVM							
Clases	π	ρ	F_1	Clases	π	ρ	F_1
acq	0.97	0.87	0.92	jobs	1.00	0.83	0.90
alum	0.93	0.73	0.82	lead	0.00	0.00	0.00
bop	0.42	1.00	0.60	lei	0.66	0.66	0.66
carcass	0.66	0.40	0.50	livestock	0.33	0.60	0.42
cocoa	0.65	0.86	0.74	lumber	0.00	0.00	0.00
coffee	0.67	0.86	0.76	meal-feed	0.00	0.00	0.00
copper	1.00	0.76	0.87	money-fx	0.74	0.85	0.79
cotton	0.88	0.88	0.88	money-supply	0.84	0.78	0.81
cpi	0.87	0.82	0.84	nat-gas	0.43	0.58	0.50
cpu	0.50	1.00	0.66	nickel	0.10	1.00	0.18
crude	0.82	0.75	0.78	orange	1.00	0.77	0.87
dlr	0.00	0.00	0.00	pet-chem	0.66	0.66	0.66
earn	0.99	0.94	0.96	platinum	0.00	0.00	0.00
fuel	0.04	0.14	0.06	potato	1.00	0.66	0.80
gas	0.58	0.87	0.70	reserves	0.45	0.91	0.61
gnp	0.38	1.00	0.55	retail	0.33	1.00	0.50
gold	0.73	0.85	0.79	rubber	0.31	0.66	0.42
grain	0.71	1.00	0.83	ship	0.88	0.61	0.72
heat	0.30	0.75	0.42	strategic-metal	0.00	0.00	0.00
housing	0.40	1.00	0.57	sugar	0.91	0.88	0.89
income	1.00	0.50	0.66	tea	0.00	0.00	0.00
instal-debt	0.00	0.00	0.00	tin	0.44	0.80	0.57
interest	0.91	0.63	0.74	trade	0.64	0.84	0.73
ipi	0.72	0.72	0.72	veg-oil	0.57	0.72	0.64
iron-steel	0.73	0.91	0.81	wpi	1.00	0.77	0.87
jet	0.00	0.00	0.00	zinc	1.00	0.80	0.88
				Macro Promedio	0.56	0.64	0.57
Exactitud 0.86							

Tabla B.12: **Método Propuesto SVM**: Exactitudes para R(52) stemmed

Naive Bayes			
Clases	π	ρ	F_1
alt.atheism	0.60	0.58	0.59
comp.graphics	0.46	0.25	0.32
comp.os.ms-windows.misc	0.43	0.54	0.48
comp.sys.ibm.pc.hardware	0.63	0.37	0.46
comp.sys.mac.hardware	0.60	0.44	0.51
comp.windows.x	0.68	0.37	0.48
misc.forsale	0.18	0.67	0.29
rec.autos	0.61	0.58	0.60
rec.motorcycles	0.61	0.74	0.67
rec.sport.baseball	0.76	0.71	0.74
rec.sport.hockey	0.89	0.70	0.79
sci.crypt	0.78	0.63	0.70
sci.electronics	0.42	0.39	0.40
sci.med	0.70	0.43	0.53
sci.space	0.71	0.57	0.63
soc.religion.christian	0.73	0.58	0.64
talk.politics.guns	0.60	0.67	0.63
talk.politics.mideast	0.85	0.66	0.74
talk.politics.misc	0.55	0.49	0.52
talk.religion.misc	0.36	0.41	0.38
Macro Promedio	0.61	0.54	0.55
Exactitud 0.54			

SVM			
Clases	π	ρ	F_1
alt.atheism	0.69	0.63	0.66
comp.graphics	0.21	0.78	0.33
comp.os.ms-windows.misc	0.78	0.64	0.70
comp.sys.ibm.pc.hardware	0.59	0.64	0.62
comp.sys.mac.hardware	0.79	0.60	0.68
comp.windows.x	0.72	0.55	0.63
misc.forsale	0.75	0.81	0.78
rec.autos	0.84	0.68	0.75
rec.motorcycles	0.93	0.79	0.86
rec.sport.baseball	0.93	0.78	0.85
rec.sport.hockey	0.94	0.77	0.85
sci.crypt	0.91	0.73	0.81
sci.electronics	0.52	0.55	0.53
sci.med	0.82	0.53	0.64
sci.space	0.87	0.66	0.75
soc.religion.christian	0.84	0.79	0.82
talk.politics.guns	0.68	0.74	0.71
talk.politics.mideast	0.96	0.65	0.77
talk.politics.misc	0.72	0.50	0.59
talk.religion.misc	0.66	0.51	0.58
Macro Promedio	0.76	0.67	0.70
Exactitud 0.67			

Tabla B.13: **Baseline**: Exactitudes para 20Newsgroups no-short.

Naive Bayes			
Clases	π	ρ	F_1
alt.atheism	0.65	0.63	0.64
comp.graphics	0.67	0.60	0.63
comp.os.ms-windows.misc	0.60	0.70	0.64
comp.sys.ibm.pc.hardware	0.58	0.58	0.58
comp.sys.mac.hardware	0.72	0.70	0.71
comp.windows.x	0.71	0.51	0.60
misc.forsale	0.82	0.64	0.72
rec.autos	0.78	0.75	0.77
rec.motorcycles	0.81	0.81	0.81
rec.sport.baseball	0.77	0.87	0.82
rec.sport.hockey	0.84	0.78	0.81
sci.crypt	0.68	0.78	0.72
sci.electronics	0.77	0.53	0.63
sci.med	0.87	0.78	0.83
sci.space	0.78	0.79	0.79
soc.religion.christian	0.70	0.76	0.72
talk.politics.guns	0.57	0.87	0.69
talk.politics.mideast	0.71	0.79	0.75
talk.politics.misc	0.58	0.56	0.57
talk.religion.misc	0.54	0.55	0.55
Macro Promedio	0.71	0.70	0.70
Exactitud 0.70			

SVM			
Clases	π	ρ	F_1
alt.atheism	0.68	0.64	0.66
comp.graphics	0.71	0.68	0.70
comp.os.ms-windows.misc	0.65	0.64	0.65
comp.sys.ibm.pc.hardware	0.56	0.65	0.61
comp.sys.mac.hardware	0.76	0.67	0.71
comp.windows.x	0.71	0.70	0.71
misc.forsale	0.86	0.81	0.83
rec.autos	0.88	0.85	0.86
rec.motorcycles	0.89	0.88	0.88
rec.sport.baseball	0.83	0.90	0.86
rec.sport.hockey	0.83	0.84	0.84
sci.crypt	0.73	0.86	0.79
sci.electronics	0.80	0.57	0.67
sci.med	0.89	0.80	0.84
sci.space	0.84	0.81	0.82
soc.religion.christian	0.76	0.91	0.83
talk.politics.guns	0.64	0.84	0.73
talk.politics.mideast	0.77	0.79	0.78
talk.politics.misc	0.66	0.52	0.58
talk.religion.misc	0.68	0.63	0.66
Macro Promedio	0.76	0.75	0.75
Exactitud 0.76			

Tabla B.14: **Método Propuesto**: Exactitudes para 20Newsgroups no-short.

Naive Bayes			
Clases	π	ρ	F_1
alt.atheism	0.62	0.55	0.58
comp.graphics	0.47	0.24	0.32
comp.os.ms-windows.misc	0.42	0.48	0.44
comp.sys.ibm.pc.hardware	0.61	0.32	0.42
comp.sys.mac.hardware	0.53	0.40	0.45
comp.windows.x	0.71	0.36	0.48
misc.forsale	0.16	0.65	0.26
rec.autos	0.63	0.53	0.58
rec.motorcycles	0.58	0.71	0.64
rec.sport.baseball	0.72	0.72	0.72
rec.sport.hockey	0.85	0.68	0.76
sci.crypt	0.74	0.64	0.69
sci.electronics	0.48	0.32	0.38
sci.med	0.72	0.47	0.57
sci.space	0.68	0.58	0.63
soc.religion.christian	0.67	0.55	0.60
talk.politics.guns	0.60	0.68	0.63
talk.politics.mideast	0.83	0.66	0.74
talk.politics.misc	0.53	0.49	0.51
talk.religion.misc	0.36	0.41	0.38
Macro Promedio	0.60	0.52	0.54
Exactitud 0.52			

SVM			
Clases	π	ρ	F_1
alt.atheism	0.75	0.61	0.67
comp.graphics	0.19	0.77	0.30
comp.os.ms-windows.misc	0.71	0.59	0.64
comp.sys.ibm.pc.hardware	0.63	0.62	0.63
comp.sys.mac.hardware	0.77	0.59	0.67
comp.windows.x	0.67	0.57	0.62
misc.forsale	0.67	0.79	0.73
rec.autos	0.85	0.63	0.72
rec.motorcycles	0.93	0.75	0.83
rec.sport.baseball	0.92	0.78	0.85
rec.sport.hockey	0.96	0.74	0.84
sci.crypt	0.91	0.73	0.81
sci.electronics	0.61	0.50	0.55
sci.med	0.82	0.55	0.66
sci.space	0.86	0.62	0.72
soc.religion.christian	0.79	0.79	0.79
talk.politics.guns	0.70	0.77	0.73
talk.politics.mideast	0.95	0.64	0.77
talk.politics.misc	0.71	0.49	0.58
talk.religion.misc	0.70	0.48	0.57
Macro Promedio	0.76	0.65	0.68
Exactitud 0.66			

Tabla B.15: **Baseline**: Exactitudes para 20Newsgroups stemmed.

Naive Bayes			
Clases	π	ρ	F_1
alt.atheism	0.65	0.65	0.65
comp.graphics	0.69	0.63	0.66
comp.os.ms-windows.misc	0.61	0.64	0.63
comp.sys.ibm.pc.hardware	0.54	0.56	0.55
comp.sys.mac.hardware	0.69	0.71	0.70
comp.windows.x	0.70	0.54	0.61
misc.forsale	0.81	0.54	0.65
rec.autos	0.79	0.72	0.76
rec.motorcycles	0.78	0.83	0.80
rec.sport.baseball	0.78	0.87	0.82
rec.sport.hockey	0.77	0.78	0.78
sci.crypt	0.69	0.82	0.75
sci.electronics	0.77	0.48	0.60
sci.med	0.81	0.74	0.78
sci.space	0.77	0.80	0.78
soc.religion.christian	0.69	0.72	0.70
talk.politics.guns	0.56	0.83	0.67
talk.politics.mideast	0.68	0.77	0.73
talk.politics.misc	0.57	0.56	0.57
talk.religion.misc	0.51	0.53	0.52
Macro Promedio	0.69	0.69	0.68
Exactitud 0.69			

SVM			
Clases	π	ρ	F_1
alt.atheism	0.70	0.66	0.68
comp.graphics	0.71	0.69	0.70
comp.os.ms-windows.misc	0.70	0.64	0.67
comp.sys.ibm.pc.hardware	0.56	0.67	0.61
comp.sys.mac.hardware	0.76	0.70	0.73
comp.windows.x	0.71	0.72	0.71
misc.forsale	0.86	0.65	0.74
rec.autos	0.85	0.85	0.85
rec.motorcycles	0.85	0.87	0.86
rec.sport.baseball	0.84	0.90	0.87
rec.sport.hockey	0.80	0.86	0.83
sci.crypt	0.74	0.86	0.79
sci.electronics	0.83	0.56	0.67
sci.med	0.87	0.78	0.83
sci.space	0.82	0.82	0.82
soc.religion.christian	0.73	0.89	0.80
talk.politics.guns	0.62	0.84	0.71
talk.politics.mideast	0.75	0.79	0.77
talk.politics.misc	0.64	0.52	0.57
talk.religion.misc	0.68	0.61	0.64
Macro Promedio	0.75	0.74	0.74
Exactitud 0.75			

Tabla B.16: **Método Propuesto**: Exactitudes para 20newsgroups stemmed.

Exactitudes				
	Naive Bayes		**SVM**	
Corpus	Baseline	2 Etapas	Baseline	2 Etapas
R(8) no-short	0.84	**0.90**	0.92	**0.93**
R(8) stemmed	0.83	**0.88**	**0.92**	**0.92**
R(52) no-short	0.81	**0.84**	**0.86**	**0.86**
R(52) stemmed	0.79	**0.84**	**0.86**	**0.86**
20newsgroups no-short	0.54	**0.70**	0.67	**0.76**
20newsgroups stemmed	0.52	**0.69**	0.66	**0.75**

Tabla B.17: Desempeño de los clasificadores con el método propuesto comparado con el Baseline

La Tabla B.17 resume los resultados de las exactitudes obtenidas en el corpora analizado. Los resultados prácticamente son los mismos a los obtenidos sin quitar palabras cortas o hacer *stemming* en general el método en dos etapas propuesto es mejor que el método tradicional.

Los conjuntos *stemeed* muestran un desempeño inferior al de *no-short* y es que de cierta manera se están eliminando términos que pudieran ayudar a distinguir un documento entre las clases.