Método eficaz para el Filtrado Adaptativo de Documentos mediante Indexado Aleatorio

Adrian Fonseca Bruzón, Aurelio López López, José Eladio Medina Pagola

Reporte Técnico No. CCC-16-002
15 de Febrero de 2016

Luis Enrique Erro 1
Sta. Ma. Tonantzintla,
72840, Puebla, México.

Método eficaz para el Filtrado Adaptativo de Documentos mediante Indexado Aleatorio

Adrian Fonseca Bruzón, Aurelio López López, José Eladio Medina Pagola

Departamento de Ciencias Computacionales

Instituto Nacional de Astrofísica, Óptica y Electrónica

Luis Enrique Erro # 1, Santa María Tonantzintla, Puebla, México. CP: 72840

Correo electrónico: {adrian, allopez}@inaoep.mx, jmedina@cenatav.co.cu

Resumen.

Ante los grandes volúmenes de información que son generados diariamente en la Web, los sistemas de Filtrado Adaptativos de Información se han convertido en una herramienta muy poderosa para los usuarios. Estos sistemas permiten a los usuarios concentrarse en la información relevante mientras que el sistema descarta toda aquella información que no es de su interés. Varios han sido los métodos de Filtrado reportados en la literatura y en su mayoría hacen uso del tradicional modelo vectorial para la información contenida en los documentos y en los perfiles de usuarios, aun cuando es conocido que este modelo presenta una serie de limitaciones. Si bien en la literatura se han reportado varias alternativas para solventar los problemas del modelo vectorial, estas son muy costosas para ser empleadas en entornos, como Filtrado Adaptativo, en el cual se requiere de actualizar con frecuencia la información almacenada. Esta propuesta doctoral se plantea desarrollar un algoritmo de Filtrado Adaptativo en el cual se emplee una representación de los documentos que no asuma independencia estadística entre los términos, cosa que sí hace el modelo vectorial, pero que no sea tan costosa en obtención y que pueda ser construida, preferiblemente, de forma incremental para que se ajuste con mayor facilidad a la tarea de Filtrado. Los resultados preliminares han estado encaminados a estudiar el Indexado Aleatorio, y sus variantes. El Indexado Aleatorio es un método para representar los documentos que puede ser obtenido de forma incremental y que puede ser una alternativa viable en la tarea del Filtrado Adaptativo.

Tabla de Contenidos

1. Introducción

Hoy día el auge de Internet ha provocado que diariamente se genere un volumen enorme de documentos de diversa naturaleza. Este auge trae aparejado que en ocasiones los usuarios corran el riesgo de quedar abrumados por este flujo creciente de información disponible [20] [1]. Para dar respuesta a esta problemática se han propuesto en la literatura varias soluciones que facilitan el acceso a la información que satisface nuestras necesidades de información sin necesidad de inspeccionar de forma manual todos los documentos. Dos de estas soluciones son la Recuperación de Información y el Filtrado de Información. Estas son tareas muy similares en el sentido de que ambas tienen la finalidad de proveer al usuario con información útil con respecto a una determinada necesidad de información. Sin embargo, existe una gran diferencia entre ellas. El Filtrado de Información está dirigido a satisfacer las necesidades de información que perduran durante un período largo de tiempo, mientras que la Recuperación de Información se centra en consultas de corta duración las cuales son, con frecuencia, descartadas una vez terminada la sesión de búsqueda.

El objetivo del Filtrado de Información (FI) es clasificar los documentos de un flujo, en relevante o no relevante, de acuerdo con el interés de un usuario particular [33]. Los sistemas de FI se dirigen a necesidades de información relativamente estables a largo plazo, aunque usualmente permiten que estos intereses puedan modificarse de forma gradual en el tiempo [27]. En tal caso se les conoce como sistemas de Filtrado de Información Adaptativos [77].

2. Filtrado Adaptativo de Información

En la tarea de Filtrado de Información se monitorea de forma continua un flujo de información con el fin de detectar aquellos documentos que se acercan a las necesidades de información de los usuarios. Esto ayuda a los usuarios a concentrarse solamente en aquellos documentos recuperados.

Uno de los componentes más importantes de un sistema de filtrado consiste en la representación, usualmente denominada *perfil*, del interés del usuario. Este perfil se mantiene durante el tiempo que dura la necesidad. En el caso del Filtrado Adaptativo, este perfil de usuario puede ser mejorado si se cuenta con retroalimentación explícita o implícita por parte del usuario.

De forma general podemos encontrar dos enfoques diferentes a la tarea de Filtrado de Información: el filtrado basado en contenido y el colaborativo [75]. Estos enfoques difieren en la forma en que son representados y comparados los perfiles de usuarios y los documentos. En el filtrado basado en contenido tanto el perfil del usuario como los documentos son representados empleando características extraídas de los propios documentos. En el caso de aquellos que siguen el enfoque colaborativo, los elementos son caracterizados por la puntuación que reciben por parte de los usuarios. En este caso, el perfil del usuario es construido mediante la valoración que emite éste sobre los documentos y su similaridad con lo expresado por otros usuarios con intereses similares. También conocidos como sistemas de Recomendación, en este enfoque el objetivo consiste en recomendar nuevos documentos para el usuario. Este proceso se realiza analizando el conjunto de usuarios que comparten intereses similares de información y en la puntuación que estos han dado a los documentos. En este trabajo nos centraremos en los sistemas de filtrado basados en contenido.

Los algoritmos de Filtrado Adaptativo de Información filtran los documentos relevantes para el usuario presentes en un flujo de información. En el filtrado adaptativo, aquellos documentos que son potencialmente relevantes deben ser entregados al usuario inmediatamente, por ende el sistema no tiene tiempo de acumularlos y ordenarlos de acuerdo a su relevancia como en los sistemas tradicionales de recuperación de información. Un sistema de filtrado adaptativo usualmente toma

una decisión binaria ante cada nuevo documento: recuperar o descartar el documento para cada usuario en particular.

Generalmente, estos sistemas comienzan con un perfil de usuario y muy pocas muestras positivas. La relevancia de los documentos es dependiente de los cambios que se producen en las necesidades de información de los usuarios. Los intereses de los usuarios pueden cambiar producto de cambios en el entorno del usuario o en su conocimiento, entre otras tantas causas. Por ello, el perfil es luego adaptado empleando la retroalimentación provista por el usuario sobre los documentos recuperados. Esta retroalimentación puede ser explícita o implícita. Es explícita si es el propio usuario el que indica cuándo un documento recuperado es realmente de su interés o no. Es implícita si es el sistema quien infiere la relevancia del documento. Para ello se vale de heurísticas como pueden ser: el tiempo que tiene abierto el documento o si el mismo es borrado sin leerlo.

En esta tarea el proceso de ajuste del perfil del usuario representa un elemento clave en el diseño. Los perfiles de los usuarios representan elementos duraderos en el tiempo que deben ser actualizados constantemente en función de los cambios producidos en los intereses del usuario a lo largo del tiempo. La calidad de un sistema de Filtrado Adaptativo de Información está sustentada en su capacidad de mantener un perfil del usuario que realmente permita alcanzar una exactitud elevada a medida que el mismo es utilizado.

En un sistema de Filtrado Adaptativo no se dispone de todo el conjunto de entrenamiento desde un inicio para construir el clasificador; sino que este se va incrementando, a medida que se van adicionando nuevas muestras. Al poder incorporar nuevas muestras al clasificador, podemos adaptar su comportamiento a los intereses cambiantes de los usuarios.

Estos algoritmos se basan en la existencia de muestras etiquetadas adquiridas por medio de la retroalimentación, implícita o explícita, realizada por el usuario para su desempeño. Sin embargo, el número de muestras etiquetadas puede ser muy pequeño o incluso nulo, fundamentalmente, en las primeras etapas del proceso de filtrado. Este fenómeno es conocido como "*Inicio Frío*" [74].

Los cambios ocurridos en los intereses de los usuarios a lo largo del tiempo es un ejemplo del fenómeno conocido en la literatura como *Concept Drift* [17]. Este es un fenómeno que de forma más general está relacionado con los ambientes en los cuales ocurren cambios a los largo del tiempo. En este contexto el término Aprendizaje Adaptativo está relacionado con la actualización en línea del clasificador durante su operación para reaccionar ante estos cambios [17].

2.1 Filtrado Adaptativo en las competiciones TREC

Los sistemas de Filtrado Adaptativo han sido ampliamente influenciados por la tarea de Filtrado de Información de las competiciones TREC [51][52]. Esta tarea se introdujo en el año 2000 en la TREC9, y continuó en los dos siguientes años. Recientemente volvió a ser convocada como parte de los TREC-2012 [66], aunque en esta ocasión dentro del análisis de microblogs. La colección de datos empleada en esta tarea únicamente se puso disponible para los equipos participantes, previa firma de un acuerdo de no divulgación. En la presente edición se volvió a convocar el filtrado de microblogs, aunque ahora orientada a la recomendación de tweets en tiempo real, más que a la adaptabilidad en sí.

La tarea fue diseñada con el objetivo de medir la capacidad de los sistemas de construir perfiles de usuario persistentes que sean capaces de separar de forma efectiva los documentos relevantes de los no relevantes en un flujo de información. En el caso de la tarea de Filtrado Adaptativo los sistemas debían comenzar su funcionamiento con tan solo una descripción del tópico y un número muy

reducido de muestras positivas (a lo sumo 3), estos debían ser capaces de construir un mejor perfil a partir de la retroalimentación obtenida en línea.

De acuerdo a los organizadores, con la tarea se intentaba simular aplicaciones de filtrado de textos en línea con restricciones de tiempo, donde el valor de un documento se degradaba rápidamente con el paso del tiempo. Por ello los sistemas debían presentar de forma inmediata a los usuarios los documentos potencialmente relevantes.

En la definición de la tarea se asumió que los usuarios revisaban de forma periódica los documentos recuperados. Por ello, los sistemas disponían de la validez (pertenencia) o no de los documentos recuperados de forma instantánea. Si bien esta es una suposición bastante simplista, facilita bastante el proceso de evaluación y comparación de los sistemas.

La configuración dada a la tarea impone varias restricciones a los sistemas:

1- Los documentos deben ser presentados tan pronto como estos arriban. Por tal razón, solamente se dispone de la información almacenada hasta el momento para realizar la evaluación y decidir si es potencialmente relevante o no.
2- No se permite almacenar los documentos sobre los cuales no se tiene una certeza de su clasificación para un momento posterior cuando se dispone de un perfil más completo que permita realizar una clasificación correcta. Durante la evaluación de un documento no se dispondrá de ninguna información sobre los futuros documentos, y el conjunto de documentos recuperados es dependiente del tiempo y el orden en que arriban los documentos.
3- El sistema solamente tendrá retroalimentación de aquellos documentos que son presentados al usuario como potencialmente relevantes. Una vez que un documento es etiquetado como No Relevante no se dispondrá de la retroalimentación que permita conocer si realmente no era un documento de interés.
4- Una vez que un documento es clasificado esta decisión es final, no se permite cambiar esta clasificación (a los efectos de medir la calidad de los sistemas).

La tarea en la competición TREC está orientada al tratamiento de necesidades de información estables y duraderas en el tiempo, dejando fuera de momento la temática relacionada el *Concept Drift*, y en consecuencia la colección fue planteada sin esta característica.

Colección de Documentos

La colección de documentos más empleada durante la evaluación de los sistemas ha sido el corpus RCV1 [34]. Esta es una colección compuesta por aproximadamente 800000 noticias periodísticas o boletines de titulares originadas en un período de tiempo entre 1996 y 1997.

Para las competiciones TREC se creó un conjunto de 100 tópicos. Los primeros 50 fueron creados de forma manual por los asesores de la NIST. Los otros 50 tópicos fueron construidos como intersecciones entre pares de categorías originales de la colección. Estos tópicos fueron escogidos de forma tal que aparentaran ser resultados significativos de una búsqueda.

Estos tópicos no todos tienen el mismo comportamiento a lo largo del tiempo ni están igualmente representados. Para ilustrar esta situación en la Figura 1 se muestra el comportamiento del flujo de documentos para los cuales se tiene información de retroalimentación de 4 tópicos de la colección.

Como se puede apreciar, las muestras no se encuentran homogéneamente distribuidas a lo largo del tiempo. Además, en tópicos como R108 y R111 se puede apreciar un fuerte desbalance entre las muestras negativas y las positivas, incluso llegando estas a tener solamente 15 muestras positivas en todo el flujo de datos. Estas características imponen por si mismas grandes retos a los sistemas que se evalúan en esta colección.

Por otro lado, del análisis de los gráficos presentados podemos ver que en estos flujos solo van apareciendo ejemplos de un solo tópico, y no se encuentra presente el fenómeno del *Concept Drift*.

Medida de Evaluación

Las competiciones TREC introdujeron dos medidas para evaluar la calidad de los sistemas: F-beta y T11SU.

F-beta

La medida F-beta es una función que depende de la precisión y la relevancia, junto con un parámetro libre beta que pondera el aporte relativo de la relevancia y la precisión en el resultado final. Cualquiera sea el valor de beta, esta medida toma valores entre 0 (mal) y 1 (bien). En el caso de las competiciones TREC el valor de beta fue fijado a 0.5, haciendo más énfasis de esta forma en la precisión sobre la relevancia. La medida finalmente empleada, una vez fijado beta, puede ser expresada mediante la ecuación:

$$T11F = \frac{1.25 * Núm.\,de\ documentos\ relevantes\ recuperados}{Núm.\,de\ docs\ recuperados + 0.25 * Núm.\,de\ docs\ relevantes}$$

Si no se recupera ningún documento como relevante el valor de la medida es 0.

T11SU

La otra medida empleada en la competición fue la T11SU, la cual se expresa en función de la utilidad lineal de un sistema:

$$T11U = 2 * Núm.\,de\ docs\ Rel\ recuperados - \ Núm.\,de\ docs\ No\ Rel\ recuperados$$

La máxima utilidad que puede obtener un sistema de filtrado de acuerdo a esta medida está dado por la expresión:

$$MaxU = 2 * Núm.\,de\ docs\ Rel$$

La expresión de la medida T11SU viene dada por la expresión:

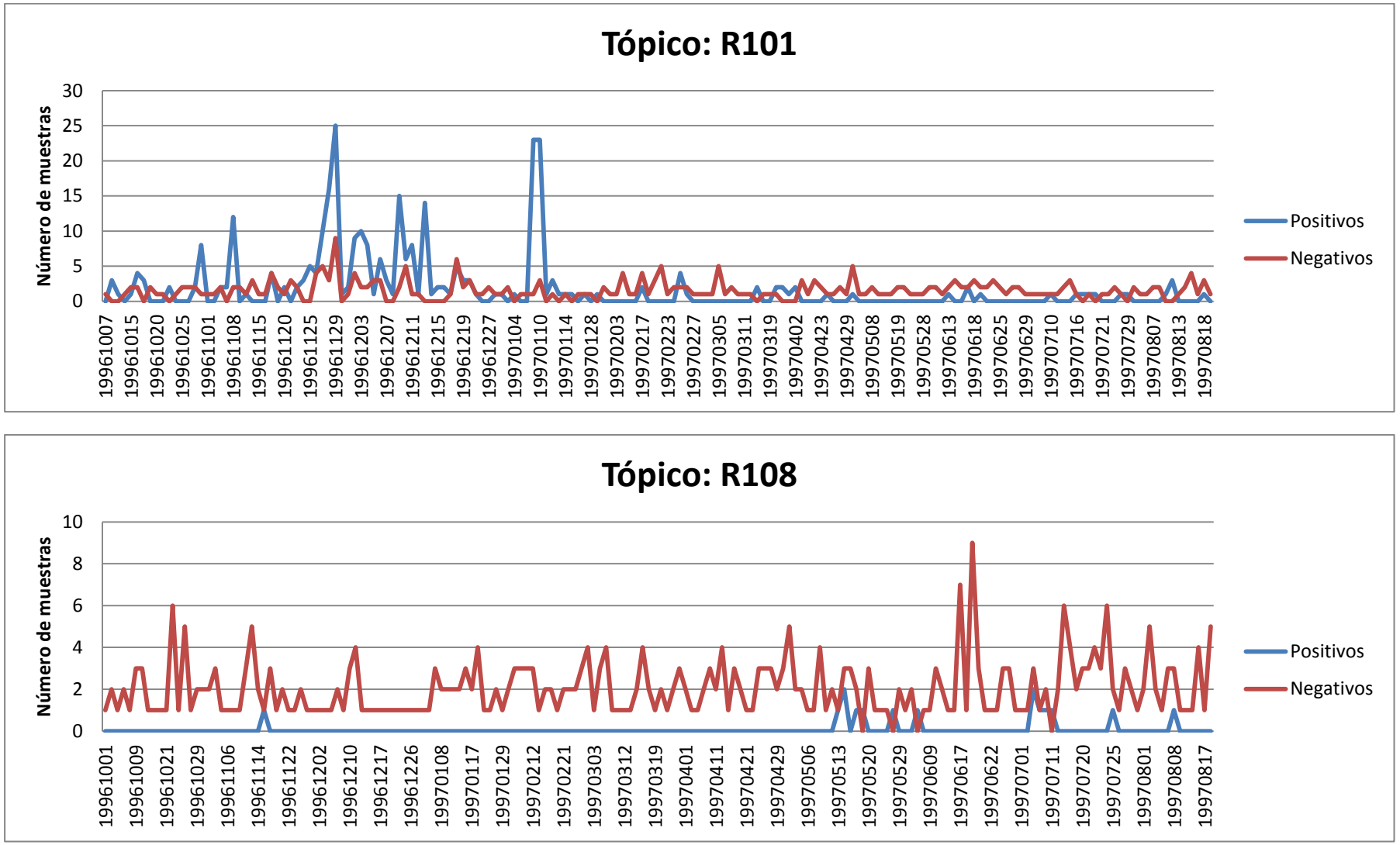
$$T11SU = \frac{max(T11NU, MinNU) - MinNU}{1 - MinNU}$$

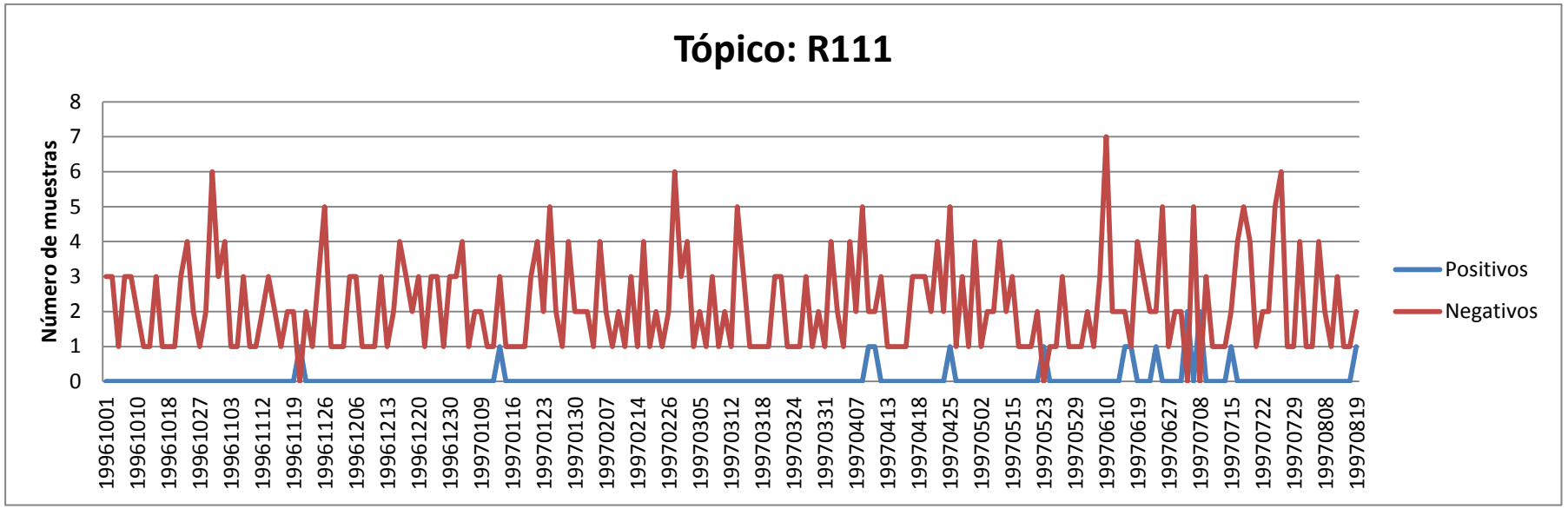
Donde T11NU es la normalización de T11U, mediante la expresión:

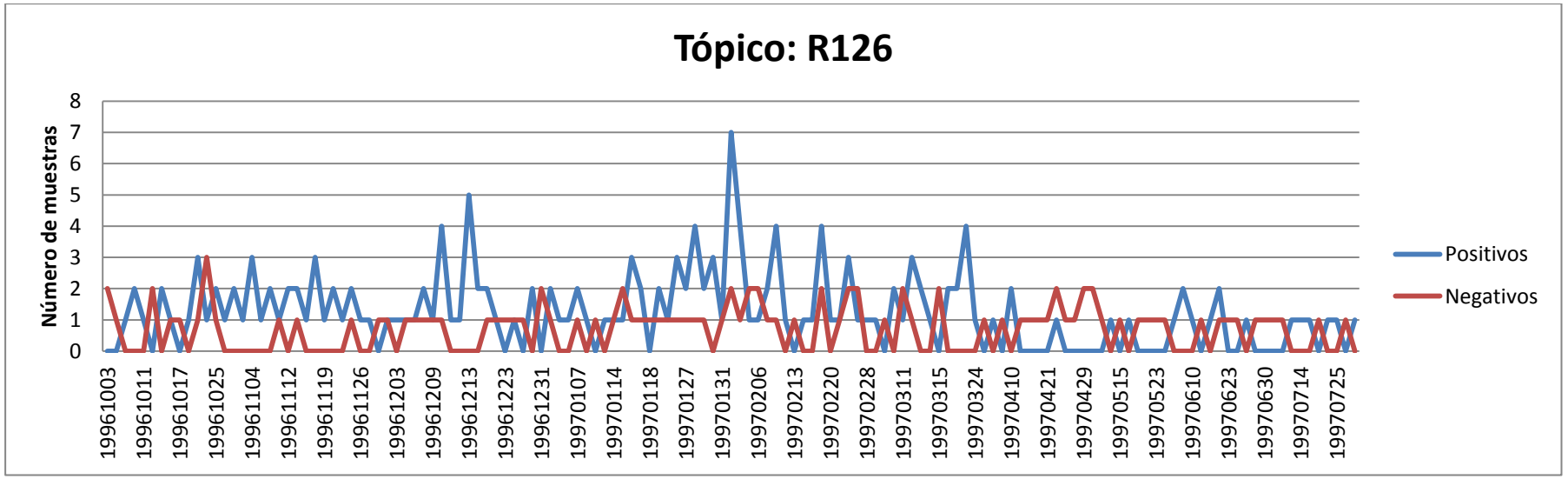
$$T11NU = \frac{T11U}{MaxU}$$

MinNU representa la mínima utilidad que un usuario toleraría durante la vida útil de un perfil. Su valor fue fijado a -0.5.

Figura 1: Comportamiento del flujo de datos de 4 tópicos de la colección empleada en la TREC 11




Tópico: R111
Número de muestras
0
1
2
3
4
5
6
7
8
19961001
19961010
19961018
19961027
19961103
19961112
19961119
19961126
19961206
19961213
19961220
19961230
19970109
19970116
19970123
19970130
19970207
19970214
19970226
19970305
19970312
19970318
19970324
19970331
19970407
19970413
19970418
19970425
19970502
19970515
19970523
19970529
19970610
19970619
19970627
19970708
19970715
19970722
19970729
19970808
19970819
Positivos
Negativos


Tópico: R126
Número de muestras
0
1
2
3
4
5
6
7
8
19961003
19961011
19961017
19961025
19961104
19961112
19961119
19961126
19961203
19961209
19961213
19961223
19961231
19970107
19970114
19970118
19970127
19970131
19970206
19970213
19970220
19970228
19970311
19970315
19970324
19970410
19970421
19970429
19970515
19970523
19970610
19970623
19970630
19970714
19970725
Positivos
Negativos

Resultados alcanzados por varios sistemas

Los resultados obtenidos en la TREC11 fueron modestos. Como parte de la competición se presentaron 14 corridas pertenecientes a 11 grupos de investigación diferente. El *baseline* en la competición fue establecido en 0.33. Este se corresponde con el valor de la medida T11SU, cuando no es recuperado ningún documento como relevante. Los resultados alcanzados de forma general por los diferentes sistemas fueron inferiores a 0.5, en los primeros 50 tópicos construidos por los asesores de la competición. El resultado en los otros 50 tópicos, los construidos por medio de la intersección entre tópicos originales de la colección RCV1 fueron aún peores, estando todos los sistemas por debajo del *baseline*. Debido a esto, la mayoría de los trabajos reportados han concentrado sus evaluaciones en los primeros 50 tópicos.

Aunque se ha seguido trabajando en la tarea luego de realizada la competición y se han reportados mejoras en cuanto a los resultados alcanzados en comparación con los reportados en la TREC 11, aún estos valores son bajos. Por poner algunos ejemplos, Zhang [76] reportó 0.52 con la medida T11SU al emplear Regresión Logística y combinarlo con el algoritmo de Rocchio. Piwowarski et. al. [48] en su trabajo emplean el clasificador de Rocchio y reportan 0.44 al emplear la medida T11F. Algarni et al. [2], reporta con la medida F1 resultados de 0.47 al emplear la minería de patrones frecuentes. Zhang et al. [74] [75] obtienen 0.504 con T11SU al emplear los metadatos presentes en los documentos. Gao et al. en [18] reportan un resultado 0.46 de F1. Aparte de los anteriores, se han reportado soluciones con algoritmos evolutivos [8] [46] [45] y mapas auto-organizados [15], aunque evaluados con otros datos y medidas.

Enfoques y Clasificadores

Los enfoques dados al Filtrado Adaptativo han seguido dos líneas de desarrollo fundamentales:

1- Afrontar la tarea como un problema de Recuperación de Información aunado a Ajuste de umbrales.
2- Categorización de Textos.

Entre los algoritmos y aproximaciones dadas encontramos el uso del algoritmo de Rocchio [3], la regresión logística [76], los algoritmos genéticos [45] [4], las redes neuronales [32], modelos bayesianos jerárquicos [73], entre otros.

La mayoría de las aproximaciones basadas en categorización de textos siguen un esquema binario. Sin embargo, este no tiene que ser necesariamente la única alternativa. En el manejo de flujos de información se ha empleado además la clasificación en una sola clase. La clasificación de una clase (*one class classification*, OCC) se enfoca en el desarrollo de clasificadores capaces de discernir la pertenencia o no de un objeto a una clase modelando solamente muestras de la clase de interés [31].

Decidir cuál modelo, binario o de una clase, es más conveniente para determinado entorno puede resultar bastante complejo. Por ejemplo, Bellinger et al. [6] llegan a la conclusión de que en ambientes desbalanceados es preferible el uso de OCC, sin embargo sus pruebas fueron realizadas con datos de una naturaleza muy diferente a los documentos textuales.

Ambos enfoques son afectados por el problema del desbalance entre las clases y para ambos se han reportados trabajos que intentan solventar esta situación. Incluso, algunos autores consideran el problema de OCC más difícil de resolver [31].

El empleo de las Máquinas de Vectores Soportes (SVM) ha sido muy estudiado en la tarea de OCC. Uno de los métodos más ampliamente empleado ha sido el método SVDD (Support Vector Data Description) [67].

Representación de los documentos

La inmensa mayoría de los sistemas reportados para la tarea de Filtrado Adaptativo emplean el tradicional modelo de Bolsa de Palabras. Los sistemas han estado más enfocados en el diseño de los perfiles de usuario y su actualización que en la propia representación de los documentos en sí.

Sin embargo, se han reportados algunos trabajos que exploran otras representaciones o extensiones al modelo de Bolsa de Palabras.

Por ejemplo, Zhang et al. [74] [75], considerando que muchos de los documentos presentes hoy en día poseen metadatos importantes que pueden ayudar en la clasificación de la información proponen considerar el valor de estos metadatos en forma de facetas, para ello emplean una representación en la cual utilizan el tradicional modelo vectorial para el texto y además mantienen el valor de los metadatos presentes en el documento.

Algarni et al. [2], motivados por las limitaciones del modelo de bolsa de palabras, propusieron el empleo de la minería de patrones para la representación de los documentos, específicamente PTM (Pattern Taxonomy Model) [71]. La idea tras el empleo de patrones radica en que estos son menos ambiguos y más discriminativos que los términos individuales, y por ello deben ofrecer mejores resultados que los términos individuales. Sin embargo, los patrones son costosos de obtener y su calidad puede ser muy pobre en las primeras etapas de la tarea de filtrado. Esta idea es nuevamente retomada por Gao et al. [18], los cuales proponen una representación que combina los patrones encontrados con el modelo LDA (Latent Dirichlet Allocation en inglés) [7].

Como ya hemos visto, la mayoría de los sistemas de Filtrado Adaptativo de Documentos existentes se han concentrado en identificar documentos relevantes empleando el tradicional modelo vectorial [77] [35] para representar los documentos. Este modelo presenta el inconveniente de asumir que existe una independencia estadística entre las diferentes palabras que componen un documento, lo cual es usualmente incorrecto [21]. Además, es incapaz de manejar los problemas de polisemia y sinonimia presentes en el idioma.

En la literatura hay reportados otros métodos para la representación de los documentos, como es el caso del LSI (Latent Semantic Indexing en inglés) [14], LDA (Latent Dirichlet Allocation en inglés) [7], PLSI (Probabilistic Latent Semantic Indexing) [25], Word2Vec [37], entre otros. Todos ellos tienen en común el hecho de no asumir que los términos que forman un documento son independientes entre sí.

Estos algoritmos son costosos computacionalmente, lo cual es una limitante fuerte para ser aplicados a problemas en los cuales se requiere de una actualización constante del espacio representación.

En la literatura se ha reportado otro método conocido como Indexado Aleatorio, el cual no asume independencia entre los términos y no requiere de cálculos computacionalmente tan costosos, por lo que pudiera ser aplicado a problemas de Filtrado de Información.

3. Indexado Aleatorio

El Indexado Aleatorio [57] fue introducido por Pentti Kanerva [30] y se basa en tres supuestos fundamentales:

- la hipótesis de la distribución, la cual puede ser expresada como: “palabras con un significado similar ocurren en contextos similares” [53] o también en esta otra forma: “palabras que ocurren en los mismos contextos tienden a tener significados similares” [47].
- el lema de Johnson – Lindenstrass [26], el cual garantiza que la proyección de un espacio vectorial de alta dimensionalidad en un subespacio menor es probable que preserve el orden del producto escalar [70].
- y en que hay muchas más direcciones casi ortogonales que direcciones realmente ortogonales en un espacio de alta dimensión, lo cual fue demostrado en [23].

Las ideas de Pentti Kanerva continuaron siendo desarrolladas por Magnus Sahlgren del Instituto Sueco de Ciencia de la Computación, el cual en [57] propone un método incremental para la construcción del Indexado Aleatorio como un proceso de dos pasos en la siguiente forma:

1. Primero, a cada contexto (por ejemplo: un documento o una palabra) en la colección se le asigna una representación única generada de forma aleatoria, denominada *Vector Índice*. Estos Vectores Índices son dispersos, de una elevada dimensionalidad (k), y están compuestos por un pequeño número (e) de 1 y -1, distribuidos aleatoriamente, con el resto de los elementos del vector en 0.
2. Luego, se generan *Vectores de Contexto* para cada una de las palabras. Para ello se recorre el texto y cada vez que aparece una palabra en el contexto se adiciona su vector índice al vector de contexto de la palabra.

El Indexado Aleatorio puede ser utilizado considerando diversos tipos de contexto, de ellos se detallan los más empleados.

3.1 Documentos como contextos

Pentti Kanerva 2000 [30] propone el Indexado Aleatorio para reducir el número de columnas en una matriz de frecuencia de ocurrencias de los términos en los documentos M_{pxq}. Para realizar esta reducción asigna a cada documento d_i un *vector índice* aleatorio único de dimensión k, $k \ll q$. Estos *vectores índices* contienen todas sus componentes en cero y a los cuales se les ha cambiado de forma aleatoria un pequeño número de componentes a +1 o -1. La matriz reducida resultante M'_{pxk} se obtiene inicializando la M’ con ceros y cada vez que el término t_j aparece en el documento d_i se adiciona el vector índice asociado a éste a la *j*-ésima fila de la matriz M’.

Los resultados obtenidos en este trabajo mostraron que esta reducción permite obtener resultados comparables a los obtenidos por LSA en la tarea de encontrar relaciones de sinonimia entre palabras.

La formulación del Indexado Aleatorio, dada por Kanerva [30] y Sahlgren [57], plantea adicionar el vector índice del contexto al vector de contexto de un término cada vez que este aparece en el contexto. En [19] se propone una modificación para emplear funciones de pesado de término durante la construcción de los vectores de contexto. Para ello proponen adicionar una sola vez el vector índice de un documento multiplicado por una función de pesado de término. Es decir, en esta forma el vector de contexto del término *t* se obtendría de la siguiente forma:

$$VC(t) = \sum_{\{d_i | t \in d_i\}} I(d_i) F(t, d_i) \quad (1)$$

donde $I(d_i)$ es el vector índice del documento d_i, y F la función de pesado de términos. Nótese que si se toma a F como la frecuencia del término t en el documento d_i se obtiene la definición original del Indexado Aleatorio.

En [58] se propone por primera vez emplear el Indexado Aleatorio como una forma de representación basada en concepto. Para ello los autores, una vez obtenidos los vectores de contexto, representan a los documentos como la suma de los vectores de contexto de los términos presentes en el mismo.

Este enfoque ha sido empleado recientemente en diferentes áreas dentro de la Minería de Textos y el Procesamiento del Lenguaje Natural, como es el caso de la comparación de similitudes entre oraciones [9] y la recuperación de información [42].

3.2 Términos como contextos

En vez de asignar un vector índice a cada documento, Magnus Sahlgren en el 2001 [55] asigna los vectores índices a los términos y define como contexto los términos que se encuentran en una ventana alrededor del término de interés. En esta propuesta, cada vez que el término x_i ocurre en el documento; los vectores índices de los términos presentes en la ventana son adicionados al vector de contexto v_i. Los resultados en esta ocasión en la tarea de detección de sinónimos superan a los obtenidos por Kanerva [30] en su estudio.

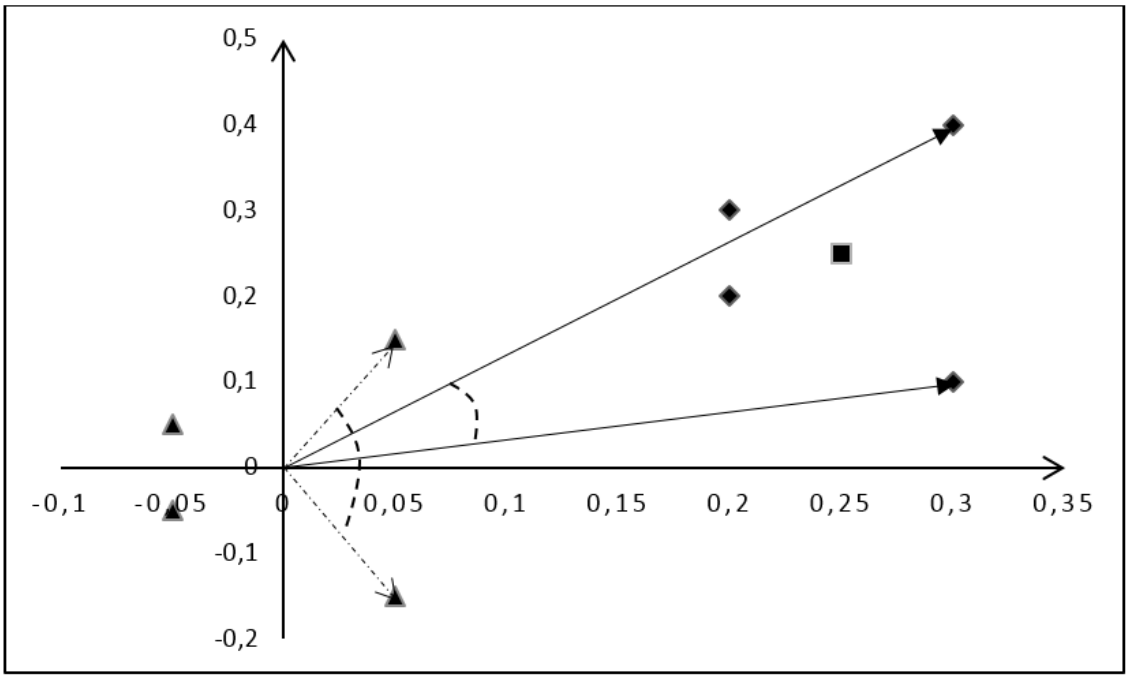


Figura 2: Efecto de la traslación de los vectores al restar el vector medio.

En la propuesta anterior todos los términos presentes en la ventana tienen igual importancia, sin tomar en consideración la distancia que media entre el término cuyo vector índice se adicionará al vector de contexto y el término de interés o la importancia del mismo. Con el objetivo de considerar esta distancia en [62] proponen ponderar por 2^{1-d} (d representa la distancia que media entre los términos dentro de la ventana) al vector índice antes de adicionarlo al vector de contexto. Esto permite que los términos más al centro de la ventana tengan una mayor importancia que aquellos que se encuentran más alejados. En [69] proponen otras funciones de pesado de términos que pueden ser empleadas para ponderar los vectores índices antes de ser adicionados a los vectores de contexto.

Entre las diversas aplicaciones que ha tenido el Indexado Aleatorio está la tarea de construcción de extractos como es el caso de los trabajos propuestos en [10] y [64] [65]. En estos trabajos los autores proponen para obtener la representación final de las oraciones, antes de sumar los vectores de contexto,

restar a estos vectores de contexto la media de todos los vectores de contexto siguiendo una idea similar a la propuesta por Higgins y Burstein en [24].

Higgins y Burstein plantean que si se toma un documento compuesto por n términos y construimos su representación como la media de los vectores de contexto que contiene, a medida que el número de términos del documento se incremente su representación tenderá a la media de los vectores de contexto:

$$\lim_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n} x_i = \bar{x} \quad (2)$$

donde x_i es el vector de contexto de *i*-ésimo término y $\bar{x}$ es el vector medio de los vectores de contexto. Según los autores si se toman dos documentos con estas características y se calcula la semejanza del coseno entre ellos este valor tenderá a 1. Para mitigar esta situación, en su trabajo proponen restar este vector de los vectores de contexto y con ello se obtendría:

$$\lim_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n} (x_i - \bar{x}) = \vec{0} \quad (3)$$

Esta transformación, provoca una traslación de los vectores con respecto al origen de coordenadas. En la figura 2 se muestra la situación descrita, en ella se han representado 4 puntos (representados por rombos) y su punto medio (representado por el cuadrado); además se han señalado dos de los vectores y el ángulo entre ellos. Igualmente se muestran los puntos una vez que se les ha restado el valor del vector medio (representados por triángulos) y los vectores entre los mismos puntos. Como puede apreciarse el ángulo entre los vectores aumenta y con ello disminuye el valor de la semejanza del coseno.

En la práctica es muy poco probable que un documento tenga un número de características que tienda al total de términos de la colección. Este tipo de documento los encontramos cuando combinamos varios en uno solo, por ejemplo para la construcción de algunos tipos de centroides; y estos, usualmente, no suelen ser comparados entre sí, sino con otros documentos.

Sellberg y Jönsson [63] proponen una variante ligeramente diferente de Indexado Aleatorio. Al igual que Sahlgren en 2001 asignan vectores índices para los términos pero no emplean ventanas de términos. A diferencia de los otros modelos vistos, no son obtenidos los vectores de contexto, sino que la representación final de los documentos es obtenida proyectando la matriz de frecuencia de ocurrencia *dxw* en la matriz de vectores índices *wxk*. Esta técnica es conocida como *Random Projection*.

En [11], los autores plantean que un aspecto a tomar en consideración en el Indexado Aleatorio, es que las posiciones aleatorias de los ´+1´ y ´-1´ pueden ser tales que dos vectores índices tengan sus respectivos ´+1´ y ´-1´ en las mismas coordenadas, pero dispuestos de forma tal que la suma entre ellos se anule. Según ellos, esto provoca que se generen vectores de contextos que no capturen correctamente la información disponible. Para evitar esta situación proponen restringir la posición de los ´+1´ para la primera mitad de los vectores y los ´-1´ para la segunda mitad. Sin embargo, esta agrupación es equivalente a tener dos veces la cantidad de elementos ´+1´, lo cual afecta la casi ortogonalidad en la cual se basa la teoría del Indexado Aleatorio. Este enfoque solamente se ha probado en la construcción automática de resúmenes.

Cataldo Musto [40] [43] propone una variante ligeramente diferente del Indexado Aleatorio. En su trabajo se asigna un vector índice a cada término. Una vez hecho esto los vectores de contexto son obtenidos mediante la suma de todos los vectores índices de los términos con los cuales coocurre. Luego la representación final de los documentos se obtiene sumando todos los vectores de contexto de los términos presentes en el documento.

En 2008, Sahlgren et. al. [60] mostraron que es posible codificar el orden de las palabras en los contextos mediante el empleo de permutaciones. Para ello los vectores de contextos en las ventanas son

actualizados en la siguiente forma: si tenemos la secuencia de términos $w_{-2}w_{-1}w_0w_1w_2$, el vector de contexto de w_0 se actualizaría con el vector obtenido a partir de la expresión: $P^{-2}(I(w_{-2}))+P^{-1}(I(w_{-1}))+P(I(w_1))+P^2(I(w_2))$. Donde P es una función de permutación aleatoria y P^{-1} su inversa. P^n significa que el vector es permutado n veces y $I(w_i)$ representa el vector índice asociado al término w_i. Una permutación sencilla que puede ser aplicada a los vectores es rotar los elementos 1 o más veces en una dirección, por ende su inversa sería la rotación el mismo número de veces en sentido contrario. Esta estrategia fue empleada en [13] para codificar relaciones entre tripletas de elementos en la forma entidad-relación-entidad y en [28] para reconocer nombres de entidades.

3.2.1 Indexado Aleatorio con Múltiples Sentidos

El indexado aleatorio no toma en consideración la polisemia de las palabras. Una palabra es polisémica si puede tener varios significados. Estas palabras suelen ocurrir en contextos diferentes, relacionados con los diferentes sentidos en los cuales son empleadas. En el Indexado Aleatorio todos estos contextos son combinados en un solo vector de contexto, lo cual puede afectar la calidad de los resultados obtenidos.

La primera propuesta para tratar de resolver esta deficiencia fue reportada por Moen, Marsi y Gambäck en [38]. La propuesta consiste en almacenar múltiples vectores por cada sentido diferente del término, estos vectores son conocidos como *vectores de sentidos*.

A diferencia de otros modelos que emplean ventanas de términos, durante la actualización de los vectores de contexto, en vez de adicionar directamente los vectores índices de los términos, primeramente calculan el *vector de la ventana* (v_{win}), el cual no es más que la suma de los vectores índices de los términos que se encuentran en la ventana. Seguidamente calculan la semejanza entre este vector y cada uno de los vectores de sentidos almacenados para el término de interés. Cada valor de semejanza es comparado contra un umbral predefinido y se analizan las siguientes opciones:

- Si ninguna de las semejanzas supera el umbral, el vector de la ventana se convierte en un nuevo vector de sentido.
- Si exactamente una semejanza supera el umbral, el vector de la ventana es adicionado al vector de sentido correspondiente.
- Si varios vectores superan el umbral de semejanza, entonces todos ellos y el vector de la ventana son combinados en un solo vector de sentido.

En este método los autores emplean un procedimiento no supervisado que consiste en los siguientes pasos:

1. Para cada término t_i del documento.
   a. Asignar un vector índice a cada término t_i.
   b. Para cada término construir el vector de la ventana (v_{win}).
   c. Calcular la semejanza de v_{win} con cada uno de los vectores de sentidos ($vs_j(t_i)$).
   d. Calcular el conjunto $S = \{vs_j(t_i) | sem(vs_j(t_i), v_{win}) \geq \gamma\}$.
   e. Actualizar la información almacenada en la siguiente forma:
      i. Si |S| = 1, $vs_j(t_i)$ += v_{win}.
      ii. Si |S| > 1, Mezclar todos los elementos de S+{ v_{win} } en un solo vector mediante la suma de vectores.
      iii. Si |S| = 0, Crear un nuevo vector de sentido para t_i con v_{win}.

Donde γ representa un umbral preestablecido para diferenciar entre sentidos y *sem* es la semejanza del coseno.

En su trabajo los autores proponen varias formas para medir la semejanza entre dos términos t_i, t_j representados en esta forma, entre ellas están:

- Centroide: para cada término calcular el vector suma de todos sus sentidos. Calcular la semejanza coseno de los vectores obtenidos.
- Promedio: Obtener el vector promedio de cada uno de los términos y luego calcular la semejanza entre ellos.

Esta representación la emplearon los autores para evaluar la similitud entre oraciones. Esta similitud la calcularon como el promedio de la mayor semejanza entre cada par de términos de las dos oraciones. Entre las estrategias empleadas se encuentran:

- Centroide: En cada oración, el término se representa por medio de la suma de todos sus vectores de sentido, luego de eliminar aquellos vectores asociados a sentidos pocos frecuentes.
- Término En Contexto: En este caso consideran una ventana de 5 + 5 alrededor del término y calculan el vector de la ventana v_{win}. El valor de la ventana es fijado a 5 + 5 por los autores para esta estrategia, sin dar una justificación del porqué de su elección. Finalmente el término es representado por el vector de sentido más semejante a v_{win}.

Los resultados obtenidos por los autores no mostraron diferencias significativas con el empleo del Indexado Aleatorio tradicional.

3.3 Indexado Aleatorio Reflexivo

El Indexado Aleatorio captura el significado semántico de los términos basado en su combinación con aquellos con los cuales suele ocurrir. Sin embargo, Cohen et al. [12] llegan a la conclusión que el Indexado Aleatorio presenta deficiencias a la hora de extraer relaciones indirectas entre las palabras. Para dar solución a esta situación realizan una extensión del modelo consistente en asignar vectores índices a los términos de los documentos, luego se obtiene la representación de los documentos como la suma lineal (posiblemente pesada) de los vectores índices de los términos que contiene. Luego estos vectores de documentos son empleados para construir los vectores de contexto de los términos. Este proceso se puede repetir varias veces. A esta extensión los autores le dan el nombre de Indexado Aleatorio Reflexivo [12] [68] basado en Términos (TRRI). En el mismo trabajo los autores presentan además el Indexado Aleatorio Reflexivo basado en Documentos (DRRI), el cual no es más que asignar los vectores índices a los documentos y seguir el proceso iterativo de entrenamiento.

Si bien es cierto que en el Indexado Aleatorio Reflexivo se pueden realizar varias iteraciones de entrenamiento, los resultados obtenidos por los autores, para la tarea de detección de relaciones indirectas, muestra que efectivamente el proceso aumenta la calidad en las primeras dos iteraciones. Luego de la tercera iteración se obtiene una disminución considerable en términos de calidad.

3.4 Aplicaciones del Indexado Aleatorio

Las primeras aplicaciones dadas al Indexado Aleatorio estuvieron encaminadas a determinar la similitud entre términos diferentes y fue aplicado a varias tareas de la minería de textos como es el caso de la construcción de tesauros bilingües para la expansión de consultas entre varios idiomas [59] [61] [56] y la construcción de lexicones bilingües de términos de forma automática [54].

Trabajos más recientes han aplicado el Indexado Aleatorio en tareas como la expansión de consultas [62], predicción de fallos en aplicaciones mediante el análisis de trazas [16], la búsqueda de respuestas [39], la construcción de resúmenes [22], la detección de eventos en blogs [29], el reconocimiento de términos de tecnologías en un corpus de publicaciones científicas [72] y la recuperación de información [42].

3.5 Trabajos Relacionados

Aunque se han presentado trabajos relacionados con la temática del Aprendizaje en Línea y el Filtrado de Información, solo uno ha empleado el Indexado Aleatorio como parte de su modelación [40] [41] [44]. En este trabajo representan tanto el perfil del usuario como los documentos por medio del Indexado Aleatorio. Luego los elementos recuperados son escogidos mediante la comparación basada en el cálculo de la similitud entre los vectores obtenidos. El perfil de los usuarios está compuesto por dos vectores, uno en el cual representan los elementos etiquetados como relevantes y otro para aquellos elementos que el usuario indica que no son de su interés. La representación final consiste en una proyección del vector de los elementos positivos sobre el vector de los elementos negativos con el objetivo de encontrar aquellos elementos que aparecen tanto como sea posible entre los elementos positivos y tan poco como se pueda entre los elementos negativos.

En [42], los autores plantean una propuesta para el filtrado de información inter-lengua. La idea del trabajo consiste en construir un modelo semántico para cada idioma (dos en su caso). Después se construye el perfil del usuario en uno de los idiomas y luego los nuevos documentos, en el mismo idioma u otro, son representados y comparados con respecto al perfil construido.

Por otro lado, en [36] evalúan la utilidad del Indexado Aleatorio en la tarea de la recuperación de patentes. Para esta tarea los autores exploran varias de las variantes de Indexado Aleatorio, así como sus parámetros. En todos los casos, los resultados obtenidos no superan a los alcanzados por la herramienta Lucene. Los autores hacen además un análisis de las posibles causas de los malos resultados obtenidos, llegando a la conclusión de que el Indexado Aleatorio hasta el momento presenta dificultades cuando es empleado en entornos donde la distribución de los términos es muy sesgada, como es el caso de la recuperación de patentes.

En [49], el Indexado Aleatorio es evaluado en la tarea de recuperación de información general. En esta propuesta los documentos son representados empleando el modelo de las ventanas de términos. Los documentos y las consultas son representados mediante la suma de los vectores de contextos pesados por la frecuencia del término en los documento. Luego se calcula la semejanza entre la consulta y cada uno de los documentos del corpus. Finalmente, los n primeros documentos más similares son recuperados. Los resultados obtenidos son ligeramente superiores a los obtenidos por el modelo de espacio vectorial tradicional.

3.6 Algunas consideraciones sobre su posible aplicación al Filtrado de Información

A diferencia de métodos como LSI, el Indexado Aleatorio no requiere de tener formada toda la matriz de coocurrencias para obtener la representación de los documentos. Por el contrario, tras acumular algunos contextos, es posible ir teniendo resultados parciales de los vectores de contexto que pueden emplearse en la representación de los documentos. Este es un modelo incremental en el cual se puede ir actualizando el modelo a medida que arriban nuevos documentos a ser procesados.

Un aspecto a considerar es el hecho de que el modelo está pensado para obtener una representación conceptual de los términos, basado en la coocurrencia. Para ello no se requiere de obtener luego una representación de los documentos, aunque puede obtenerse en aquellas tareas que se requieran.

En un entorno de Filtrado Adaptativo, un algoritmo se enfrenta tanto a la llegada de nueva información y posiblemente nuevos términos. Esto conlleva a que crezca la dimensionalidad del problema conforme transcurre el tiempo. El Indexado Aleatorio permite mantener acotado el número de dimensiones de nuestro problema, por lo que una de las dos fuentes de aumento de la dimensionalidad está controlada. Por otro lado, dado que no se requiere de preservar los documentos para poder adquirir nuevo

conocimiento y con ello modelar mejor el espacio, podemos descartarlos una vez se ha terminado el indexado, con lo cual mantendríamos controlada la otra fuente de aumento de la dimensionalidad.

4. Propuesta

La presente sección muestra la motivación, el problema de investigación, las preguntas de investigación, objetivos y las principales contribuciones relacionadas con esta propuesta. Además, se describe la metodología a utilizar, el plan de publicaciones y el cronograma de trabajo a desarrollar.

4.1 Motivación

La mayoría de los trabajos reportados en la literatura hacen uso del tradicional modelo vectorial para la representación de los documentos, pese a que este modelo no logra capturar las relaciones semánticas que se establecen entre los términos de un documento [5].

Por otro lado el Filtrado Adaptativo de Documentos pudiera verse favorecidos por el empleo de métodos en los cuales la suposición de independencia entre los términos no esté presente y con ello lograr una mayor eficacia en su desempeño.

El Indexado Aleatorio no asume la independencia entre los términos ni es tan costoso como las técnicas antes mencionadas. El mismo ha sido empleado con éxito en varias tareas del Procesamiento del Lenguaje Natural y la Minería de Textos.

Es por ello que consideramos pertinente desarrollar una representación para los documentos que sea compacta (en el sentido empleado en el Indexado Aleatorio de no representar en memoria todos los términos) y que a su vez no asuma independencia entre los términos presente en el documento.

Si bien se han reportados varios trabajos relacionados con el Indexado Aleatorio, aun no se ha realizado una comparación entre estas variantes, ni con otras formas de representación de los documentos, en tareas afines al Filtrado de Información que permita evaluar cuál, o cuáles, son las que mejor se ajustan a las necesidades propias de la tarea.

La tarea de Filtrado Adaptativo impone grandes retos, entre ellos se encuentra la poca homogeneidad en la distribución de las muestras, el fuerte desbalance presente entre lo que es de interés y lo que no, así como la escasez de información para la correcta modelación del perfil de los usuarios. Esta última situación puede mantenerse incluso durante todo el funcionamiento del sistema cuando el interés del usuario es muy específico y se tienen pocos documentos que satisfagan su necesidad de información. En este contexto, debe evaluarse el comportamiento del Indexado Aleatorio ante estas situaciones y buscar alternativas que permitan resolver estas limitaciones durante su construcción.

4.2 Problema de Investigación

En el filtrado de información aún se requieren de investigaciones que permitan dar solución a la escasez de información durante las primeras etapas del filtrado, el desbalance presente entre lo que es de interés y lo que no para el usuario y el manejo del incremento del espacio de representación producto de la aparición de nuevos términos. En la presente investigación buscamos contribuir a su solución mediante el desarrollo de un método en el cual se logre mayor eficacia ante el fenómeno del Inicio en Frío y se logre una dimensionalidad de la representación baja, a pesar del incremento del volumen de documentos.

4.3 Preguntas de Investigación

1. ¿Cómo aplicar el Indexado Aleatorio en la tarea de Filtrado de Información?

2. ¿Qué recursos externos pueden permitir resolver el problema de la escasez de información para construir el Indexado Aleatorio en las etapas iniciales del filtrado?
3. ¿Qué efecto tiene en la solución desarrollada el desbalance entre las clases Relevantes y No Relevantes?

4.4 Hipótesis

En el contexto del Filtrado Adaptativo de Documentos, es posible obtener una representación del perfil del usuario basada en el Indexado Aleatorio que logre una eficacia aceptable durante las etapas iniciales del proceso de filtrado y logre mantener la eficiencia a medida que se adicionan nuevos documentos a la definición del perfil del usuario, manteniendo la eficacia en términos de la calidad del clasificador.

4.5 Objetivo General

Diseñar un método para la tarea del Filtrado Adaptativo de Documentos, en el cual la representación del perfil del usuario no suponga independencia entre los términos ni se vea afectada su eficiencia, en cuanto a costo computacional, por el incremento del volumen de documentos que lo definen y obtenga resultados con una eficacia equivalente o mejor a la reportada en la literatura.

4.6 Objetivos específicos

- Estudio comparativo de las diversas variantes del Indexado Aleatorio en el proceso de la clasificación binaria, para seleccionar de estas la de mejor eficacia para la tarea.
- Elaborar una representación de los perfiles de usuario para el Filtrado Adaptativo de Documentos basada en la variante anteriormente seleccionada.
- Diseñar y evaluar una estrategia que permita disminuir el impacto del Inicio Frío en la construcción de la representación y con ello obtener mayor eficacia en las etapas iniciales del proceso de filtrado.
- Estudiar el impacto que tiene el desbalance entre las clases Relevantes y No Relevantes en una representación basada en el Indexado Aleatorio y diseñar una estrategia que permita tratar dicho desbalance con una eficacia equivalente o mejor a lo reportado en la literatura.

4.7 Contribuciones

Las principales contribuciones esperadas al término de esta investigación doctoral son las siguientes:

- Estudio comparativo de las diversas variantes del Indexado Aleatorio.
- Algoritmo de Filtrado Adaptativo de Documentos empleando una representación para los perfiles y documentos que no suponga independencia entre los términos.
- Estrategia para combatir el Inicio en Frío (escasez de información) en el funcionamiento del algoritmo de Filtrado y en la construcción de la representación.
- Algoritmo para disminuir el impacto del desbalance entre las clases en el funcionamiento del algoritmo de Filtrado y en la construcción de la representación.
- Método que incorpore los elementos previos que mejore la eficacia de representaciones anteriores en la tarea.

4.8 Metodología

1. Análisis de las potencialidades de las diferentes variantes del Indexado Aleatorio.
   a. Comparación de las diversas variantes del Indexado Aleatorio en la tarea de Clasificación binaria de textos.

b. Comparación de las diversas variantes del Indexado Aleatorio con otras alternativas al modelo vectorial.
c. Estudiar el impacto de los parámetros en los resultados alcanzados por el Indexado Aleatorio.
d. Estudiar el comportamiento del Indexado Aleatorio en entornos de clasificación binaria donde las clases se encuentran desbalanceados.
    i. Estudiar si los parámetros del Indexado Aleatorio pueden influir favorablemente en el comportamiento.
e. Estudiar el comportamiento del Indexado Aleatorio ante la presencia de un volumen de información variable para su construcción.
    i. Estudiar si existe una relación entre el tamaño de los vectores y la cantidad total de términos del espacio que permita mantener la calidad del Indexado Aleatorio.
f. Evaluar el impacto que tiene el empleo de un modelo de Indexado Aleatorio pre-construido en los problemas anteriores.

2. Proponer algoritmo de Filtrado Adaptativo de Documentos.
    a. Evaluar el comportamiento de algoritmos de Filtrado Adaptativo ante el uso o no del Indexado Aleatorio en la construcción de los perfiles de usuario y la representación de los documentos.
    b. Proponer un algoritmo de Filtrado Adaptativo a partir del estudio realizado en la etapa 1 de la investigación.
    c. Evaluar una estrategia para la selección de los parámetros de la representación.

3. Estrategia para encarar el Inicio en Frío.
    a. Evaluar el empleo de un modelo pre-construido para mejorar la calidad de la representación.
    b. Evaluar el empleo de la clasificación semisupervisada, con el fin de incluir nuevas muestras al clasificador y con ello mejorar la modelación del usuario.
    c. Estudiar la integración en el modelo de información proveniente de otros usuarios con intereses similares para mejorar la representación inicial.

4. Algoritmo para disminuir el impacto del desbalance entre las clases Relevantes y No Relevantes.
    a. Estudiar el empleo de la clasificación semisupervisada para disminuir el impacto del desbalance entre las clases.
    b. Diseñar una estrategia que permita hacer énfasis en la clase de interés durante la retroalimentación de los usuarios.

4.9 Diseño experimental

El diseño experimental de la presente propuesta doctoral está compuesto por 5 experimentos principales.

1- Comparativa de las diferentes variantes del Indexado Aleatorio en la tarea de la categorización binaria de textos.

Este experimento tiene por objetivo comparar las diferentes variantes del Indexado Aleatorio entre ellas, así como otras técnicas reportadas en la literatura como alternativas al modelo de Bolsa de Palabras en la tarea de categorización binaria de textos. Se seleccionó la categorización binaria de textos como marco común para evaluar las distintas formas de representación de los documentos por estar acorde con la tarea de Filtrado Adaptativo, dado que para cada documento en el Filtrado Adaptativo se debe tomar la decisión binaria de aceptar o no un determinado documento para un perfil en particular.

En el experimento se emplean las variantes fundamentales del Indexado Aleatorio descritas en la sección 3 y se comparan además con las técnicas LDA, LSI y Word2Vec.

Este experimento se corresponde con el primer objetivo específico y la primera etapa de la metodología propuesta.

Hipótesis científica: Se pueden alcanzar resultados aceptables con el empleo del Indexado Aleatorio en comparación con los alcanzados con otras técnicas de representación de los documentos en la categorización binaria de textos.

Hipótesis experimental: El Indexado Aleatorio permite obtener resultados comparables, o superiores a los obtenidos por LDA, LSI o Word2Vec.

Diseño del experimento:

- Datos: Subcolección de 8 clases pertenecientes a colección Reuter-21578[1].
- Clasificador: Clasificador basado en centroides.
- Validación mediante la técnica de validación cruzada con 5 particiones tomando 4 para entrenamiento y para prueba.
- Medida de Evaluación: Para la evaluación se seleccionó la medida F_1 Macro-promediada.

2- Análisis del comportamiento del Indexado Aleatorio ante diversos escenarios para su construcción.

Este experimento tiene por objetivo estudiar la capacidad del Indexado Aleatorio de preservar las distancias entre las muestras al ser construido bajo diversas configuraciones en la disponibilidad de los datos.

En el experimento se seguirá la metodología propuesta por Zadeh [50] y está relacionado con el primer objetivo específico de la presente propuesta y la primera etapa de la metodología propuesta.

Hipótesis científica: Se puede mitigar el efecto de la escasez de información y el desbalance entre las clases durante la construcción del Indexado Aleatorio si es empleado un modelo pre-construido a partir de una fuente de datos externa.

Hipótesis experimental: Si tomamos un repositorio de documentos, como Wikipedia, y construimos el Indexado Aleatorio a partir de este, al emplearlo podemos mitigar el efecto que tiene la escasez de datos de entrenamiento o el desbalance entre las clases.

Diseño del experimento:

- Datos: Colecciones Reuter-21578 y RCV1.
- Validación: Correlación entre ordenes de Spearman.
- Recurso externo: Wikipedia en idioma inglés.

3- Evaluación de la calidad del Algoritmo de Filtrado Propuesto.

[1] http://kdd.ics.uci.edu/databases/reuters21578/README.txt

En este experimento se seguirán las especificaciones de las competiciones TREC para ser comparable los resultados con los reportados en el estado del arte.

Este experimento se relaciona con el segundo objetivo específico y con la segunda etapa de la investigación.

Hipótesis científica: El empleo de una representación que no asuma independencia entre los términos permite mejorar los resultados alcanzados en la tarea del Filtrado Adaptativo.

Hipótesis experimental: El empleo de una representación basada en el Indexado Aleatorio permite mejorar los resultados alcanzados en la tarea del Filtrado Adaptativo.

Diseño del experimento:

- Datos: Colecciones Reuter-21578 y RCV1.
- Medida de Evaluación: T11SU.

4- Evaluación de la estrategia para disminuir el impacto del Inicio en Frío.

Este experimento tiene por objetivo evaluar la estrategia para disminuir el Inicio en Frío. En el mismo se seguirán las especificaciones de las competiciones TREC para que los resultados alcanzados sean comparables con los reportados en el estado del arte. En el análisis se hará énfasis en la calidad que se obtiene con la estrategia diseñada en el momento de construir el perfil del usuario.

Este experimento se relaciona con el tercer objetivo específico y con la tercera etapa de la investigación.

Hipótesis científica: El empleo de recursos externos puede disminuir la incidencia del Inicio en Frío en la tarea del Filtrado Adaptativo.

Hipótesis experimental: El empleo de recursos externos (como pueden ser modelos pre-construidos, documentos externos o información proveniente de otros usuarios) puede disminuir la incidencia del Inicio en Frío en la tarea del Filtrado Adaptativo.

Diseño del experimento:

- Datos: Colecciones Reuter-21578 y RCV1.
- Medida de Evaluación: T11SU.

5- Evaluación de la estrategia propuesta para combatir el desbalance entre las clases.

Este experimento tiene por objetivo evaluar la estrategia para combatir el desbalance entre las clases. En el mismo se seguirán las especificaciones de las competiciones TREC para que los resultados obtenidos sean comparables con los reportados en el estado del arte.

Este experimento se relaciona con el cuarto objetivo específico y con la cuarta etapa de la investigación.

Hipótesis científica: Por medio del empleo de muestras semisupervisadas es posible combatir el problema del desbalance en la tarea de Filtrado Adaptativo.

Hipótesis experimental: El empleo de muestras no etiquetadas puede disminuir el impacto del desbalance en la tarea del Filtrado Adaptativo.

Diseño del experimento:

- Datos: Colecciones Reuter-21578 y RCV1.
- Medida de Evaluación: T11SU.

5. Resultados preliminares

En esta sección se presenta de forma resumida los resultados preliminares obtenidos hasta el momento en esta propuesta doctoral.

5.1 Comparativa de las versiones de Indexado Aleatorio

El objetivo de este experimento es comparar el desempeño de las variantes de uso del Indexado Aleatorio en un marco común cercano a la tarea de interés. Para el mismo seleccionamos la colección Reuter-21578. De esta colección se han creado varios subconjuntos, de ellos los más conocidos son:

- Reu10: Contiene las diez categorías con un mayor número de muestras.
- Reu90: Contiene las 90 categorías para las cuales se cuenta con al menos un documento de muestra y uno de prueba.
- Reu115: Contiene el conjunto de las 115 categorías que al menos tienen un documento de muestra.

En particular en este experimento trabajaremos con el primero de estos subconjuntos. En la tabla 2 se muestra el número de muestras presente en cada una de las clases que componen este conjunto.

Clase	Número de Muestras
earn	3753
acq	2131
wheat	264
money-fx	600
corn	206
trade	449
grain	527
interest	389
crude	510
ship	276

Tabla 2: Número de documentos por clase

Con el fin de evitar el problema del desbalance entre las clases, en este experimento ignoraremos las clases earn y acq, considerando únicamente las 8 clases restantes. Nótese que estas dos clases juntas tienen casi el 65% de las muestras disponibles.

Durante el preprocesamiento, se eliminaron las palabras vacías (stop-words) y se empleó un proceso de lematización a los términos. Los documentos fueron pesados empleando TF-IDF.

Para evaluar la calidad empleamos un proceso de validación cruzada con 5 particiones.

En todos los casos la representación de los documentos fue construida empleando todas las muestras del conjunto de entrenamiento.

Para cada clase, el perfil del usuario fue representado por medio de dos vectores. Uno de ellos para representar los documentos de interés del usuario y el otro para representar aquellos que no son de su interés. Cada uno de estos vectores fue construido adicionando todos los vectores de los documentos que pertenecen, o no, a la clase en el conjunto de entrenamiento.

Durante el proceso de clasificación, un documento es clasificado como Relevante para un perfil si su semejanza con el vector de los documentos relevantes supera a la semejanza obtenido con respecto al vector de los no relevantes.

Para la evaluación empleamos las tradicionales medidas *precisión* (proporción de documentos clasificados como relevantes que en realidad son relevantes) y *relevancia* (proporción de documentos relevantes clasificados como tal). Estas medidas suelen ser combinadas en la popular medida F_1 mediante la expresión:

$$F_1 = \frac{2 * precisión * relevancia}{precisión + relevancia} \quad (5)$$

Dado que la medida F_1 se calcula por separado para cada una de las clases, consideramos como medida global el promedio de todos los valores de F_1, comúnmente conocido como Macro-F_1.

Variante de Indexado Aleatorio	Promedio de Macro-F_1
RI	0.762
RI-MV	0.762
wRI	0.696
wRI-MV	0.696
RRI	0.706
RRI-MV	0.706
TRI	0.693
TRI-MV	0.692
LSI	0.825
LDA	0.680
Word2Vec	0.644

Tabla 3: Promedio de los Macro-F_1 para las distintas variantes de Indexado Aleatorio

En la Tabla 3 se muestra el promedio de los valores de Macro-F_1 obtenidos. En la misma RI representa el Indexado Aleatorio cuando se considera los documentos como contextos; de la misma manera wRI cuando se consideran los términos como contextos y se emplea una ventana alrededor del término de interés y TRI cuando no es empleada ninguna ventana. RRI se refiere al uso del Indexado Aleatorio Reflexivo. Los modelos seguidos del sufijo "-MV" representan el resultado obtenido cuando la media de los vectores de contextos es restada a los vectores de contexto antes de obtener la representación de los documentos.

Para el Indexado Aleatorio consideramos vectores de longitud 5000, con 5 posiciones seleccionadas como +1 y 5 posiciones como -1 cuando se generaron los vectores índices. Para la variante wRI se consideró una ventana de tamaño 2 alrededor del término. Para el Indexado Aleatorio Reflexivo solo se llevó a cabo una iteración.

En la misma, además, se reflejan los valores obtenidos por las técnicas LSI, LDA y Word2Vec.

A partir del análisis de los resultados mostrados en la table podemos notar varios comportamientos. Primero, al comparar las diversas formas del Indexado Aleatorio, los mejores resultados son obtenidos con la variante que considera a los documentos como contextos. En este caso, esta variante es mejor que el resto de las otras variantes del Indexado Aleatorio en aproximadamente 8% - 9 %.

Otro elemento relevante es que los resultados obtenidos con el Indexado Aleatorio Reflexivo son superior a los obtenidos con el Indexado Aleatorio cuando son empleados los términos como contextos. El Indexado Aleatorio Reflexivo intenta capturar las relaciones indirectas que se establecen entre los términos, y su utilidad no es la misma en otras tareas.

Podemos notar que, en la mayoría de los casos, cuando le sustraemos vector medio de los contextos, antes de obtener la representación final de los documentos, no se obtienen ganancias consistentes. Por ende, no encontramos una razón que justifique esta operación adicional sin una mejora real en la calidad.

Por otra parte, el Indexado Aleatorio obtuvo resultados similares a los obtenidos por las otras técnicas más ampliamente empleadas en tareas de Minería de Textos, tales como LSI y LDA. Inclusive, en nuestros experimentos, el Indexado Aleatorio produjo resultados comparables al LSI, y superiores a los obtenidos por las técnicas LDA y Word2Vec.

Por último, la principal ventaja del Indexado Aleatorio es que solamente consideramos en su construcción vectores de 5000 elementos. Este aspecto toma particular importancia cuando nuestro objetivo son los ambientes en línea, donde cada nuevo documento puede contener nuevos términos. Con el Indexado Aleatorio, el problema de la aparición frecuente de nuevos términos no afecta la eficiencia debido a que los documentos siempre son representados por vectores de una dimensión fija. Cuando trabajamos en ambientes en línea, donde el conjunto de entrenamiento crece frecuentemente, el Indexado Aleatorio tiene una ventaja adicional sobre otras técnicas como LSI. Específicamente, esta ventaja consiste en el hecho de que con el Indexado Aleatorio no se requiere de almacenar los documentos anteriores para poder adicionar nueva información al modelo, a diferencia de LSI, LDA o Word2Vec. Inclusive, el Indexado Aleatorio es menos costoso, dado que su procesamiento involucra únicamente operaciones simples sobre vectores, como es la adición, mientras que otras técnicas emplean la Descomposición en Vectores Singulares, u otro método de optimización.

5.2 Capacidades del Indexado Aleatorio

El siguiente resultado preliminar está relacionado con el experimento 2.

Para el mismo se empleó la subcolección de las 10 clases más representadas en la colección Reuter-21578. Como parte del experimento, al igual que en los resultados preliminares mostrados anteriormente, eliminamos las dos clases mayoritarias presentes en reu10.

Estos resultados preliminares muestran la capacidad del Indexado Aleatorio para preservar la semejanza del coseno entre el espacio original y el espacio reducido construido. Para la evaluación se siguió una metodología similar a la propuesta por Qasemi Zadeh en [50].

Para la experimentación se tomó de cada una de las clases presente en el conjunto de datos un total de 11 documentos pertenecientes a la clase y 10 documentos no pertenecientes a ella. Luego se calculó la semejanza entre cada documento perteneciente a la clase con los otros 10, así como con los 10 no pertenecientes a la clase. Los documentos fueron ordenados de acuerdo al valor de semejanza obtenido.

Luego se construyó el modelo del Indexado Aleatorio a partir de todo el conjunto de datos presente en la colección variando la dimensión de los vectores y la cantidad de unos y menos unos presentes en los

vectores índices. Una vez obtenido el modelo se representaron los documentos y se procedió a calcular las semejanzas como fue descrito anteriormente.

Para determinar la capacidad de preservar las semejanzas del Indexado Aleatorio se procedió a determinar el coeficiente de correlación de Spearman entre los elementos ordenados por sus semejanzas al emplear o no el Indexado Aleatorio.

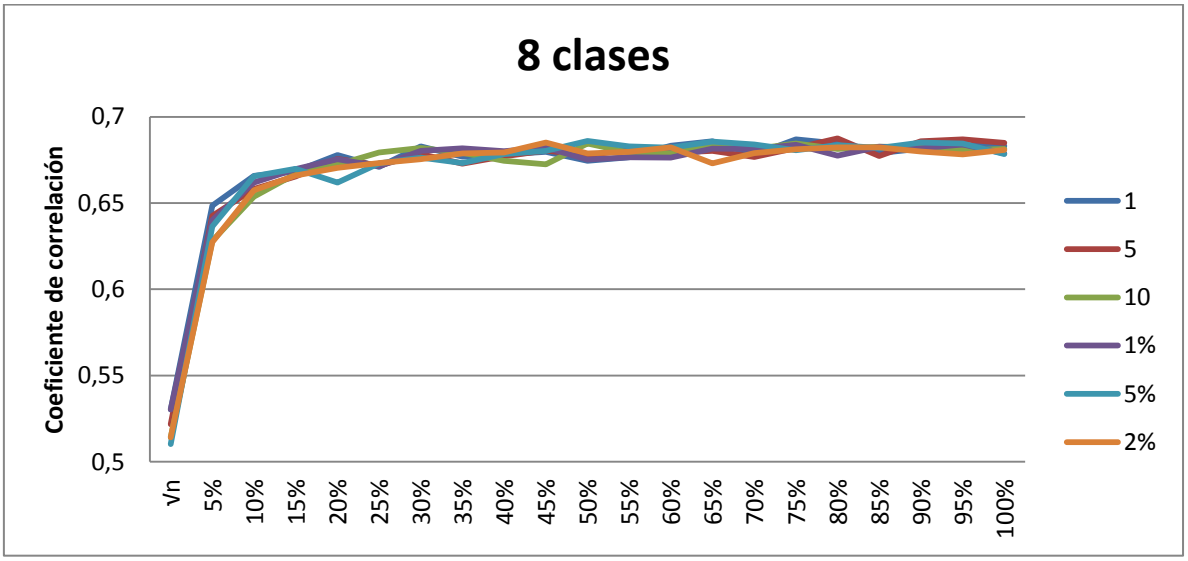
Para disminuir el impacto del sesgo en la selección todo el proceso fue repetido 5 veces.

Con este experimento buscamos estudiar cuánto podemos reducir el tamaño de los vectores de representación en función de la cantidad de términos diferentes presentes en el espacio de representación de los documentos. Si una vez concluido el experimento notamos que el Indexado Aleatorio es muy sensible a la variabilidad de los parámetros, entonces eso dificulta su posible empleo en la tarea que nos ocupa. En caso contrario, si el Indexado Aleatorio es poco sensible a los valores de sus parámetros, o al menos a partir de un determinado valor, entonces eso es buen indicador para su empleo en nuestra tarea principal.

Los parámetros del Indexado Aleatorio fueron variados en la siguiente forma:

Sea *n* la cantidad de términos total presentes en el espacio de representación. La dimensión de los vectores se comenzó en $\sqrt{n}$, luego el 5% de **n**, así se fue aumentando el valor de 5 en 5 hasta llegar al 100% de *n*. Una vez establecida la dimensión (k) de los vectores. La cantidad de 1s y -1s tomó los valores: 1, 5 y 10; así como el 1%, 2% y 5% de k.

Los resultados alcanzados se muestran en las gráficas siguientes.



Del análisis de las gráficas anteriores podemos notar que los resultados alcanzados son muy estables aun cuando se emplean tamaños de vectores de tan solo un 5% o 10% del total de términos de la colección. Además cuando adicionamos más información se logran valores más altos en cuanto a la correlación. Por otro lado, puede notarse que la cantidad de unos presentes en los vectores índices tiene poca incidencia, aunque cuando son empleadas las 10 clases se aprecia que con tamaño de vectores muy pequeños, con valores de más de un 2% de posiciones en 1 o -1 se puede ver afectada la calidad.

Una vez analizados los resultados obtenidos en este segundo experimento, podemos llegar a la conclusión de que los resultados que se obtienen con el Indexado Aleatorio tienen un comportamiento estable a partir de determinado valor de los parámetros. Esto facilite en cierta medida el volumen de experimentos que es preciso realizar para determinar el valor más adecuado de los parámetros.

Referencias

[1] M-Dyaa Albakour, Craig Macdonald, and Iadh Ounis. On sparsity and drift for effective real-time filtering in microblogs. In *CIKM '13: Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pages 419–428, New York, USA, October 2013. ACM.

[2] Abdulmohsen Algarni, Yuefeng Li, Sheng-Tang Wu, and Yue Xu. Text mining in negative relevance feedback. *Web Intelli. and Agent Sys.*, 10(2):151–163, April 2012.

[3] James Allan. Incremental relevance feedback for information filtering. In *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 270–278. ACM, 1996.

[4] Nurulhuda Firdaus Mohd Azmi, Fiona Polack, and Jon Timmis. Immune inspired adaptive information filtering: Focusing on profile adaptation. In *Bio-Inspired Models of Networks, Information, and Computing Systems*, pages 242–247. Springer, 2012.

[5] Jörg Becker and Dominik Kuropka. Topic-based vector space model. In *Proceedings of the 6th International Conference on Business Information Systems*, pages 7–12, 2003.

[6] Colin Bellinger, Shiven Sharma, and Nathalie Japkowicz. One-class versus binary classification: Which and when? In *In proceedings of 11th International Conference on Machine Learning and Applications (ICMLA)*, volume 2, pages 102 – 106, 2012.

[7] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

[8] Abdelhamid Bouchachia, Arthur Lena, and Charlie Vanaret. Online and interactive self-adaptive learning of user profile using incremental evolutionary algorithms. *Evolving Systems*, 5(3):143–157, 2014.

[9] Maya Carrillo, Darnes Vilarino, David Pinto, Mireya Tovar, Saul León, and Esteban Castillo. Buap: three approaches for semantic textual similarity. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, pages 631–634. Association for Computational Linguistics, 2012.

[10] Niladri Chatterjee and Shiwali Mohan. Extraction-based single-document summarization using random indexing. In *Tools with Artificial Intelligence, 2007. ICTAI 2007. 19th IEEE International Conference on*, volume 2, pages 448–455. IEEE, 2007.

[11] Niladri Chatterjee and Pramod Kumar Sahoo. Random indexing and modified random indexing based approach for extractive text summarization. *Computer Speech and Language*, 29:32–44, 2015.

[12] Trevor Cohen, Roger Schvaneveldt, and Dominic Widdows. Reflective random indexing and indirect inference: A scalable method for discovery of implicit connections. *Journal of Biomedical Informatics*, 43(2):240 – 256, 2010.

[13] Trevor Cohen, Roger W Schvaneveldt, and Thomas C Rindflesch. Predication-based semantic indexing: Permutations as a means to encode predications in semantic space. In *AMIA Annual Symposium Proceedings*, volume 2009, page 114. American Medical Informatics Association, 2009.

[14] S Dumais, G Furnas, T Landauer, S Deerwester, S Deerwester, et al. Latent semantic indexing. In *Proceedings of the Text Retrieval Conference*, 1995.

[15] Doina Alexandra Dumitrescu and Simone Santini. Using context to get novel recommendation in internet message streams. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15 Companion, pages 783–786, Republic and Canton of Geneva, Switzerland, 2015. International World Wide Web Conferences Steering Committee.

[16] Ilenia Fronza, Alberto Sillitti, Giancarlo Succi, Mikko Terho, and Jelena Vlasenko. Failure prediction based on log files using random indexing and support vector machines. *The Journal of Systems and Software*, 86(1):2–11, 2013.

[17] João Gama, Indre Žliobaite, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. A survey on concept drift adaptation. *ACM Computing Surveys (CSUR)*, 46(4):44, 2014.

[18] Yang Gao, Yue Xu, and Yue Li. Pattern-based topics for document modelling in information filtering. *IEEE Transactions on Knowledge and Data Engineering*, 27(6):1629 – 1642, June 2015.

[19] James Gorman and James R Curran. Random indexing using statistical weight functions. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 457–464. Association for Computational Linguistics, 2006.

[20] Uri Hanani, Bracha Shapira, and Peretz Shoval. Information filtering: Overview of issues, research and systems. *User Modeling and User-Adapted Interaction*, 11(3):203–259, 2001.

[21] BS Harish, DS Guru, and S Manjunath. Representation and classification of text documents: A brief review. *IJCA, Special Issue on RTIPPR (2)*, pages 110–119, 2010.

[22] Martin Hassel and Jonas Sjöbergh. Towards holistic summarization: Selecting summaries, not sentences. In *Proceedings of LREC 2006*, Genoa, Italy, 2006.

[23] Robert Hecht-Nielsen. Context vectors: general purpose approximate meaning representations self-organized from raw data. *Computational intelligence: Imitating life*, pages 43–56, 1994.

[24] Derrick Higgins and Jill Burstein. Sentence similarity measures for essay coherence. In *Proceedings of the 7th International Workshop on Computational Semantics*, pages 1–12, 2007.

[25] Thomas Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57. ACM, 1999.

[26] William B Johnson and Joram Lindenstrauss. Extensions of lipschitz mappings into a hilbert space. *Contemporary mathematics*, 26(189-206):1, 1984.

[27] Gareth JF Jones and Peter J Brown. Context-aware retrieval for ubiquitous computing environments. In *Mobile and ubiquitous information access*, pages 227–243. Springer, 2004.

[28] Siddhartha Jonnalagadda, Robert Leaman, Trevor Cohen, and Graciela Gonzalez. A distributional semantics approach to simultaneous recognition of multiple classes of named entities. In *Computational Linguistics and Intelligent Text Processing*, pages 224–235. Springer, 2010.

[29] David Jurgens and Keith Stevens. Event detection in blogs using temporal random indexing. In *Proceedings of the Workshop on Events in Emerging Text Types*, eETTs '09, pages 9–16. Association for Computational Linguistics, 2009.

[30] Pentti Kanerva, Jan Kristofersson, and Anders Holst. Random indexing of text samples for latent semantic analysis. In *Proceedings of the 22nd annual conference of the cognitive science society*, volume 1036, 2000.

[31] Shehroz S Khan and Michael G Madden. One-class classification: taxonomy of study and review of techniques. *The Knowledge Engineering Review*, 29(03):345–374, 2014.

[32] Teuvo Kohonen, Samuel Kaski, Krista Lagus, Jarkko Salojärvi, Jukka Honkela, Vesa Paatero, and Antti Saarela. Self organization of a massive document collection. *Neural Networks, IEEE Transactions on*, 11(3):574–585, 2000.

[33] Carsten Lanquillon and Ingrid Renz. Adaptive information filtering: detecting changes in text streams. In *Proceedings of the eighth international conference on Information and knowledge management*, pages 538–544. ACM, 1999.

[34] David D Lewis, Yiming Yang, Tony G Rose, and Fan Li. Rcv1: A new benchmark collection for text categorization research. *The Journal of Machine Learning Research*, 5:361–397, 2004.

[35] Pasquale Lops, Marco De Gemmis, and Giovanni Semeraro. Content-based recommender systems: State of the art and trends. In *Recommender systems handbook*, pages 73–105. Springer, 2011.

[36] Mihai Lupu. On the usability of random indexing in patent retrieval. In *Graph-Based Representation and Reasoning*, number 8577 in Lecture Notes in Computer Science, pages 202–216. Springer International Publishing, 2014.

[37] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *Proceedings of Workshop at ICLR*, 2013.

[38] Hans Moen, Erwin Marsi, and Björn Gambäck. Towards dynamic word sense discrimination with random indexing. In *Proceedings of the Workshop on Continuous Vector Space Models and their Compositionality*, pages 83–90. Association for Computational Linguistics, 2013.

[39] Piero Molino, Pierpaolo Basile, Annalina Caputo, Pasquale Lops, and Giovanni Semeraro. Exploiting distributional semantic models in question answering. In *Semantic Computing (ICSC), 2012 IEEE Sixth International Conference on*, pages 146–153. IEEE, 2012.

[40] Cataldo Musto. Enhanced vector space models for content-based recommender systems. In *Proceedings of the Fourth ACM Conference on Recommender Systems*, RecSys '10, pages 361–364. ACM, 2010.

[41] Cataldo Musto, Pasquale Lops, Marco De Gemmis, and Giovanni Semeraro. Random indexing for content-based recommender systems. In *Proceedings of the 2nd Italian Information Retrieval (IIR) Workshop*, 2011.

[42] Cataldo Musto, Fedelucio Narducci, Pierpaolo Basile, Pasquale Lops, Marco de Gemmis, and Giovanni Semeraro. Cross-language information filtering: Word sense disambiguation vs. distributional models. In *AI* IA 2011: Artificial Intelligence Around Man and Beyond*, pages 250–261. Springer, 2011.

[43] Cataldo Musto, Giovanni Semeraro, Pasquale Lops, and Marco de Gemmis. Random indexing and negative user preferences for enhancing content-based recommender systems. In *E-Commerce and Web Technologies*, pages 270–281. Springer, 2011.

[44] Cataldo Musto, Giovanni Semeraro, Pasquale Lops, and Marco de Gemmis. Contextual evsm: A content-based context-aware recommendation framework based on distributional semantics. In C. Huemer and P. Lops, editors, *EC-Web 2013*, number 152 in LNBIP, pages 125–136. Springer-Verlag, 2013.

[45] Nikolaos Nanas, Stefanos Kodovas, Manolis Vavalis, and Elias Houstis. Immune inspired information filtering in a high dimensional space. In Emma Hart, Chris McEwan, Jon Timmis, and Andy Hone, editors, *Artificial Immune Systems*, volume 6209 of *Lecture Notes in Computer Science*, pages 47–60. Springer Berlin Heidelberg, 2010.

[46] Nikolaos Nanas, Manolis Vavalis, and Lefteris Kellis. Immune learning in a dynamic information environment. In PaulS. Andrews, Jon Timmis, NickD.L. Owens, Uwe Aickelin, Emma Hart, Andrew Hone, and AndyM. Tyrrell, editors, *Artificial Immune Systems*, volume 5666 of *Lecture Notes in Computer Science*, pages 192–205. Springer Berlin Heidelberg, 2009.

[47] Patrick Pantel. Inducing ontological co-occurrence vectors. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 125–132. Association for Computational Linguistics, 2005.

[48] Benjamin Piwowarski, Ingo Frommholz, Yashar Moshfeghi, Mounia Lalmas, and Keith van Rijsbergen. Filtering documents with subspaces. In Cathal Gurrin, Yulan He, Gabriella Kazai, Udo Kruschwitz, Suzanne Little, Thomas Roelleke, Stefan R¼ger, and Keith van Rijsbergen, editors, *Advances in Information Retrieval*, volume 5993 of *Lecture Notes in Computer Science*, pages 615–618. Springer Berlin Heidelberg, 2010.

[49] Rajendra Prasath, Sudeshna Sarkar, and Philip O'Reilly. RI for IR: Capturing term contexts using random indexing for comprehensive information retrieval. In *Human-Inspired Computing and Its Applications*, pages 104–112. Springer, 2014.

[50] Behrang QasemiZadeh and Siegfried Handschuh. Random indexing explained with high probability. In Pavel KrÃ¡l and VÃ¡clav MatouÅ¡ek, editors, *Text, Speech, and Dialogue*, volume 9302 of *Lecture Notes in Computer Science*, pages 414–423. Springer International Publishing, 2015.

[51] Stephen E. Robertson and Ian Soboroff. The TREC 2001 filtering track report. In *Proceedings of The Tenth Text REtrieval Conference, TREC 2001*, November 2001.

[52] Stephen E Robertson and Ian Soboroff. The trec 2002 filtering track report. In *TREC*, volume 2002, page 5, 2002.

[53] Herbert Rubenstein and John B Goodenough. Contextual correlates of synonymy. *Communications of the ACM*, 8(10):627–633, 1965.

[54] M. Sahlgren and J. Karlgren. Automatic bilingual lexicon acquisition using random indexing of parallel corpora. *Nat. Lang. Eng.*, 11(3):327–341, September 2005.

[55] Magnus Sahlgren. Vector-based semantic analysis: Representing word meanings based on random labels. In *In ESSLI Workshop on Semantic Knowledge Acquistion and Categorization*, 2001.

[56] Magnus Sahlgren. Automatic bilingual lexicon acquisition using random indexing of aligned bilingual data. In *LREC*, 2004.

[57] Magnus Sahlgren. An introduction to random indexing. In *Proceedings of the Methods and Applications of Semantic Indexing Workshop at the 7th International Conference on Terminology and Knowledge Engineering, TKE 2005*, August 2005.

[58] Magnus Sahlgren and Rickard Cöster. Using bag-of-concepts to improve the performance of support vector machines in text categorization. In *Proceedings of the 20th international conference on Computational Linguistics*, page 487. Association for Computational Linguistics, 2004.

[59] Magnus Sahlgren, Preben Hansen, and Jussi Karlgren. English-japanese cross-lingual query expansion using random indexing of aligned bilingual text data. In *The Philosophical Writings of Gottlob Frege*, 2002.

[60] Magnus Sahlgren, Anders Holst, and Pentti Kanerva. Permutations as a means to encode order in word space. In Bradley C. Love, Ken. McRae, and Vladimir M. Sloutsky, editors, *Proceedings of the 30th Annual Conference of the Cognitive Science Society*, pages 1300–1305, July 2008.

[61] Magnus Sahlgren and Jussi Karlgren. Vector-based semantic analysis using random indexing for cross-lingual query expansion. In *Evaluation of Cross-Language Information Retrieval Systems*, pages 169–176. Springer, 2002.

[62] Magnus Sahlgren, Jussi Karlgren, Rickard Cöster, and Timo Järvinen. Sics at clef 2002: Automatic query expansion using random indexing. In *Advances in Cross-Language Information Retrieval*, pages 311–320. Springer, 2003.

[63] Linus Sellberg and Arne Jönsson. Using random indexing to improve singular value decomposition for latent semantic analysis. In *LREC*, pages 2335–2338, 2008.

[64] Christian Smith and Arne Jönsson. Automatic summarization as means of simplifying texts, an evaluation for swedish. In *Proceedings of the 18th Nordic Conference of Computational Linguistics (NoDaLiDa-2010), Riga, Latvia*, 2011.

[65] Christian Smith, Arne Jönsson, and IT Santa Anna. Enhancing extraction based summarization with outside word space. In *IJCNLP*, pages 1062–1070, 2011.

[66] Ian Soboroff, Iadh Ounis, J Lin, and I Soboroff. Overview of the trec-2012 microblog track. In *Proceedings of TREC*, volume 2012, 2012.

[67] D Tax, Alexander Ypma, and R Duin. Support vector data description applied to machine vibration analysis. In *Proc. 5th Annual Conference of the Advanced School for Computing and Imaging (Heijen, NL*, pages 398–405. Citeseer, 1999.

[68] Vidya Vasuki and Trevor Cohen. Reflective random indexing for semi-automatic indexing of the biomedical literature. *Journal of biomedical informatics*, 43(5):694–700, 2010.

[69] Miao Wan, Arne Jönsson, Cong Wang, Lixiang Li, and Yixian Yang. Web user clustering and web prefetching using random indexing with weight functions. *Knowledge and Information Systems*, 33(1):89–115, 2012.

[70] Dominic Widdows and Trevor Cohen. The semantic vectors package: New algorithms and public tools for distributional semantics. In *Semantic Computing (ICSC), 2010 IEEE Fourth International Conference on*, pages 9–15. IEEE, 2010.

[71] Sheng-Tang Wu. *Knowledge discovery using pattern taxonomy model in text mining*. PhD thesis, Queensland University of Technology, 2007.

[72] Behrang Q. Zadeh and Siegfried Handschuh. Evaluation of technology term recognition with random indexing. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*, pages 4027–4032, 2014.

[73] Lanbo Zhang and Yi Zhang. Hierarchical bayesian models with factorization for content-based recommendation. *arXiv preprint arXiv:1412.8118*, 2014.

[74] Lanbo Zhang, Yi Zhang, and Qianli Xing. Filtering semi-structured documents based on faceted feedback. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 645–654. ACM, 2011.

[75] Lanbo Zhang, Yi Zhang, and Qianli Xing. Learning from labeled features for document filtering. *CoRR*, abs/1412.8125, 2014.

[76] Yi Zhang. Using bayesian priors to combine classifiers for adaptive filtering. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 345–352. ACM, 2004.

[77] Yi Zhang. *Text Mining: Classification, Clustering, and Applications*, chapter Adaptive Information Filtering. Chapman & Hall/CRC Press, 2009.