Clasificación de Textos Cortos Usando Representaciones Distribucionales de los Términos

Por

Juan Manuel Cabrera Jiménez

Tesis sometida como requisito parcial para obtener el grado de

MAESTRO EN CIENCIAS EN LA ESPECIALIDAD DE CIENCIAS COMPUTACIONALES

en el

Instituto Nacional de Astrofísica, Óptica y Electrónica

Tonantzintla, Puebla

Supervisada por:

Dr. Manuel Montes y Gómez
Investigador del INAOE
Dr. Hugo Jair Escalante Balderas
Investigador del INAOE

Resumen

La cantidad de documentos cortos que está disponible se ha incrementado considerablemente en los últimos años gracias a los avances tecnológicos. En este contexto se ha motivado el desarrollo de mecanismos automáticos que faciliten su acceso, organización y análisis. Debido a la longitud de los documentos y a las representaciones tan dispersas de los documentos, la aplicación directa de los método de representación estándar de la categorización de texto no es una solución viable al problema. En este trabajo se describe el uso de las representaciones distribucionales de los términos (DTRs, por sus siglas en inglés) para la clasificación de los textos cortos para superar, en cierta medida, los problemas longitud/disperso. Las DTRs son una forma de representar términos, por medio de la información contextual dada por la ocurrencia en un documento y la co-ocurrencia estadística entre términos. Combinamos las DTRs de los términos que aparecen en los textos cortos para generar mejores representaciones de documentos que se pueden utilizar con las técnicas de aprendizaje automático. De esta manera, un documento no está representado por los términos que ocurren en él, sino por un vector de pesos contextuales, que indican la asociación de términos con los documentos en el corpus o con términos en el vocabulario; en lugar de representar un documento únicamente por el conjunto de términos que aparecen en él. La evaluación se realizó en tres colecciones, utilizando una variedad de métodos de clasificación y en

dos distintos escenarios: i) en la clasificación de textos cortos, y ii) en la clasificación de textos cortos y conjunto datos reducidos. Los resultados experimentales demostraron que el uso de las DTRs es beneficioso para mejorar el rendimiento de los clasificadores en la categorización de textos cortos y también cuando se tiene una combinación de textos cortos y un conjunto de entrenamiento reducido. En particular, la representación ocurrencia-documento superó a las otras representaciones evaluadas.

Abstract

The amount of short documents that are available has increased considerably in recent years due to technological advances. In this context it has motivated the development of automatic mechanisms to facilitate their access, organization and analysis. Due to the tiny legth of documents and of the extremely sparse document representations, the direct application of standard text categorization methods is not an effective solution to the problem. In this work describes the use of distributional representations of terms (DTRs) for the classification of short texts to overcome, to some extent, the small-length/high-sparsity issues. The DTRs are a way of representing terms, using contextual information, given by the document occurrence and term co-occurrence statistical. We combine the DTRs of the terms appearing in short texts to generate better document representations that can be used with standard machine learning techniques. Thus, a document is not represented by the terms that occur in it, but for contextual weight vector, indicating the association of terms with documents in the corpus or terms in the vocabulary; instead of representing a document only by the set of terms in it. The evaluation was performed in three collections, using a variety of classification methods and two different scenarios: i) in the classification of short texts, and ii) in the classification of short texts and when there are few labeled documents. The experimental results show that the use of DTRs is beneficial for improving the performance of

classifiers in short text categorization and also when one has a combination of short texts and small training set. In particular, the document-occurrence representation outperformed the other representations we evaluated.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

La cantidad de información en formato digital que está disponible gracias a los avances tecnológicos se ha incrementado considerablemente en los últimos años. Gracias a Internet y a la facilidad de crear, compartir y divulgar información, se puede encontrar todo tipo de documentos como son: imágenes, videos, audios y textos, siendo este último de particular interés para esta tesis. Realizar la búsqueda, organización y análisis de la información de forma manual es un proceso complicado e ineficiente, debido a la gran cantidad de documentos de textos que se genera diariamente.

Por lo anterior, y con la finalidad de aprovechar la información contenida en los documentos de manera efectiva, se han realizado estudios en distintas líneas de investigación, entre las que destacan recuperación de información, generación de resúmenes automáticos, búsqueda de respuestas, clasificación de textos, entre muchas otras. Todo esto con el objetivo de obtener resultados de calidad e invertir el menor tiempo posible en organizar, buscar o recuperar la información deseada. No cabe duda que una organización de los documentos con la mayor calidad posible, es una valiosa herramienta cuando llega el momento de buscar información,

tomar decisiones o realizar alguna otra tarea.

La clasificación de textos (TC, por sus siglas en inglés) es una de las áreas encargadas de la organización de la información. También conocida como clasificación de documentos o categorización de textos y se define como "la asignación de un documento a una o más categorías predefinidas en función de su contenido" (Sebastiani, 2002). El método de solución más popular de clasificación de textos es supervisado, lo que significa que no solo se conocen previamente las categorías sino que también debe contarse con un conjunto de entrenamiento. Dicho conjunto de entrenamiento, es un conjunto de documentos previamente etiquetados con las categorías a las que pertenecen. Para asignar de forma automática la clase a cada documento, es necesario entrenar un clasificador usando el conjunto de entrenamiento, generando un modelo computacional que posteriormente se utiliza para asignar la clase de forma automática a un nuevo documento de acuerdo a ciertas variables. Algunos ejemplos de aplicación de la clasificación automática de textos son: organizar noticias de acuerdo a la temática (deportes, sociales, etc), organizar los documentos de acuerdo su autor (poemas), etc.

1.1. Planteamiento del problema

Los métodos utilizados dentro de la clasificación automática de textos han mejorado en cuanto a la calidad, llegando a compararse en cuanto a exactitud con los profesionales entrenados (Sebastiani, 2002), y además el tiempo invertido por un sistema de clasificación automática de textos, es mucho menor. Varios métodos de clasificación de textos tienden a dar resultados aceptables para documentos extensos, debido a que existe una mayor probabilidad de distinguir la clase a la que pertenece cada documento (Pinto), gracias a las frecuencias de los términos

en las clases. Sin embargo, cuando los documentos contienen un número reducido de palabras los métodos no son efectivos, ya que las frecuencias de los términos son demasiado bajas.

La cantidad de información contenida en la Web tiende a aumentar, mientras que el número de palabras de cada documento tiende a reducir. Algunos ejemplos de textos cortos son: los resúmenes científicos, noticias, blogs, correos electrónicos. Un ejemplo de la necesidad de procesamiento de los textos cortos se da en la organización de los artículos científicos existentes en bibliotecas digitales y en repositorios en linea, ya que la mayoría está limitado a mostrar solo los resúmenes científicos, complicando la tarea de organizar la información en categorías. Por otro lado, la necesidad e importancia de estudio de los textos cortos se ha incrementado debido a la actual forma en que gran parte de la gente usa el lenguaje, es decir, el lenguaje reducido que provee la tendencia tecnológica como los correos electrónicos, *blogs, chats*, redes sociales.

Los atributos seleccionados para la clasificación de textos son generalmente las palabras de los documentos que se usan en la colección (vocabulario), y cada documento es representado usando esos atributos. Los documentos cortos tienen el problema de la baja frecuencia de ocurrencia de las palabras en los documentos. Por lo que al representar a los documentos por las palabras que contiene puede que no sea suficiente para obtener una caracterización indicada del contenido de los mismos, y por ende imposibilita la representación de las clases. Además, si añadimos al problema de baja frecuencia de términos la longitud de documentos, el resultado es una representación *dispersa*, es decir, se tiene un amplio vocabulario y cada documento es representado por pocos atributos que además no aportan información relevante. Otro de los problemas que es ocasionado por los textos cortos es la ambigüedad de las palabras (*sinonimia* y *polisemia*). En resumen, los

textos cortos tienen dos principales problemas: la *baja frecuencia de términos* y la *longitud de los documentos*. Por lo mencionado con anterioridad, surgió el interés por investigar cómo organizar de forma sencilla y adecuada los textos cortos.

1.2. Solución propuesta

Una de las soluciones propuestas al problema de clasificación de textos cortos dentro del estado del arte es agregar a la representación de los documentos términos que no existan en el documento pero que tengan una cierta relación a la representación del documento, esto con la finalidad de solventar el problema de la longitud de los textos cortos. Cabe resaltar que el procedimiento se puede realizar de distintas formas, y una de ellas es la utilización de recursos externos. Sin embargo, es complicado obtener un conjunto externo que esté relacionado con el objetivo, además, son dependientes del dominio, por lo que en este trabajo no usamos recursos externos.

La solución propuesta es un método basado en la intuición de que el significado de una palabra está dado por los documentos en que ocurre o por los otros términos con los que co-ocurre (Harris, 1968), en otras palabras, la idea es buscar las relaciones entre los términos o documentos que existen en la colección y posteriormente representar (enriquecer representaciones) a cada documento utilizando las relaciones encontradas. Para buscar las relaciones existentes entre los términos se usan estadísticas de ocurrencia o co-ocurrencia de los términos, calculadas dentro del mismo conjunto de entrenamiento.

La solución propuesta se divide en tres etapas:

1. Obtener la representación de los términos, la idea es encontrar relaciones entre términos que proporcionen información relevante.

2. Obtener la representación de cada documento, en esta etapa se enriquece al documento, es decir, se agregan los términos relacionados encontrados en la etapa anterior.

3. Realizar el proceso de clasificación, donde se entrena algún algoritmo de clasificación y se genera el modelo.

Cabe mencionar que un *término* puede interpretarse como cualquier conjunto de caracteres o unidad de lenguaje como una frase o una palabra. En particular en este trabajo por términos nos referimos a palabras, aunque los métodos a desarrollar se pueden aplicar para cualquier tipo de término.

Un punto importante relacionado con las DTRs es que tienen un inconveniente, para obtener DTRs informativas se necesitan grandes cantidades de información (documentos extensos) en la etapa de entrenamiento, dado que las DTRs se basan en estadísticas de ocurrencia/co-ocurrencia. Sin embargo, existen algunos escenarios en los que si se cuentan con documentos extensos para la etapa de entrenamiento, aunque para la etapa de prueba solo hayan textos cortos y es precisamente este tipo de escenarios los que pretendemos abordar.

1.3. Objetivos

Objetivo general

Desarrollar e implementar un método basado en representaciones distribucionales de términos para mejorar el rendimiento de la clasificación de textos cortos.

Objetivos específicos

- Estudiar el alcance de las representaciones clásicas en la clasificación de textos cortos.
- Implementar y desarrollar una representación basada en DTRs para la clasificación de textos cortos.
- Implementar y desarrollar un método de pesado supervisado para la representación de los textos cortos.
- Evaluar y analizar la relevancia del método propuesto para abordar la clasificación de textos cortos.

1.4. Organización del documento

El presente documento está organizado de la siguiente manera: en el capítulo 2 se describe la teoría y conceptos básicos dentro del área de clasificación de textos, que permitirán entender de mejor forma el contenido del presente trabajo. En el capítulo 3 se presenta una revisión de los trabajos relacionados con esta investigación. En el capítulo 4 se explica el método propuesto. Posteriormente, en el capítulo 5 se presenta un análisis de los conjuntos de datos que se utilizarán para la evaluación del método y además se explica porqué se eligieron estos conjuntos. En el capítulo 6 se definen los experimentos realizados para evaluar el método propuesto, además se muestran los resultado obtenidos y se realiza un análisis de los mismos. Finalmente en el capítulo 7 se presentan las conclusiones del trabajo de investigación y las posibles líneas para un trabajo futuro.

Capítulo 2

Marco teórico

El objetivo de esta sección es introducir al lector los conceptos y definiciones relacionados con el presente trabajo de investigación. Se espera que el contenido de este capítulo contribuya a lograr una mejor comprensión de esta tesis. En la primera sección se describen las nociones básicas de la clasificación de textos. Posteriormente, en las siguientes secciones se presentan algunas de las diferentes maneras de representar a los documentos, así como las implicaciones que tienen. Después, se describen los procesos involucrados en la clasificación de textos, como son la extracción de características, métodos de aprendizaje y por último la forma en que se evalúa la tarea de clasificación.

2.1. Clasificación de Textos

El aprendizaje computacional (ML, por sus siglas en inglés) en general, busca construir sistemas computacionales que mejoren automáticamente mediante la experiencia.

La clasificación de textos, también conocida como clasificación de documentos

o categorización de textos, es la asignación de texto libre a una o más categorías predefinidas en función de su contenido (Sebastiani, 2002). La clasificación consiste en asociar un valor booleano a cada par $(d_j, c_i) \in D \times C$, donde $D = d_1, \cdots, d_n$ es el conjunto de documentos y $C = c_1, \cdots, c_m$ es el conjunto predefinido de clases. El valor V (Verdadero) es asociado a un par (d_j, c_i) si el documento d_j pertenece a la clase c_i, mientras que el valor F (Falso) es asignado en caso contrario. Se tiene que aproximar una función desconocida $\phi : D \times C \rightarrow \{V, F\}$ que asocia una clase a un documento d_j de tal forma que la decisión tomada es lo más cercana posible a la que le correspondería en la función $\Phi : D \times C \rightarrow \{V, F\}$ a la cual llamamos *clasificador*, ϕ (asociación automática) y Φ (asociación manual) son similares tanto como se pueda (Sebastiani, 2002).

La clasificación automática de textos tiene el objetivo de proveer de forma automática la clase apropiada a un documento. Cuando se usa aprendizaje computacional, el objetivo es entrenar a un clasificador con ejemplos de documentos previamente etiquetados para que posteriormente éste pueda asignar la clase de forma automática a nuevos documentos. Para generar el clasificador se realiza un proceso inductivo, el cual analiza los atributos de un conjunto de instancias preclasificadas bajo C y determina qué atributos debe tener una instancia desconocida para que pertenezca a la categoría. Es por esto que es necesario un conjunto de datos iniciales tal que el valor $\Phi(d_j, c_i)$ sea conocido para cada $(d_j, c_i) \in D \times C$, el cual comúnmente es conocido como conjunto de entrenamiento (T_r).

Los sistemas informáticos no pueden manipular el texto plano por lo que se debe procesar, ordenar y guardar en una estructura para que estos sistemas puedan manipular la información. La forma en que se guarda dicha estructura es conocida como representación de documentos. Una vez representado el documento, la estructura es utilizada para entrenar un clasificador y construir un modelo y evaluar

dicho modelo, sección. En las siguientes secciones se presentarán los dos requeridos dentro de la clasificación de textos: la representación de los documentos y la construcción del clasificador.

2.2. Representación de documentos

Uno de los primeros pasos en la clasificación de documentos es el pre-procesado de los documentos, los cuales se deben transformar a una representación con la que un algoritmo de aprendizaje pueda llevar a cabo la tarea de clasificación. Regularmente, para esta tarea se realizan algunos de los siguiente procesos:

- **Limpieza de documentos:** Consiste en remover todo aquello que se considere ruido, por ejemplo: etiquetas HTML o XML, estas normalmente se usan para organizar las colecciones de documentos en distintas categorías o de alguna forma distinta. Sin embargo, estas etiquetas se tienen que remover para que el algoritmo de aprendizaje únicamente tome en consideración la información del contenido del documento. También se considera la eliminación de encabezados, separadores, tablas, caracteres extraños entre otros.

- **Eliminar palabras vacías:** También conocidas como *stopwords* (en inglés), son palabras muy frecuentes y, por lo general no aportan información del contenido del documento (por ejemplo: artículos, pronombres, preposiciones y conjunciones).

- **Lematización de palabras:** Un lematizador obtiene las raíces morfológicas de las palabras, eliminando declinaciones por conjugación, número, género. Ejemplo: doctor <- doctora, doctores. También es posible que dos palabras no relacionadas se lleven a la misma raíz (*overstemming*) o que dos palabras

relacionadas se lleven a distintas raíces (*understemming*). Sin embargo, en la mayoría de los casos se aplica una aproximación a este proceso debido a la dificultad. Un ejemplo de aproximación es realizada por el lematizador creado por *Porter* (Porter, 2006) que no busca la raíz de las palabras, más bien elimina los sufijos de los términos en inglés. Para efectos de este trabajo de tesis se usó *porter stemmer*.

2.2.1. Indexado

El indexado denota la actividad de hacer el mapeo de un documento d_j a una representación compacta de su contenido. El esquema más usado para la representación de los documentos es un vector con términos ponderados, este concepto fue tomado del modelo de espacio vectorial propuesto para la recuperación de información por Salton (Salton, 1991).

Modelo de espacio vectorial

La representación de documentos más usada es la llamada modelo de espacio vectorial (VSM, por sus siglas en inglés). VSM fue desarrollado para el sistema de recuperación de información SMART (Salton, 1971) por Salton y colegas. La idea central del modelo vectorial es que el contenido de los documentos puede representarse con los términos que aparecen en ellos. Dichos documentos son representados como vectores de m elementos, donde m son el número de términos de la colección.

Formalmente, un documento d_j puede ser representado como un vector $\overrightarrow{d_j} = (w_{j1}, \cdots, w_{j|m|})$, donde m es el diccionario de términos, es decir, el conjunto de términos que ocurren en algún documento d_j, mientras que w_{ji} representa la im-

portancia que tiene el término i en el documento j. Una forma de representar lo anteriormente mencionado es como sigue:

$$D_{VSM} = \begin{pmatrix} w_{11} & \cdots & w_{1m} \\ \vdots & \ddots & \vdots \\ w_{n1} & \cdots & w_{nm} \end{pmatrix}$$

donde cada fila representa un documento y cada columna representa los términos existentes en toda la colección de documentos.

El peso w_{ji} puede ser calculado de diversas formas, las tres más usadas en la clasificación de textos son: *Booleano, frecuencia de términos y* $tf \cdot idf$:

- **Booleano:** Es el pesado más simple, representa la presencia o ausencia de un término en el documento.

$$w_{kj} = \begin{cases} 1 & \text{si } t_k \text{ ocurre en } d_j \\ 0 & \text{en otro caso} \end{cases} \tag{2.1}$$

- **Frecuencia de términos:** Consiste en asignar el número de veces que el término k ocurren en el documento d_j

$$w_{kj} = TF(t_k, d_j) \tag{2.2}$$

  donde $TF(t_k, d_j)$ es la frecuencia de ocurrencia del término t_k en el documento d_j.

- **TF · IDF** : Consiste en la combinación de la frecuencia del término en el documento, con la frecuencia del mismo término en el resto de los documentos de la colección, y se calcula de la siguiente forma:

$$w_{kj} = TF(t_k, d_j) \times IDF(t_k) \tag{2.3}$$

donde IDF es la "frecuencia inversa" del término t_k en toda la colección de documentos. IDF es calculado de la siguiente forma:

$$IDF(t_k) = log\left(\frac{|D|}{df_k}\right)$$

donde df_k es el número de documentos que contienen el término t_k y $|D|$ es el número de documentos en la colección. Este pesado es comúnmente utilizado para eliminar el impacto de términos poco frecuentes a nivel de documento, pero que existen en la mayoría de los documentos.

La representación de los documentos empleada en el modelo VSM, considera únicamente las palabras presentes en ellos, sin realizar ningún análisis adicional ni conservar el orden de las palabras, por cual a este tipo de representación se les conoce como bolsa de palabras (BOW). A pesar de la simplicidad de este modelo, dentro de la clasificación de textos es un modelo robusto que funciona muy bien de forma general. Otros investigadores han utilizado otras representaciones más elaboradas como es el caso de *indexado semántico latente* (LSI), presentado en la siguiente sección. Para conocer un poco más acerca de otras representaciones se puede consulta el documento de Turney y Pantel (Karlgren and Sahlgren, 1996; Turney and Pantel, 2010) en el cual realizan un resumen de las diferentes formas que existen para representar los documentos.

Indexado semántico latente

El indexado semántico latente (LSI, Latent Semantic Indexing) es un método para extraer y representar el significado-uso de las palabras mediante cálculos estadísticos aplicados a un amplio conjunto de textos (Deerwester and Dumais, 1990). Este método también es conocido como análisis semántico latente (LSA)

cuando es aplicado a la similitud de palabras (Praks, 2003; Venegas, 2006). Cabe señalar que LSI no hace uso de ningún recurso lingüístico previo, no toma en cuenta el orden de las palabras, las relaciones sintácticas o lógicas, por el contrario, está basado en análisis de estadísticas de ocurrencia en el texto. La idea es que en grandes cantidades de información existen interrelaciones semánticas débiles entre palabras que son mejoradas con reducción de dimensión determinada por la descomposición en valores singulares (SVD, por sus siglas en inglés).

En general LSI se ha usado para diversas tareas como son la de encontrar el significado latente de los términos, reducción de ruido, co-ocurrencias de alto orden, y reducción de dispersión (Turney and Pantel, 2010).

Originalmente este método fue utilizado en la recuperación de información (Breiman, 2001; Deerwester and Dumais, 1990; Praks, 2003), en la reducción de dimensión (Landauer et al., 1998), y en los últimos años de ha adaptado e implementado en otras áreas como en la clasificación de textos (Cardoso-Cachopo and Oliveira, 2007; Zelikovitz, 2004). Por lo anterior, en este trabajo nos comparamos con LSI.

2.2.2. Reducción de la dimensión

Uno de los grandes problemas dentro de la clasificación de textos es que los atributos son generalmente las palabras que contienen lo documentos, por lo que generalmente se trabaja con una alta dimensión (miles de palabras). Como se mencionó anteriormente, los atributos más utilizados en TC son las diferentes palabras existentes en la colección de documentos. En la mayoría de los casos, el manejar toda esta cantidad de información es computacionalmente costoso. Además se sabe que es un proceso necesario para poder quitar términos irrele-

vantes y/o redundantes. Las características irrelevantes y redundantes pueden ser removidas sin afectar el desempeño de la tarea (Eikvil and Aas, 1999).

En la categorización de documentos, la alta dimensionalidad del espacio de términos puede ocasionar un sobre-ajuste en el proceso de aprendizaje. Es decir, que el algoritmo de aprendizaje no captura la generalidad de los documentos y se especializa únicamente en los documentos con los que se entrenó. Para resolver este problema se han desarrollado diferentes métodos que ayudan a seleccionar un subconjunto de términos, el cual debe describir igual o mejor a cada documento que el conjunto original de términos. A este proceso de selección del subconjunto de términos se le conoce como *selección de atributos* (Eikvil and Aas, 1999).

Los esquemas más frecuentes de selección de atributos en la categorización de textos son: *frecuencia de términos (TF), frecuencia de documento (DF), entropía, los cuales se describen a continuación.*

Frecuencia de Términos (TF)

La frecuencia de términos es uno de los métodos más simples. La idea básica detrás de este método es que las palabras poco frecuentes no aportan información útil para la clasificación. Consiste en calcular la frecuencia de ocurrencia de los términos en el *corpus* y eliminar aquellos que tengan una frecuencia menor a un umbral establecido.

Frecuencia de Documento (DF)

La idea básica de este esquema es que los términos que aparecen en pocos documentos no aportan información para predecir la clase. La frecuencia de documento de un término es el número de documentos en que cada término aparece, de esta forma solo basta con eliminar aquellos términos que contengan una DF

menor a cierto umbral.

Entropía

Es una medida comúnmente utilizada en teoría de la información, la cual caracteriza la (im)pureza de una colección arbitraria de ejemplos. Dada una colección S, la cual contenga dos clases positiva $c_{\oplus}$ y negativa $c_{\ominus}$, la entropía de S es:

$$Entropia(S) \equiv -p_{\oplus} log_2 p_{\oplus} - p_{\ominus} log_2 p_{\ominus}$$

Debemos observar que la entropía es 0, si todos los elementos de S pertenecen a la misma clase y es 1 cuando la colección S contiene un número equitativo de ejemplos positivos y negativos.

Hemos hablado de la entropía para se tienen dos clases, sin embargo, para cuando las clases pueden tomar k diferentes valores, entonces la entropía relativa está definida como :

$$Entropia(S) = \sum_{i=1}^{k} -P(C_i) log_2 P(C_i) \tag{2.4}$$

donde p_i es la porción de instancias de S que pertenecen a la clase i. Una vez que se tiene la entropía de cada término, esta se puede utilizar como un esquema de pesado (Debole and Sebastiani, 2003; Ure, 2005), o bien, se ordenan los atributos de menor a mayor y se seleccionan lo k términos más relevantes, es decir, aquellos con valor de entropía cercanos a cero. Para mayor información consultar (Mitchell, 1997).

Ganancia de Información (IG)

Es un sinónimo para la divergencia de Kullback–Leibler, al cual también se le conoce como divergencia de la información o entropía relativa. La idea básica es

descubrir qué tan bien un término puede dividir al conjunto de datos con base a las clases. De manera formal se define como la reducción esperada en la entropía causada por la división de los ejemplos de acuerdo a un atributo (Mitchell, 1997). Para un conjunto de categorías $C = c_1, \cdots, c_m$, la ganancia de información del atributo o término t se define como:

$$IG(t) = -\sum_{i=1}^{m} P(C_i)log(P(c_i)) + P(t)\sum_{i=1}^{m} P(c_i|t)log(P(c_i|t)) - P(\bar{t})\sum_{i=1}^{m} P(c_i|\bar{t})logP(c_i|\bar{t}) \quad (2.5)$$

donde m es el numero de clases, $P(c_i)$ es la probabilidad de la clase c_i, $P(t)$ es la probabilidad de seleccionar un documento en el que aparezca el término t, $P(c_i|t)$ es la probabilidad condicional de que pertenezca a la clase c_i que contengan el término t, por otro lado, $P(\bar{t})$ es el número de documentos en los cuales no ocurre el término t y $P(c_i|\bar{t})$ es la fracción de documentos de la clase c_i en los que no ocurre el término t.

Normalmente para la clasificación de textos se calcula la IG para cada término y se eliminan aquellos que estén por debajo de algún umbral establecido, aunque cabe mencionar que también puede servir como pesado de términos (Debole and Sebastiani, 2003; Ure, 2005). Para mayor información consultar (Mitchell, 1997).

2.3. Algoritmos de aprendizaje

La función principal de estos algoritmos es inducir un modelo para asignar de forma automática una categoría a cada documento, todo esto basado en el conjunto de entrenamiento. Diferentes algoritmos se han usado en la tarea de clasificación

de textos, en esta tesis se seleccionaron cinco algoritmos de aprendizaje representativos de la variedad de métodos existentes: clasificador probabilista (Bayesiano simple), basado en instancias (k-Vecinos más cercanos), kernel (Máquinas de vectores de soporte), árboles (RandomForest) y ensambles (AdaBoost), los cuales son descritos brevemente a continuación.

2.3.1. Bayesiano simple

El clasificador *Naive simple* (NB) es un clasificador probabilista, que se basa en el teorema de Bayes con una fuerte suposición de independencia entre características, es decir, el Bayesiano simple asume que la presencia/ausencia de una característica particular de una clase no esta relacionada con la presencia/ausencia de alguna otra característica dada la clase. A pesar de los supuestos que esto conlleva se ha utilizado mucho debido a los resultados satisfactorios logrados dentro de la clasificación de textos (Frank and Bouckaert, 2006; Gu et al., 2009; He and Ding, 2007; Mccallum and Nigam, 1997).

El clasificador Naive simple es construido usando el conjunto de entrenamiento para estimar la probabilidad de cada clase dado las características de un nuevo ejemplo. Para esto se usa el teorema de Bayes:

$$P(c_j|d) = \frac{P(c_j)P(d|c_j)}{P(d)} \tag{2.6}$$

donde P(d) es la probabilidad de que se elija el documento d aleatoriamente y ésta no afecta en la decisión de la clase, por lo que se puede omitir. La parte simple o ingenua de este modelo es el supuesto de independencia de términos, es decir, asumimos que las características son condicionalmente independiente dada

la clase, por lo que se simplifica el cálculo resultando:

$$P(d|c_j) = \prod_{i=1}^{M} P(t_i|c_j) \tag{2.7}$$

donde M es el tamaño del vocabulario, t_i denota un término o atributo. Sustituyendo esta ecuación en 2.6 resulta:

$$P(c_j|d) \propto P(c_j) \prod_{i=1}^{M} P(w_i|c_j) \tag{2.8}$$

La probabilidad *a priori* de la clase $P(c_j)$ se puede estimar utilizando el conjunto de entrenamiento como:

$$P(c_j) = \frac{N_j}{N} \tag{2.9}$$

donde N_j es el número de documentos que pertenecen a la clase c_j y N es el número total de documentos en el conjunto de entrenamiento. $P(w_i|c_j)$ se puede calcular como:

$$P(w_i|c_j) = \frac{N_{ij}}{\sum_{k=1}^{M} N_{k_j}} \tag{2.10}$$

donde N_{ij} es el número de veces que el término i ocurren en los documentos de la categoría c_j. Para resolver el problema de probabilidad cero se usa la estimación de Laplace, conocida como: Add-One Smoothing.

$$P(w_i|c_j) = \frac{1 + N_{ij}}{M + \sum_{k=1}^{M} N_{k_j}} \tag{2.11}$$

Para asignarle la clase a un documento nuevo d lo que se hace es calcular:

$$clase_d = argmax_r P(c_r|d) \tag{2.12}$$

2.3.2. K-Vecinos más cercanos

Es un método de aprendizaje basado en instancias, también conocido como *lazy learning* o *memory learning*, debido a que los datos de entrenamiento se

procesan solo hasta que se requiere hacer una predicción, y la relevancia de los datos se mide en función de una medida de distancia. Este método de clasificación ha sido ampliamente usado en el análisis de texto (Chen et al., 2003; Pandey and Chakraverty, 2011).

La idea principal de este método (K-Vecinos más cercanos, kNN), es almacenar el conjunto de entrenamiento (D_T), de tal modo que al clasificar un nuevo documento se busca en los documentos almacenados los k casos más cercanos (similares) usando la ecuación de la similitud coseno (2.13) y se le asigna al documento la clase al de la mayoría de esos k documentos.

La similitud del coseno está dada por la siguiente ecuación:

$$sim(d_i, d_j) = \frac{d_i \cdot d_j}{||d_i|| \times ||d_j||} \tag{2.13}$$

donde d_i y d_j son los vectores de los documentos a comparar.

2.3.3. AdaBoost

Es un método de clasificación tipo Boosting, en el cual combinan diferentes clasificadores de forma iterativa para obtener mejores resultados (Freund and Schapire, 1996). El clasificador final está compuesto de varios clasificadores "débiles" los cuales se han generado a través del proceso de aprendizaje (Cortes and Vapnik, 1995; Freund, Y. Schapire, 1996).

AdaBoost es adaptativo en el sentido de que los clasificadores construidos posteriormente se ajustan a favor de los ejemplos mal clasificados por los clasificadores anteriores. AdaBoost es sensible al ruido en los datos y valores atípicos, sin embargo, puede ser menos susceptible al problema de sobre-ajuste que la mayoría de algoritmos de aprendizaje. Los clasificadores que utiliza pueden ser débiles, es de-

cir, mostrar una tasa de error considerable, pero siempre y cuando su rendimiento no es al azar (que resulta en una tasa de error de 50 % para la clasificación binaria, que mejorará el modelo final.

Algoritmo 2.1 AdaBoost

Entrada: Instancias D;

Algoritmo de aprendizaje L;

Número de iteraciones T

Salida: H(x)=CombinaSalidas(h_t(x))

1: $D_1 = D$; % Se inicializan los ejemplos con el mismo peso
2: **Para** $t = 1, \cdots, T$ **hacer**
3: $h_t = L(D_t)$; % Se entrena el clasificador con la distribución D_t
4: $\epsilon_t = Pr_{x\ D_{t,y}} I[h_t(x) \neq y]$; % Medida de error de h_t
5: $D_{t+1} = AjustarDistribucion(D_t, \epsilon_t)$
6: **Fin Para**

2.3.4. Random Forest

Al igual que AdaBoost Random Forests es un clasificador tipo ensamble, que consiste de varios árboles de decisión, tal que cada uno de los árboles depende de los valores de un vector aleatorio independientemente y con la misma distribución para todos los árboles. Cada árbol emite su voto para la clase más popular para una entrada x dada, y al final la salida de RF se realiza usando voto mayoritario. (Breiman, 2001)

Algunas de las principales ventajas de este método son:

- Proporcionar una buena capacidad predictiva aún cuando exista mucho ruido en la información.

- El problema de sobre ajuste en los datos no es tan severo
- Establece un ranking de relevancia de las variables.

Ahora bien, los árboles se construyen a partir de muchos conjuntos de datos similares generados mediante *bootstrap*, es decir, haciendo re-muestreo con remplazo sobre el conjunto de entrenamiento. Con esto, se consigue suavizar el error de predicción y además de crear los árboles a partir de muestras independientes. Cabe mencionar que para cada nodo no se selecciona la mejor variable, sino que se selecciona al mejor subconjunto aleatorio de m atributos.

Algoritmo 2.2 Random Forests

Entrada: IDT árbol de decisión, T número de iteraciones, S conjunto de entrenamiento, μ tamaño de la sub-muestra, N número de atributos usados en cada nodo.

Salida: $M_t; t = 1, \cdots, T$

1: **Para** $t = 1, \cdots, T$ **hacer**
2: S_t= muestra con μ instancias de S con remplazo
3: Construir el clasificador M_t usando IDT(N) en S_t
4: **Fin Para**

2.3.5. Máquinas de vectores de soporte

Las máquinas de vectores de soporte (SVM, por sus siglas en inglés) es un método de clasificación supervisada propuesto Vapnik et. al (Cortes and Vapnik, 1995). Este método tiene su base en la teoría de aprendizaje estadístico. La idea detrás de este método consiste buscar hiperplanos que maximicen el margen entre dos clases, separando los ejemplos de entrenamiento positivos de los negativos.

La figura 2.1 muestra la idea principal en datos linealmente separables, la distancia que existe entre las lineas punteadas se le llama *margen* y los puntos más cercanos al hiperplano son llamados *vectores de soporte*.

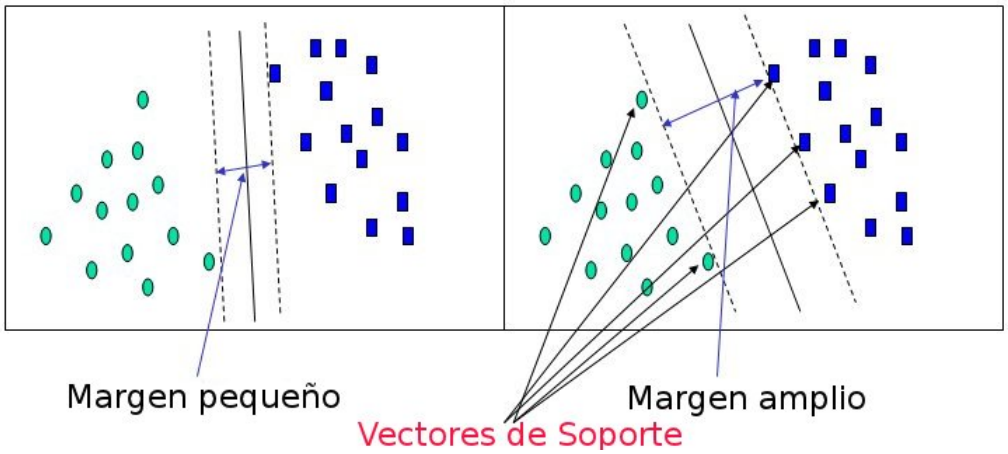


Figura 2.1: SVM, el hiperplano seleccionado es el que maximiza los márgenes, es decir, la distancia entre los puntos de entrenamiento más cercanos.

Un punto importante relacionado con SVM, es que transforma el espacio original en otro espacio de alta dimensionalidad llamado *espacio de características*, el cual busca maximizar la distancia entre los puntos negativos de los positivos, Figura 2.2.

SVM ha mostrado conseguir un buen desempeño sobre una amplia variedad de problemas, en particular dentro de la clasificación de textos también se han conseguido buenos resultados (Cardoso-Cachopo and Oliveira, 2003; Joachims, 1998). Para más información consultar (Joachims, 1998).

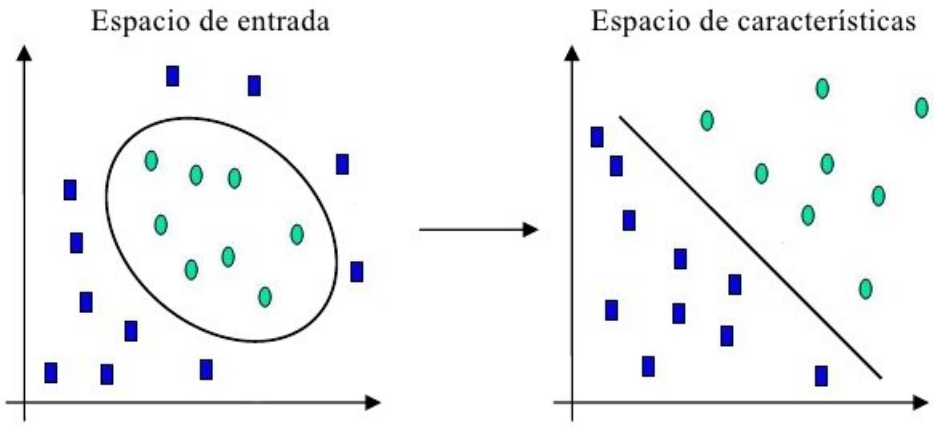


Figura 2.2: Transformación de los datos originales (no lineales) a un espacio de mayor dimensión

2.4. Evaluación

En la mayoría de las ocasiones se requiere evaluar de alguna forma a nuestros algoritmos de aprendizaje computacional, debido a que se requiere saber que tan bueno es respecto a otros que han demostrado ser buenos o simplemente para saber que tan exactos son. La forma más común de realizar esta evaluación es mediante la predicción sobre nuevas instancias, es decir, predecir la clase de ejemplos diferentes al conjunto con el cual se realizó el entrenamiento, al que se le llama *conjunto de evaluación o pruebas*.

Muchas medidas han sido propuestas y usadas, cada una de ellas evalúa algún aspecto específico del desempeño del sistema de categorización. En esta sección describiremos algunas medidas de evaluación que han sido reportadas en la literatura y que fueron usadas en este trabajo de investigación.

Consideremos el problema de la clasificación con dos clases (A,B). En la tabla se muestran los resultados de la predicción que realiza el sistema de clasificación.

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

donde:

- a - es el número de documentos correctamente asignados a su categoría (verdaderos positivos)
- b - es el número de documentos incorrectamente asignados a su categoría (falsos positivos)
- c - es el número de documentos incorrectamente rechazados a su categoría (falsos negativos)
- d - es el número de documentos correctamente rechazados a su categoría (verdaderos negativos)

Dados estos valores, definimos las siguiente medidas: Precisión (P), Recuerdo (R), *F-measure*.

$$R = \frac{a}{a+c} \tag{2.14}$$

$$P = \frac{a}{a+b} \tag{2.15}$$

$$Fmeasure = \frac{2 \cdot P \cdot R}{P+R} \tag{2.16}$$

Donde el *recuerdo* se puede interpretar como la probabilidad de que si un documento d_p pertenece a una categoría C, el sistema lo asigne a la categoría correcta. Mientras que la *precisión* es la probabilidad de que un documento d_p sea etiquetado con la categoría C y éste realmente pertenezca a esa categoría. Por último

F-measure, es una medida que combina la precisión y el recuerdo. Si bien estas medidas nos dan una idea del desempeño del clasificador por categoría, existen otras dos medidas que nos ofrecen una visón global del desempeño del clasificador: *macro promedio y micro promedio*. La primera se llama *micro-promedio*, en la cual se le da la misma importancia a cada documento, es decir, las categorías pequeñas tienen poco impacto en el resultado final (Eikvil and Aas, 1999).

$$microR = \frac{\sum_{j=1}^{k} a}{\sum_{j=1}^{k}(a+c)} \tag{2.17}$$

$$microP = \frac{\sum_{j=1}^{k} a}{\sum_{j=1}^{k}(a+b)} \tag{2.18}$$

$$microFmeasure = \frac{2 \cdot microP \cdot microR}{microP + microR} \tag{2.19}$$

La segunda, *macro-promedio*, donde se le da la misma relevancia a las clases.

$$macroR = \frac{1}{k}\sum_{i=1}^{k}\frac{a_j}{a_j + c_j} \tag{2.20}$$

$$macroP = \frac{1}{k}\sum_{i=1}^{k}\frac{a_j}{a_j + b_j} \tag{2.21}$$

$$macroFmeasure = \frac{2 \cdot macroP \cdot macroR}{macroP + macroR} \tag{2.22}$$

Algunas ocasiones no se cuenta con el conjunto de evaluación independiente del conjunto de entrenamiento, por lo que se tiene que aplicar una técnica llamada *validación cruzada*. Esta es una técnica utilizada para evaluar los resultados mediante un análisis estadístico y garantizar que los datos de entrenamiento son independientes de las de prueba (Kohavi, 1995). La validación cruzada consiste en dividir el conjunto de datos D en k particiones mutuamente excluyentes $D_1, D_2, \cdots, D_k$, conteniendo cada uno aproximadamente el mismo número de ejemplos. Una vez

realizado esto, se entrena al clasificador con k-1 subconjuntos y el restante se utiliza para evaluar los resultados. Este proceso se realiza k veces, y al final se obtiene un promedio de los resultados obtenidos.

Capítulo 3

Trabajos relacionados

La mayoría de la información empresarial se origina y almacena como texto, y gracias a la Web esta información se puede difundir de forma fácil, brindando diversas oportunidades para el uso de ésta información. Recientemente, ha surgido la necesidad de la clasificación de textos cortos, debido al crecimiento inmensurable de este tipo de documentos, gracias a la popularidad de los servicios Web. Gran parte de la información a ser procesada es tomada de repositorios donde los documentos frecuentemente son textos cortos, como son: los resúmenes científicos, noticias, *feeds*, entre otros. Un ejemplo de la necesidad de procesamiento de los textos cortos es la organización de los artículos científicos existentes en bibliotecas digitales y en repositorios en linea, ya que la mayoría está limitado a mostrar solo los resúmenes científicos, complicando la tarea de organizar la información en categorías. Por otro lado, la necesidad del estudio de los textos cortos se ha incrementado gracias a la actual forma en que gran parte de la gente usa el lenguaje, es decir, el lenguaje reducido que provee la tendencia tecnológica como son los correos electrónicos, *blogs, chats* y redes sociales.

En años recientes varios estudios han reconocido la relevancia y complejidad

de la clasificación de textos cortos (STC) (Makagonov et al., 2004) y además, se han propuesto diferentes métodos de diferente naturaleza que serán expuestas más adelante en este capítulo . En este capítulo presentamos y dividimos el trabajo relacionado en tres secciones. Primero, presentamos aquellos que utilizan una representación basada en la bolsa de palabras, en donde algunos trabajos realizan desde una selección de atributos hasta una expansión de términos a los documentos. La idea principal en la que se basan la mayoría de estas soluciones, es agregar nuevos términos no existentes en el documento original, que estén de alguna forma relacionados a los términos de éste, con el objetivo de solventar el problema de longitud del documento. La forma de encontrar los términos que se deben agregar a cada documento, es lo que diferencia a cada trabajo, algunos buscan nuevos términos en recursos externos, como por ejemplo: *WordNet*, Wikipedia, etc (Fan and Hu, 2010a; Rosas et al., 2010; Shim et al., 2008; Zelikovitz, 1999), y otros dentro del mismo conjunto de entrenamiento (Cai et al., 2008; Fan and Hu, 2010b; He et al., 2011; Rosa and Ellen, 2009; Xi-wei, 2010). En el segundo enfoque se encuentran los métodos que llevan la representación de los documentos a otro espacio de representación, donde el objetivo es descubrir nuevas relaciones no obtenidas en la representación original (Phan et al., 2011; Pu and Yang, 2006; Rosas et al., 2010; Shea et al., 2008; Zelikovitz, 2004). La metáfora que subyace es que por medio del cambio de espacio de representación, se obtiene una representación adecuada de las relaciones existentes entre las palabras en un corpus textual, en el cual estas relaciones son muy débiles debido al gran número de palabras. Dentro del tercer enfoque, están aquellos trabajos que realizan una modificación/adaptación de algoritmos de clasificación para poder lidiar con los textos cortos.

En el resto del capítulo detallaremos cada uno de los grupos de trabajo relacionado, y se mencionarán algunas ventajas y desventajas que tienen estos grupos,

así como las diferencias que existen con el método propuesto en esta tesis.

3.1. Representación basada en bolsa de palabras

La mayoría de los trabajos realizados para la clasificación de textos cortos se enfocan en el enriquecimiento del contenido de los documentos, para tratar de solventar el problema de longitud de los documentos cortos. El enriquecimiento consiste en agregar términos no existentes dentro del documento original, pero que estos términos tengan una relación con los términos contenidos en el documento original. Por un lado, algunos se basan en la ayuda de recursos externos (Fan and Hu, 2010a; Nagarajan et al., 2007; Wang et al., 2009; Xi-wei, 2010), con la finalidad de enriquecer las representaciones, es decir, agregan términos existentes en el recurso externo que están relacionados de alguna forma a los del documento.

En (Shen et al., 2009) la idea propuesta para solventar el problema de textos cortos es obtener otro conjunto de datos, en este caso una taxonomía y buscar relaciones entre términos. Posteriormente, le asignan mayor importancia a aquellas relaciones de términos que co-ocurran en la misma clase que a las que co-ocurran en diferentes clases; además estima la relevancia de la relación de los términos basado en la distribución de los términos sobre diferentes categorías jerárquicas todo esto con la ayuda de una taxonomía. La ventaja de este método es que permite encontrar co-ocurrencias muy relacionadas a la categoría al usar la taxonomía, y además asigna mayor relevancia a las co-ocurrencias que pertenecen a una sola clase. Sin embargo, tiene una fuerte dependencia a un recurso externo y los resultado obtenidos no muestran una mejora significativa con respecto a los clasificadores base.

La clasificación de textos cortos se ha abordado en su mayoría haciendo uso de

recursos externos. Aunque se han obtenido resultados aceptables existe un problema con este tipo de trabajos, el problema radica principalmente en conseguir un recurso externo que pueda ayudar en la clasificación. El recurso externo debe tener una adecuada relación temática con el corpus objetivo, de lo contrario introducirá ruido (ambigüedad de las palabras) (Fan and Hu, 2010b). Es precisamente este argumento el que es presentado en los siguientes trabajos (Cai et al., 2008; Fan and Hu, 2010b; He et al., 2011; Rosa and Ellen, 2009; Sriram et al., 2010), donde se centraron en la clasificación de textos cortos sin enriquecer los documentos con recursos externos.

La idea de (Fan and Hu, 2010a) está orientada a expandir los documentos, su hipótesis era que al agregar términos a los documentos el rendimiento de la clasificación de textos cortos mejorará. La expansión la realizó utilizando el conjunto de entrenamiento como se propone en (Fan and Hu, 2010b). El autor realiza dos procesos: agrega nuevas características, "términos" a los documentos, y modifica los pesados de acuerdo a la relación de los términos que contiene el documento y los nuevos términos que se agregan. El proceso consiste en obtener 3 vectores, el primero es el original, el cual contiene las características del documento, el segundo son las características extraídas de otro recurso externo pero que también se encuentra en el original, y por último son las características que no están en el vector original pero están en la librería externa. Una de las ventajas es que pueden asignarles una mayor relevancia a cada vector dependiendo de la tarea y controlar el grado de expansión de los vectores. Sin embargo, los resultados mostrados por su método no muestran ser significativamente mejores que BOW y además, cuentan con tres parámetros libres que deben ajustarse para cada corpora que se tenga.

Mientras algunos se basaron en la idea de expandir los documentos, otros au-

tores se enfocaron en la idea de utilizar los atributos más relevantes, realizando una selección de atributos (Rosa and Ellen, 2009) o utilizando recursos estilísticos (Sriram et al., 2010). (Rosa and Ellen, 2009) planteó que realizando una buena selección de atributos se podrían obtener desempeños aceptables. Los autores se centraron en el análisis de los mensajes enviados por medio del *chat* militar interno de los Estados Unidos, con la idea de filtrar los textos con información de importancia para la naval. Realizaron una comparación de cuatro diferentes métodos de selección de atributos y cuatro clasificadores. Sin embargo, no se lograron superar los resultados obtenidos por el método base (BOW). Finalmente, (Sriram et al., 2010) tiene la hipótesis de que usando recursos estilísticos como *slangs*, énfasis en las palabra, símbolos como el de porcentaje y el de pesos, mejorará el desempeño de la clasificación de *Tweets*. Los resultados indicaron que el método que propusieron mejoró significativamente los resultados base.

3.2. Cambio de espacio de representación

En general, por cambio de espacio de representación se entenderá a aquellos métodos que producen características "artificiales" a partir de las características originales (uniendo características, creando nuevas, etc).

Los siguientes trabajos están inspirados en la idea de aplicar alguna transformación a la representación de los documentos, para llevar un espacio de documentos-términos a otro espacio, en donde puedan descubrir relaciones entre términos o alguna otra relación y en algunos casos no necesariamente tiene una interpretación clara. La idea intuitiva de estos trabajos es buscar relaciones entre los términos que no están tan claras en la representación original, debido al gran número de términos. Por lo que al cambiar la representación y llevarlo a otro espacio de

atributos se fortalecen estas relaciones y permite reducir el ruido existente.

La idea detrás de lo propuesto por (Rosas et al., 2010) es realizar un indexado por sentido de las palabras, ya que los términos se ven afectados por la sinonimia y la polisemia. Por lo que se necesita capturar el concepto en lugar del término y para esto utilizaron tres métodos de desambiguación del sentido de las palabras (WSD, por sus siglas en inglés) como son: *CIAOSENSO*[1], *LESK*[2] y el *Método del sentido más frecuente*[3]; esto con la ayuda de WordNet. El objetivo dentro de este trabajo es, determinar si la incorporación de la información obtenida mediante los métodos de WSD (al que llaman conceptos) en la representación de documentos, ayuda a mejorar la tarea de categorización de textos cortos. Los autores comparan tres enfoques, 1) el enfoque tradicional basado en términos (BOW), 2) el basado en "conceptos"[4] y 3) la combinación de estos dos. De manera general lo que hacen es, para cada término contenido en el documento se desambigua utilizando alguno de los algoritmos anteriormente mencionados, el resultado (el sentido del término) es sustituido por el término original, si no se puede desambiguar únicamente se elimina el término original, a esto se le llamará "conceptos". El último enfoque, combina los dos anteriores "términos + conceptos", dando los mejores resultados. Por lo que se puede decir que el uso de la representación semántica en la representación de los documentos, a través de métodos basados en conocimiento, puede

[1]CIAOSENSO: Sistema basado en la idea de densidad conceptual. Para esto, utiliza la longitud del camino más corto que conecta dos synsets en la taxonomía de sustantivos que utiliza WordNet.

[2]LESK: Procedimiento que determina los sentidos de las palabras que ocurren en un contexto particular basándose en una medida de solapamiento entre las definiciones de un diccionario y dicho contexto.

[3]Sentido más frecuente: Es la técnica más simple de desambiguación asignando a una palabras el sentido que ocurre más a menudo de todos los posibles sentidos de esa palabra

[4]En (Rosas et al., 2010) **conceptos** es el sentido regresado por algún WSD.

ser benéfico y robusto ante una reducción de vocabulario. Sin embargo, debido a los resultados presentados no se puede determinar un WSD óptimo para esta tarea, debido a que no existió alguno que fuera el mejor en la mayoría de los experimentos.

(Phan et al., 2008, 2011) utilizaron un recurso externo con la idea de buscar que los documentos cortos estén más relacionados temáticamente y para que al expandir los documentos, los clasificadores tengan mayor poder de generalización para la toma de decisiones para datos futuros. La idea principal de este trabajo radica en que, para cada tarea de clasificación, se tiene que obtener un conjunto muy grande de datos llamados "*universal dataset*", y después construir un clasificador que tome tanto los datos de entrenamiento como los temas descubiertos dentro del *universal dataset* por algún método. Para realizar el descubrimiento de los temas en el conjunto externo, se utilizaron dos métodos, *Probabilistic Latent Semantic Analysis* (pLSA) y Latent Dirichlet Allocation (LDA). De manera general, pLSA es un técnica estadística para el análisis de co-ocurrencia de datos muy similar a lo visto en la sección 2.2.1 al cual se le agregó el modelo probabilístico, para mayor detalle consultar (Hofmann, 1999). Por otro lado, LDA es un modelo de tres niveles jerárquico bayesiano, en el que se modela cada elemento de una colección como una mezcla finita sobre un conjunto subyacente de los temas, para mayor información ver (Blei et al., 2003). En LDA cada documento puede ser visto como una mezcla de temáticas diferentes. Es muy similar a LSA, excepto que en LDA la distribución de la temática se asume que tiene una distribución Dirichlet, para mayor información ver (Girolami and Kabán, 2003). Una de las conclusiones importantes está relacionada con el conjunto externo, donde comentan que es una de las más grandes dificultades, ya que los temas que descubren esta íntimamente ligado con la naturaleza de este conjunto de datos y en muchas ocasiones no es

fácil descubrir temas que ayuden a la tarea.

Una de las técnicas con las que han intentado enfrentar este problema de clasificación de textos cortos sin utilizar recursos externos, se basa en el uso del método conocido como *transductive learning* (Zelikovitz, 2004). Donde se intenta sacar provecho de los documentos de prueba, para enriquecer las representaciones y de esta forma solventar el problema de *representaciones dispersas*. En la metodología de *transductive learning*, se unen los conjuntos de entrenamiento (C_T) y prueba (C_P) en uno solo (C_{TP}). (Marquez and Finella, 2005) usa la metodología *transductive learning* en combinación con LSI, con la idea principal de compactar la representación obteniendo un espacio k-dimensional relevante mediante LSI, y de esta forma las representaciones de los documentos mejoren.

Finalmente, (Pu and Yang, 2006) utilizaron tanto análisis de componentes principales (ICA, por sus siglas en inglés) como LSA. Primero, usa LSA para el pre-procesamiento de texto, es decir, para crear una representación reducida que contenga los conceptos más relevantes y después utilizan ICA como clasificador (Comon, 1994). Todos los experimentos realizados fueron sobre textos chinos. ICA[5] se utiliza en TC con el supuesto de que un corpus de documentos es generado por una combinación de varios tópicos temáticos. Esto es, los componentes independientes obtenidos por ICA definen una agrupación de documentos estadísticamente independiente, permitiendo su clasificación temática. El argumento en el cual se basan para usar estos dos métodos radica en que ICA por si solo no obtiene buenos resultados, por lo que también proponen el uso de LSA.

[5]ICA: es un método para representar un conjunto de observaciones multi-variados como una combinación lineal de variables latentes desconocidas que son estadísticamente independientes, el cual se ha utilizado para el análisis de textos (Hyvrinen, 1999).

3.3. Modificación/adaptación del proceso de clasificación

Otros investigadores no han intentado mejorar la representación de los documentos, en su lugar se enfocan en mejorar la forma en que el clasificador genere el modelo para la toma de decisiones, (Faguo et al., 2010; Ramírez-de-la Rosa et al.; Zelikovitz, 1999).

(Zelikovitz, 1999) usa el conjunto de entrenamiento y además un conjunto externo de documentos que utiliza como puente para encontrar un documento que sea muy similar a un documento etiquetado y al documento que se quiera etiquetar. Una característica relevante de éste método es que el conjunto externo debe tener una cercanía temática al conjunto de entrenamiento. El método que propone se basa en la idea que un ejemplo de entrenamiento puede servir para clasificar a un documento de prueba, solo si existe un documento dentro del conjunto externo que sea muy similar a un documento de entrenamiento y uno de prueba, al cual llamaron "*second-order*". En otras palabras, dado un ejemplo de prueba se genera una consulta, y se recuperan K documentos más parecidos dentro de la base de datos sin etiquetar. Posteriormente se busca dentro del conjunto de entrenamiento aquellos documentos que sean más parecidos a los k-documentos regresados anteriormente y finalmente se toma un consenso para la asignación de la clase. Utilizaron tres conjuntos de datos para evaluar su método (Noticias, resúmenes científicos, páginas Web), resultando en bajos porcentajes de error. Sin embargo, un tema importante es el relacionado con el conjunto externo de datos, ya que este afecta considerablemente los resultados, por lo que proponen mejorar la forma de obtener este recurso externo.

(Ramírez-de-la Rosa et al.) propusieron un método donde la clase de un documento no se determina únicamente por el contenido del documento, si no que también por la clase asignada a los documentos que se le parecen. ste método consta de tres etapas: i) El entrenamiento se realiza usando un clasificador basado en prototipos. ii) Una etapa pre-prueba: donde se buscan los K-vecinos más cercanos a cada documento dentro del conjunto de prueba y dependiendo de la clase asignada a sus vecinos. iii) Finalmente, se realiza un consenso entre la clase asignada a los vecinos y la clase asignada por el clasificador basado en el contenido del documento. El principal problema de este método es encontrar el número de vecinos adecuado, la ponderación acerca de la decisión (a quien hacerle más caso, a los vecinos o al clasificador), así como el manejo de representaciones tan dispersas que generan los textos cortos.

La idea del método propuesto por (Faguo et al., 2010) es crear un método que esté basado en reglas y que estas coincidan con el texto, es decir, si los términos característicos de las reglas aparecen en el mismo orden dentro del documento, entonces este documento cumple las reglas y se le asigna esta clase. Proponen un clasificador de textos cortos basado en dos etapas: en la primera se realiza una extracción de características mediante el método de χ^2. Primero realizan un proceso de etiquetado POS (Partes de la oración) y se realiza el cálculo de la frecuencia de los términos. Después, se aplica la selección de atributo χ^2 [6] La principal ventaja de este enfoque es que no requiere de recursos externos, sin embargo, de acuerdo a los resultados presentados (Faguo et al., 2010) únicamente mejora en cuanto a la medida de *recuerdo* mientras que en la precisión es baja. Otro de lo problemas que podría tener es cuando los conjuntos tienen clases muy

[6] χ^2: es una medida de dependencia entre la palabra F y la categoría C. Si la palabra F y la categoría C son independientes, el resultado será cero.

similares, debido a que las reglas creadas mediante partes de la oración pueden ser muy similares, al igual que los términos y por lo consiguiente no tener reglas que discriminen adecuadamente.

3.4. Discusión

Los trabajos presentados anteriormente están divididos en tres enfoques principales, aquellos que utilizan la representación de bolsa de palabras, los que modifican algún proceso de clasificación y los que realizan un cambio de espacio de representación. Algunos trabajos que utilizan la representación de bolsa de palabras se enfocaron a utilizar un recurso externo, con la idea de que este conjunto les ayude a enriquecer la representación de los documentos cortos, y así solventar el problema de la longitud, obteniendo resultados aceptables. Sin embargo, el problema principal de este enfoque es obtener el recurso externo, debido a que se debe considerar la calidad del conjunto externo, de lo contrario se puede introducir ruido y no beneficiar la clasificación. Otra limitante, en el caso de los que usan taxonomías es el hecho de que la información puede ser inconsistente, es decir, en caso de algunos dominios las palabras buscadas pueden no encontrarse dentro de estos recursos externos. En suma, cada uno de estos métodos solo funcionan en la tarea-contexto específica con la que fueron creados.

Debido a que los métodos que entran en el enfoque de modificación/adaptación del proceso de clasificación no modifican la representación, estos trabajos no son excluyentes de usarse en combinación con los otros dos enfoques ni mucho menos con nuestro método, por lo que podrían usarse conjuntamente en experimentos futuros.

Los trabajos que están en la categoría de cambio de espacio de representación

y usan recursos externos tienen los mismos problemas que anteriormente se mencionaron. Por otro lado, el resto de los trabajos que no hace uso de recursos externos tienen la ventaja de no depender del dominio (Pu and Yang, 2006; Zelikovitz, 2004), pero se utilizó LSI, que tiene el inconveniente de reajustar algunos parámetros cada vez que se cambie de dominio y es propenso al sobre-ajuste (Blei et al., 2003).

El método propuesto en esta tesis se ubica en el enfoque de cambio de representación, pero a diferencia de la mayoría de los trabajos expuestos anteriormente, el método que proponemos no hace uso de recursos externos. Además, no tiene parámetros libres necesarios, como LSI y LDA, el cual es calculado empíricamente. Además, las relaciones de términos encontradas por LSI se basan en todo el conjunto de datos, lo que implica que el enriquecimiento de los documentos se termina representado con base a lo que es relevante en todo el corpus y no a cada documento.

El método propuesto se basa en el uso de las representaciones distribucionales de términos (DTRs) para la expansión o enriquecimiento de la representación de los documentos cortos. Las DTRs son herramientas para la representación de términos que se basan en estadísticas de ocurrencia en documentos y co-ocurrencia con otros términos. El método más cercano al nuestro es LSI, debido a los dos toman co-ocurrencias entre los término para generar la representación y los dos cambian el espacio de representación. Por otro lado, las diferencias existentes con respecto a nuestro método son: i) el método propuesto no tiene parámetros libres como los tiene LSI, ii) La representación generada por LSI no se puede interpretar fácilmente y iii) Se espera que el método propuesto sea más robusto que LSI, ya que LSI requiere de un gran conjunto de documentos para obtener resultados aceptables (Deerwester and Dumais, 1990; Paper, 2000).

Capítulo 4

Clasificación de textos cortos mediante representaciones distribucionales de los términos

En este capítulo se describe el método que se desarrolló para la clasificación de textos cortos basados en representaciones distribucionales de los términos, así como una variante del mismo. Primero se presenta una breve introducción al método propuesto, donde se explica de manera general el método y la idea detrás del mismo. Posteriormente se presenta con mayor detalle cada etapa del método propuesto, y finalmente, una variante de éste método, en donde se menciona la hipótesis de la cual se parte.

4.1. Introducción

Las representaciones distribucionales de los términos (DTRs), son herramientas para la representación de los términos que se basan en la estadística de ocur-

rencia y co-ocurrencia de los términos (LAVELLI et al., 2004). La idea detrás de las DTRs es que el significado de un término puede ser deducido por su contexto; donde el contexto de un término está determinado por otros términos que co-ocurren con frecuencia o por aquellos documentos en el cual el término ocurre.

Las DTRs han sido ampliamente usadas en lingüística computacional en tareas que incluyen el procesamiento de términos, como por ejemplo: agrupamiento de términos (Lewis and Croft, 1990), construcción automática de tesauros (Chen et al., 1995; Crouch, 1990), desambiguación del sentido de las palabras(Gale et al., 1992; Gallant, 1991), recuperación de información (Ruiz, 2010) y recientemente en la recuperación de información multimedia de forma multimodal (Escalante et al., 2011).

Para obtener DTRs informativas se necesitan grandes cantidades de información (documentos extensos) con los que no se cuenta en la mayoría de las ocasiones para la clasificación de textos cortos. Sin embargo, existen escenarios en los que si se cuenta con documentos suficientemente extensos para la etapa de entrenamiento, aunque en la etapa de prueba solo se cuente con documentos cortos. Algunos de los escenarios donde se presenta este tipo de situaciones son: artículos científicos, noticias, blogs, descripción de productos. Es precisamente en los dos primeros escenarios mencionados donde pretendemos abordar la problemática de textos cortos. Donde se plantea el escenario siguiente: suponemos que tenemos los documentos completos (artículos, noticias) para la etapa de entrenamiento y solamente los resúmenes o títulos de los documentos (artículos, noticias) para clasificar, a esta configuración le llamaremos Documento-Título (DT), mientras que la configuración estándar la denotaremos por DD.

A continuación se presentarán algunos ejemplos de escenarios que cumplen nuestra suposición: **1) artículos científicos**: Existen centros de investigación en

donde diariamente se reciben cientos de resúmenes de publicaciones que deben ser clasificadas de acuerdo a ciertas clases predefinidas, por lo que se podría realizar el entrenamiento utilizando los artículos completos de las categorías con los que se puede contar en un momento determinado y clasificar únicamente con los resúmenes que van arribando. *2)**noticias:*** El entrenamiento de la clasificación de noticias se puede realizar utilizando las noticias completas (obtenidas con anterioridad) y la etapa de clasificación sería utilizando un resumen o título de la noticia.

En este trabajo proponemos el uso de DTRs para la clasificación de textos cortos. La idea central es expandir el contenido de los documentos mediante DTRs, con el fin de solventar el problema de pocos términos y bajas frecuencias. El método propuesto consiste de 3 pasos:

- Obtener las DTRs de términos. La idea intuitiva es encontrar relaciones de los términos con otros términos o con los documentos y generar una representación de cada término.

- Obtener la representación de documentos. La idea es enriquecer los documentos a partir de las relaciones encontradas en el paso anterior.

- Proceso de clasificación. Consiste en proporcionar al algoritmo de clasificación las representaciones de los documentos enriquecidas para generar el modelo.

Para dar solución a la problemática planteada en este trabajo, se estudian dos formas de obtener DTRs de términos, las cuales están basadas en la co-ocurrencia con otros términos en el vocabulario o en la ocurrencia de los términos en los documentos de la colección. Al resultado del cálculo de las DTRs se denominan

conceptos, por que generan una representación del significado de cada término con respecto a los documentos u otros términos. Una vez que se obtienen los conceptos, se procede a generar la representación de los documentos, esto se realiza simplemente combinando los conceptos de los términos que contenga cada documento.

El esquema general del método propuesto es presentado en la figura 4.1. En este esquema existen dos procesos principales: la **Representación de términos** y **Representación de documentos** (BoC).

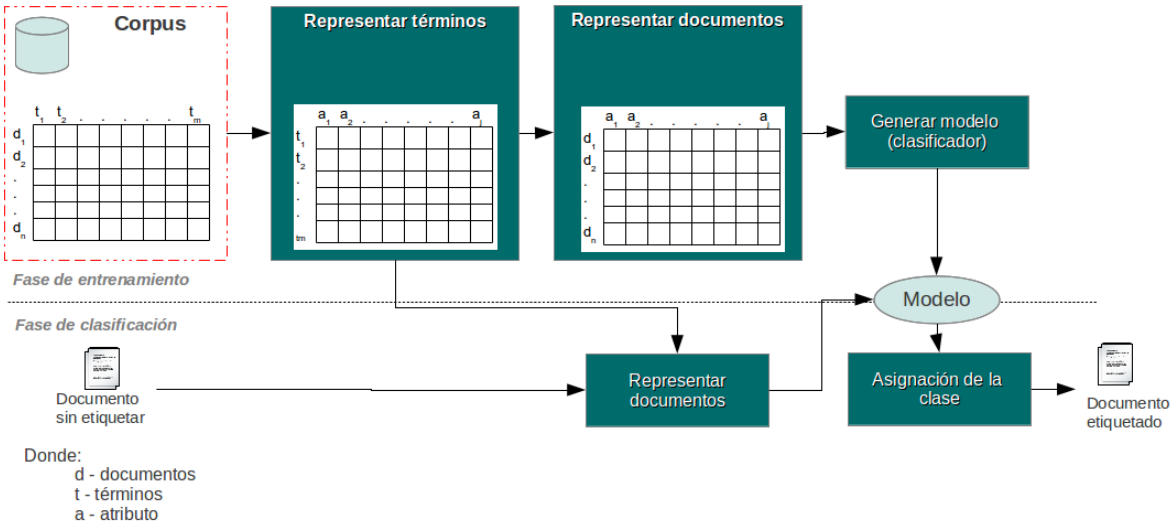


Figura 4.1: Esquema general del método propuesto para la tarea de clasificación de textos cortos

En términos generales, la **representación de términos** consiste en realizar el cálculo de las estadísticas de co-ocurrencia/ocurrencia de los términos sobre el conjunto de entrenamiento. Donde cada fila de la matriz resultante de DTRs representa el concepto de un término existente en la colección de entrenamiento, y cada columna (a_i) representa la co-ocurrencia con otro término o la ocurrencia en

un documento. Posteriormente, el proceso de **representación de documentos**, es un proceso en que se combinan los conceptos que aparecen en el documento original.

4.2. Método propuesto

En las secciones siguientes se provee una descripción más detallada de cada uno de los procesos involucrados en el método propuesto figura 4.1.

4.2.1. Representaciones distribucionales de los términos (DTRs)

En este trabajo consideramos dos DTRs como son: representación basada en ocurrencia de documentos (*document ocurrence representation*, DOR) y representación basada en co-ocurrencias entre términos (*term coocurrence representation*, TCOR), las cuales se explican a continuación:

- **Representación basada en ocurrencia de documentos (DOR).** La representación basada en ocurrencia de documentos es considerada el dual del pesado TF-IDF, ver ecuación (2.3). DOR se basa en la idea de que la semántica de un término puede ser representada por una distribución de ocurrencias sobre los documentos de la colección.

  Un término t_j es representado como un vector de ponderación de documentos $\vec{w_j} = \langle w_{j1}, \ldots, w_{jN} \rangle$, donde cada elemento esta asociado a un documento, indicando la relevancia del documento para la representación del término, donde N es el número de documentos en la colección y $0 \leq w_{kj} \leq 1$ representa la contribución del documento d_k a la semántica de t_j:

$$f(d_k, t_j) = df(d_k, t_j) \cdot log \frac{|T|}{\#\tau(d_k)} \tag{4.1}$$

donde $\#\tau(d_k)$ es el número de términos diferentes en el diccionario T que aparecen en el documento d_k y

$$df(d_k, t_j) = \begin{cases} 1 + log\#(d_k, t_j) & if\#(d_k, t_j) > 0 \\ 0 & \text{caso contrario} \end{cases} \tag{4.2}$$

donde $\#(d_k, t_j)$ denota el número de veces que el término t_j ocurre en el documento d_k. Los pesos son normalizados usando la *normalización del coseno*, finalmente resultando:

$$w_{kj} = \frac{f(d_k, t, j)}{\sqrt{\sum_{s=1}^{N} f(d_s, t_j)^2}} \tag{4.3}$$

Intuitivamente, mientras más frecuente sea el término t_j en el documento d_k, más importante es d_k para definir la semántica de t_j; Mientras más términos contenga d_k, menos contribuye a la caracterización de la semántica de t_j.

- **Representación basada en co-ocurrencia de términos (TCOR).** La representación de co-ocurrencia de términos es similar a DOR, aunque la idea principal es de que la semántica de un término t_j se puede obtener a partir de la asociación de un término con el vocabulario de la colección. En TCOR, cada término $t_j \in T$ es representado por un vector de pesos $\vec{w_j} = \langle w_{j1}, \ldots, w_{jM} \rangle$, donde M es el número de términos en la colección y $0 \leq w_{kj} \leq 1$ representa la contribución del término t_k a la semántica de t_j.

$$f(t_k, t_j) = tff(t_k, t_j) \cdot log\frac{|M|}{\#\tau(d_k)} \tag{4.4}$$

donde $\#\tau(d_k)$ es el número de diferentes términos en el diccionario M que co-ocurren con t_j en al menos un documento y

$$tff(t_k, t_j) = \begin{cases} 1 + log\#(t_k, t_j) & if\#(t_k, t_j) > 0 \\ 0 & \text{caso contrario} \end{cases} \tag{4.5}$$

donde $\#(t_k, t_j)$ denota el número de veces que el término t_j co-ocurren con el término t_k. Los pesos son normalizados usando la *normalización del coseno*.

$$w_{kj} = \frac{f(d_k, t, j)}{\sqrt{\sum_{s=1}^{N} f(d_s, t_j)^2}} \tag{4.6}$$

La idea intuitiva detrás de este tipo de pesado es que mientras más t_k y t_j co-ocurran, más importante es t_k para describir el término t_j; Por otro lado, mientras más sea el número de términos que co-ocurran con t_k, menos contribuirá en la caracterización del término t_j.

4.2.2. Representación de documentos

Hasta ahora, hemos descrito como se realizan los cálculos de las DTRs, sin embargo, estas representaciones solo son a nivel de términos y no a nivel documentos, que es lo que se requiere.

Sahlgren y Coster (Sahlgren and Cöster, 2004) introducen una representación que llaman *bolsa de conceptos* (BoC) basada en la intuición de que el significado de los documentos puede expresarse como la unión del significado de sus términos. En nuestro caso, utilizando información de ocurrencia y co-ocurrencia, se genera un vector de concepto para cada término, por lo tanto la representación del documento será la suma ponderada de los vectores de concepto de sus términos.

$$d_i^{dtr} = \sum_{t_j \in d_i} \alpha_j \cdot \vec{w_{t_j}} \tag{4.7}$$

donde α_j es la contribución del concepto t_j a la representación del documento d_i^{dtr}. En este trabajo usamos tres diferentes pesados:

- **Booleano:** bajo esta representación si un término aparece en el documento se considera su representación para la suma únicamente una vez, es decir, $\alpha_j = 1$.

$$d_i^{dtr} = \sum_{t_j \in d_i} \vec{w_{t_j}} \tag{4.8}$$

- **TF:** consiste en dar mayor relevancia a los conceptos que ocurran con mayor frecuencia en el documento, es decir, se considera las veces que el término aparece en el documento.

$$d_i^{dtr} = \sum_{t_j \in d_i} tf_{DTRs}(i, t_j) \times \vec{w_{t_j}} \tag{4.9}$$

  donde $tf_{DTRs}(i, t_j)$ es el número de veces en que t_j aparece en el documento j.

- **TF-IDF:** Consiste en la combinación de la frecuencia del concepto en el documento, con la frecuencia del mismo concepto en el resto de los documentos de la colección, y se calcula de la siguiente forma:

$$d_i^{dtr} = \sum_{t_j \in d_i} tf_D(i, t_j) \times \vec{w_{t_j}} \times log(\frac{N_D}{N_k}) \tag{4.10}$$

  donde N_k es el número de documentos que contienen el término t_j y N_D es el número de documentos en la colección.

4.2.3. Pesado supervisado

A continuación se presenta una variante del método propuesto basado en la entropía al que llamaremos TCOR-E. En los pesados anteriores, la importancia

asignada a cada término se basaba únicamente en saber si ese término existe en el documento (Booleano), las veces que aparece en cada documento (TF) o las veces que aparece dentro de un documento con respecto a otros documentos (TF-IDF). Sin embargo, dentro de la clasificación de textos, la mayoría de las veces la importancia de un término esta relacionada a las categorías. Es decir, sin un término aparece en los documentos de una clase dada y solo en esa clase, ese término es relevante para dicha clase; por el contrario, si el término aparece en todas las clases no es relevante para ninguna clase y se podría considerar ruido. Por lo cual, este esquema de pesado se basa en la idea de que los términos pueden ser ponderados de manera supervisada, es decir, los términos que estén muy relacionados a una sola clase deben ser muy relevantes para la clasificación, a diferencia de aquellos términos que se presenten en varias clases.

Por tanto, se propone una variante del método propuesto, la cual pondera la relevancia de los términos por clases. Cabe señalar que se le dice *pesado supervisado* debido a que toma en consideración las categorías de los documentos de entrenamiento. La modificación al método se presenta en la fórmula (4.7), ya que el α_j es un escalar que afecta a la representación del término de forma global. Sin embargo, con TCOR-E se requiere de un vector ($\vec{\beta_j}$), debido a que se desea ponderar cada una de las relaciones entre términos existentes en la representación de cada concepto, resultando la ecuación de la siguiente manera:

$$d_i^{dtr} = \sum_{t_j \in d_i} \vec{\beta_j} \cdot \vec{w_{t_j}} \tag{4.11}$$

Para TCOR-E se requiere del cálculo de la entropía de las co-ocurrencia de los términos, con la finalidad de que aquellas relaciones entre términos que ocurran mucha dentro de una clase (entropía = 0) sean más relevantes; por otro lado,

aquellas que ocurran en todas las clases (entropía = 1) Se realizaron los cálculos de entropía (ver sección 2.2.2) de las co-ocurrencias de los términos t_i y t_j para cada una de las categorías de la siguiente forma:

$$Entropia(t_i, t_j) = \sum_{c \in C} -p_{ij}^{c} log_2 p_{ij}^{c} \tag{4.12}$$

donde C son las distintas categorías que existen en la colección. El log_2 se cambia por un $log_{|C|}$ debido a que necesitamos que el valor de la entropía sea entre [0,1], siendo 1 cuando las co-ocurrencias están presentes en todas las categorías y 0 para el caso contrario. p_{ij}^{c} está definido de la siguiente forma:

$$p_{ij}^{c} = \frac{\#(t_{ij}, c)}{\sum_{x \in C} \#(t_{ij}, x)} \tag{4.13}$$

donde $\#(t_{ij},)$ es el número de veces que el término i y j co-ocurren en la clase c.

Una vez finalizado se obtiene una matriz de pesado supervisado basado en la entropía (M_E) de $t \times t$, donde t son los términos de la colección, resultando la ecuación:

$$d_i^{dtr} = \sum_{t_j \in d_i} \vec{M}_E(t_j) \cdot \vec{w_{t_j}} \tag{4.14}$$

4.3. Resumen

Los métodos propuestos están basados en el enfoque de enriquecimiento de los documentos cortos con la ayuda de las representaciones distribucionales de los términos, además, se realiza sin ayuda de ningún recurso externo. Los métodos constan de tres procesos fundamentales:

- Dos métodos para calcular las (co-)ocurrencias de los términos, los cuales nos permiten estimar relaciones entre términos y así generar conceptos de los términos.

- Un método para representar a los documentos en función de los conceptos descubiertos por cada término.
- Un esquema de pesado, el cual se basa en la idea de que las co-ocurrencias que se presenten en una sola clase son más relevantes, que aquellos que se presentan en más de una clase.

Con esto se cumplen los objetivos específicos 2 y 3 (1.3), los restantes objetivos serán presentados en el capítulo 6.

Capítulo 5

Colecciones utilizadas para experimentación

En el presente capítulo se hablará acerca de los conjuntos de datos que se utilizarán para realizar los experimentos. Entre los puntos más importantes, hablaremos acerca del porqué se escogieron los conjuntos de datos y de las características de cada uno de estos conjuntos, así como del análisis que se realizó a cada corpus de documentos cortos. El capítulo esta dividido de la siguiente forma. En la primera sección se presenta un análisis de las características de cada uno de los conjuntos de datos utilizados. Después, se presenta una modificación de los conjuntos de datos, con la finalidad de analizar el comportamiento de los métodos establecidos en el estado del arte y nuestro método dentro de la clasificación de textos cortos. Finalmente, en la sección 5.5, se realiza un análisis de estos conjuntos de datos con la ayuda de medidas de evaluación de corpus, propuestos por (Pinto et al., 2010).

5.1. R8

En esta tesis se usa el subconjunto R8 de Reuters-21578. El cual consta de 8 de las clases más grandes de Reuters-21578, donde los documentos solo pertenecen a una clase. El subconjunto R8 ha sido previamente utilizado por (Cardoso-Cachopo and Oliveira, 2007; Ingaramo et al., 2008; Ramírez-de-la Rosa et al.). En la tabla 5.1 se presentan los detalles generales del conjunto de datos R8 y algunas de las características más relevantes.

Tabla 5.1: Características principales del Corpus R8

Clases	Docs. de		Características	Docs. de	
	pruebas	entrenamiento		pruebas	entrenamiento
acq	1594	695	Tamaño del vocabulario	14865	8760
crude	251	119	Número de documentos	4559	2179
earn	2824	1076	Vocabulario por Doc.	40.9	39.2
grain	41	10	Núm. Categorías	8	8
interest	189	81			
money-fx	207	87			
ship	107	36			
trade	249	75			
Total	5459	2179			

5.2. EasyAbstracts

EasyAbstracts es una colección de resúmenes de artículos científicos relacionados al área de sistemas inteligentes, y es considerada más compleja debido a que solo cuenta con 48 documentos. El conjunto de EasyAbstracts fue utilizado en tareas de clasificación de textos por (Ingaramo et al., 2010; Rosas et al., 2010; Rosso and Luis, 2011). Esta colección esta compuesta de documentos pertenecientes a resúmenes de artículos científicos de 4 revistas internacionales en los siguientes

ámbitos: Machine Learning (ML), Heuristic in Optimization (HO), Autonomus Intelligent Agent (AIA) y Automated Reasoning (AR). La distribución de los documentos en cada categoría y las características principales de la colección se presentan en la tabla 5.2.

Tabla 5.2: Características principales del conjunto de datos EasyAbstracts

Categorías	**Documentos**
AIA	11
ho	11
ml	15
ar	11
Total	43

Características	**Valores**
Vocabulario del corpus	1136
Núm. de documentos	48
Vocabulario por documento	60.3
Núm. categorías	4

5.3. CICLIng2002

Este conjunto de datos está compuesto de 48 resúmenes científicos del área de lingüística computacional, los cuales corresponden a artículos presentados en la conferencia CICLIng2002 en el 2002. A pesar del tamaño tan pequeño de este conjunto de datos, ha sido usado en diferentes experimentos por (Ingaramo et al., 2008, 2010; Makagonov et al., 2004; Rosas et al., 2010). La distribución y las características de este conjunto de datos son presentadas en la Tabla 5.3.

Tabla 5.3: Características principales del conjunto de datos CICLIng2002

Categorías	Documentos
Linguistics	11
Lexicon	11
Ambiguity	15
Text Processing	11
Total	43

Características	Valores
Vocabulario del corpus	813
Núm. de documentos	48
Vocabulario por documento	45.06
Núm. categorías	4

5.4. Subconjuntos derivados de los conjuntos de datos

El propósito de obtener subconjuntos de datos, es poder evaluar la clasificación de textos con documentos muy cortos (nueve términos en promedio), lo cual añade dificultad a esta tarea. Además, nos permitirá evaluar la tarea de clasificación de textos cortos en escenarios donde se tienen documentos extensos para realizar el entrenamiento (por ejemplo, artículos completos) y evaluar al clasificador usando documentos cortos (títulos o resúmenes).

Generamos tres nuevos subconjuntos de datos: R8-Títulos, EasyAbstracts-Títulos y CICLIng2002-Títulos, para analizar el desempeño de la clasificación de textos cortos. Estos subconjuntos consisten únicamente de los títulos de las noticias o resúmenes científicos y se utilizarán como conjunto de prueba mientras que los documentos completos se utilizarán para realizar el entrenamiento. La tabla 5.4 muestra la información de los datos para estas tres nuevas colecciones de datos, donde se puede observar la reducción considerable de vocabulario y longitud de cada documento. En esta tabla podemos observar tres aspectos importantes, el promedio de la longitud de los documentos (número de términos por documento),

el promedio del vocabulario de los documentos (número de términos diferentes por documento) y el vocabulario del conjunto de datos completo. En el caso de R8, se tiene un promedio de 39.2 términos diferentes por cada documento y para R8-Títulos tiene 6.6, por lo que hubo una reducción de 83.8 %. EasyAbstracts original tiene en promedio 60.3 y EasyAbstracts-Títulos tiene 5.85 dando un porcentaje de 90.2 % de reducción. Finalmente CICLIng2002 tiene 45.06 y CICLIng2002-Títulos tiene 4.8, mostrando un porcentaje de reducción de 89.3 %. Por lo tanto la dificultad de los conjuntos de datos se incrementa considerablemente al reducir el vocabulario de los documentos y reducir aún más la longitud de los documentos.

Tabla 5.4: Principales características de los subconjuntos de datos basados en títulos

Conjunto de datos	Clases	Docs	Pals. x docs.	Vocabulario por doc.	Vocabulario corpus
R8-T	8	2179	6.7	6.6	3676
EasyAbstracts-T	4	48	5.95	5.85	206
CICLIng2002-T	4	48	4.8	4.8	180

5.5. Análisis de los conjuntos de datos seleccionados

Se requiere medir la complejidad de los conjunto de datos para tener una idea bajo qué circunstancias un método puede tener un mejor desempeño. En este trabajo se requiere medir la complejidad de un conjunto de datos en términos de: qué tanta semejanza existe entre las clases y que tan cortos son los documentos de la colección. Para calcular esto ya se han propuesto diferentes medidas,

entre ellas las propuestas por Pinto et al. (2010) y que serán utilizadas en este trabajo enfocándonos en dos medidas: ***amplitud de dominio*** y ***longitud de documentos***.

5.5.1. Amplitud de dominio

La *amplitud de dominio* es relevante, ya que nos permitirá saber que tanta relación temática existe entre las distintas categorías. Por lo que podría ser más fácil clasificar los documentos si los vocabularios de cada categoría son distintos (soccer, cocina), caso contrario a que si son muy parecidos (soccer, americano), ya que sería mas complicado diferenciar la categoría correcta ver figura 5.1 .

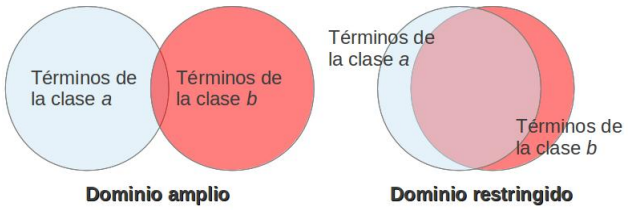


Figura 5.1: Ejemplo de traslape de términos en las diferentes clases de un conjunto de datos. Un dominio amplio indica que existe poco traslape de términos entre las clases. Por otro lado, un dominio restringido significa que existe mucho traslape entre los términos.

Esta medida puede ser calculada de dos formas de acuerdo a (Pinto et al., 2010):

- **Basado en el modelado estadístico del lenguaje (SLM):** SLM es un método para estimar la distribución de las palabras/cadenas del lenguaje. El cálculo de probabilidad de una cadena S de longitud n, comúnmente llamada n-gramas, intenta reflejar la frecuencia relativa de S en una sentencia. En

otras palabras, intenta capturar las características de escritura de un idioma con el fin de predecir la siguiente palabra dada una secuencia.

La idea de utilizar SLM es de generar el modelo para una clase C_c y observar que tan bien se puede generar el lenguaje de otra clase con el modelo creado, mientras mejor se pueda generar el lenguaje de otra clase dado el modelo más restringido será el dominio. La fórmula de SLM es la siguiente:

$$SLMB(D) = \sqrt{\frac{1}{k}\sum_{i=1}^{k}\Big(Perplexity(C_i^*|\overline{C}_i^*) - \mu(Perplexity(C^*))\Big)^2} \quad (5.1)$$

donde

$$\mu(Perplexity(C^*)) = \frac{\sum_{i=1}^{k} Perplexity(C_i^*|\overline{C}_i^*)}{k} \quad (5.2)$$

donde D es un conjunto de datos etiquetado con k diferentes clases $C^* = C_1^*, C_2^*, \cdots, C_k^*$, y $\overline{C}_1^*$ es el modelo del lenguaje de todas las clases excepto C_1^*. Lo que intentan realizar es ver que tan parecido es el modelo del lenguaje $\overline{C}_1^*$ con el C_1^*.

- **Basado en la dimensión de vocabulario (SVB):**

  Esta medida parte del supuesto que los subconjuntos pertenecen a un dominio restringido, si comparten un número alto de términos del vocabulario, comparado con los subconjuntos que no pertenecen, ver figura 5.1. En caso de que pertenezcan a un dominio "amplio", se espera que la desviación estándar del tamaño del vocabulario de los subconjuntos, sea mayor que el obtenido por los subconjuntos de dominio restringido. La descripción formal se presenta enseguida.

Dado un corpus D previamente etiquetado con k categorías $C^* = C_1^*, C_2^*, \cdots, C_k^*$, y $\overline{C}_1^*$, si $|V(D)|$ es la cardinalidad del vocabulario de todo el corpus y $|V(C_i^*)|$ el tamaño de vocabulario de la categoría C_i^*.

$$SVB = \sqrt{\frac{1}{k}\sum_{i=1}^{k}\left(\frac{|V(C_i^*)| - |V(D)|}{|D|}\right)^2} \tag{5.3}$$

5.5.2. Longitud del documento

Longitud del documento es otra medida, que captura algunas características como: la longitud de un texto y la proporción entre el tamaño del vocabulario de un documento y la longitud de un documento. Esta medida puede ser calculada de diferentes formas, (Pinto et al., 2010) propone tres formas.

Dado un corpus D que consta de n documentos $D = d_1, \cdots, d_n$ las medidas pueden ser calculadas de la siguiente forma:

- **Longitud de los documentos (DL):**

$$DL(D) = \frac{1}{n}\sum_{i=1}^{n}|d_i| \tag{5.4}$$

donde $|d_i|$ es la longitud del documento i, y n es el número total de documentos de la colección. DL mide el promedio de longitudes de los documentos.

- **Longitud del vocabulario (VL):**

$$VL(D) = \frac{1}{n}\sum_{i=1}^{n}|V(d_i)| \tag{5.5}$$

donde $|V(d_i)|$ es el número de términos diferentes que contiene el documento i, y n es el número total de documentos de la colección. VL mide el promedio de términos diferentes que contiene una colección de documentos.

- **Proporción entre el promedio del vocabulario y la cardinalidad del documento (VDR):**

$$VRD(D) = \frac{log(VL(D))}{log(DL(D))} \tag{5.6}$$

VRD mide la relación existente entre las longitudes de los documentos y las longitudes de los vocabularios de los documentos.

5.5.3. Resultados y Discusión

Primero hablaremos acerca de los resultados obtenidos relacionados con la *amplitud del vocabulario*. En la Tabla 5.5 se puede observar los valores obtenidos mediante las fórmulas (5.1) y (5.3), estos resultados están ordenados según el valor obtenido mediante SLMB y además por tipo de conjunto, es decir, aquellos que contienen el documento completo (a los que les asignaremos un prefijo "-D") y los que contienen solo los títulos (prefijo "-T"). Mientras más pequeños sean las cantidades el dominio será más restringido (*narrow*), y mientras más grandes, los dominios serán más amplios (*broad*). Los resultados finales muestran que tanto SLMB como SVB son muy parecidos en cuanto al ordenamiento final de los resultados obtenidos, por lo que podemos asegurar con más confianza que CICLIng2002 es un corpus de dominio más restringido en comparación con los otros dos, por lo menos en la configuración de documentos completos (-D) y para la configuración de solo títulos (-T) es EasyAbstracts.

Por otro lado, en la Tabla 5.6 observamos los resultados relacionados a la longitud del documento. Como se esperaba, los conjuntos de datos que contienen únicamente títulos son los que obtuvieron los resultados más bajos. El corpus con valor más pequeño considerando solo los títulos es el EasyAbstracts-T, y cuando consideramos los conjuntos de datos originales, R8-Train-D es el más corto. R8-

Corporas	Broadness	
	SLMB	SVB
Documentos completos		
CICLIng2002-D	18.9239	1.7319
EasyAbstracts-D	28.0862	2.67034
R8-Test-D	241.711	3.02914
R8-Train-D	603.95	3.67
Documentos solo títulos		
EasyAbstracts-T	8.5324	4.4444
CICLIng2002-T	16.9613	3.62552
R8-Test-T	211.542	9.44226

Tabla 5.5: Resultados relacionados a la amplitud de dominio. Los resultados están ordenados de forma descendente de dominio restringido a dominio amplio. Los conjuntos EasyAbstracts y CICLIng2002 son los que obtuvieron resultados similares (dominio restringido), mientras que R8 en cualquiera de sus variantes es de dominio amplio (comparado con CICLIng2002 y EasyAbstracts).

	Long. documento		
	DL	VL	VDR
Documentos completos			
R8-Train-D	66.32	15.438	0.8865
CICLIng2002-D	70.4583	48.3958	0.9117
EasyAbstracts-D	94.6458	66.5833	0.9227
R8-Test-D	122.604	72.6003	0.8910
Documentos solo títulos			
EasyAbstracts-T	6.0208	5.9583	0.9941
CICLIng2002-T	7	6.9375	0.9953
R8-Test-T	8.6030	8.4694	0.9927

Tabla 5.6: Resultados de los conjuntos de datos relacionados a la longitud del documento, mientras más bajo se el valor, los documentos serán más pequeños. Los resultados están ordenados de forma descendente de acuerdo a DL.

Train-D tiene en promedio 66.32 términos por documento, pero solo consta de 15.34 términos distintos por documento. El corpus R8-Test-D es el que contiene los documentos más largos de todas las colecciones consideradas, tiene 122.60 términos en promedio por documento de los cuales 72.60 son diferentes.

Al observar estas dos medidas de evaluación de corpus podemos imaginarnos las diferentes situaciones a las que nos enfrentaremos en la clasificación de textos cortos. Por un lado, el conjunto CICLIng2002-D es de dominio restringido y también contiene textos muy cortos, por lo que esta colección es considerada la más complicada de estas tres. Después, EasyAbstracts al compararse con CICLIng2002-D es de dominio menos restringido y además, la longitud de sus

documentos es ligeramente mayor. Finalmente, el menos complejo de los tres corpus es R8-D, debido a que tiene un dominio más amplio y los documentos son relativamente extensos. En cuanto a los conjuntos que constan solo de los títulos tienen un comportamiento muy similar.

Cabe mencionar que otro aspecto importante relacionado con la dificultad de los conjuntos de datos CICLIng2002 y EasyAbstracts es el número de documentos con los que se cuentan (48 en total).

En resumen, la principal motivación por la cual se eligieron estos tres conjuntos de datos fue:

1. La combinación de resultados presentados de los tres conjuntos de datos nos permite abordar de mejor manera y saber bajo que situaciones los métodos se pueden desempeñar. Por ejemplo, R8-Train-D y R8-Test-T son de dominio amplio, documentos cortos y cuentan con una gran cantidad de documentos, todo esto en comparación de los otros dos conjuntos de datos de la misma configuración. Por su parte EasyAbstracts-D es de dominio medianamente restringido, documentos extensos pero pocos documentos. EasyAbstracts-T es de dominio muy restringido, documentos muy cortos y pocos datos de entrenamiento. CICLIng2002-D dominio muy restringido, documentos cortos y pocos documentos al igual que CICLIng2002-T.

Capítulo 6

Resultados experimentales

En este capítulo se presentan los resultados experimentales obtenidos con los métodos propuestos, de igual forma se realiza la comparación con el trabajo más parecido del trabajo relacionado a nuestra propuesta y además con el método más usado y popular en la clasificación de textos (BOW). El presente capítulo está organizado de la siguiente forma: en la primera sección se plantean las condiciones utilizadas para los experimentos. Posteriormente, se presentan los resultados de los experimentos con la finalidad de comprobar el desempeño del método propuesto en esta tesis. Para finalizar, se presentan una serie de experimentos adicionales relacionados con documentos cortos y además con conjunto de entrenamiento reducido.

6.1. Condiciones experimentales

El pre-procesamiento de los datos fue el mismo para todos los experimentos. Se aplicó truncamiento de las palabras utilizando el algoritmo *Porter* (Porter, 2006), explicado en la sección 2.2. Además, se removieron las palabras vacías y se aplicó

un método para reducir la dimensión de las representaciones de los documentos, eliminando aquellos términos que tienen una frecuencia menor a dos (ver Sección 2.2.2).

Para generar resultados de referencia que sirvan como parte de la comparación del método propuesto (*baseline*), se utilizó BOW, (sección 2.2.1). Además de realizar ésta comparación, se realizó una comparación con el método LSI que es el más parecido al método propuesto.

El objetivo de este trabajo de tesis, es evaluar la representación generada de los documentos en términos de qué tan útil es para la clasificación de textos cortos y no propiamente los algoritmos de aprendizaje. Por esta razón, se utilizaron cinco distintos algoritmos de aprendizaje cada uno de distinta naturaleza que, además, son muy utilizados en la clasificación de textos por sus resultados obtenidos. Estos algoritmos de aprendizaje son: Máquinas de Vectores de Soporte (SVM), Bayesiano simple (NB), AdaBoost (AB), k-Vecinos más cercanos (kNN) y Random Forest (RF), (sección 2.3). Las implementaciones usadas para estos algoritmos son las que provee WEKA (Garner, 1995), con los parámetros por omisión, excepto SVM el cual utilizó el kernel lineal, debido a que es con el que mejores resultados se ha obtenido (Ure, 2005).

Para realizar la evaluación del desempeño del método propuesto se utilizó la medida macro-F1, ver sección (2.4), debido a que expresa de forma más adecuada los resultados de desempeño en situaciones con conjuntos de datos con clases desequilibradas. Además, para la evaluación se utilizaron los conjuntos de datos explicados anteriormente, ver sección (5.4), estos conjuntos fueron utilizados bajo dos condiciones experimentales:

- **Documento-Documento:** En esta configuración se entrenó y evaluó el clasificador usando el documento completo (título + cuerpo).

- **Documento-Título o clasificación de títulos:** En esta configuración se entrenó el clasificador usando el documento completo y se evaluó el clasificador utilizando únicamente los títulos (textos cortos). La idea detrás de esta configuración es simular que se tiene un corpus de entrenamiento con documentos más extensos que los documentos de prueba y que los documentos de prueba son muy cortos (en promedio 8 términos).

En las primeras secciones se presentarán y analizarán los resultados obtenidos mediante la representación BOW, comparando la configuración Documento-Documento y la Documento-Título, con la finalidad de observar los problemas relacionados con los textos cortos. Posteriormente los resultados serán comparados con los obtenidos mediante el método propuesto y, finalmente, con el método más parecido al nuestro, *LSI* sección 3.

6.2. Clasificación de textos cortos con bolsa de palabras (BOW)

El primer experimento consiste en evaluar BOW en la tarea de clasificación de textos cortos, analizando la clasificación de títulos con la finalidad de "*determinar hasta que punto los métodos de representación de documentos clásicos son adecuados para la clasificación de textos cortos*", objetivo específico 1, 1.3.

En la tabla 6.1 se muestran los resultados obtenidos por las distintas configuraciones (columnas) DD y DT. En la tercera columna se muestra el decremento que sufre en términos del Macro-F1 al clasificar los títulos. En las filas se muestran los diversos clasificadores que se usaron con los distintos conjuntos de datos y también el promedio de los clasificadores, los máximos y mínimos resultados logrados. Para el conjunto R8 los resultados demuestran la complejidad de la clasificación de títulos. De manera similar ocurre con el corpus EasyAbstracts, donde se puede observar el considerable decremento (hasta un 64 % como máximo y 15 % mínimo) y para CICLIng2002 desde un decremento máximo de 65 % y un mínimo de 1 %. Los mejores resultados se obtuvieron mediante la configuración DD. Por otra parte, se puede ver que la configuración DT se ve considerablemente afectado en un promedio de 36.59 % con el pesado binario, un 33.94 % con el pesado TF y 44.05 % con TF-IDF.

Tabla 6.1: Resultados obtenidos con la representación BOW en la tarea de clasificación de textos cortos.

R8

	Binario			TF			TFIDF		
	DD	DT	Decremento	DD	DT	Decremento	DD	DT	Decremento
AdaBoost	0.64	0.18	**-72.74 %**	0.64	0.18	**-72.74 %**	0.64	0.18	**-72.74 %**
kNN	0.69	0.39	**-43.98 %**	0.47	0.34	**-27.53 %**	0.47	0.34	**-27.53 %**
Naive Bayes	0.87	0.66	**-24.16 %**	0.82	0.34	**-58.97 %**	0.82	0.34	**-59.13 %**
RandomForest	0.80	0.54	**-32.21 %**	0.80	0.57	**-29.02 %**	0.82	0.74	**-10.46 %**
SVMLineal	0.91	0.83	**-7.85 %**	0.90	0.73	**-19.29 %**	0.90	0.70	**-22.59 %**
Promedio	0.78	0.52	**-28.37 %**	0.72	0.43	**-40.29 %**	0.73	0.46	**-36.98 %**
Máx. global	0.91	0.83	**-7.77 %**	0.90	0.73	**-18.88 %**	0.90	0.74	**-17.77 %**
Min. global	0.64	0.18	**-61.70 %**	0.47	0.18	**-61.70 %**	0.47	0.18	**-61.70 %**

EasyAbstracts

AdaBoost	0.41	0.27	**-34.34 %**	0.40	0.25	**-37.70 %**	0.40	0.25	**-37.70 %**
kNN	0.21	0.11	**-46.14 %**	0.14	0.09	**-38.74 %**	0.14	0.09	**-38.74 %**
Naive Bayes	0.70	0.40	**-42.89 %**	0.74	0.35	**-53.09 %**	0.79	0.37	**-52.93 %**
RandomForest	0.57	0.24	**-57.82 %**	0.49	0.22	**-54.34 %**	0.53	0.19	**-64.01 %**
SVMLineal	0.69	0.59	**-15.64 %**	0.90	0.16	**-82.05 %**	0.85	0.30	**-64.67 %**
Promedio	0.51	0.32	**-59.927 %**	0.53	0.21	**-40.29 %**	0.54	0.24	**-55.71 %**
Máx. global	0.70	0.59	**-61.11 %**	0.90	0.35	**-18.88 %**	0.85	0.37	**-56.47 %**
Min. global	0.21	0.11	**-35.70 %**	0.14	0.09	**-61.70 %**	0.14	0.09	**-35.71 %**

CICLIng2002

AdaBoosts	0.36	0.27	**-22.76 %**	0.36	0.27	**-22.76 %**	0.31	0.20	**-35.32 %**
kNN	0.29	0.10	**-65.62 %**	0.14	0.16	10.62 %	0.13	0.09	**-31.31 %**
Naive Bayes	0.43	0.33	**-23.50 %**	0.43	0.39	**-10.50 %**	0.37	0.14	**-61.30 %**
RandomForest	0.40	0.25	**-38.01 %**	0.31	0.30	**-1.10 %**	0.22	0.12	**-46.91 %**
SVMLineal	0.45	0.35	**-21.14 %**	0.54	0.48	**-11.91 %**	0.21	0.14	**-35.52 %**
Promedio	0.38	0.26	**-32.64 %**	0.35	0.32	**-10.11 %**	0.24	0.13	**-44.35 %**
Máx. global	0.45	0.35	**-22.22 %**	0.54	0.48	**-11.11 %**	0.37	0.20	**-45.947 %**
Min. global	0.29	0.10	**-65.51 %**	0.14	0.16	**-14.28 %**	0.13	0.09	**-30.76 %**

Por otro lado, se puede observar que el comportamiento tiene relación con lo que se esperaba de acuerdo al análisis realizado a los conjuntos de datos en el capítulo anterior 5. El análisis realizado suponía que el conjunto más complejo de los tres utilizados sería el CICLIng2002 y el más fácil sería R8. Con el conjunto CICLIng2002 se obtuvo el menor desempeño (0.40 valor máximo de F1), esto se puede deber a tres factores relevantes, *dominio restringido*, *longitud del documento*, y además al conjunto reducido de entrenamiento con el que se cuenta. Mientras que con el conjunto R8 se alcanzaron resultados satisfactorios de hasta 0.91 de medida F1 global, en EasyAbstracts se obtuvo un 0.85 de F1. Otro aspecto importante que se puede observar es que los mejores resultados obtenidos fueron utilizando los algoritmos SVM y el Bayesiano simple, y los peores resultados fueron obtenidos por kNN y AdaBoost; un posible motivo por el cual se hayan obtenido estos resultados, en el caso de kNN todos los vecinos son muy cercanos (similares) y AdaBoost se sobre ajusta por el mismo motivo que kNN, no puede discriminar los documentos por que la similitud entre documentos es muy grande. Por otra parte, los clasificadores SVM y Bayesiano simple no fueron consistentes a lo largo de los experimentos, por lo que se puede reafirmar el hecho de utilizar algoritmos de clasificación de distinta naturaleza para la evaluación de las representaciones propuestas para la clasificación de textos cortos.

6.3. DTRs para la clasificación de textos cortos

El objetivo de este segundo experimento es evaluar las representaciones distribucionales de los términos en la clasificación de textos cortos, y además realizar la comparación de las DTRs contra el método *base*.

En la tabla 6.2, se muestran los resultados obtenidos por las tres colecciones,

las cuatro representaciones (BOW, DOR, TCOR, y TCOR-E, este último es la variante propuesta de TCOR) y además con los tres distintos pesados. Los resultados sombreados indican que se obtuvo un mejor desempeño que el método base (BOW), y los resultados resaltados en negritas fueron los significativamente mejores que BOW. Se realizó la validación cruzada con 10 pliegues, y para calcular la significación estadística se utilizó la prueba T-Test con una confianza de 95 %. Para obtener la representación de DTRs se realizaron entrenando con los documentos completos y probando con los títulos (clasificación de títulos), esto es un escenario de clasificación de textos cortos con más información para el desarrollo del método. Como se puede observar en la tabla 6.2 la mayoría de los resultados están sombreados y en negritas, lo que significa que en la mayoría de los casos se obtuvieron mejores resultados cuando se utilizan las DTRs.

En la gráfica 6.1 podemos darnos una idea más clara y resumida de lo que está sucediendo. Esta gráfica representa el decremento de desempeño que se obtiene entre usar la configuración DD y DT; mientras más al centro de la red se encuentre un punto, mayor fue su decremento en la clasificación de títulos. Por lo tanto podemos observar que las DTRs tienen mejor desempeño que BOW, al mantenerse más estables sin importar el clasificador que se utilice.

Como se mencionó en la sección 5.5 el conjunto EasyAbstracts tiene una complejidad alta con respecto al R8 y a su vez CICLIng2002 tiene una mayor complejidad que la de EasyAbstracts. Los resultados obtenidos coinciden con estas medidas, como se puede observar en la tabla 6.2, para el conjunto EasyAbstracts se obtuvieron comportamientos muy similares a los del conjunto R8, aunque con resultados un poco más bajos; la mayoría de los resultados fueron significativamente mejores que BOW. Sin embargo, para el caso de la colección CICLIng2002, los resultados no fueron los mejores para el pesado Booleano y TF, pero para el

Tabla 6.2: Resultados de la clasificación de textos cortos usando DTRs y la configuración DT. Los resultados sombreados son los que obtuvieron mejores resultados que la representación BOW, y los resultados resaltados en negrita fueron los significativamente mejores que BOW.

R8

Pesados	Binario				TF				TFIDF			
Clasificadores	BOW	DOR	TCOR	TCOR-E	BOW	DOR	TCOR	TCOR-E	BOW	DOR	TCOR	TCOR-E
AB	0.175	0.645	0.668	0.638	0.175	0.632	**0.651**	**0.659**	0.175	**0.591**	**0.667**	**0.676**
KNN	0.386	**0.899**	**0.897**	**0.921**	0.337	**0.908**	**0.902**	**0.884**	0.337	**0.746**	**0.754**	**0.856**
NB	0.656	**0.881**	**0.893**	**0.880**	0.336	**0.874**	**0.886**	**0.878**	0.336	**0.785**	**0.854**	**0.855**
RF	0.543	**0.786**	**0.774**	**0.835**	0.565	**0.805**	**0.823**	**0.827**	0.736	**0.798**	**0.819**	**0.817**
SVM	0.834	**0.930**	**0.891**	**0.925**	0.728	**0.928**	**0.901**	**0.929**	0.699	**0.897**	**0.784**	**0.877**
Mínimo	0.175	0.644	0.668	0.638	0.175	0.632	0.651	0.67	0.175	0.591	0.667	0.676
Máximo	0.834	0.93	0.897	0.925	0.728	0.928	0.902	0.93	0.736	0.897	0.854	0.877
Promedio	0.518	0.828	0.824	0.839	0.428	0.829	0.832	0.850	0.456	0.763	0.775	0.816

EasyAbstract

AB	0.268	0.185	0.201	0.192	0.255	0.272	0.245	0.306	0.250	0.263	0.292	0.255
KNN	0.114	**0.600**	**0.482**	**0.627**	0.086	**0.666**	**0.712**	**0.700**	0.086	**0.571**	**0.541**	**0.578**
NB	0.402	**0.568**	**0.586**	**0.596**	0.345	**0.603**	**0.590**	**0.683**	0.370	**0.578**	**0.603**	**0.595**
RF	0.239	**0.495**	**0.332**	**0.470**	0.223	**0.507**	**0.582**	**0.566**	0.192	**0.588**	**0.550**	**0.503**
SVM	0.585	0.660	**0.639**	**0.722**	0.161	**0.728**	**0.733**	**0.713**	0.301	**0.622**	**0.589**	**0.563**
Mínimo	0.114	0.185	0.201	0.299	0.086	0.272	0.245	0.294	0.086	0.266	0.292	0.291
Máximo	0.585	0.660	0.639	0.656	0.345	0.728	0.733	0.700	0.370	0.622	0.603	0.594
Promedio	0.321	0.502	0.448	0.544	0.213	0.555	0.572	0.562	0.240	0.524	0.515	0.488

CICLIng2002

AB	0.274	0.188	0.244	0.278	0.274	0.129	0.224	0.274	0.199	0.201	0.232	0.271
KNN	0.099	**0.450**	**0.395**	**0.368**	0.156	**0.478**	**0.399**	**0.424**	0.089	**0.493**	**0.44**	**0.501**
NB	0.332	**0.473**	0.415	0.401	0.386	**0.426**	0.471	0.366	0.143	**0.506**	**0.399**	**0.392**
RF	0.249	0.184	0.369	**0.422**	0.304	0.279	0.374	0.367	0.119	**0.418**	**0.291**	**0.276**
SVM	0.354	**0.526**	0.414	**0.412**	0.48	0.504	0.502	0.440	0.135	**0.528**	**0.442**	**0.448**
Mínimo	0.099	0.184	0.244	0.237	0.156	0.129	0.224	0.258	0.089	0.201	0.232	0.239
Máximo	0.354	0.526	0.415	0.467	0.480	0.509	0.502	0.472	0.199	0.528	0.442	0.527
Promedio	0.261	0.364	0.367	0.365	0.320	0.363	0.394	0.393	0.137	0.429	0.361	0.411

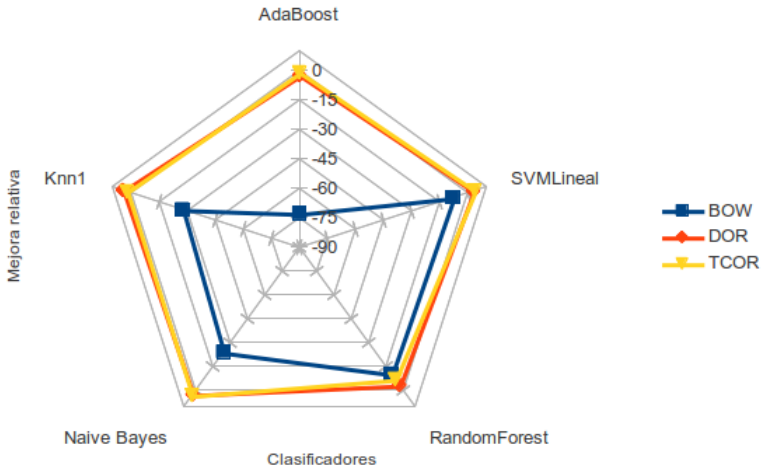


Figura 6.1: Mejora Relativa del corpus R8. Se muestran los mejores resultados obtenidos por BOW y por las DTRs (DOR); mientras mas al centro de la red se encuentre un punto mayor fue su decremento en la clasificación de títulos.

pesado TF-IDF, el método propuesto si fue significativamente mejor que BOW.

Una explicación a los resultados obtenidos con el método propuesto en la tabla 6.2 con el corpus CICLIng2002 y los pesados Booleano y TF, se puede deber a que tiene un dominio más estrecho, por lo tanto un poco más complejo como se resaltan en la tabla 5.5. Además que es un conjunto con muy pocos documentos, por lo que se complica aun más la tarea.

Hasta este punto, tanto DOR, TCOR, TCOR-E son muy parecidos, en varios casos DOR tiene mejor desempeño que TCOR y esto se puede deber a que los conjuntos de datos no son tan extensos como para que TCOR pueda capturar de una mejor forma las co-ocurrencias y por lo tanto ayuden a la expansión de documentos en esta tarea. Por lo que, DOR es más robusto cuando se tienen conjuntos más pequeños de entrenamiento.

Al comparar los resultados de TCOR y TCOR-E, se puede observar que tienen un comportamiento muy similar, de hecho, ninguno es significativamente mejor que el otro, por lo que se podría concluir que el pesado supervisado propuesto (TCOR-E) no mejoró lo obtenido por el esquema tradicional de TCOR. Por el contrario, si bien no mejoró el resultado general, en la mayoría de las ocasiones sí mejoró el F1 por clase, ver tabla 6.3, debido a que TCOR-E brinda mayor relevancia a las clases, es decir, intenta descubrir los términos más importantes por clase, lo que mejora los resultados a nivel de las clases.

Tabla 6.3: Resultados con la medida F1 por clases obtenidas por las representaciones TCOR y TCOR-E, utilizando el pesado TF en el corpus R8. Las celdas resaltadas en negritas son resultados de TCOR que mejoraron al TCOR-E y las celdas obscuras son donde TCOR-E mejoró a TCOR.

Representación	Clasificadores	trade	ship	money-fx	interest	grain	earn	crude	acq
TCOR	AdaBoost	0	0	0	0	0	0.891	0	0.662
TCOR	KNN	0.81	0.783	0.759	0.755	0.8	0.941	0.87	0.899
TCOR	Naive Bayes	**0.677**	**0.685**	**0.455**	**0.721**	**0.323**	**0.948**	0.868	0.909
TCOR	RandomForest	0.601	0.374	0.497	0.611	**0.303**	0.909	**0.72**	0.828
TCOR	SVM	0.824	0.655	0.69	0.574	0.706	0.948	**0.895**	0.918
TCOR-E	AdaBoost	0	0	0	0	0	0.901	0	0.705
TCOR-E	KNN	0.831	0.812	0.744	0.779	0.8	0.965	0.885	0.942
TCOR-E	Naive Bayes	0.639	0.571	0.407	0.708	0.291	0.945	0.89	0.91
TCOR-E	RandomForest	0.61	0.495	0.503	0.642	0.182	0.922	0.716	0.865
TCOR-E	SVM	0.839	0.667	0.775	0.788	0.857	0.967	0.855	0.946

6.4. Clasificación de textos cortos mediante el método de indexado semántico latente (LSI)

Después de realizar el análisis de la clasificación de textos cortos y de observar el comportamiento del método que proponemos, se procede a realizar la comparación con uno de los métodos más parecidos al nuestro, encontrado en el trabajo relacionado: LSI, ver sección (2.2.1).

Como se mencionó en la sección 2.2.1, LSI tiene un parámetro libre (k) el cual sirve para controlar el tamaño de la proyección de los atributos más relevantes que se deben elegir. Por lo que en los experimentos realizados se fueron eligiendo distintos valores de k, donde $1 \leq k \geq min(\#d, \#t)$ recordemos que tiene que ser menor o igual al valor mínimo entre el número de documentos y el número de términos. Para el conjunto R8 fueron elegidos los siguientes valores: 500,1000,2000,3000 y 5000, para EasyAbstracts y CICLIng2002: 20 y 40. Cabe mencionar que para este experimento se realizaron todos los experimentos (diferentes pesados), pero solo mostramos los obtenidos con TF por ser los mejores.

Las figuras 6.2,6.3 y 6.4 muestran los resultados obtenidos por LSI. Como se puede observar en la figura 6.2, LSI tiene un bajo desempeño comparado con las DTRs y BOW en la colección R8, esto se puede deber a que, como se menciona en la literatura, LSI requiere de un conjunto extenso de datos de entrenamiento para obtener mejores resultados (Deerwester and Dumais, 1990; Paper, 2000). Si partimos de esta hipótesis, debe ser claro que para los dos siguientes conjuntos (EasyAbstracts y CICLIng2002) de datos LSI debería tener un desempeño similar o peor, como se logra ver en las figuras 6.3 y 6.4.

De lo anterior se puede concluir que las DTRs obtienen mejores resultados sin necesidad de tener un conjunto tan extenso de datos de entrenamiento como

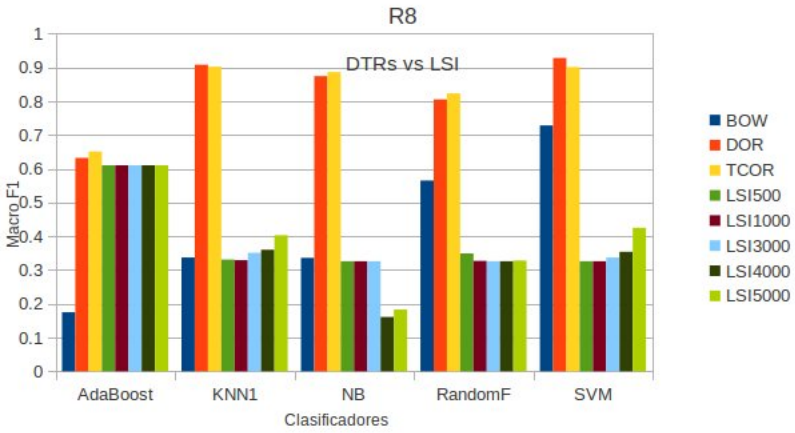


Figura 6.2: Gráficas comparativas del método propuesto frente a LSI en el conjunto de datos R8

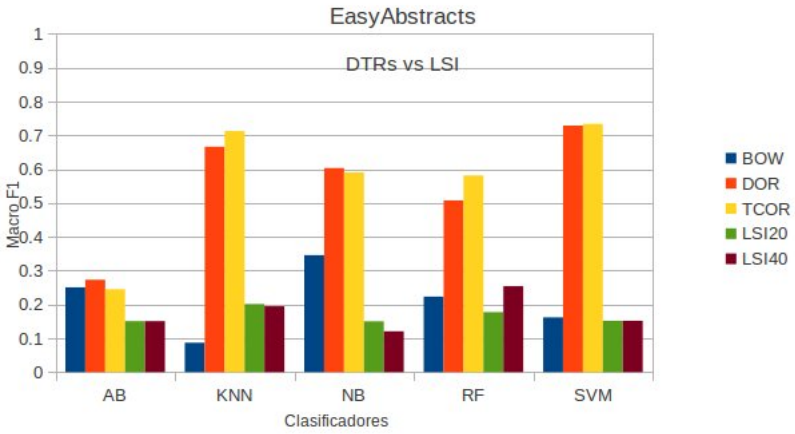


Figura 6.3: Gráficas comparativas del método propuesto frente a LSI en el conjunto de datos EasyAbstracts

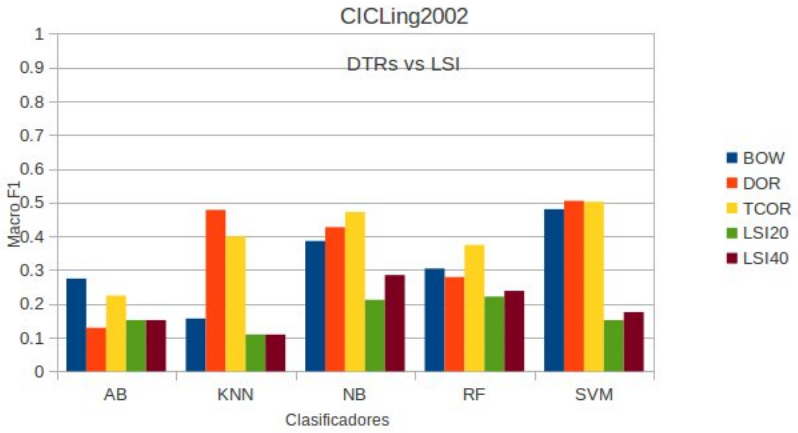


Figura 6.4: Gráficas comparativas del método propuesto frente a LSI en el conjunto de datos CICLIng2002

lo requiere LSI. Debido a estos resultados se buscó probar la estabilidad de los métodos propuestos cuando son aplicados a condiciones extremas, es decir, con muy pocos datos de entrenamiento, ver sección (6.5)

6.5. Clasificación de textos cortos y conjuntos de datos reducidos mediante DTRs

Tradicionalmente, las técnicas de aprendizaje supervisado utilizan un gran número de ejemplos etiquetados para generar un clasificador con buen desempeño, pero en la práctica esto puede resultar muy tedioso, debido a que es muy complicado y costoso obtener un conjunto etiquetado manualmente. Existen diferentes formas de solventar este problema, algunos usan información adicional como puede ser un conjunto de datos no etiquetados (Ramírez-de-la Rosa et al.), uti-

lizando algunas técnicas semi-supervisadas, el cual toma ventaja de documentos no etiquetados para que iterativamente genere mejores modelos de clasificación (Cabrera, 2010; Xu et al., 2008). Otro solución es aplicar el método de aprendizaje transductivo. Sin embargo, estos enfoques no son apropiados para la clasificación de documentos cortos. Por lo tanto, el objetivo general de este experimento es evaluar la efectividad y robustez de las DTRs en un escenario más retador, que consiste de tener textos cortos, y además un conjunto de datos de entrenamiento reducido para generar clasificadores con buen desempeño. Cabe mencionar que existe un trabajo que se enfocó a resolver el mismo problema y con el cual nos compararemos (Ramírez-de-la Rosa et al.), por lo que utilizaremos las mismas colecciones presentadas en el trabajo anteriormente mencionado.

En esta sección se analizarán las DTRs en la clasificación de textos cortos y conjuntos de entrenamiento reducidos. Primero, se proponen y explican los conjuntos de datos que serán usados. Posteriormente, se realiza la comparación y análisis de los resultados obtenidos del *baseline* y las DTRs. Debido a los resultados no satisfactorios anteriormente expuestos acerca de LSI, en esta sección se omitirá dicha representación.

Con la finalidad de evaluar la tarea de entrenamiento con un conjunto reducido, generamos variantes de las tres colecciones anteriormente mencionadas. Se crearon para el caso del conjunto R8 tres conjuntos reducidos: R8-train-50, R8-train-20, R8-train-10, los cuales incluyen el 50 %, 20 %, y 10 % de las instancias del conjunto de entrenamiento original. Por otra parte, para las otras colecciones restantes (EasyAbstracts y CICLIng2002) fueron generadas solo dos colecciones reducidas al 50 % y 20 %: EasyAbstracts-50, EasyAbstracts-20, CICLIng2002-50, CICLIng2002-20 respectivamente, esto debido a que estas colecciones originalmente son pequeñas, por lo que, si tomáramos el 10 % solo se entrenaría con un

documento por clase.

Tabla 6.4: Características principales de los conjuntos de datos de entrenamiento reducido

Características	Vocabulario del conjunto	Número de Documentos	Vocabulario prom. por doc.
R8-10	2000	550	38.33
R8-20	2929	1095	38.54
R8-50	4986	2732	38.98
EasyAbstracts-20	537	12	61.83
EasyAbstracts-50	833	26	59.24
CICLIng2002-20	330	12	43.09
CICLIng2002-50	608	26	44.17

Para mantener la distribución original del conjunto de datos los porcentajes de reducción fueron aplicados por clase. La tabla 6.4 presenta algunas de las características principales de estas nuevas colecciones. La construcción de estos conjuntos de datos fueron seleccionando documentos aleatoriamente de cada clase, este proceso se repitió cinco veces para cada conjunto de datos. Además se utilizaron dos distintos pesados (TF y TFIDF), se omitió el binario debido a que en el peor de los casos el pesado TF tiende a ser igual al Booleano.

Resultados

En la figura 6.5 se muestran los mejores resultados obtenidos por las distintas representaciones y las distintas reducciones de conjunto de entrenamiento del corpus R8 con los pesados TF y TF-IDF. Estas gráficas muestran que el método

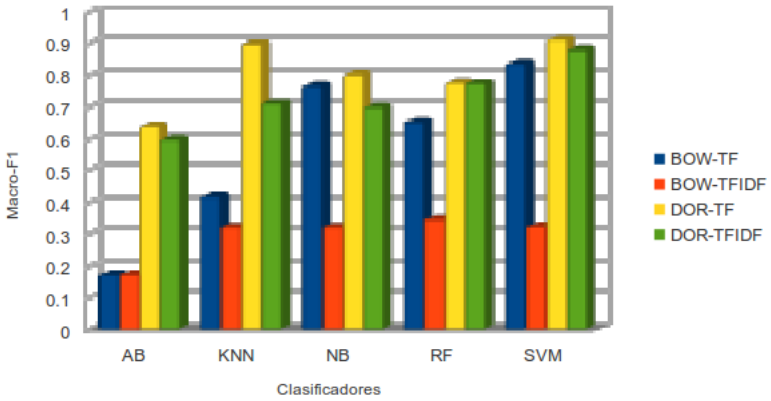
propuesto mejora la clasificación para los tres distintas particiones realizadas al conjunto de datos R8. Sin embargo, BOW tiene un bajo desempeño a medida que el conjunto de datos es reducido y sus mejores resultados son obtenidos mediante el pesado TF.

En cuanto a los otros dos corpora, el resultado fue muy similar. Para el caso de EasyAbstracts como se pueden ver en la figura 6.6(a) y 6.6(b). Tanto DOR y TCOR tienen un comportamiento muy similar, sin embargo, cuando se realizaron los experimentos utilizando la colección CICLIng2002 figuras 6.7(a) y 6.7(b), nos pudimos percatar que, aunque se tiene una mejora utilizando las DTRs no se tiene una diferencia tan marcada como en los dos casos anteriores. Esto se puede deber al dominio restringido que tiene y la dificultad que presenta este corpus como se mencionó en la sección 5.5 y en (Ingaramo et al., 2008). Por otro lado, podemos ver que uno de los clasificadores que tuvo buenos resultados en la mayoría de los experimentos fue el Naive Bayes (NB) y SVM, reafirmando los resultados mencionados en la literatura acerca de estos dos clasificadores en la clasificación de textos y ahora también en la clasificación de textos cortos.

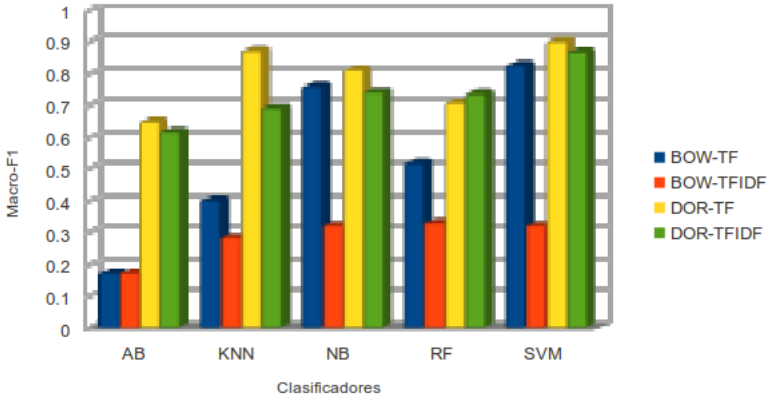
Estos resultados confirman que las DTRs tienen un buen desempeño cuando se tienen textos cortos y además un conjunto de datos reducidos simultáneamente. En adición a estas conclusiones, se puede ver que mejora los resultados obtenidos presentados en (Ramírez-de-la Rosa et al.), tabla 6.5.

Tabla 6.5: Tabla de comparación de clasificación de textos cortos y conjunto de entrenamiento reducido

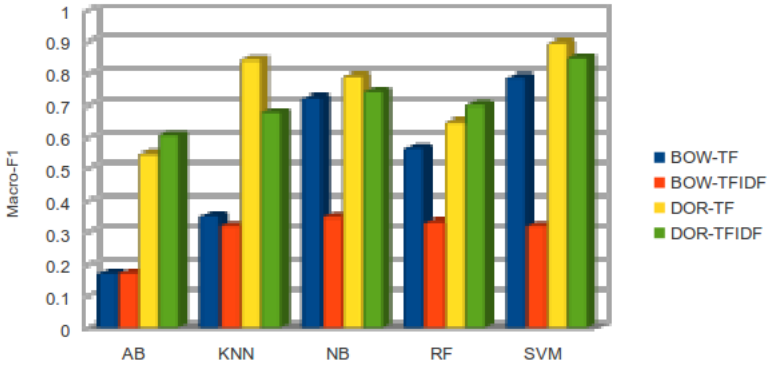
	BOW	Ramírez-de-la Rosa et al.	DOR	TCOR
R8-red50	0.572	0.672	0.749	0.770
R8-red20	0.538	0.657	0.792	0.788
R8-red10	0.524	0.639	0.809	0.813



(a) R8 reducido al 50 %

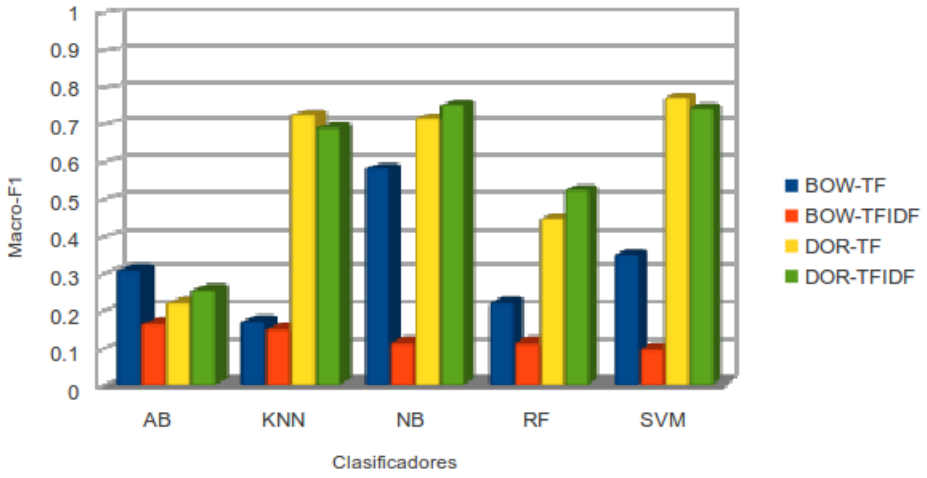


(b) R8 reducido al 20 %

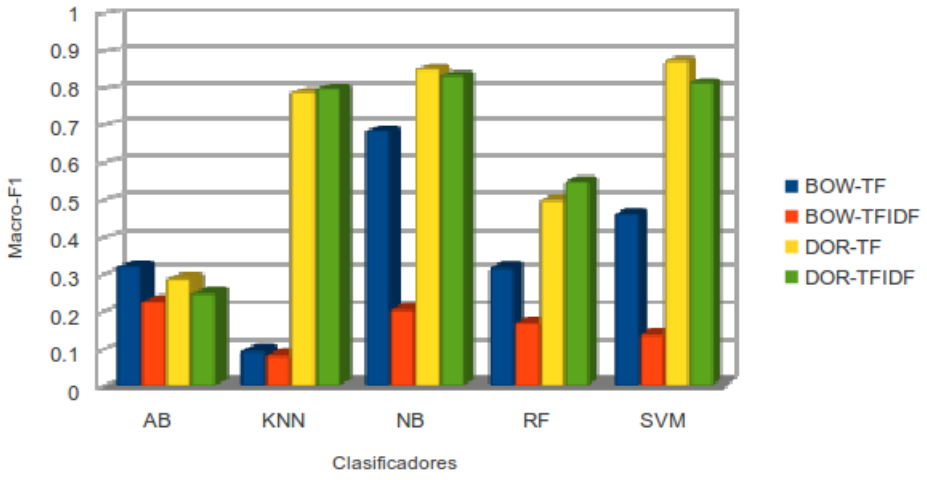


(c) R8 reducido al 10 %

Figura 6.5: Gráficas de los resultados obtenidos del conjunto de datos R8 con conjuntos de datos de entrenamiento reducidos en 10 %(c), 20 %(b) y 50 % (a).

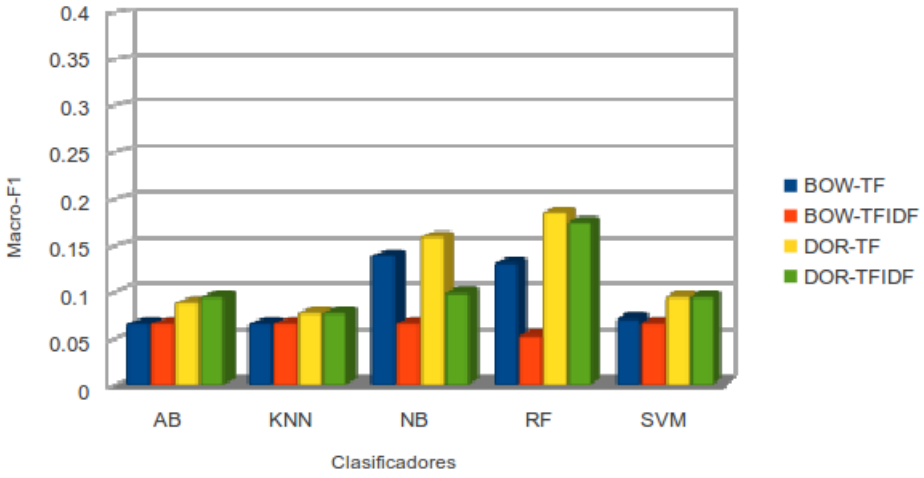


(a) EasyAbstracts reducido al 20 %

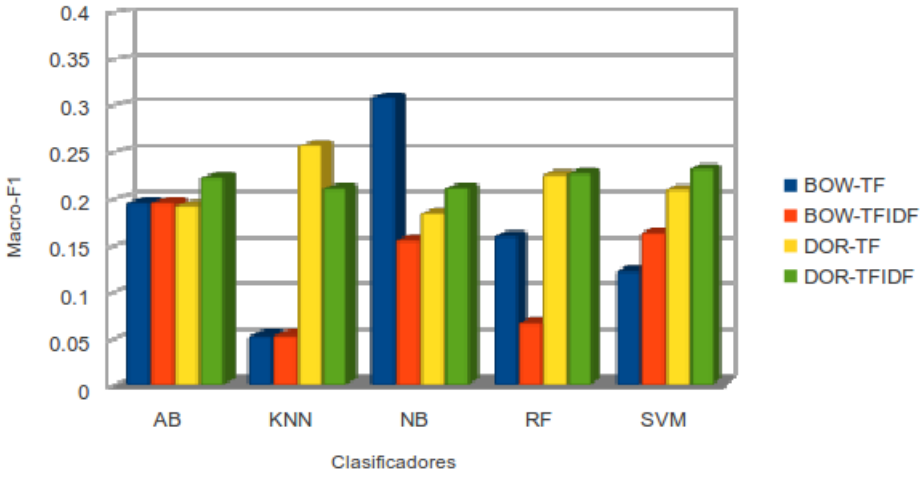


(b) EasyAbstracts reducido al 50 %

Figura 6.6: Gráficas de los resultados obtenidos de los conjuntos de datos EasyAbstracts con conjuntos de datos de entrenamiento reducidos en 20 %(a) y 50 % (b) y CICLIng2002: 20 %(a) y 50 % (b)



(a) CICLIng2002 reducido al 20 %



(b) CICLIng2002 reducido al 50 %

Figura 6.7: Gráficas de los resultados obtenidos de los conjuntos de datos EasyAbstracts con conjuntos de datos de entrenamiento reducidos al 20 %(a) y 50 % (b) y CICLIng2002: 20 %(a) y 50 % (b)

Capítulo 7

Conclusiones y trabajo futuro

7.1. Conclusiones

En este trabajo proponemos métodos basados en representaciones distribucionales de los términos (DTRs) para la clasificación de textos cortos, con el objetivo de mejorar el desempeño de la clasificación. El método se basa en la hipótesis de la expansión de los textos cortos mediante el enriquecimiento de los documentos y la modificación de los pesos orientados al contexto, ayudará a mejorar la efectividad de la clasificación de textos cortos. El funcionamiento general del método consiste en, crear una representación de cada término existente al que se le llamó "*conceptos*", estos conceptos se pueden obtener de dos formas DOR y TCOR, y son creados a partir de relaciones encontradas dentro de una colección, sección 4. Se obtienen los conceptos generados de cada uno de los términos del documento y se combinan estos conceptos para generar la nueva representación, por lo que en esta parte podemos decir que se expandió el contenido de los documentos.

Se evaluaron las representaciones propuestas usando tres colecciones de documentos de distinta complejidad cada una y de diferente naturaleza como son:

dos colecciones de resúmenes de artículos científicos y una colección de noticias. Además se realizaron experimentos comparando los métodos propuestos con BOW, LSI y en una condición adversas como es: conjunto de entrenamiento reducido. Los métodos presentados obtuvieron resultados aceptables, para la tarea de clasificación de textos cortos, obteniendo desde un 30 % hasta un 40 % de mejora con respecto a BOW y LSI. En la tarea de clasificación de textos cortos con conjunto de entrenamiento reducido, el promedio fue de 30 %. Aunado al esquema de clasificación de textos cortos se propuso un esquema de pesado supervisado TCOR-E, sección 4.2.2, con el cual se obtuvieron resultados similares a usar TCOR para la representación de los términos, como se muestra en la sección 6.2.

Las principales conclusiones que se obtuvieron de los experimentos realizados son:

- Los resultados obtenidos demuestran que la representación basada en bolsa de palabras no logra un buen desempeño en las clasificación de textos cortos.
- El uso de las representaciones distribucionales de los términos para la representación de documentos cortos permite obtener resultados satisfactorios, solventando los dos problemas principales de los textos cortos: bajas frecuencias y representaciones dispersas.
- En los dos escenarios (clasificación de textos cortos y clasificación de textos cortos con un conjunto de entrenamiento reducido) el método muestra una mejora significativa sobre dos (R8 y EasyAbstracts) de los tres conjuntos de datos evaluados. Además, los clasificadores con los mejores resultados obtenidos para la clasificación de textos cortos utilizando los dos escenarios fueron: k-Vecinos más cercanos, Máquinas de vectores de soporte y Naive

Bayes, reafirmando el buen desempeño que tienen estos clasificadores en otras tareas de clasificación de textos. Por lo consiguiente, al utilizar el método propuesto no se requiere de un conjunto de entrenamiento extenso para la obtener un buen desempeño en comparación de LSI.

- Los mejores resultados obtenidos mediante el método propuesto fueron obtenidos mediante el pesado TF.

- En el segundo escenario (conjunto de entrenamiento reducido) cuando se utiliza el tercer corpus (CICLIng2002) no se logró superar significativamente al método de referencia (BOW), debido a que es el conjunto de datos con mayor dominio restringido. Por lo que al tener un corpus de dominio restringido y además tener un conjunto de entrenamiento reducido, el método utilizado no nos permite encontrar las relaciones "conceptos" con un nivel suficiente de discriminación; causando que no se tenga un buen desempeño.

- Con respecto a la creación de la representación de los términos:

  - DOR demostró obtener mejores resultados en la mayoría de los casos y más aun cuando se tiene un conjunto de entrenamiento reducido.

  - TCOR captura y crea de mejor forma los conceptos de los términos cuando se tiene un amplio corpus de entrenamiento en comparación con DOR. Por lo consiguiente, tiene problemas cuando se reduce el conjunto de entrenamiento.

  - El método supervisado creado (TCORE), aunque no supera significativamente a los otros dos, en varios casos obtiene mejores resultados que TCOR.

- Finalmente, con base en los resultados obtenidos en la evaluación experimental realizada en este trabajo y además con la ayuda de las medidas de evaluación de los conjuntos de datos, es posible predecir las situaciones en las que el método que se propone es funcional. Se puede obtener un desempeño aceptable cuando: 1) la longitud por documento tiene como mínimo 6 términos pero se cuenta con un conjunto de entrenamiento con documentos más extensos. 2) Si el corpus es de dominio restringido, se recomienda tener un corpus de entrenamiento amplio, para poder crear los conceptos con una mayor confianza y así obtener mejores resultados.

7.2. Trabajo futuro

Durante la investigación que se llevó a cabo surgieron algunas inquietudes que se consideran para trabajo futuro:

- **Analizar el impacto de la redundancia** al combinar las representaciones DOR y TCOR o BOW y DOR o BOW y TCOR. Debido a que se podría analizar si los conceptos encontrados por DOR son complementarios a los encontrados por TCOR o qué impacto en la discriminación de las clases tienen los términos en conjunto con los conceptos (BOW + [DOR o TCOR]).

- **Estudio del método TCOR-E**, analizar bajo que circunstancias TCOR-E puede ser mejor que TCOR o incluso en que condiciones TCOR-E tiene un buen desempeño.

- **Estudiar el desempeño de las DTRs en el agrupamiento** de textos cortos.

- **Analizar el impacto de las representaciones de los documentos enriqueciéndola con información multilingüe**. Realizar el cálculo de las representaciones en un conjunto de entrenamiento de idioma X y un conjunto externo en otro idioma Y.

- **Transductive cross-domain**, donde el cálculo de las representaciones se realice sobre la conjunción del conjunto de entrenamiento y pruebas, en una tarea donde no existen documentos de entrenamiento etiquetados en ese dominio. No obstante existen documentos etiquetados en otro dominio pero de alguna forma los dominios están relacionados y que se podría utilizar para ayudar a la clasificación de documentos destino.

Bibliografía

D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003.

L. Breiman. Random forests. *Machine Learning*, 45:5–32, 2001.

R. G. Cabrera. Categorización semi-supervisada de documentos usando la web como corpus. *Procesamiento de Lenguaje Natural*, 46(0), 2010.

J. Cai, Y. Tang, and R. Hu. Spam Filter for Short Messages Using Winnow. *2008 International Conference on Advanced Language Processing and Web Information Technology*, pages 454–459, 2008.

A. Cardoso-Cachopo and A. Oliveira. An empirical comparison of text categorization methods. In M. Nascimento, E. de Moura, and A. Oliveira, editors, *String Processing and Information Retrieval*, volume 2857 of *Lecture Notes in Computer Science*, pages 183–196. Springer Berlin / Heidelberg, 2003.

A. Cardoso-Cachopo and A. Oliveira. Combining LSI with other classifiers to improve accuracy of single-label text categorization. In *First European Workshop on Latent Semantic Analysis in Technology Enhanced Learning*, Netherlands, 2007.

H. C. Chen, T. Yim, D. Fye, and B. R. Schatz. Automatic thesaurus generation for an electronic community system. *Journal of the American Society for Information Science*, 46:175–193, 1995.

Z. Chen, T. Liu, and S. Liu. An Evaluation on Feature Selection for Text Clustering. In *Proceedings of the Twentieth International Conference on Machine Learning*, Washington DC, 2003.

P. Comon. Independent component analysis, a new concept? *Signal Processing*, 36(3):287 – 314, 1994. ISSN 0165-1684. doi: 10.1016/0165-1684(94)90029-9. <ce:title>Higher Order Statistics</ce:title>.

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

C. Crouch. An approach to the automatic construction of global thesauri. *Information Processing and Management*, 26(5):629–640, 1990.

F. Debole and F. Sebastiani. Supervised term weighting for automated text categorization. In *Proceedings of the 2003 ACM symposium on Applied computing*, SAC '03, pages 784–788, New York, NY, USA, 2003. ACM.

S. Deerwester and S. Dumais. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

L. Eikvil and K. Aas. Text Categorisation: Survey. *Norwegian Computing Center Mimeograph*, 1999.

H. Escalante, M. Montes, and E. Sucar. Multimodal indexing based on semantic cohesion for image retrieval. *Information Retrieval*, 15(1):1–32, 2011.

Z. Faguo, Z. Fan, and Y. Bingru. Research on Short Text Classification Algorithm Based on Statistics and Rules. *Third International Symposium on Electronic Commerce and Security*, (2):3–7, July 2010.

X. Fan and H. Hu. A New Model for Chinese Short-text Classification Considering Feature Extension. pages 7–11. Ieee, Oct. 2010a.

X. Fan and H. Hu. Construction of High-quality Feature Extension Mode Library for Chinese Short-text Classification. pages 87–90. Ieee, Aug. 2010b.

E. Frank and R. Bouckaert. Naive bayes for text classification with unbalanced classes. *Knowledge Discovery in Databases: PKDD 2006*, pages 503–510, 2006.

Y. Freund and R. Schapire. Experiments with a new boosting algorithm. pages 148–156. MORGAN KAUFMANN PUBLISHERS, INC., 1996.

R. Freund, Y. Schapire. Experiments with a New Boosting Algorithm. *Proceedings of the Thirteenth International Conference on Machine Learning*, pages 148–156, 1996.

W. Gale, K. Church, and D. Yarowsky. A method for disambiguating word senses in a large corpus. *Computers and the Humanities*, 26:415–439, 1992. 10.1007/BF00136984.

S. I. Gallant. A Practical Approach for Representing Context and for Performing Word Sense Disambiguation Using Neural Networks. *Neural Computation*, 3 (3):293–309, Sept. 1991.

S. R. Garner. Weka: The Waikato environment for knowledge analysis. In *In Proc. of the New Zealand Computer Science Research Students Conference*, pages 57–64, 1995.

M. Girolami and A. Kabán. On an equivalence between plsi and lda. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, SIGIR '03, pages 433–434, New York, NY, USA, 2003. ACM.

P. Gu, Q. Zhu, and C. Zhang. A multi-view approach to semi-supervised document classification with incremental Naive Bayes. *Computers & Mathematics with Applications*, 57(6):1030–1036, Mar. 2009.

Z. S. Harris. *Mathematical structures of language*. Wiley, 1968.

F. He and X. Ding. Improving Naive Bayes Text Classifier Using Smoothing. pages 703–707, 2007.

X. He, C. Zhu, and T. Zhao. Research on short text classification for web forum. In *Fuzzy Systems and Knowledge Discovery (FSKD), 2011 Eighth International Conference on*, volume 2, pages 1052 –1056, july 2011.

T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '99, pages 50–57, New York, NY, USA, 1999. ACM.

A. Hyvrinen. Survey on independent component analysis. *Neural Computing Surveys*, pages 94–128, 1999.

D. Ingaramo, D. Pinto, P. Rosso, and M. Errecalde. Evaluation of internal validity measures in short-text corpora. pages 555–567, 2008.

D. Ingaramo, M. Errecalde, P. Rosso, U. Nacional, and D. S. Luis. A General Bio-inspired Method to Improve the Short-Text Clustering Task. *Computational Linguistics and Intelligent Text Processing*, pages 661–672, 2010.

T. Joachims. Text categorization with support vector machines: learning with many relevant features. In C. Nédellec and C. Rouveirol, editors, *Proceedings of ECML-98, 10th European Conference on Machine Learning*, number 1398, pages 137–142, Chemnitz, DE, 1998. Springer Verlag, Heidelberg, DE.

J. Karlgren and M. Sahlgren. From Words to Understanding. *Information Retrieval*, pages 294–311, 1996.

R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. *International joint Conference on artificial intelligence*, 1995.

T. Landauer, P. Foltz, and D. Laham. An introduction to latent semantic analysis. *Discourse Processes*, 25(2):259–284, 1998.

A. LAVELLI, F. SEBASTIANI, and R. ZANOLI. Distributional Term Representations: An Experimental Comparison. *Italian Workshop on Advanced Database Systems*, 2004.

D. D. Lewis and W. B. Croft. Term clustering of syntactic phrases. In *Proceedings of the 13th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '90, pages 385–404, New York, NY, USA, 1990. ACM.

P. Makagonov, M. Alexandrov, and A. Gelbukh. Clustering Abstracts instead of Full Texts. *Notes*, pages 129–135, 2004.

S. Z. Marquez and Finella. Transductive Learning For Short-Text Classification Problems Using Latent Semantic Indexing. *International Journal of Pattern Recognition and Artificial Intelligence*, 19(2):143–163, 2005.

A. Mccallum and K. Nigam. A Comparison of Event Models for Naive Bayes Text Classification. *Dimension Contemporary German Arts And Letters*, 1997.

T. Mitchell. *Machine Learning*. PhD thesis, 1997.

M. Nagarajan, A. Sheth, M. Aguilera, and K. Keeton. Altering Document Term Vectors for Classification - Ontologies as Expectations of Co-occurrence. *Re-CALL*, pages 1225–1226, 2007.

U. Pandey and S. Chakraverty. A Review of Text Classification Approaches for E-mail Management. *International Journal of Engineering*, 3(2), 2011.

F. Paper. Latent Semantic Indexing : An overview. pages 1–16, 2000.

X.-H. Phan, L.-M. Nguyen, and S. Horiguchi. Learning to classify short and sparse text & web with hidden topics from large-scale data collections. *Proceeding of the 17th international conference on World Wide Web - WWW '08*, page 91, 2008.

X.-H. Phan, C.-T. Nguyen, D.-T. Le, L.-M. Nguyen, S. Horiguchi, and Q.-T. Ha. A hidden topic-based framework towards building applications with short web documents. *IEEE Transactions on Knowledge and Data Engineering*, 23(7): 961–976, 2011.

D. Pinto. *On Clustering and Evaluation of Narrow Domain Short-Text Corpora*. PhD thesis.

D. Pinto, P. Rosso, and H. Jimenez-Salazar. A Self-enriching Methodology for Clustering Narrow Domain Short Texts. *The Computer Journal*, pages 1–18, Sept. 2010.

M. F. Porter. An algorithm for suffix stripping. *Program: electronic library & information systems*, 40(3):211–218, 2006.

P. Praks. Latent Semantic Indexing for Image Retrieval Systems. In *In: SIAM Conference on Applied Linear Algebra*, Williamsburg, USA, 2003. International Linear Algebra Society (ILAS).

Q. Pu and G.-W. Yang. Short-Text Classification Based on ICA and LSA. pages 265–270. 2006.

G. Ramírez-de-la Rosa, M. Montes-y Gómez, T. Solorio, and L. Villaseñor-Pineda. A document is known by the company it keeps: neighborhood consensus for short text categorization. *Language Resources and Evaluation*, pages 1–23. ISSN 1574-020X. 10.1007/s10579-012-9192-1.

K. D. Rosa and J. Ellen. Text Classification Methodologies Applied to Micro-Text in Military Chat. *2009 International Conference on Machine Learning and Applications*, pages 710–714, Dec. 2009.

V. Rosas, M. L. Errecalde, and P. Rosso. Un Analisis Comparativo de Estrategias para la Categorización Semantica de Textos Cortos. *Sociedad Española para el Procesamiento del Lenguaje Natural*, 44:11–18, 2010.

P. Rosso and S. Luis. Clustering Iterativo de Textos Cortos con Representaciones basadas en conceptos. *Sociedad Española para el Procesamiento del Lenguaje Natural*, pages 19–26, 2011.

M. C. Ruiz. *Representaciones Vectoriales Orientadas a Conceptos y Estructura para Recuperación de Información*. PhD thesis, 2010.

M. Sahlgren and R. Cöster. Using bag-of-concepts to improve the performance of support vector machines in text categorization. *Proceedings of the 20th international conference on Computational Linguistics - COLING '04*, pages 1–7, 2004.

G. Salton. *The SMART Retrieval System*. Prentice-Hall, 1971.

G. Salton. Developments in automatic text retrieval. *Science*, 253:974 – 979, 1991.

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

J. O. Shea, Z. Bandar, K. Crockett, and D. Mclean. A Comparative Study of Two Short Text Semantic Similarity Measures. *Artificial Intelligence*, pages 172–181, 2008.

D. Shen, J. Wu, B. Cao, J.-T. Sun, Q. Yang, Z. Chen, and Y. Li. Exploiting term relationship to boost text classification. *Proceeding of the 18th ACM conference on Information and knowledge management - CIKM '09*, page 1637, 2009.

K.-s. Shim, C.-Y. Ock, D.-M. Kim, H.-S. Choe, and C.-H. Kim. Finding Similar Texts Using U-WIN. *2008 International Conference on Advanced Language Processing and Web Information Technology*, (1):43–48, 2008.

B. Sriram, D. Fuhry, E. Demir, and H. Ferhatosmanoglu. Short Text Classification in Twitter to Improve Information Filtering. *Performance Evaluation*, pages 841–842, 2010.

P. D. Turney and P. Pantel. From Frequency to Meaning: Vector Space Models of Semantics. *J. Artif. Intell. Res. (JAIR)*, 37:141–188, Mar. 2010.

L. A. Ure. *Thesis Automatic Text Categorization of documents in the High Energy Physics domain.* 2005.

R. Venegas. La similitud léxico-semántica en artículos de investigación científica en español: Una aproximación desde el Análisis Semántico Latente. *Revista signos*, 39:75–106, 00 2006.

J. Wang, Y. Zhou, L. Li, B. Hu, and X. Hu. Improving short text clustering performance with keyword expansion. In H. Wang, Y. Shen, T. Huang, and Z. Zeng, editors, *The Sixth International Symposium on Neural Networks (ISNN 2009)*, volume 56 of *Advances in Intelligent and Soft Computing*, pages 291–298. Springer Berlin / Heidelberg, 2009.

Y. T. W. Xi-wei. Feature Extension for short text. *Proceedings of the Third International Symposium on Computer Science and Computational Technology*, (August):338–341, 2010.

Z. Xu, R. Jin, K. Huang, M. R. Lyu, and I. King. Semi-supervised text categorization by active search. In *Proceeding of the 17th ACM conference on Information and knowledge mining - CIKM '08*, page 1517, New York, New York, USA, 2008. ACM Press.

S. Zelikovitz. Improving Short-Text Classification Using Unlabeled Background Knowledge to Assess Document Similarity. *Test*, 1999.

S. Zelikovitz. Transductive lsi for short text classification problems. *American Association for Artificial Intelligence*, 2004.