ANÁLISIS DE LA COMBINACIÓN DE MODELOS DE FILTRADO DE INFORMACIÓN.

Olga Lucía Monroy Vecino

UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA
INSTITUTO TECNOLÓGICO DE ESTUDIOS SUPERIORES DE MONTERREY (MEXICO)
MAESTRÍA EN CIENCIAS COMPUTACIONALES
BUCARAMANGA
2003

ANÁLISIS DE LA COMBINACIÓN DE MODELOS DE FILTRADO DE INFORMACIÓN.

OLGA LUCÍA MONROY VECINO

Informe final de trabajo de grado
como requisito para optar al título de Magister en Ciencias Computacionales

Director
José de Jesús Pérez Alcázar
PhD. en Informática

UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA
INSTITUTO TECNOLÓGICO DE ESTUDIOS SUPERIORES DE MONTERREY (MÉXICO)
MAESTRÍA EN CIENCIAS COMPUTACIONALES
BUCARAMANGA
2003

Nota de Aceptación:

_____

_____

_____

_____
Presidente del Jurado

_____
Jurado

_____
Jurado

Bucaramanga, 03 de Junio de 2003

AGRADECIMIENTOS

Expreso mis sinceros agradecimientos a:

La Universidad Autónoma de Bucaramanga.

Doctor José de Jesús Pérez Alcázar, director de la Tesis, por su orientación y tiempo.

Los profesores y compañeros de la Maestría en Ciencias Computacionales.

Ingeniero Carlos Fernando Carvajal, por su colaboración en la programación y prueba de los algoritmos.

Y muy especialmente a Dios y a mi querida familia.

CONTENIDO

		pág.
INTRODUCCIÓN		10
1. CONCEPTOS GENERALES		14
1.1 RECUPERACIÓN DE INFORMACIÓN		14
1.1.1 Efectividad y Eficiencia.		15
1.1.2 Relevancia.		16
1.1.3 Modelos de recuperación de Información.		17
1.1.3.1	Vista lógica o representación de los documentos.	17
1.1.3.2	Evolución de los sistemas de recuperación de información.	18
1.1.3.3	Taxonomía.	19
1.1.3.4	Descripción de un modelo de Recuperación de Información.	22
1.1.3.5	Modelos Teóricos alternativos.	24
1.1.3.6	Modelos Algebraicos alternativos.	25
1.1.3.7	Modelos Probabilísticos alternativos.	27
1.2 FILTRADO DE INFORMACIÓN		29
1.3 SISTEMAS DE RECOMENDACIÓN		31
1.4 DSI. DISEMINACIÓN SELECTIVA DE INFORMACIÓN		33
2. MODELOS DE FILTRADO		35
2.1 FILTRADO BASADO EN CONTENIDO		35
2.2 FILTRADO COLABORATIVO		40
2.2.1 Algoritmos de Filtrado Colaborativo basados en Memoria.		43

pág.
2.2.2 Algoritmos de Filtrado Colaborativo basados en Modelo. 45

2.2.2.1 Redes Bayesianas. 45

2.2.2.2 Agrupamiento o Clustering. 45

2.2.2.3 Aproximación basada en reglas. 46

3. INTEGRACIÓN DE FILTRADO COLABORATIVO Y POR CONTENIDO 47

3.1 VISIÓN GENERAL DE LA INTEGRACIÓN DE FILTRADO COLABORATIVO Y POR CONTENIDO 47

3.2 UNA ALTERNATIVA DE INTEGRACIÓN. 50

3.2.1 Integración modelo colaborativo y basado en contenido orientado por objetivos. 51

3.2.1.1 Arquitectura del Modelo. 52

3.2.1.2 Funcionalidad. 53

3.2.2 Uso de información social y basada en contenido para hacer recomendaciones. Recomendación como un problema de clasificación. 54

3.2.3 Filtrado de información Colaborativo Basado en Contenido, con el uso de agentes cooperativos. 56

3.3 EJEMPLOS DE SISTEMAS BASADOS EN CONTENIDO, COLABORATIVOS O HÍBRIDOS 57

4. IMPLEMENTACION DE MODELOS DE INTEGRACION 59

4.1 HERRAMIENTAS UTILIZADAS 59

4.1.1 Base de datos de EachMovie. 59

4.1.2 Java y Oracle. 59

4.2 MÉTRICAS 60

4.3 PROPUESTAS DE INTEGRACIÓN DE MODELOS DE FILTRADO 63

4.3.1 Algoritmo de Filtrado Colaborativo. 63

pág.
4.3.2 Algoritmo de Filtrado por Contenido. 63

4.3.3 Modelo híbrido de Recomendación de Información: Filtrado Colaborativo Mejorado por Contenido. 66

4.3.3.1 Descripción del Sistema. 67

4.3.3.2 Resultados obtenidos. 70

4.3.4 Modelo Combinado para Recomendación de información, usando Filtrado basado en Contenido y Colaborativo. 72

4.3.4.1 Descripción del sistema. 72

4.3.4.2 Resultados obtenidos. 74

4.3.5 Combinación de Filtrado Basado en Contenido y Colaborativo manteniendo los dos modelos separados. 76

5. ANÁLISIS DE RESULTADOS 78

6. CONCLUSIONES 85

7. TRABAJO FUTURO 87

8. REFERENCIAS BIBLIOGRÁFICAS 89

BIBLIOGRAFÍA 93

ANEXO A. ALGORITMOS

LISTA DE TABLAS

pág.
Tabla1. Aplicaciones de un Sistema Basado en Contenido. 47
Tabla 2. Aplicaciones de un Sistema Colaborativo. 48
Tabla 3. Posibles consultas en un sistema combinado. 49
Tabla 4. Error Absoluto Medio, manteniendo el número de usuarios constante. 78

LISTA DE FIGURAS

pág.

Figura 1. Sistema básico de Recuperación de Información. 14
Figura 2. Presicion and Recall. 16
Figura 3. Vista lógica de un documento, desde el texto completo hasta un conjunto de términos indexados. 18
Figura 4. Taxonomía de los Modelos de Recuperación de Información. 20
Figura 5. El coseno de θ se adopta como la sim(d_j,q). 23
Figura 6. Modelo de Redes Neuronales. 27
Figura 7. Ejemplo de Red Bayesiana. 28
Figura 8. Modelo de Red de Inferencia. 29
Figura 9. Filtrado Basado en Contenido. El perfil de un usuario se construye del contenido de los documentos que él ha calificado. 37
Figura 10. El árbol de decisión muestra si el programa de televisión "Melrose Place" será visto, teniendo como padres los programas "Friends" y "Beverly Hills". Las barras en la parte inferior del árbol indican la probabilidad de que el programa "Melrose Place" sea o no visto, condicionado a que los programas padres sean vistos. 46
Figura 11. Arquitectura de METIOREW. 52
Figura 12. Variación de la Media del error Absoluto, con el incremento en el número de ítems, para el filtrado colaborativo. 64
Figura 13. Variación de la Media del error Absoluto, con el incremento en el número de ítems, para el filtrado por contenido. 65
Figura 14. Operación de las matrices UxD y DxD. 65
Figura 15. Descripción general del sistema. 66
Figura 16. Variación de la Media del error Absoluto, con el incremento en el número de ítems, para el modelo híbrido de filtrado colaborativo, mejorado por contenido. 71
Figura 17. Esquema del método combinado CF (Filtrado Colaborativo) y CBF (Filtrado basado en contenido). 73
Figura 18. Variación de la Medio del error Absoluto, con el incremento en el número de ítems, para el modelo de combinación de filtrado colaborativo y por contenido. 75
Figura 19. Variación de la Media del error absoluto, con el incremento en el número de ítems, combinando filtrado colaborativo y por contenido, manteniendo los dos modelos separados. 77
Figura 20. Resultados comparativos de los cinco modelos estudiados. 79

INTRODUCCIÓN

Tomando las palabras de Peter Denning, Presidente de ACM[1] (1.982), como punto de partida, se observa la problemática que ocupa a este trabajo: "La Visión de los computadores personales, estaciones de trabajo y redes locales, han enfocado su atención principalmente a la generación de información - proceso de producir documentos y diseminarlos. Es el momento de centrar la atención en recibir información - proceso de controlar y filtrar información, para que llegue a las personas que deban usarla."

El tema de la sobrecarga de información se ha discutido ampliamente en la literatura referente a la administración y recuperación de información, pero aun sigue siendo un problema sin resolver. La información para cada usuario llega de muy diversas fuentes y en diferentes formatos: en el área profesional, hay memorandos, libros, artículos técnicos, conferencias, publicaciones, páginas web, mensajes de voz y de correo electrónico. Para mantenerse informado de los eventos en el mundo, hay artículos de periódicos, revistas, reportes del tiempo, anuncios, etc. En el área de entretenimiento, se debe seleccionar de una innumerable colección de películas, videos, CD's de música, presentaciones de teatro, musicales, ofertas de viajes, eventos recreacionales, etc., y aún pueden enumerarse otros campos más.

Cualquiera de estos flujos de información representa más datos de los que una persona normal puede manejar. En todos los casos descritos, las personas pueden sacar provecho de esta información, y necesitar solo una parte muy específica de todo este flujo, pero puede significarle un esfuerzo considerable para localizarla y en muchos casos, la búsqueda puede no ser exitosa. La tarea de examinar grandes volúmenes de información es repetitiva y pesada, lo que la hace candidata para ser automatizada usando software de computador. Esta ha sido la herramienta utilizada por muchos años en campos como la recuperación y el filtrado de información, aplicando nuevas tecnologías, apuntando a resolver o minimizar esta sobrecarga.

Las aplicaciones más recientes emplean tecnologías como el filtrado colaborativo automático o la representación y evaluación de documentos, mediante diversos métodos automatizados, para ayudarle a los usuarios en sus tareas de recepción y análisis de información. En internet pueden encontrarse ejemplos de herramientas creadas para facilitarle a los usuarios las tareas de búsqueda de información en muy diversas áreas, que llegan más allá de responder a las consultas realizadas por un individuo, recomendándole ítems que puedan ser de su interés. En el desarrollo de todo el documento, se hará referencia al término ítem, por ser la

[1] Association for Computing Machinery, primera Sociedad en Computación, fundada en 1.947.

forma más general, que cobija cualquier tipo de recurso, sea un documento textual (un abstract, un artículo completo, un periódico, etc.), una página web, una película, un libro, etc.

En los últimos años y particularmente en la última década, la definición de Recuperación de Información ha cambiado considerablemente, con la expansión de la WWW y la llegada de modernas interfaces gráficas para usuarios y de dispositivos de almacenamiento masivos, por lo tanto, la recuperación tradicional sobre libros de texto, ha dado paso a nuevos proceso y a nuevas fuentes, que hagan más efectivo el proceso y ayuden a limitar los resultados. Para el mejoramiento de este proceso, se hace necesaria la utilización de una estrategia de recuperación de información, que se define como una técnica mediante la cual se obtiene una medida de relevancia entre una consulta y uno o varios ítems.

Estas estrategias pueden ser manuales (sistema booleano, conjuntos Fuzzy o redes de inferencia), automáticas (modelo del vector espacial o indexación semántica latente) o adaptativas (modelos probabilísticos, algoritmos genéticos o redes neuronales). En el desarrollo del documento se explicarán algunos de estos conceptos, pero el estudio se centra en el uso del vector espacial como una estrategia de recuperación de información automática utilizada para el filtrado basado en contenido.

En muchos casos, el solo análisis del contenido de un ítem no es suficiente para determinar que tan interesante será este ítem para un usuario particular o para un grupo de usuarios con intereses comunes. Los computadores son rápidos en el procesamiento de información, pero son generalmente tontos comparados con la capacidad humana de tomar decisiones sobre el contenido de la información. Por tanto, es de gran utilidad emplear el filtrado colaborativo, que aplica la velocidad de los computadores con la inteligencia de los humanos, haciendo uso de la experiencia o de los conocimientos de otros individuos en áreas específicas de conocimiento. En el documento se hace referencia a algunos de los muchos sistemas de recomendación y sistemas de filtrado de información comerciales, para diversos usos, por ejemplo, artículos de revistas o de periódicos, libros, videos, música, restaurantes, chistes, entre otros.

La documentación está dividida en ocho capítulos empezando con una visión general de la temática alrededor de la recuperación de información hasta las recomendaciones para trabajos futuros en esta área.

En el capítulo primero, se busca hacer una revisión de conceptos básicos de Recuperación de Información como antesala a la tarea de Filtrado de Información y de sistemas que permiten, además de responder a una consulta de un usuario, recomendarle un conjunto de ítems que también pueden ser de su interés, que es lo que se conoce como Sistemas de Recomendación. Se tratará brevemente el concepto de Diseminación Selectiva de Información, como herramienta que utiliza

el filtrado de información y le lleva los resultados de este proceso a él o los usuarios que se hayan suscrito a este servicio, de forma automática, es decir, la información le llega al usuario sin que ellos tengan que disparar una consulta, valiéndose del perfil que cada uno haya definido con anterioridad.

Posteriormente, en el capítulo dos, se habla de los modelos más utilizados y ampliamente conocidos para la realización de filtrado de información: Filtrado de información Basado en Contenido (Cognitivo) y el Filtrado Colaborativo (Social), y de las diferentes clases de algoritmos pertenecientes a este grupo; así como de las debilidades que presenta cada uno de los modelos, que motivan a la realización del presente estudio, buscando alternativas para trabajar con las bondades de cada uno de los dos, combinando o integrando los dos procesos para minimizar aquellos puntos que los hacen débiles a cada uno por separado.

Esta integración se trata en los dos capítulos siguientes. En el capítulo tres, se muestra una visión general de la integración de los dos modelos ya mencionados y de aquí se desprenden algunos ejemplo de propuestas de trabajo combinado o soluciones híbridas, donde se pueden ver diferentes maneras para utilizar el contenido de la información junto con la experiencia de otros individuos que ya se ha recolectado y almacenado. Al finalizar este capítulo, se describen brevemente algunos sistemas que muestran estas propuestas y que se pueden estudiar mediante la documentación que se consigue libremente en internet.

El capítulo cuatro, presenta la descripción e implementación de tres propuestas de integración que son en su orden: "Content-Boosted Collaborative Filtering" desarrollada por Prem Melville, Raymond J. Mooney y Ramadass Nagarajan, del departamento de ciencias computacionales de la Universidad de Texas; en segundo lugar, "Combining Content-based and Collaborative filtering", planteada por Gabriela Polcicová y Pavol Návrat, del Departamento de Ciencias Computacionales e Ingeniería de la Universidad Slovaka de Tecnología y por último, la propuesta hecha en la Tesis "Improvement to Collaborative Filtering Algorithms" de Anuja Gokhale, del Instituto Politécnico de Worcester. Para cada una de las tres propuestas, se describe el uso del filtrado por contenido y el colaborativo, la integración de los dos, los resultados utilizando como datos de prueba, la base de datos de Each Movie, que se describe al comienzo de este capítulo, donde se muestra la forma de obtener libremente esta base de prueba para ser utilizada por los investigadores en el área del filtrado de información. Para determinar que tan buenos pueden ser los resultados de cualquier propuesta es necesario la utilización de métricas que permitan evaluar que tan precisas son las respuestas que puede arrojar un sistema que filtre y recomiende información. Por lo tanto, en la sección 4.2, se habla de las métricas más ampliamente utilizadas para la evaluación de este tipo de algoritmos.

Seguidamente, el capítulo cinco muestra los resultados comparativos de las tres propuestas, que ya en el capítulo cuatro se mostraron y explicaron en forma

individual. Para los tres casos, las pruebas se efectuaron sobre el mismo conjunto de datos, evaluando los resultados de la medición hecha por el cálculo de la media del error absoluto, trabajando con 20 ítems, luego con 40, 60, 80, 100, 120 y 140, manteniendo constante el número de usuarios.

El capítulo seis, resume las conclusiones de lo que fue el desarrollo del trabajo y de los resultados obtenidos. De los pros y contras de las propuestas que se desarrollaron, se desprenden una serie de puntos que se describen en el capítulo siete como trabajos futuros. Algunos son mejoras a los algoritmos desarrollados, otros sugieren la utilización de diferentes métricas que permitan medir otros aspectos de los modelos, el empleo de nuevas formas de representación de los documentos y la adición de la retroalimentación con usuarios reales para seguir afinando los modelos.

Por último, en el capítulo ocho, se presenta la lista de los autores consultados para la investigación y el desarrollo del trabajo. Un buen número de las referencias se encuentra como páginas de internet, de libre acceso, que permiten profundizar más en los temas relativos a la recuperación de información y filtrado.

3. CONCEPTOS GENERALES

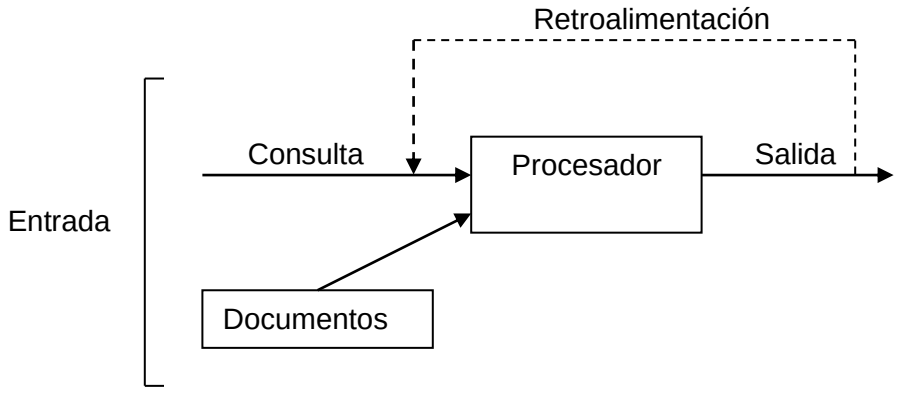
1.1 RECUPERACIÓN DE INFORMACIÓN [9]

El problema de almacenamiento y recuperación de información ha atraído la atención de muchos, desde los años cuarenta, ya que el vasto monto de información de que se dispone, no puede ser explotado rápida y exitosamente y esta dificultad es cada vez mayor. El efecto de este problema es que una gran cantidad de información relevante no es descubierta y produce duplicación de trabajo y esfuerzo. A pesar del uso de computadores para esta tarea, el problema de recuperación efectiva continúa aún sin resolver.

El propósito de una estrategia de recuperación automática es traer todos los documentos relevantes y al mismo tiempo recuperar la menor cantidad posible de información no relevante. La figura 1 muestra un sistema típico de recuperación de información, que aunque parece demasiado sencillo, permite crear una amplia discusión.

La entrada: El principal problema aquí es obtener una representación de los documentos y de las consultas de tal forma que puedan ser usadas por el computador. La representación del documento podría ser una lista de palabras extractadas que se consideran significativas.

Figura 1. Sistema básico de Recuperación de Información.



La retroalimentación: Cuando el sistema de recuperación es en línea, el usuario puede cambiar su petición durante una sesión de búsqueda, mejorando la siguiente recuperación.

El procesador: Es la parte del sistema que involucra la estructuración de la información en una forma apropiada y la ejecución de la estrategia de búsqueda en respuesta a la consulta.

Los documentos: Forman parte de la entrada pero son usados en el proceso de recuperación.

La salida: Generalmente un conjunto de citaciones o números de documentos. Aquí termina el proceso, a menos que el sistema permita hacer una evaluación de los resultados.

Para que el sistema de recuperación sea aplicable, debe evaluarse en términos de la calidad del mismo, que se mide en diversos aspectos, tales como la velocidad de procesamiento, el espacio de almacenamiento requerido, o por los resultados comparativos entre los documentos recuperados y documentos relevantes para una consulta específica. El presente trabajo se centra principalmente en la evaluación de la recuperación.

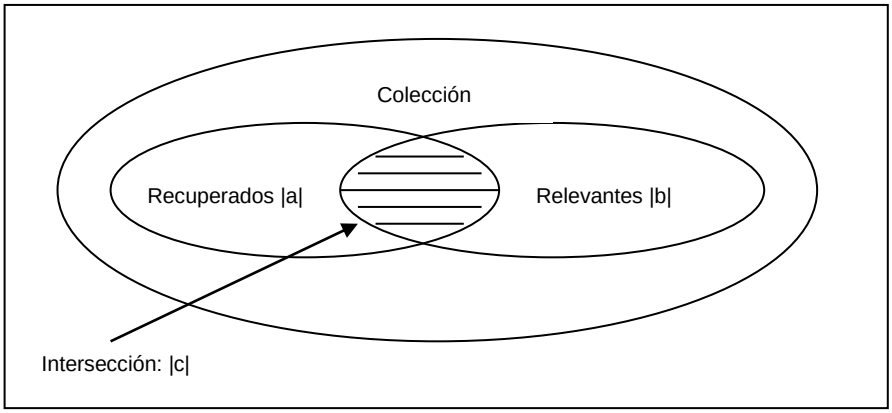
1.1.1 Efectividad y Eficiencia. Muchas de las investigaciones y desarrollos en recuperación de información están orientados al mejoramiento de la efectividad y eficiencia de este proceso. La eficiencia se mide generalmente en términos de recursos de computador. Esta medida debe tomarse en conjunción con la efectividad para obtener una idea de los beneficios en términos de costos. La efectividad es comúnmente medida usando las métricas de “Precision and Recall”. Algunos autores traducen estos términos como Precisión y exhaustividad, pero no es una traducción uniforme, por lo que en este documento se emplean los términos originales en inglés, como se encuentran con mayor frecuencia en la literatura. Precision, determina cuántos documentos recuperados son relevantes y Recall, cuántos documentos relevantes se han recuperado.

Tomando los documentos de la colección de pruebas, los relevantes y una consulta particular, se aplica dicha consulta al sistema que se desea evaluar. Sea |b| el número de documentos relevantes, |a| el número de documentos recuperados, como se ilustra en la figura 2, las medidas de Precision and Recall se definen como:

$$\Pr ecision = \frac{|c|}{|a|}$$

$$\mathrm{Re}\, call = \frac{|c|}{|b|}$$

Figura 2. Presicion and Recall. [33]



Precision and Recall mide la habilidad del sistema para recuperar documentos relevantes y al mismo tiempo descartar los no relevantes. Se ha debatido fuertemente si Precision and Recall son valores apropiados para ser usados como medida de efectividad, a pesar de ser las métricas más ampliamente usadas y de fácil entendimiento.

1.1.2 Relevancia. La relevancia es una noción subjetiva, ya que diferentes usuarios pueden diferir sobre la relevancia o no relevancia de un documento en particular, para una consulta dada. Las diferencias no son suficientemente grandes como para invalidar los experimentos que se han hecho con colecciones de documentos de una disciplina particular, para los que se conocen las consultas de prueba y la correspondiente relevancia de los resultados, que han sido dadas por un grupo de expertos en esa disciplina.

En el campo de la Recuperación de Información, se asume que si una estrategia de recuperación se comporta bien bajo un número grande de condiciones experimentales, se comportará en forma similar en una situación operacional, donde la relevancia no se conoce de antemano.

La retroalimentación de relevancia es la estrategia de reformulación de consultas más popular. En un ciclo de retroalimentación, el usuario examina una lista de documentos que fueron recuperados, y marca los que él considera relevantes. Generalmente solo se examinan los primeros 10 documentos (los mejor rankeados). La idea principal consiste en seleccionar términos importantes, o expresiones, y ligados a los documentos que se marcaron como relevantes, para resaltar la importancia de estos términos en la formulación de una nueva consulta. [31]

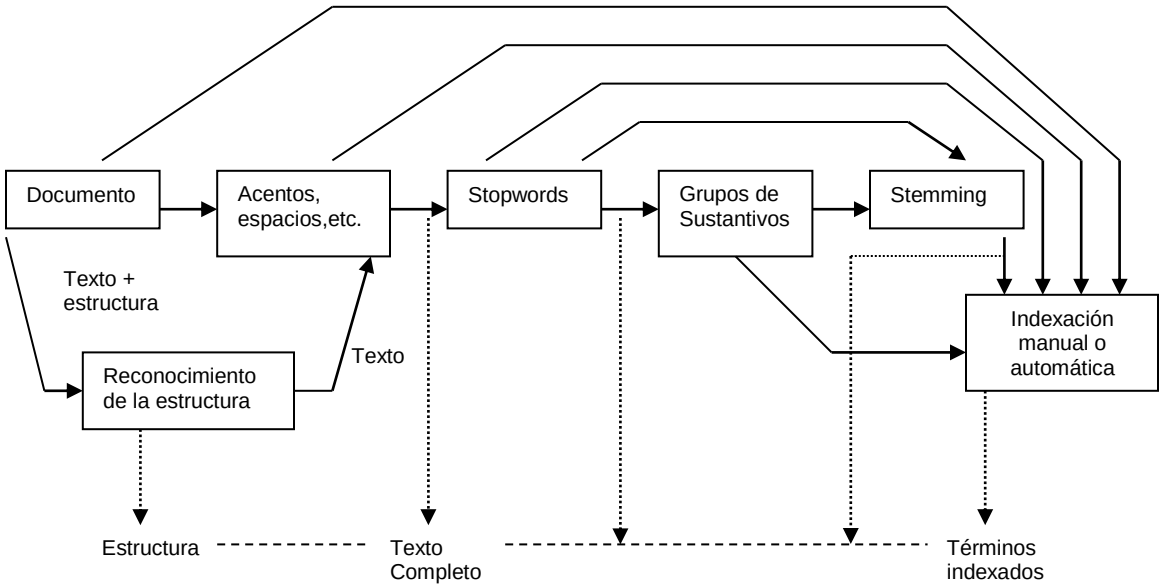
1.1.3 Modelos de recuperación de Información. [31] Un proceso de recuperación de información se ve afectado directamente por dos aspectos principales: Las tareas que debe realizar el usuario y la manera en que se representan los documentos, que se conoce como la vista lógica de los mismos. El usuario tiene que traducir sus necesidades de información a una consulta escrita en el lenguaje que le imponga el sistema. Sus tareas pueden ser de dos tipos distintos, de recuperación de información o datos y de navegación. Los sistemas de hipertexto, facilitan la navegación rápida y los sistemas modernos de bibliotecas digitales e interfaces Web intentan combinar los dos tipos de tareas para mejorar las capacidades de recuperación.

1.1.3.1 Vista lógica o representación de los documentos. Generalmente los documentos se representan como un conjunto de términos indexados o palabras claves. Estas palabras claves pueden obtenerse automáticamente (directamente del texto) o generados por un especialista y esta lista muestra una vista lógica del documento.

Para grupos muy grandes de documentos es necesario reducir el conjunto de palabras claves representativas. Esto se logra mediante la eliminación de stopwords (artículos, conjunciones, pronombres, etc.), el uso de stemming (proceso que reduce distintas palabras a su raíz gramatical común) y la identificación de grupos de sustantivos (elimina adverbios, adjetivos y verbos). (Ver figura 2).

La vista más completa de un documento, es por supuesto el texto completo, pero su uso es computacionalmente muy costoso. La vista lógica más concisa la puede generar un especialista humano, pero puede llevar a una recuperación muy pobre. Los sistemas de recuperación adoptan vistas lógicas intermedias de los documentos, como se aprecia en la figura 3.

Figura 3. Vista lógica de un documento, desde el texto completo hasta un conjunto de términos indexados.



1.1.3.2 Evolución de los sistemas de recuperación de información. [31] Por aproximadamente 400 años, el hombre ha organizado la información para que pueda ser recuperada y usada posteriormente, por ejemplo, mediante la tabla de contenido de un libro. Con el crecimiento del volumen de información se hizo necesario la creación de estructuras de datos para la recuperación de información más rápida. Una de estas estructuras, es una colección de palabras seleccionadas o conceptos, asociados a punteros a la información relacionada (o documentos), que es lo que se conoce como un índice. Los índices, son el corazón de todos los sistemas modernos de recuperación de información, y permiten el acceso rápido a los datos, y el procesamiento ágil de las consultas.

Por mucho tiempo los índices se crearon en forma manual, en una organización jerárquica. En la actualidad, todavía la mayoría de las bibliotecas utilizan alguna forma de categorización jerárquica para clasificar sus libros (o documentos). Con el empleo de computadores modernos se hizo posible la construcción de grandes índices de forma automática, que proporciona una vista del problema de recuperación más enfocado al sistema en sí, que a las necesidades del usuario.

Es importante resaltar que el problema de la recuperación de información tiene dos diferentes enfoques: El primero de ellos centrado en la máquina (computer - centered) y el segundo centrado en el usuario (Human – centered). En el enfoque centrado en las máquinas, el problema de la Recuperación de Información consiste principalmente en la construcción de índices eficientemente, el

procesamiento de las consultas del usuario con alto rendimiento, y el desarrollo de algoritmos de evaluación (ranking algorithms) que mejoren la calidad de las respuestas.

En el enfoque centrado en el usuario, el problema de recuperación de información consiste principalmente en el estudio del comportamiento del usuario, el entendimiento de sus principales necesidades y la determinación de cómo tal entendimiento afecta la organización y operación del sistema de recuperación. El desarrollo de este trabajo enfoca su atención hacia los sistemas de recuperación centrados en las máquinas, que son dominantes en el mercado.

Las bibliotecas fueron unas de las primeras instituciones en adoptar los sistemas de recuperación de información, inicialmente desarrollados por las mismas instituciones y luego por proveedores comerciales. La primera generación de estos sistemas automatizó tecnologías previas (catálogo de tarjetas – consultas en ficheros) y permitían consultas basadas en autor y título. En la segunda generación, se incrementó la funcionalidad de la búsqueda, permitiendo consultar por encabezados de materias, palabras claves y algunas consultas más complejas. La tercera generación se enfoca a la mejora de las interfaces gráficas, formas electrónicas, características de hipertexto y arquitecturas de sistemas abiertos.

Con la llegada y amplia difusión de los recursos Web y los avances en tecnologías de computación, han ocurrido algunos cambios fundamentales:

- El acceso a numerosas fuentes de información es ahora mucho más económico y fácil para investigadores y audiencias mucho mayores.

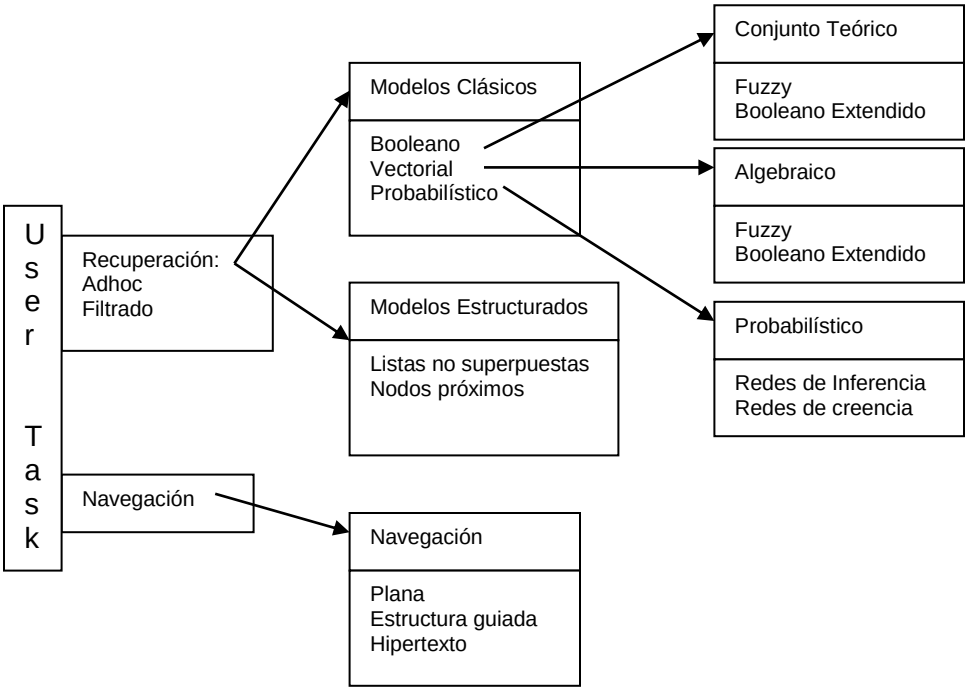
- Los avances en las comunicaciones digitales facilitan el acceso a amplias redes, logrando que las fuentes de información estén disponibles aunque se encuentren localizadas a grandes distancias y puedan ser consultadas rápidamente (en pocos segundos).

- La libertad de colocar cualquier información que alguna persona juzgue que pueda ser de utilidad, ha contribuido ampliamente a la popularidad de la Web.

1.1.3.3 Taxonomía. [31] Existen tres modelo clásicos en la recuperación de información, conocidos como Booleano, Vectorial y Probabilístico. El modelo Booleano representa los documentos y las consultas de usuario como un conjunto de términos indexados (Modelo de Conjunto Teórico). En el modelo Vectorial, los documentos y las consultas se representan como vectores en un espacio de dimensión t, y se dice que el modelo es Algebraico. En el modelo probabilístico, la representación de documentos y consultas se basa en la teoría de probabilidad, por tanto se dice que el modelo es Probabilístico. Para cada uno de estos tipos

clásicos, se han propuesto modelos alternos; para el Conjunto Teórico: Fuzzy[2] y Modelo Booleano Extendido; para el Algebraico: Vectorial Generalizado, Indexación Semántico Latente y Redes Neuronales; y para modelos probabilísticos, se distinguen las redes de inferencia y las redes de creencia. Esta clasificación se ilustra en la figura 4.

Figura 4. Taxonomía de los Modelos de Recuperación de Información.



Más allá de las referencias al contenido de los textos, el modelo puede permitir referencias a la estructura presente en los textos. En este caso, se habla de modelo estructurado. En esta clase se distinguen dos modelos: listas no superpuestas y el modelo de los nodos próximos.

En el proceso de búsqueda de información, la tarea del usuario puede ser de navegación en lugar de Recuperación, como cuando sus intereses no están muy bien definidos o son muy abiertos. Se conocen tres modelos para la navegación: Plano, de Estructura Guiada y de Filtrado. Ambas tareas, recuperación y navegación son en lenguaje de WWW, acciones que traen o jalan información (pulling actions). Una alternativa es hacer la recuperación de manera automática y

[2] Lógica Fuzzy es un superconjunto de la lógica convencional (booleana), que se ha extendido para manejar el concepto de verdad parcial, valores de verdad entre "completamente cierto" y "completamente falso". Este concepto fue introducido por el Dr. Lotfi Zadeh en los años 60's.

permanente, usando agentes de software que envíen (push) la información hacia el usuario. Por ejemplo, la información útil para el usuario puede ser extractada periódicamente de un servicio de noticias. En este caso, se dice que el sistema de recuperación de información ejecuta una tarea especial que consiste en filtrar la información relevante, para la posterior revisión por parte del usuario.

En un sistema de recuperación de información tradicional, los documentos permanecen relativamente estáticos, mientras se hacen nuevas consultas; este modo de operación es el que se conoce como Recuperación y es la tarea más común del usuario. Cuando las consultas permanecen relativamente estáticas mientras el sistema se alimenta con nuevos documentos, se habla de filtrado. En una tarea de filtrado, el perfil del usuario describe sus preferencias, tal perfil se compara después con los documentos entrantes, intentando determinar cuales de esos documentos pueden ser de interés para este usuario particular. Por ejemplo, esta propuesta puede ser usada para seleccionar un nuevo artículo entre cientos de artículos que son enviados cada día.

La tarea de filtrado simplemente le indica al usuario los documentos que pueden ser de su interés. La determinación de cuáles son realmente relevantes es totalmente reservada para el usuario. Una variación de este procedimiento es calificar (*ranking*) los documentos filtrados y mostrar este *ranking* al usuario. Esto ayuda a que el usuario solo examine los primeros de la lista y se disminuya su trabajo. Aún si el *ranking* no se le muestra al usuario, el sistema puede calcularlo para determinar la relevancia potencial de los documentos. Por ejemplo, los documentos con un *ranking* superior a un umbral dado serán seleccionados y los otros descartados. El modelo más utilizado para adoptar el *rankeo* de documentos es el del vector espacial, debido a su simplicidad.

En el proceso de filtrado, el paso crucial no es el *rankeo* de los documentos, sino la construcción de un perfil de usuario que realmente refleje sus preferencias. Una propuesta simple para construir el perfil es describirlo mediante palabras claves y pedirle al usuario que provea las palabras necesarias. Esto requiere mucho trabajo del usuario. Una alternativa más elaborada es recoger información del usuario sobre sus preferencias y usar esta información para construir su perfil dinámicamente. Al comienzo el usuario da una serie de palabras claves las cuales describen inicialmente (y en forma primitiva) sus preferencias. Cuando llegan nuevos documentos, el sistema usa este perfil para seleccionar documentos que sean potencialmente de su interés y se los muestra al usuario. Luego el usuario en un ciclo de retroalimentación indica cuáles son relevantes y cuáles no. El sistema usa esta información para ajustar la descripción del perfil de usuario que refleja las nuevas preferencias. El perfil cambia constantemente, se espera que se estabilice y no se presenten cambios drásticos a menos que el usuario cambie sus intereses repentinamente.

1.1.3.4 Descripción de un modelo de Recuperación de Información. Un modelo de Recuperación de Información está conformado por la cuádrupla [D, Q, f, R (q_i, d_j)] donde:

- D es un conjunto compuesto por las vistas lógicas o representaciones de los documentos.

- Q es un conjunto compuesto por las vistas lógicas o representaciones de las necesidades de usuario, que se nombran como consultas (Queries).

- f es una estructura para el modelamiento de las representaciones de los documentos, las consultas y sus relaciones.

- R (q_i, d_j) es una función de rankeo que asocia un número real con una consulta $q_i \in Q$ y una representación de un documento $d_j \in D$. Este ranking define un ordenamiento entre los documentos que se relacionan con la consulta.

La estructura de modelamiento determina la construcción de la función de rankeo. Por ejemplo para el modelo booleano clásico, la estructura se compone de un conjunto de documentos y de operaciones estándar entre conjuntos. Para el modelo clásico de Vector Espacial, la estructura tiene un vector espacial de dimensión t y operaciones estándar de álgebra lineal entre vectores. Para el modelo probabilístico clásico, la estructura utiliza conjuntos, operaciones estándar de probabilidad y el teorema de Bayes.

A continuación se describe brevemente cada uno de los tres modelos clásicos para la recuperación de información:

1. Modelo Booleano: Es un modelo simple de recuperación basado en la teoría de conjuntos y el álgebra booleana. Su estructura es fácil de entender por un usuario común de un sistema de recuperación de información. Las consultas se especifican como expresiones booleanas con una semántica precisa. Por su simplicidad y formalismo, este modelo fue muy utilizado en años anteriores y adoptado por muchos de los primeros sistemas bibliográficos comerciales.

Este modelo, a pesar de ser aún el más utilizado en los sistemas comerciales de bases de datos de documentos, tiene importantes debilidades. Primero, su estrategia de recuperación se basa en un criterio de decisión binario, es decir, un documento es relevante o no relevante, sin contemplar diferentes grados de relevancia, que podrían mejorar el rendimiento de la recuperación. Por tanto, este modelo es más un sistema de recuperación de datos que un modelo de recuperación de información. Segundo, mientras las expresiones booleanas tienen una semántica precisa, las necesidades de información con frecuencia son difíciles de traducir a este tipo de expresiones. De hecho, no es fácil para la

mayoría de usuarios expresar sus consultas en términos de expresiones booleanas.

El modelo booleano considera que los términos indexados están presentes o ausentes de un texto. Como resultado, los pesos de los términos en el índice tienen valores binarios de 0 y 1. Una consulta que se compone de términos indexados unidos por tres conectores: And, Or y Not. Así, una consulta es esencialmente una expresión booleana convencional.

2. Modelo Vectorial. Este modelo reconoce que el uso de pesos binarios es demasiado limitado y propone una estructura en la que se puedan hacer correspondencias parciales. Esto se logra calculando pesos ($w_{i,j}$) a los términos para hallar el grado de similitud entre cada documento almacenado y la consulta del usuario. El modelo vectorial tiene en cuenta documentos que satisfacen la consulta parcialmente, dependiendo del número de coincidencias, ordenando los documentos recuperados en forma descendente de acuerdo al grado de similitud,. El principal efecto es que el conjunto de documentos rankeados es más preciso, en el sentido que corresponde mejor a las necesidades de información del usuario, mostrando cuales se aproximan mejor a la consulta realizada, que los documentos que halla un modelo booleano.

Un documento d_j se representa como un vector de dimensión t, así:

$$\vec{d_j} = (w_{1,j}, w_{2,j}, w_{3,j}, \ldots . w_{t,j})$$

y una consulta de usuario q, como:

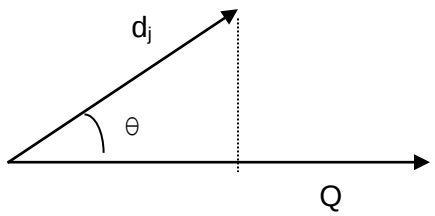
$$\vec{q} = (w_{1,q}, w_{2,q}, w_{3,q}, \ldots . w_{t,q})$$

El modelo vectorial propone evaluar el grado de similaridad de un documento dj y una consulta q ($sim(d_j,q)$), como la correlación entre los dos vectores:

$$\vec{d_j} \text{ y } \vec{q}$$

Este valor se calcula mediante el coseno del ángulo entre los dos vectores.

Figura 5. El coseno de θ se adopta como la $sim(d_j,q)$.



Ya que los pesos $w_{i,j} \geq 0$ y $w_{i,q} \geq 0$, la similaridad d_j y q varía entre 0 y 1. Por lo tanto este modelo en lugar de predecir si un documento es relevante o no, rankea los documentos de acuerdo al grado de similaridad con la consulta.

3. Modelo Probabilístico. Este modelo se fundamenta en que hay un conjunto de documentos relevantes que satisfacen la consulta hecha por un usuario y no otro. Este conjunto de documentos es la "respuesta ideal". El tratamiento de las consultas se reduce a un proceso de especificar las propiedades de un conjunto ideal de respuesta que es análogo a interpretar el problema de la recuperación de información como un problema de agrupamiento o clustering. Estas propiedades se desconocen, se sabe que la semántica de los términos en el índice debe usarse para caracterizar estas propiedades.

Una primera predicción permite generar una descripción probabilística del conjunto ideal de respuesta, que se emplea para recuperar un primer conjunto de documentos. El usuario analiza los documentos recuperados y decide cuáles son relevantes y cuáles no. Con esta información, el sistema refina la descripción del conjunto ideal de respuesta. Repitiendo este proceso varias veces, se espera llegar a tener una descripción muy cercana de este conjunto de respuestas.

El modelo probabilístico se basa en asumir el siguiente principio probabilístico: Dada una consulta de un usuario q y un documento d_j en la colección, el modelo trata de estimar la probabilidad de que el usuario encontrará interesante (relevante) al documento d_j. El modelo asume que la probabilidad de relevancia depende solo de las representaciones de la consulta y de los documentos. El modelo asume que hay un subconjunto de todos los documentos que son preferidos por el usuario como conjunto de respuesta para su consulta. Estos documentos forman el conjunto R de relevantes y los documentos que no están ahí se consideran no relevantes.

1.1.3.5 Modelos Teóricos alternativos.

1. Modelo del Conjunto Fuzzy: La representación de documentos y consultas a través de conjuntos de palabras claves, lleva a descripciones que solo se relacionan parcialmente con el contenido de los respectivos documentos y consultas. Como resultado la correspondencia de un documento con los términos de la consulta es aproximada o vaga. Esto puede modelarse considerando que cada término de la consulta define un conjunto fuzzy y que cada documento tiene un grado de pertenencia o membresía (usualmente > 1) en este conjunto. Esta interpretación del proceso de recuperación (en términos de conceptos de la teoría fuzzy) es el fundamento básico de diversos modelos de conjuntos fuzzy para la representación, estudiados por años. Estos modelos han sido discutidos principalmente en la literatura dedicada a la

teoría fuzzy y no son populares entre la comunidad de recuperación de información.

2. Modelo Booleano extendido: La idea central es extender el modelo Booleano con las características del modelo vectorial. Considere una operación de conjunción booleana en una consulta dada por: q=Kx ∧ Ky. De acuerdo al modelo booleano, un documento que contenga el término Kx o el término Ky es tan irrelevante como otro documento que no contenga ninguno de los dos términos. Sin embargo, este criterio de decisión binario, frecuentemente no está de acuerdo con el sentido común. Por tal razón se introduce el uso de pesos para los términos, usando los factores tf – idf que se explicarán en el numeral 2.1.

1.1.3.6 Modelos Algebraicos alternativos.

1. Modelo del Vector Espacial Generalizado. Sea $\vec{k}_i$ un vector asociado con el término del índice k_i. La independencia de los términos del índice en el modelo vectorial implica que el conjunto de vectores $\{\vec{k}_1, \vec{k}_2, .. \vec{k}_t\}$ es linealmente independiente y forman una base para el subespacio de interés. La dimensión de este espacio es el número t de términos del índice en la colección. En el modelo del vector espacial generalizado, dos vectores de términos del índice pueden ser no ortogonales. Estos vectores están formados de componentes más pequeños de colecciones particulares. Dado el conjunto $\{k_1, k_2, .. k_n\}$ de índices de términos en una colección, y los pesos $w_{i,j}$ asociados con cada pareja término-documento $[k_i, d_j]$, si todos estos pesos son binarios, todos los posibles patrones de co-ocurrencia (dentro de los documentos) pueden ser representados por un conjunto de 2^t términos principales dados por $m_1 = (0,0,\ldots,0)$, $m_2 = (1,0,\ldots,0)$, . . . $m_{2^t} = (1,1,\ldots,1)$. $g_i(m_j)$ retorna el peso {0,1} del término del índice k_i en el término principal m_j.

Así, el término principal m_1 (para el cual $g_i(m_1) = 0$ para todo i) apunta a los documentos que no contienen ninguno de los términos del índice. El término principal m_2 (para el cual $g_1(m_2) = 1$ para i =1, y $g_i(m_2) = 0$, para i >1) apunta a los documentos que contienen solamente el término del índice k1. Por tanto, el término principal m_{2^t} apunta a los documentos que contienen todos los términos del índice.

La idea central en el modelo de Vector Espacial Generalizado es introducir un conjunto de parejas de vectores ortogonales $\vec{m}_i$ asociados con el conjunto de términos principales y adoptar este conjunto de vectores como la base para el subespacio de interés.

2. Modelo de Indexación Semántico Latente (LSI – Latent Semantic Indexing)[3]. Las ideas en un texto están más relacionadas a los conceptos descritos en él que con los términos del índice usados para describirlo. Por tanto, el proceso de correspondencia entre documentos y una consulta dada, podría basarse en la correspondencia de conceptos en lugar de igualación de términos del índice. Por ejemplo, un documento puede ser recuperado porque comparte conceptos con otro documento que es relevante para una consulta dada.

La idea principal en este modelo es representar cada vector documento y consulta en un espacio dimensional menor que es asociado con conceptos. Esto se logra representando los vectores de términos del índice en este espacio dimensional. La recuperación en este espacio reducido puede ser superior a la recuperación en el espacio de índice de términos. El modelo de indexación semántico latente introduce una conceptualización interesante del problema de la recuperación de información en la teoría de la descomposición en valores singulares.

3. Modelo de Redes Neuronales. En un sistema de recuperación de información, los vectores de los documentos deben ser comparados con los vectores de las consultas para calcular un ranking, mediante la correspondencia de términos indexados. Se considera el uso de redes neuronales como un modelo alternativo, por su reconocido buen desempeño en la correspondencia de patrones.

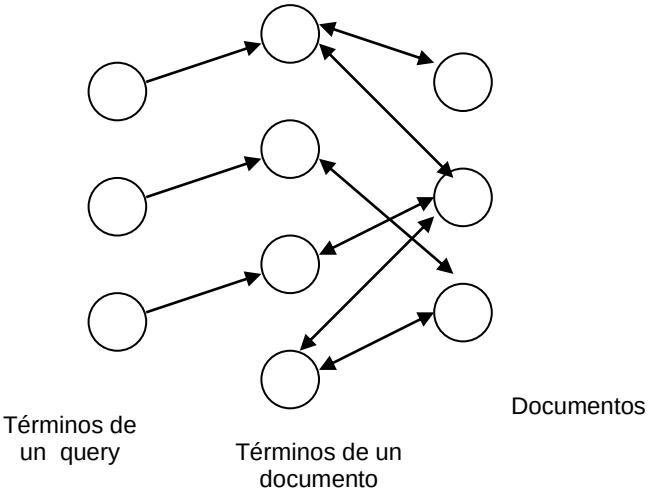
El cerebro humano está compuesto de billones de neuronas, cada una puede ser vista como una unidad de procesamiento básico, que al ser estimulada por una señal de entrada puede emitir señales de salida como una acción reactiva. Las señales emitidas por una neurona alimentan a otras neuronas (a través de conexiones llamadas sinapsis) que a su vez pueden emitir otras señales de salida. Este proceso se puede repetir a través de muchas capas de neuronas y se conoce como un proceso de activación dividido. Como resultado, la información de entrada es procesada y puede llevar al cerebro a ordenar reacciones físicas como respuesta.

Una red neuronal es una representación gráfica muy simplificada de la malla de redes interconectadas en el cerebro humano. Los nodos del grafo son las unidades de procesamiento y los arcos representan las conexiones de sinapsis.

Para la recuperación de información, una red neuronal consta de tres capas: Una para los términos de la consulta, una para los términos del documento, y la tercera para los documentos en si.

[3] LSI es una técnica de recuperación de información basada en el análisis espectral de la matriz de términos de un documento, cuyo éxito implícito no ha tenido una predicción ni explicación rigurosas. Bajo ciertas condiciones, LSI mejora el rendimiento en la recuperación al capturar las semánticas subyacentes de los documentos. [34]

Figura 6. Modelo de Redes Neuronales.



El proceso de inferencia lo inician los nodos de los términos de la consulta, enviando señales a los nodos de los términos de los documentos. Estos nodos a su vez pueden generar señales para los nodos de los documentos. La red neuronal no para después de la primera fase de propagación, ya que los nodos de los documentos pueden generar nuevas señales que se dirigen de regreso a los nodos de los términos de los documentos. Estos nodos con este estímulo pueden enviar nuevas señales repitiendo el proceso. Las señales se van haciendo más débiles en cada iteración y el proceso de activación dividido finalmente se detiene.

Los documentos activados, no necesariamente contienen términos de la consulta, pueden haber sido activados por contener términos relacionados, por esto, el proceso completo puede ser interpretado como la activación de un Tesauro[4] en construcción.

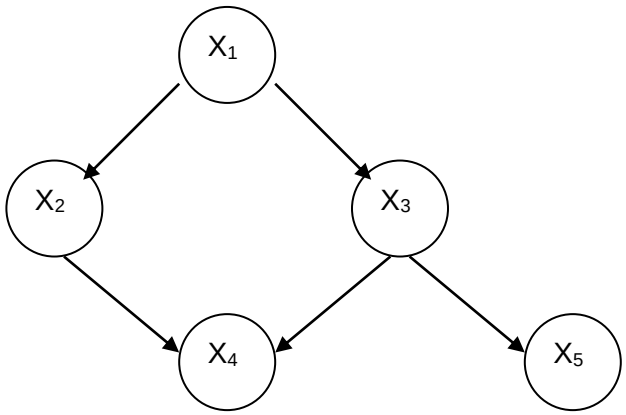
1.1.3.7 Modelos Probabilísticos alternativos. Una opción para cuantificar la relevancia de los documentos es el uso de la teoría de la probabilidad y sus diferentes ramas. Una de esas ramas es la concerniente a las Redes Bayesianas de Creencia (Belief networks), que combinan distintas fuentes de evidencia (consultas anteriores, ciclos de retroalimentación pasados, y la formulación de distintas consultas) como soporte a la calificación de un documento dado.

Esta combinación de diferentes fuentes de evidencia pueden utilizarse para mejorar el rendimiento de la recuperación (mejorar la calidad de la lista rankeada de documentos recuperados).

[4] Lista precompilada de palabras importantes en un dominio de conocimiento específico, y por cada palabra de esta lista, un conjunto de términos relacionados.

1. Redes Bayesianas: Son grafos acíclicos dirigidos en el cual los nodos representan variables aleatorias, los arcos, relación causal entre esas variables y la fuerza de esas influencias causales se expresan por medio de probabilidades condicionales. Los padres de un nodo, son considerados las directas causas para este. Esta relación causal se representa como un enlace dirigido de cada nodo padre al nodo hijo. Las raíces de la red son los nodos sin padres.

Figura 7. Ejemplo de Red Bayesiana.



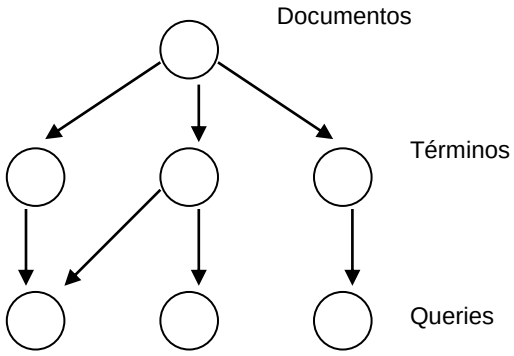
La figura 7 ilustra una red Bayesiana para una distribución de probabilidad de conjunto $P(x_1, x_2, x_3, x_4, x_5)$. Así:

$$P(x_1, x_2, x_3, x_4, x_5) = P(x_1)\, P(x_2|\, x_1)\, P(x_3|\, x_1)\, P(x_4|\, x_2, x_3)\, P(x_5|\, x_3)$$

La probabilidad $P(x_1)$ es llamada la probabilidad principal para la red y puede ser usada para modelar el conocimiento previo sobre las semánticas de la aplicación.

2. Modelo de Redes de Inferencia (Inference Network): Este modelo toma una vista epistemológica del problema de la recuperación de información, asociando variables aleatorias con los términos del índice, los documentos y las consultas del usuario. Una variable aleatoria asociada con un documento d_j representa el evento de la observación de ese documento (el modelo asume que los documentos son observados en la búsqueda de documentos relevantes. La observación del documento d_j afirma una creencia sobre las variables aleatorias asociadas con sus términos indexados. Así la observación de un documento es la causa para una creencia incrementada en las variables asociadas con sus términos indexados.

Figura 8. Modelo de Red de Inferencia.



1.2 FILTRADO DE INFORMACIÓN [9]

El término filtrado de información es usado frecuentemente para describir procesos que involucran la entrega de información a las personas que la necesitan, en aplicaciones como el correo electrónico o sistemas de distribución de multimedia, entre otros, pero no existe una clara diferenciación con otros procesos relacionados como recuperación, direccionamiento, categorización y extracción. Para definir lo que es el filtrado de información, se enuncian las siguientes características:

- Un sistema de filtrado de información es un sistema diseñado para datos no estructurados o semiestructurados, en contraste con una aplicación de base de datos.

- Generalmente manejan información textual, de hecho, hablar de datos no estructurados es usado generalmente como sinónimo de datos textuales. Sin embargo, pueden manejar otros tipos de datos como imágenes, voz y video, como parte de los sistemas de información multimedia.

- Los sistemas de filtrado involucran grandes cantidades de datos.

- Los sistemas de filtrado manejan cadenas entrantes de datos, difundidas por fuentes remotas o enviadas directamente por otras fuentes. El filtrado también se ha empleado para describir procesos de acceso y recuperación de información de bases de datos remotas, en este caso, los datos entrantes son el resultado de búsquedas. Estas tareas han permitido a los desarrolladores de sistemas generar “Agentes Inteligentes” para la búsqueda en bases de datos remotas y heterogéneas.

- El filtrado se basa en la descripción de las preferencias de un individuo o un grupo, comúnmente llamada perfil, que representan generalmente intereses a largo plazo.

- El filtrado puede trabajar con lo que queda después de remover datos, como los sistemas que eliminan basura. Por tanto, los perfiles pueden expresar no solo lo que la gente quiere, sino también lo que no quiere.

Muchas de estas características sin embargo, son virtualmente las mismas de otros sistemas de información basados en texto.

Un sistema de Recuperación de Información tiene la función de "presentarle al usuario aquellos documentos que mejor satisfagan sus necesidades de información". Generalmente, el objetivo de un sistema de recuperación es proporcionarle al usuario información de un recurso de conocimiento, que le ayudará al usuario en la administración de problemas.

Las principales diferencias entre filtrado y recuperación de información son:

- La recuperación de información se ocupa de usos simples del sistema, por una persona con un objetivo y búsqueda que se realiza una vez, el filtrado de información hace usos repetidos del sistema por persona o personas con objetivos o intereses a largo plazo.

- La recuperación de información se ocupa de la colección y organización de textos, el filtrado en cambio, trabaja en la distribución de textos a grupos o individuos.

- La recuperación de información se ocupa típicamente de la selección de texto de una base de datos relativamente estática, el filtrado trata principalmente con la elección o eliminación de textos de cadenas de datos dinámicas.

Además de estas distinciones basadas en los modelos de recuperación y filtrado de información, existen otras diferencias contextuales que pueden ser relevantes para propósitos de investigación.

- Diferencias relacionadas con los textos: Para el filtrado de información, la oportunidad de un texto es con frecuencia de marcada importancia, para la recuperación de información no.

- Diferencias relacionadas con el usuario: Recuperación de información tiene grupos de usuarios estudiados, muy bien definidos, en dominios específicos igualmente bien definidos, especialmente en ciencia y tecnología. Estos

usuarios en su mayoría han sido altamente motivados en su comportamiento en la búsqueda de información. Sin embargo, el filtrado frecuentemente se ocupa de comunidades de usuarios indefinidos, tales como personas buscando entretenimiento desde sus hogares y con dominios altamente variados. Además la motivación no siempre se tiene en cuenta.

- Diferencias ambientales: Tal vez la diferencia más significativa, tiene que ver con el hecho de que el filtrado en muchas situaciones se ocupa en alto grado de la privacidad; la recuperación de información, por varias razones, no ha puesto mucha atención a este problema.

Finalmente, el filtrado de información involucra muchos aspectos económicos y sociales, asociados con la producción y distribución de textos, que han sido de relativo poco interés para la recuperación de información.

1.3 SISTEMAS DE RECOMENDACIÓN [11]

Dada la dificultad para tomar decisiones cuando no se tiene la suficiente experiencia personal en un tópico, es frecuente confiar en recomendaciones hechas por otras personas, en forma oral, en cartas de recomendación, en revisiones hechas de libros o películas que aparecen impresas en los periódicos, o publicaciones generales como por ejemplo una guía de restaurantes.

Los sistemas de recomendación asisten y aumentan este natural proceso social. En un sistema de recomendación típico, las personas proveen las recomendaciones como entradas que el sistema agrega y direcciona a recipientes apropiados. El valor del sistema radica en hacer buenos apareamientos entre los recomendadores y quienes están buscando recomendaciones.

Los desarrolladores del primer sistema de recomendación, Tapestry [12], acuñaron la frase “Filtrado Colaborativo” y muchos otros lo han adoptado.

Tapestry, es un sistema experimental de correo electrónico, desarrollado por Xerox en el Centro de Investigaciones en Palo Alto. La motivación de Tapestry viene con el creciente uso del correo electrónico y la búsqueda de un mecanismo de filtrado que evite el gran volumen de documentos no deseados que se reciben a diario.

Los sistemas de recomendación también tienen algunas implicaciones sociales, que tienen que ver con algunos problemas de incentivos [11]. Primero, una vez que se ha establecido un perfil de interés, es sencillo navegar libremente consumiendo las evaluaciones provistas por otros. Pero el problema no está enteramente resuelto, aun si las evaluaciones se obtienen implícitamente de los recursos existentes o monitoreando el comportamiento del usuario. Los sistemas futuros probablemente necesitarán ofrecer algún incentivo para la provisión de

recomendaciones, haciéndolo un pre-requisito para recibir recomendaciones o mediante el ofrecimiento de alguna compensación monetaria. Segundo, si cualquiera pueda proveer recomendaciones, los dueños de los contenidos, pueden generar montañas de recomendaciones positivas para sus materiales y negativas para sus competidores. Probablemente, los sistemas futuros tomarán precauciones para desalentar este fenómeno ("vote early and often").

Otro aspecto que ocupa a los sistemas de recomendación es la privacidad. En general, entre más información individual se tenga sobre las recomendaciones, mejor será la evaluación que se haga sobre las mismas. Sin embargo, las personas pueden no desear que sus hábitos o gustos sean ampliamente conocidos. Algunos sistemas de recomendación permiten la participación anónima o bajo un seudónimo, pero esto no es una solución completa, puesto que algunos usuarios pueden desear una combinación entre privacidad y reconocimiento por sus esfuerzos.

GroupLens Research[5] es un grupo de Investigación del Departamento de Ciencias Computacionales e Ingeniería de la Universidad de Minnesota, cuyos miembros están involucrados en numerosos proyectos relacionados con el campo del filtrado de información, filtrado colaborativo y sistemas de recomendación. El grupo, liderado por los profesores Jhon Riedl y Joseph Konstan, inició su exploración en el filtrado colaborativo automático desde 1.992 y ha expandido su alcance para integrar métodos basados en contenido, así como para mejorar la tecnología actual en filtrado colaborativo.

MovieLens[6] es un sistema de recomendación para películas, basado en tecnología GroupLens. No es solamente un servicio para quienes se interesan en películas, sino también una fuente de datos experimental y un marco de trabajo para investigaciones relacionadas con sistemas de recomendación. El conjunto de datos disponible[7] consiste de 100.000 ratings dados por 943 usuarios con 1682 películas. Para probar los algoritmos que se discutirán en los capítulos siguientes, se utilizará la base de datos de EachMovie[8].

Una aplicación de los sistemas de recomendación y especialmente de los de filtrado de información, ampliamente difundida, especialmente utilizada por los sistemas bibliográficos de universidades, centros de investigación o de centros de documentación es la Diseminación Selectiva de información, que se describe a continuación. Esta tarea se realiza mediante el servicio de alerta (entrega periódica de tablas de contenido de publicaciones periódicas) o mediante listados de bases de datos (bibliografías).

---

[5] http://www.grouplens.org/
[6] http://www.movielens.umn.edu
[7] http://www.grouplens.org/data/
[8] http://www.research.compaq.com/SCR/eachmovie/

1.4 DSI. DISEMINACIÓN SELECTIVA DE INFORMACIÓN

La diseminación selectiva de información es un procedimiento mediante el cual se suministra periódicamente a cada usuario o grupo de usuarios las referencias de los documentos que corresponden a sus intereses cognoscitivos. Este servicio puede proporcionarse a un usuario particular, basándose en un perfil de recuperación de información individual, o bien, a un conjunto de personas que tienen intereses comunes, con un perfil colectivo definido. La eficacia de este servicio depende especialmente, de una adecuada definición del perfil del usuario mediante la selección de un conjunto de palabras clave que reflejen la temática en cuestión.

El término "diseminación selectiva de información" tiene más de cuatro décadas de existencia, ya que fue acuñado en 1958 por Hans Peter Luhn, miembro de la entonces *Advance Systems Development Division* de IBM, quien definió el término en su ensayo "*Business Intelligence System*" de la siguiente manera: "Aquel servicio dentro de una organización dedicado a la canalización de nuevos elementos (ítems) de información, a partir de cualquier fuente, hasta aquellos puntos dentro de la organización donde la probabilidad de su utilidad en el trabajo diario o interés sea elevada". [35]

Por su parte, la ISO (Organización Internacional para la Normalización) define DSI como "un proceso activo que comunica regularmente la nueva información disponible sobre una materia definida por un perfil de búsqueda", entendiendo por perfil de búsqueda "la descripción de una solicitud en un lenguaje documental indicando y definiendo el área de interés de un usuario"[9].

Mediante la canalización de información actualizada y personalizada, el servicio de DSI puede impactar positivamente sobre la toma de decisiones en trabajos individuales o de grupo, al permitir ajustar oportunamente la dirección de una investigación conforme se conozca el avance desarrollado por otros especialistas de la misma materia.

En el mercado se encuentran numerosos servicios de diseminación selectiva de información electrónicos. Existen bases de datos especializadas en tablas de contenido, como Swetscan, Ebsco, Uncover y Current Contents. Ebsco, Uncover (servicio de documentación electrónica de Knight-Ridder7) y Current Contents ofrecen además el servicio DSI personalizado, mediante el cual el usuario inserta su estrategia de búsqueda en el sistema de manera que ésta se actualiza al ritmo

[9] La cita fue extraída por [35] de Sotolongo Aguilar, g.r., "Experiencias en el idict sobre la diseminación selectiva de información a partir de bases de datos extranjeras". En: **Actualidades de Información Científica y Técnica**, Academia de Ciencias de Cuba, 18 (4), agosto 1987, pp. 41-42.

que se actualiza la base. Uncover y Ebsco permiten además que el servicio obedezca a estrategias de búsqueda a partir de temáticas (mediante palabras clave), mientras que Current Contents sólo trabaja a partir de títulos de revista, esto es, Current Contents sólo actualiza al usuario con las tablas de contenido de las publicaciones que al usuario le interesen, mientras que Uncover y Ebsco proveen no sólo las tablas de contenido sino también las citas bibliográficas de los artículos concernientes a la temática de su interés.

La mayoría de las bases de datos del sistema Dialog cuentan con un servicio de diseminación selectiva de información que funciona de la misma manera que Uncover: a partir de una estrategia de búsqueda personalizada, cuyos resultados son referencias bibliográficas o texto completo. La característica común de todos estos servicios es que trabajan en función de un perfil de actualización personalizado**,** pues responden a las necesidades de información específicas de un usuario.

2. MODELOS DE FILTRADO

El filtrado de información trata con la entrega de ítems seleccionados de una colección muy grande que el usuario probablemente encontrará interesante o útil y que puede verse como una tarea de clasificación. Un usuario modelo induce al sistema para que filtre o clasifique los ítems no calificados en una clase positiva *u* (relevantes para el usuario) y una clase negativa $\bar{u}$ (irrelevantes para el usuario) [30]. Los dos modelo básicos de filtrado que se describirán y utilizarán en el desarrollo de este trabajo son: El filtrado basado en Contenido (o cognitivo) y el Filtrado Colaborativo (o social).

2.1 FILTRADO BASADO EN CONTENIDO

El pilar de ésta técnica de filtrado es el análisis del contendido de los ítems que se quieren clasificar. Formalmente un ítem se describe como un vector $X = (x_1, x_2, x_3, \ldots, x_n)$ de n componentes. Estos componente pueden tener atributos binarios, nominales o numéricos y se derivan del contenido de los ítems o de la información acerca de las preferencias de los usuarios. [30]

La tarea ahora es seleccionar una función basada en un conjunto de prueba con m vectores de entrada que pueda clasificar cualquier ítem no calificado, como positivo o negativo y retornar un valor binario o numérico. En este caso se usa un umbral para determinar si el ítem es relevante o no relevante para el usuario.

El sistema de filtrado basado en contenido selecciona ítems basado en la correlación entre el contenido de un ítem y las preferencias del usuario en forma opuesta al filtrado colaborativo que escoge ítems basado en la correlación entre personas con preferencias similares.

El contenido de un documento puede ser representado como un conjunto de términos, que se extraen de dichos documentos, ejecutando una serie de pasos de análisis gramaticales. Primero se remueven todos los “stop words” o palabras no útiles (palabras que ocurren con mucha frecuencia o que no aportan información, como por ejemplo, los artículos). Las palabras restantes se reducen a las raíces, removiendo los sufijos, mediante el algoritmo de Porter[10] (Algoritmo

[10] Algoritmo de Stemming de Porter. Proceso para remover los sufijos de palabras en Inglés, principalmente usado como parte de un proceso de normalización que se realiza generalmente en la configuración de un sistema de Recuperación de Información. Descrito por Martin Porter en 1.980.

de Stemming). Por ejemplo, las palabras “computer”, “computers” y “computing”, se pueden reducir a “comput”.

La remoción de sufijos por medios automáticos [32] es una operación especialmente útil en el campo de la recuperación de información. Generalmente se dispone de una colección de documentos, cada uno descrito por una serie de palabras. Se puede decir que un documento se representa por un vector de palabras o términos. Los términos con una raíz común por lo regular tienen significados similares, por ejemplo:

CONNECT CONNECTED CONNECTING CONNECTION CONNECTIONS	CONNECT

El rendimiento de un sistema de recuperación de información se mejora, si grupos de términos como el anterior, se compactan en un solo término. Esto se consigue eliminando los distintos sufijos: -ED, -ING, -ION, -IONS, dejando solo el término CONNECT. Este proceso reduce el número total de términos en el sistema y por lo tanto, reduce el tamaño y complejidad de los datos, que es una ventaja importante.

El algoritmo de Porter se puede obtener libremente en la página del autor[11], en diferentes versiones: ANSI C, Java, Perl y otras.

La naturaleza de la tarea de remoción de sufijos, varía considerablemente si se utiliza un diccionario de raíces o una lista de sufijos y dependiendo del propósito de la tarea. Se tienen en cuenta dos aspectos principales: El primero es el criterio para remover los sufijos de dos palabras W1 y W2, y producir una sola raíz S, por ejemplo, W1 = “CONNECTION” y W2 = “ CONNECTIONS”, estas dos palabras pueden unirse en un solo término, pero si W1 =”RELATE” y W2 = “RELATIVITY”, no es razonable que se remuevan los sufijos para llevarlos a la raíz S=RELATE, puesto que las dos palabras tienen significados muy diferentes, y S no da suficiente información acerca de la palabra W2.

El siguiente punto es que el uso de listas de sufijos y reglas, no asegura una tasa de exactitud del 100%, por ejemplo, si las palabras SAND y SANDER se reducen a una, muy probablemente la misma regla aplicará para WAND y WANDER, y aquí se produce un error, porque la sílaba –ER de WANDER se trata como un

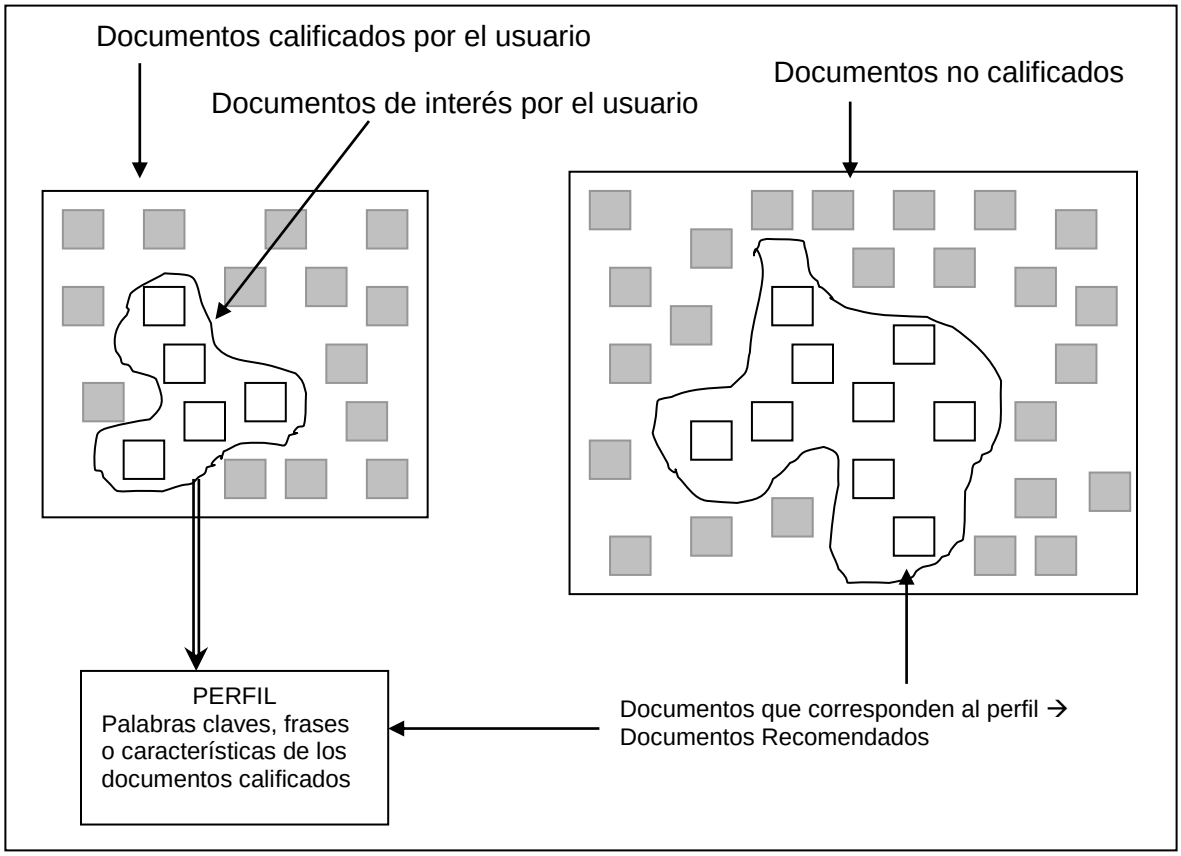
[11] Porter, M. The Porter Stemming Algorithm. http://www.tartarus.org/~martin/PorterStemmer/

sufijo, cuando de hecho es parte de su raíz. Igualmente, un sufijo puede alterar completamente el significado de una palabra.

De la misma forma en que se representa un documento, llevándolo a un conjunto de palabras claves, se opera con los términos que describen las preferencias o áreas de interés de un usuario, que es lo que constituye su perfil.

El perfil del usuario se representa con los mismos términos y se construye analizando los ítems que éste encontró interesantes (Figura 9). Estos ítems se determinan usando retroalimentación implícita o explícita. La retroalimentación explícita requiere que el usuario evalúe los documentos en una escala, mientras la implícita infiere los intereses del usuario, observando sus acciones, lo cual es más conveniente para el usuario, pero más difícil de implementar.

Figura 9. Filtrado Basado en Contenido. El perfil de un usuario se construye del contenido de los documentos que él ha calificado.



Existen muchas maneras de representar los términos, para ser usados como base para el componente de aprendizaje. Un método de representación que es usado

frecuentemente es el modelo del vector espacial. En el numeral 1.1.3.4, se hizo una breve descripción de este método, que se amplía en esta sección. Este modelo, definido por Salton[12] en 1.968, es ampliamente usado en operaciones de Recuperación de información, así como también en operaciones de categorización automática, filtrado de información, etc. [33].

En el modelo vectorial se intenta recoger la relación de cada documento D_i, de una colección de *N* documentos, con el conjunto de las *m* características de la colección. Formalmente un documento puede considerarse como un vector que expresa la relación del documento con cada una de estas características. Es decir, este vector identifica en que grado el documento D_i satisface cada una de las m características. c_{ik} es un valor numérico que expresa en que grado el documento D_i posee la característica *k*. Para este trabajo, el término "característica" es la ocurrencia de determinadas palabras o términos en el documento. Por tanto, el primer paso debe ser seleccionar aquellos términos útiles que permitan discriminar unos documentos de otros. Por supuesto, no todas las palabras contribuyen con la misma importancia en la caracterización del documento. [33]

Desde el punto de vista de la recuperación de información existen palabras casi vacías de contenido semántico, como los artículos, preposiciones o conjunciones, que son poco útiles en el proceso. También son poco importantes aquellas palabras que por su frecuencia de aparición en toda la colección de documentos pierden su poder de discriminación. En recuperación de información todas ellas forman parte del conjunto de palabras vacías (stop words), que se eliminan en el proceso de indexación.

Además de la eliminación de las palabras vacías, en el proceso se pueden incluir aplicaciones léxicas como lematización o extracción de raíces (como el algoritmo de Porter o de Stemming). Una vez seleccionado el conjunto de términos, es necesario obtener el valor de cada elemento del vector del documento. El caso más simple es utilizar una aproximación binaria, de forma que si en el documento D_i aparece el término *k*, el valor de c_{ik} sería 1 y en caso contrario sería 0.

Pero una palabra puede aparecer más de una vez en el mismo documento y unas palabras pueden ser más significativas que otras, entonces el cálculo del valor de cada término no es tan simple como un valor binario. Por otra parte es importante normalizar los vectores para no privilegiar documentos largos frente a otros documentos menos extensos.

$$\vec{d}_i = \frac{1}{\sqrt{\sum_{j=1}^{m} w_{ij}^2}} (w_{i1}, w_{i2}, ..., w_{im}) \quad (1)$$

[12] Salton, G. Automatic Information Organization and Retrieval. MacGraw-Hill, N.Y. (1.968).

Para determinar que tan representativo es un término en un documento, se calcula el número de veces que aparece en dicho documento, obteniéndose la frecuencia del término en el documento, *tf (term frequency).*

Por otra parte, si la frecuencia de un término en toda la colección de documentos es extremadamente alta, se opta por eliminarlo del conjunto de términos de la colección. Puede decirse que la capacidad de recuperación de un término es inversamente proporcional a su frecuencia en la colección de documentos. Esto es lo que se conoce como *idf (inverse document frequency).*

Así, para calcular el peso de cada elemento del vector que representa el documento se tiene en cuenta la frecuencia inversa del término en la colección, combinándola de alguna forma con la frecuencia del término dentro de cada documento. Normalmente se utiliza el producto simple.

$$w_{ij} = tf_i \bullet idf_j \quad (2)$$

w_{ij} expresa el peso del término *j* en el documento *i*.

$$idf_j = \log \frac{N}{n_j} \quad (3)$$

Donde *N* es el total de documentos en el sistema, n_j el número de documentos en el cual aparece el término indexado k_j, idf_j es la frecuencia inversa del documento *kj*.

El proceso realizado para los documentos también puede aplicarse a las consultas, puesto que también están formadas por términos y por lo tanto también puede verse como un documento más. La consulta se representa por un vector de igual número de elementos que los vectores de los documentos. Cada elemento de ese vector expresa el grado en que cada uno de los términos de la colección representa las necesidades informativas de la persona que hace la consulta.

$$Q \rightarrow \vec{q} = \frac{1}{\sqrt{\sum_{j=1}^{m} p_j^2}} (p_1, p_2, ..., p_m) \quad (4)$$

Para resolver una consulta realizada por un usuario, se establece el grado de semejanza entre el vector consulta y el vector de cada uno de los documentos; aquellos cuyo grado de similitud sea más elevado se ajustarán mejor a las necesidades expresadas en la consulta. No obstante, es el usuario el que debe

decidir la relevancia de los documentos recuperados, siendo esta una característica completamente subjetiva del mismo.

El método más simple de calcular la similitud entre una consulta y un documento, utilizando el modelo vectorial, es realizar el producto escalar de los vectores que los representan. El índice de similitud más utilizado es el coseno del ángulo formado por ambos vectores. [33]

Para una consulta *Q*, el índice de similitud con un documento D_i es:

$$Sim(Q, D_i) = \frac{\sum_{j=1}^{m} p_j d_{ij}}{\sqrt{\sum_{j=1}^{m} p_j^2 \bullet \sum_{j=1}^{m} d_{ij}^2}} \quad (5)$$

Los resultados del cálculo del índice de similitud entre la consulta y todos los documentos permite ordenar los resultados en forma decreciente, para ofrecerle primero los documentos que el sistema considera más similares con la consulta.

2.2 FILTRADO COLABORATIVO

En los últimos años, el filtrado colaborativo automatizado se ha convertido en la técnica más popular para reducir la sobrecarga de información, y con frecuencia se usa como complemento de los sistemas de filtrado de información basados en contenido. El filtrado colaborativo se considera exitoso en internet, como puede verse en sistemas como Amazon.com (la más grande tienda de libros en Internet), CDNow.com (la más grande tienda de CD en la Web) y MovieFinder.com (uno de los sitios de películas más visitado en Internet). [6]

Estas dos técnicas de filtrado (colaborativo y por contenido), usan diferentes tipos de datos para hacer una recomendación. Las herramientas de filtrado por contenido seleccionan la información correcta para los usuarios correctos, comparando la representación de los contenidos de los documentos con la representación de los intereses del usuario. Estas herramientas son efectivas en la localización de documentos textuales relevantes en un tópico usando técnicas como Vector–Espacio, Agentes “Inteligentes” y Visualización de Información.

- Vector-Espacio: Es la estrategia más comúnmente usada, en la que los documentos y las consultas son llevados a un vector de términos. Cada dimensión del vector representa el tf-idf (term frequency-inverse document frequency) para un término. Los documentos son calificados de acuerdo a la

cercanía con los términos de la consulta. Esta cercanía se determina mediante el cálculo de la similaridad entre los dos vectores, que generalmente se obtiene como el coseno del ángulo entre los dos. (Numeral 2.1)

- Agentes "inteligentes": Los agentes son programas de computador que emplean técnicas de Inteligencia Artificial para prestar asistencia activa a un usuario con tareas basadas en computador. El agente adquiere sus competencias aprendiendo del usuario así como de otros agentes que asisten a otros usuarios. Los agentes pueden asistir a los usuarios de diferentes maneras: Ejecutando tareas en favor del usuario, entrenando o enseñando al usuario, enseñando a diferentes usuarios que colaboran o monitoreando eventos y procedimientos. El conjunto de tareas o aplicaciones en las que un agente puede asistir a un usuario es virtualmente ilimitada: Filtrado de información, recuperación de información, administración de correos, planeación de reuniones, selección de libros, películas, música, etc. [8]

- Visualización de Información: La idea básica es que el contenido de los textos fuentes pueda ser transformado a una representación espacial que preserve las características de la información de los documentos. La representación espacial puede luego ser desplegada y analizada de manera que se evite el procesamiento de lenguaje y que reduzca la carga de trabajo mental del analista. El resultado es una interacción con el texto que se asemeje más de cerca de la percepción y acción con el mundo natural que con las abstracciones del lenguaje natural. [7]

Los sistemas de filtrado colaborativo trabajan recolectando los juicios humanos (conocidos como ratings) para ítems en un dominio dado y emparejando usuarios que comparten las mismas necesidades de información o los mismos gustos. Así los juicios analíticos de un usuario pueden ayudar a otros usuarios a decidir que consumir. [6]

El filtrado colaborativo tiene tres ventajas adicionales sobre los sistemas basados en contenido, para el filtrado de información:

1. Soporte para el filtrado de ítems cuyo contenido no se pueda analizar fácilmente por procesos automáticos. Los humanos determinan relevancia, calidad e interés de un ítem, por lo tanto, el filtrado puede ejecutarse en ítems que son difíciles de analizar por el computador, como películas, ideas, sentimientos, personas y políticas.

2. La habilidad para filtrar ítems basados en la calidad y gustos, midiendo que tan bien un ítem satisface las necesidades o intereses del usuario, más allá del simple contenido. Una búsqueda basada en contenido podría traer todos los artículos relacionados con un tema en particular, pero combinándolo con

filtrado colaborativo, la búsqueda podría retornar solo aquellos artículos relevantes que están bien escritos.

3. La habilidad para proveer recomendaciones inesperadas, de ítems que son valiosos para el usuario pero que no contienen la información que el usuario estaba esperando.

El potencial del filtrado colaborativo para mejorar las herramientas de filtrado de información es muy grande, sin embargo, para alcanzar este potencial, se debe combinar con la tecnología existente de filtrado de información basado en contenido, ya que por si solo no es muy eficiente para localizar un contenido específico.

En un sistema de Filtrado Colaborativo se distinguen claramente tres pasos [26]:

1. Creación del perfil de usuario (Ratings). Es posible recolectar las preferencias de los usuarios de dos modos: Implícito, por ejemplo al extraer los temas de interés de las páginas revisadas por el usuario, o explícito, como los ratings numéricos (calificaciones a ciertos ítems que agrega el usuario). Cualquier interacción con el sistema, podría potencialmente afectar el resultado de la recomendación. Es útil darle al usuario una retroalimentación sobre la calidad de su perfil, por cuanto los ratings dados por él, tienen mayor peso sobre las recomendaciones que aquellos valores calculados implícitamente. A mayor número de ítems calificados por el usuario, mayor es la probabilidad de obtener mejores recomendaciones. Al entregar los resultados acompañados de una retroalimentación, el usuario podría mejorar su perfil de forma explícita para buscar mejores recomendaciones en el futuro.

2. El Filtrado Colaborativo localiza personas con perfiles similares (vecinos). Es aquí donde el sistema muestra su verdadero valor sobre las recomendaciones hechas de viva voz por las personas, al permitir examinar miles de candidatos con diferentes preferencias y escoger de entre todo el conjunto los que sean más semejantes al usuario actual. Si los vecinos seleccionados por el sistema son los mejores para dicho usuario, de acuerdo a sus necesidades, las recomendaciones resultantes serán las mejores posibles.

3. Los ratings de los vecinos se combinan para formar recomendaciones. En la mayoría de los casos, la predicción es simplemente un promedio ponderado de los ratings de usuarios. Aunque sería más útil, si cada usuario pudiera saber que tan “cercanos“ o que tan “buenos” son sus vecinos, en términos de la cantidad y distribución de sus ratings.

El filtrado colaborativo aplica la velocidad de los computadores con la inteligencia de los humanos, pero presenta los siguientes inconvenientes: [20]

1. El problema del estado inicial (Early rater problem): El filtrado colaborativo no puede hacer predicciones para un ítem que acabe de entrar al sistema, porque ningún usuario lo ha calificado. Además, las predicciones que se hacen sobre ítems nuevos (con muy pocas calificaciones) son muy inexactas. En forma similar, las predicciones para usuarios nuevos también son muy pobres. Este problema también es conocido como "Cold Start problem"[13] o problema del arranque en frío.

2. Problema de los datos dispersos (Sparsity problem): En muchos dominios de información, el número de ítems excede en gran medida la cantidad que un individuo puede asimilar, por lo tanto, las matrices que contienen todos los ratings de usuarios para todos los ítems, son muy dispersas. Esto dificulta encontrar ítems que hayan sido calificados por suficientes personas para realizar las predicciones colaborativas.

3. El problema de la Oveja negra (Gray sheep): En una comunidad de usuarios pequeña o mediana, hay usuarios que no se benefician de un sistema de filtrado colaborativo puro, porque sus opiniones no coinciden con la mayoría del grupo de personas, por lo tanto no se encontrarán usuarios similares.

El filtrado basado en contenido no presenta estos inconvenientes, debido a que recomienda ítems basados en el perfil del usuario, construido analizando el contenido de los ítems que este ha calificado. Sin embargo, también puede ser poco efectivo, porque tiene la dificultad para distinguir entre información de baja y alta calidad, que pertenezcan al mismo tema, además, el número de ítems de cada tópico es cada vez mayor, decrementando la efectividad de estos sistemas.

2.2.1 Algoritmos de Filtrado Colaborativo basados en Memoria. [1] Estos algoritmos utilizan la base de datos de usuarios e ítems completa, para generar una predicción. Mediante técnicas estadísticas, estos sistemas buscan un conjunto de usuarios, conocidos como los *vecinos*, que tienen una historia de similitudes con el usuario en particular (usuarios que hayan calificado diferentes ítems de forma similar o tengan las mismas tendencias de compra de ítems).

Una vez se ha formado el *vecindario* del usuario, estos sistemas usan diferentes algoritmos para combinar las preferencias de los vecinos para producir una predicción o una lista de los n recomendados para el usuario activo. Estas técnicas también conocidas como el vecino más cercano (nearest-neighbor) o filtrado colaborativo basado en usuario, son muy populares y ampliamente usadas en la práctica.

[13] A. Schein, A. Popescul, L. Ungar y D. Pennock. Methods and Metrics for Cold-Start Recommendations. (SIGIR 2002).

La base de datos de usuarios consiste de un conjunto de votos o calificaciones $V_{i,j}$ correspondiente el voto del usuario *i* para el ítem *j*. Si I_i es el conjunto de ítems que el usuario i ha calificado, se calcula la media de las calificaciones como: [2]

$$\bar{v}_a = \frac{1}{|I_a|}\sum_{j \in I_a} v_{a,j} \qquad (6)$$

Para calcular la calificación o el voto que un usuario *a* le daría a un ítem *j* ($P_{a,j}$) se utiliza la suma ponderada de otros usuarios.

$$P_{a,j} = \bar{v}_a + k\sum_{i=1}^{n} w(a,j)(v_{i,j} - \bar{v}_i) \qquad (7)$$

Donde n es el número de usuarios en la base de datos de filtrado colaborativo con pesos diferentes de cero. Los pesos *w(a,i)* pueden reflejar distancia, correlación o similaridad entre cada usuario i y el usuario activo. *k* es un factor de normalización tal que el valor absoluto de los pesos sume la unidad.

Correlación: Esta formulación general aparece en la literatura publicada en el contexto del proyecto GroupLens, donde el coeficiente de correlación de Pearson fue definido como la base para los pesos.

Los pesos con los que se trabajará en el algoritmo de filtrado colaborativo, corresponden a la correlación entre usuarios, y en lo sucesivo se notará como *corr(a,i)*.

La correlación para el usuario *a* y el usuario *i* es:

$$corr(a,i) = \frac{\sum_j (v_{a,j} - \bar{v}_a)(v_{i,j} - \bar{v}_i)}{\sqrt{\sum_j (v_{a,j} - \bar{v}_a)^2 \sum_j (v_{i,j} - \bar{v}_i)^2}} \qquad (8)$$

Las sumatorias sobre *j* son sobre los ítems para los cuales tanto el usuario a como el usuario i dieron su calificación o votación.

Vector Similaridad: En el campo de la recuperación de información, generalmente se mide la similitud entre dos documentos, tratando cada uno como un vector de frecuencias de palabras y calculando el coseno del ángulo formado por los dos vectores. Para el filtrado colaborativo, se puede adoptar este formalismo, dando al usuario el papel del documento, los títulos serian las palabras y las calificaciones serian las frecuencias de las palabras.

2.2.2 Algoritmos de Filtrado Colaborativo basados en Modelo.[1] Los algoritmos en esta categoría, primero desarrollan un modelo de las calificaciones de usuario, tomando una aproximación probabilística y hallando el filtrado colaborativo como el cálculo del valor esperado de una predicción, dadas las calificaciones dadas por el usuario a otros ítems.

Para el usuario activo, el sistema intenta predecir las calificaciones para ítems no observados aún. Asumiendo que las calificaciones o votos son valores enteros en un rango de 0 a m, se plantea la siguiente operación para el cálculo de la probabilidad de que el usuario activo *a* obtenga un valor de voto particular para el ítem *j*, dados los votos observados previamente. [2]

$$P_{a,j} = E(v_{a,j}) = \sum_{i=0}^{m} \Pr(v_{a,j} = i \mid v_{a,k}, k \in I_a)\, i \qquad (9)$$

El proceso de construcción del modelo es ejecutado por diferentes algoritmos de autoaprendizaje (machine learning) tales como: Redes Bayesianas, Agrupamiento (clustering) y basado en reglas.

2.2.2.1 Redes Bayesianas. [2] Formula un modelo probabilístico, con cada nodo de la red correspondiente a un ítem en el dominio y los arcos representando la información del usuario. Los estados de cada nodo corresponden al posible voto para cada ítem. El algoritmo de aprendizaje busca sobre varios modelos de estructuras en términos de dependencias para cada ítem. En la red resultante, cada ítem tendrá un conjunto de ítems padres que son las mejores predicciones para sus votos. Cada tabla de probabilidad condicional está representada por un árbol de decisión codificando la probabilidad condicional para ese nodo. Un ejemplo de estos árboles se muestra en la figura 9, para datos de audiencia de los programas de televisión.

El modelo puede ser construido fuera de línea y puede durar horas o días. El modelo resultante es muy pequeño, muy rápido y tan exacto como el método de los vecinos más cercanos. Este modelo puede ser práctico para ambientes en los que el conocimiento de las preferencias de usuario cambia lentamente con respecto al tiempo necesario para construir el modelo, pero no para ambientes en los que los modelos de preferencias deban actualizarse rápidamente o frecuentemente.

2.2.2.2 Agrupamiento o Clustering. [1]. Este modelo identifica grupos de usuarios que muestren preferencias similares. Una vez creados los grupos, las predicciones para un individuo se pueden obtener promediando la opinión de los otros usuarios pertenecientes a su grupo. Esta técnica generalmente produce menos recomendaciones individuales y su nivel de exactitud es inferior al de los algoritmos que usan los vecinos más cercanos. El modelo calcula la probabilidad

de que un usuario esté en una clase o grupo C y a partir de esto calcula la probabilidad condicional de las calificaciones. La figura 10 muestra un ejemplo para este modelo.

Figura 10. El árbol de decisión muestra si el programa de televisión “Melrose Place” será visto, teniendo como padres los programas “Friends” y “Beverly Hills”. Las barras en la parte inferior del árbol indican la probabilidad de que el programa “Melrose Place” sea o no visto, condicionado a que los programas padres sean vistos.

2.2.2.3 Aproximación basada en reglas. Aplica algoritmos de descubrimiento de reglas de asociación para encontrar la relación entre los ítems que han sido seleccionados simultáneamente y luego genera la recomendación de ítems basado en la fuerza de asociación entre los ítems. Esencialmente, estas técnicas tratan con el descubrimiento de asociación entre dos conjuntos de productos (ítems), tales que la presencia de un producto en una transacción particular implica que los productos del otro conjunto también están presentes en la misma transacción. [3]

3. INTEGRACIÓN DE FILTRADO COLABORATIVO Y POR CONTENIDO

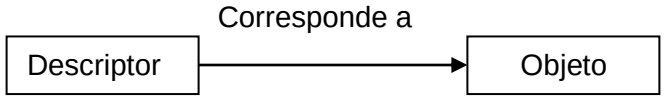
Como se describió en los numerales 2.1 y 2.2, los dos modelos de filtrado más ampliamente estudiados y utilizados, tienes ventajas que pueden ser aprovechadas y debilidades que se pueden intentar minimizar al trabajar con una propuesta que integre los dos modelos. En la sección 3.1 se describe la visión general de esta integración y en la sección 3.2, se nombran las ventajas de cada uno de los modelos y su utilización en un sistema híbrido, así como algunos ejemplos que aplican esta propuesta.

3.1 VISIÓN GENERAL DE LA INTEGRACIÓN DE FILTRADO COLABORATIVO Y POR CONTENIDO [15]

Para mostrar un panorama general de las posibilidades en diversos campos de investigación, se presenta un modelo de representación con tablas en un sistema de bases de datos relacional. Como resultado se observan nuevos casos de aplicación y una arquitectura que soporta la formulación de una gama de consultas.

El objetivo es introducir descriptores basados en contenido, en un sistema colaborativo. Las entidades son: Usuario, Descriptor (palabra clave) y Objeto (que pueden ser documentos textuales, multimedia, hipertextos, etc).

Proceso Basado en Contenido:



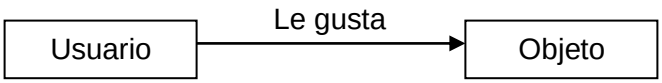
En la siguiente tabla, las filas corresponden a las entradas y las columnas a las salidas.

Tabla 1. Aplicaciones de un Sistema Basado en Contenido.

	Descriptor	**Objeto**
Descriptor	Tesauro Asociativo	Recuperación de Información / Filtrado
Objeto	Indexación	Consulta por ejemplo

En la tabla 1, la columna de la derecha, es el foco de interés.

Proceso Colaborativo:



En la siguiente tabla, las filas corresponden a las entradas y las columnas a las salidas.

Tabla 2. Aplicaciones de un Sistema Colaborativo.

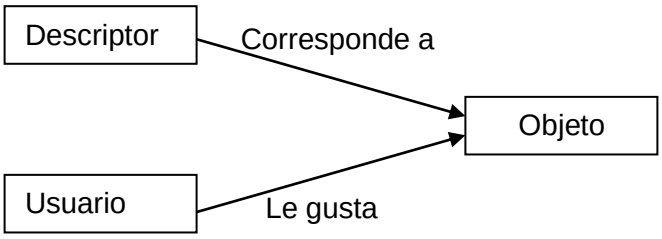
	Usuario	**Objeto**
Usuario	Corresponcia (Matchmaker)	Filtrado colaborativo activo
Objeto	Encontrar expertos	Filtrado colaborativo automático

El filtrado colaborativo activo, maneja los usuarios individualmente, ya sea enviándole recomendaciones o recolectando recomendaciones de ellos.

En el filtrado colaborativo automático (o pasivo), los usuarios llenan sus perfiles de usuario con ratings sobre algunos objetos, y luego este perfil se usa solamente para consultar el sistema.

Uniendo los dos procesos:

Si se tiene un sistema con la información basada en contenido y la información colaborativa como relaciones separadas, se tiene el siguiente modelo:



La tabla 3 une las tablas 1 y 2.

Tabla 3. Posibles consultas en un sistema combinado

	Descriptor	**Objeto**	**Usuario**
Descriptor	Tesauro	Recuperación de Información / Filtrado	Importar perfil, Encontrar expertos
Objeto	Indexación	Consulta por ejemplo	Encontrar expertos
Usuario	Exportar Perfil	Filtrado colaborativo automático	Correspondencia

.

En un sistema combinado, hay tres posibilidades para inferir relaciones objeto – objeto. Puede hacerse vía descriptores (basado en contenido), vía usuarios (colaborativo) o con ambos. Los siguientes son algunos posibles escenarios de aplicación:

- Usuarios de un sistema de recomendación interesados en la recuperación de objetos. Este proceso puede alimentarse con descriptores basados en contenido, una selección de objetos relevantes, una lista de usuarios conocidos con gustos similares o una combinación de estos.

- Aplicaciones de mercadeo interesadas en obtener información acerca de usuarios. Puede consultar el sistema usando palabras claves, objetos (productos), así como con prototipos de perfiles de mercadeo.

- Usuarios interesados en mercadeo. Las funciones de correspondencia han probado ser de especial interés en numerosos servicios de web, tales como friendfinder (www.friendfinder.com), grupos de chat, etc. Los usuarios pueden alimentar el sistema describiendo un perfil deseado o usando su propio perfil para buscar usuarios con intereses similares. Esta misma función puede usarse para buscar expertos en un tópico dado.

- Los usuarios pueden exportar su propio perfil, para de esta manera importarlo en otro sistema de recomendación.

En la siguiente sección se describe una alternativa de integración, donde se trabaja con los dos modelos de filtrado (colaborativo y por contenido), que se conoce como un sistema híbrido, con aplicación a diversos campos de investigación. Esta integración puede realizarse con distintas orientaciones como se observa en los numerales 3.2.1, 3.2.2 y 3.2.3.

3.2 UNA ALTERNATIVA DE INTEGRACIÓN. [14]

Un sistema de recomendación basado en filtrado por contenido puro, recomienda ítems similares a aquellos que el usuario ha dado una alta calificación en el pasado, en cambio, un sistema de recomendación con filtrado colaborativo puro identifica los usuarios con gustos similares, para recomendarle al usuario los ítems que a ellos les han gustado. Un sistema híbrido puede incorporar las ventajas de ambos sistemas y minimizar las desventajas que presenta cada modelo en su forma pura.

Un sistema de recomendación basado en contenido puro, presenta los siguientes problemas:

- Generalmente solo se tiene acceso a un análisis superficial de cierta clase de contenidos, y en algunos dominios de ítems, con la tecnología actual, no hay métodos de extracción muy precisos (como en el caso de películas, música, restaurantes). Aun en documentos de texto, las representaciones capturan solo ciertos aspectos del contenido, y hay muchos otros que podrían influir en la experiencia de un usuario. Por ejemplo, para páginas Web, las técnicas de recuperación de información ignoran completamente cualidades estéticas, toda la información multimedia y factores que afectan a la red, como el tiempo que se requiere para cargar la página.

- Un segundo problema es la sobre-especialización de la información, puesto que el sistema puede solo recomendar ítems calificados altamente, por lo tanto, el usuario está limitado a ver solo ítems similares a los que el ya calificó.

- Finalmente, hay un problema común a la mayoría de los sistemas de recomendación, la reducida retroalimentación por parte de los usuarios. Calificar documentos es una tarea costosa para los usuarios, por lo tanto entre menos ratings se requieran, mejor. En un sistema de recomendación basado en contenido puro, los ratings propios del usuario son el único factor que influenciará el rendimiento futuro del sistema.

Un sistema de recomendación colaborativo, más que calcular la similitud entre ítems, calcula la similitud entre usuarios. Para cada usuario, se busca un conjunto de usuarios conocido como los “vecinos más cercanos”, con quienes se encontró la más alta correlación de acuerdo a los ratings pasados. Los valores para los ítems que aun no han sido vistos por ningún usuario, se predicen basados en la combinación de los ratings conocidos proporcionados por los vecinos más cercanos. Un sistema de recomendación colaborativo puro, presenta algunas dificultades, que se describieron en el numeral 2.2.

En un sistema híbrido, se mantienen los perfiles de usuarios basados en análisis de contenido y compara directamente estos perfiles para determinar usuarios

similares y realizar la recomendación colaborativa. Los usuarios reciben ítems cuando estos son calificados altamente de acuerdo a su perfil o cuando son calificados altamente por usuarios con perfiles similares. Esta aproximación evita las limitaciones mencionadas para las recomendaciones basadas en contenido y las colaborativas puras, mientras adiciona los siguientes beneficios:

- Haciendo recomendaciones colaborativas, se pueden usar las experiencias de otros como una base, por encima de los métodos incompletos e imprecisos de análisis por contenido de los que se dispone.
- Haciendo recomendaciones basadas en contenido, se puede trabajar con ítems que no han sido evaluados por ningún usuario.

- El perfil del usuario se puede construir del contenido de los ítems para hacer buenas recomendaciones a los usuarios, aun si no hay otros usuarios similares a él. También se pueden descartar ítems.

- Se pueden hacer recomendaciones entre usuarios que no hayan calificado ítems en común, extendiendo el alcance de los sistemas colaborativos a bases de datos que cambian rápidamente o que son muy grandes con respecto al tamaño del grupo de usuarios.

- Con la utilización de retroalimentación de grupo, se requiere potencialmente menos ciclos para alcanzar el mismo nivel de personalización.

En los siguientes numerales se observa la utilización de las ventajas de la integración de los modelos básicos de filtrado, intentando minimizar los problemas que presenta cada modelo individualmente.

3.2.1 Integración modelo colaborativo y basado en contenido, orientado por objetivos.[25] Para explicar esta integración, se presenta un ejemplo de aplicación llamado METIOREW [25], que un Sistema de Recomendación para Web, basado en contenido y colaborativo, orientado por objetivos, entendiendo por objetivo la expresión de una necesidad de información.

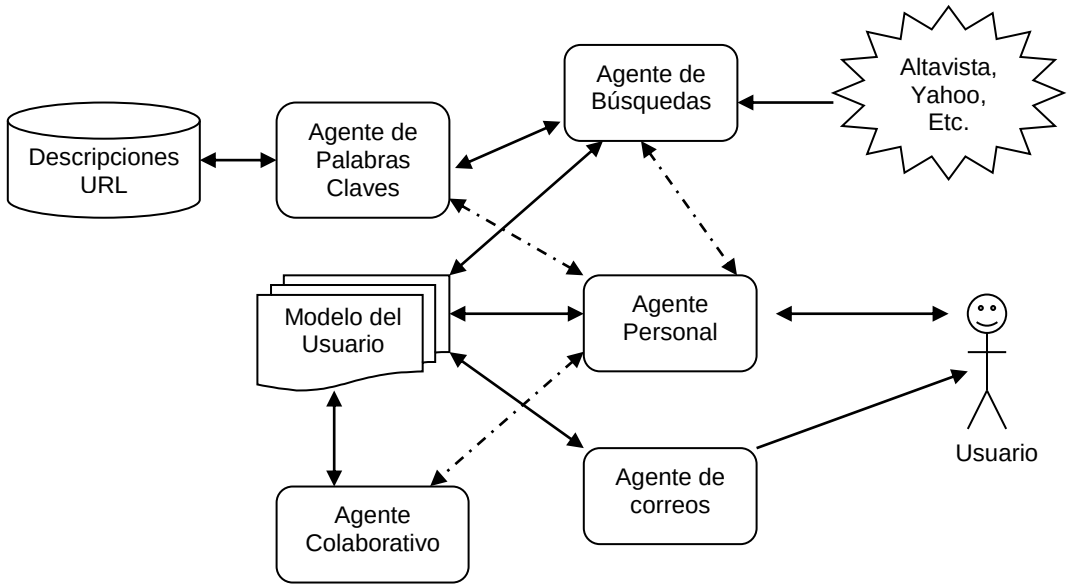
Para resolver el problema de cómo recomendar ítems a un usuario nuevo, se solicita al usuario que ingrese un objetivo (en lenguaje natural, usado como una etiqueta) y una lista de palabras claves que ayudan a crear el modelo inicial para este objetivo. Como el modelo no es aún suficientemente fuerte, el sistema busca otros usuarios similares (inicialmente usando la lista de palabras que ingresó el usuario).

Para el usuario nuevo se manejan dos modelos en paralelo para hacer las recomendaciones: su propio modelo y el modelo más semejante encontrado en el sistema. Este último es usado hasta que el modelo del usuario sea significativo.

3.2.1.1 Arquitectura del Modelo. La meta final de METIOREW es encontrar las páginas web más relevantes para el objetivo actual del usuario. Las páginas se obtienen de los motores de búsqueda en web, navegación supervisada y de recuperación colaborativa.

En los siguientes párrafos se explica la arquitectura del sistema mostrada en la figura 11. Para el desarrollo de METIOREW se adoptó una propuesta basada en agentes con objetivos específicos y que comparten información.

Figura 11. Arquitectura de METIOREW.



Agente Personal. El agente personal controla la identificación del usuario, administra los objetivos, supervisa la navegación, lleva la historia de las actividades, realiza la generación de recomendaciones y la recepción de retroalimentación del usuario.

Agente de Búsqueda. Este es un agente para un objetivo de cada usuario que busca información en un índice web como Altavista o Yahoo. El agente construye las consultas usando las palabras claves más representativas del modelo de usuario para este objetivo. Obtiene una lista de documentos que guarda para las recomendaciones cuando las necesite el usuario. El agente de palabras claves hace una descripción del contenido de los documentos para conocer la relevancia real de estos para el usuario.

Agente de palabras claves. Este agente recibe una página Web y genera un conjunto de palabras claves que lo describen, usando TF (Term Frequency) y

aplicando algunas heurísticas como “remover las palabras más y menos repetidas”, para dejar las mejores m palabras.

Agente Colaborativo. Su objetivo es ofrecer información relevante al usuario tomando como referencia documentos que han sido evaluados por otros usuarios con objetivos similares. Solo propone los documentos que tengan un grado de similaridad superior a un umbral predefinido. El agente busca los objetivos más similares comparando los modelos y para cada objetivo recupera una lista de páginas que serán ordenadas de acuerdo al grado de relevancia.

Agente de Correos. Existe un agente de correos para cada usuario, que se activa con un temporizador definido por el usuario (por ejemplo una vez al día o una vez a la semana). Su misión es examinar la lista de recomendaciones generada por los agentes de búsqueda y colaborativos y enviar los n mejores punteros para cada objetivo al correo del usuario. Esto le permite al usuario definir varios objetivos, mejorar su modelo en sesiones diferentes, habilitando al sistema para recuperar documentos sin la interacción del usuario.

3.2.1.2 Funcionalidad. Cuando el usuario inicia una sesión con un nuevo objetivo, el agente personal le pide que introduzca una descripción textual de su objetivo actual y una lista inicial de palabras claves que lo describen. Luego los agentes de búsqueda y colaborativo tratan de encontrar páginas relacionadas. El usuario puede empezar a navegar libremente en la Web de una forma supervisada. Si encuentra documentos relevantes puede hacer una retroalimentación que se usa para actualizar el modelo. Cuando el usuario solicita una recomendación, el agente personal busca los nuevos documentos encontrados para este objetivo y se los propone en una lista.

Como el modelo inicial (modelo real) está restringido solamente a una lista de palabras claves, METIOREW trata de mejorarlo usando el modelo de otro usuario (modelo externo) con el objetivo más similar al del usuario actual. Las recomendaciones posibles de los dos modelos se usan para hacer nuevas recomendaciones. Cada retroalimentación relevante sirve para mejorar el modelo real. El modelo externo se usa hasta que el real sea suficientemente independiente (al menos 10 retroalimentaciones positivas). Después de evaluar algunos documentos, el modelo del usuario se refina y consta de las palabras claves iniciales (que tienen un peso importante porque fueron dadas directamente por el usuario) y las nuevas obtenidas de los documentos evaluados por el usuario. Los agentes de búsqueda y colaborativo usan la información del modelo para encontrar nuevos documentos relacionados que se guardan en el repositorio para el usuario, que es consultado por los agentes personal y de correos para generar recomendaciones. Los documentos en el repositorio se ordenan por grado de relevancia para el objetivo.

El modelo del usuario guarda la información necesaria para personalizar la interacción con el usuario. Como el sistema es orientado a objetivos, cada usuario puede tener varios modelos dependiendo de las diferentes necesidades de información, y así trabajar en diversas sesiones con objetivos distintos.

3.2.2 Uso de información social y basada en contenido para hacer recomendaciones. Recomendación como un problema de clasificación [28] Hay muchos factores que pueden influenciar la toma de decisiones de una persona, los sistemas de recomendación idealmente, deben intentar modelar tantos de estos factores como sea posible, a fin de agregarle confiabilidad. En un primer enfoque, el usuario del sistema provee ratings de algunos ítems y el sistema realiza sugerencias de otros ítems que le pueden gustar al usuario, basado en los ratings de otros usuarios. Este es el método del filtrado social. En el segundo enfoque, el sistema acepta información describiendo la naturaleza de un ítem y basado en una muestra de las preferencias del usuario, aprende a predecir cuáles ítems le gustaran al usuario. Este proceso es el filtrado basado en contenido. El objetivo de ambos modelos es aprender una función que pueda tomar una descripción de un usuario y un ítem y predecir las preferencias de un usuario de acuerdo a un ítem.

La propuesta de integración enmarca el filtrado colaborativo como un problema de clasificación más que de ratings de ítems, empleando la información social en el proceso de aprendizaje inductivo (análisis de contenido).

El proceso de recomendación se formaliza como el problema del aprendizaje de una función que tome como entrada un usuario y un ítem y produzca como salida una etiqueta indicando si el ítem le agrada (o si lo recomienda) o si no le agrada.

$$f(\langle usuario, ítem \rangle) \rightarrow \{ le\ agrada, no\ le\ agrada \}$$

Como un problema de clasificación, el interés es saber si el ítem le agrada o no al usuario y no una calificación exacta. La salida del sistema no es una lista ordenada de ítems (por ejemplo películas), sino un conjunto de ítems que el sistema predice que le gustarán al usuario.

Para el sistema de aprendizaje, se puede representar el rating usuario/ítem de varias maneras, la primera, muestra la función colaborativa, la segunda, la función que representa el contenido y la tercera, las características híbridas que forman el sistema de recomendación.

- Funciones Colaborativas: Como una representación inicial, se usan funciones que toman en cuenta separadamente las características del usuario y las del ítem (para el ejemplo, los ítems son películas). Por ejemplo, un grupo de

usuarios identificados por que les gusta la misma película: María, Jhon y Luis les gusta Titanic. El sistema agrupa a estos usuarios en una sola función llamada ”Usuarios a los que le gusta la película X”. Por ejemplo, {María, Jhon, Luis}) sería el valor de la función para la película Titanic. Si el conjunto de datos maneja ratings numéricos, entonces se considera que a un usuario le gusta una película si su rating está por encima de un umbral escogidoT_1 (Si la escala es de 0 a 5, T_1 puede ser 4, que es el valor más frecuente).

Otro factor importante es guardar la pista de las películas favoritas del usuario Y, por ejemplo, a María le gustan las películas Twister, Eraser y Face/Off. Para esto se crea un atributo “películas que le gustan al usuario” que codifica esta información. Como resultado cada rating usuario/película se convierte en una tupla de dos funciones con valores de conjuntos. El primer atributo es un conjunto que contiene las películas que le gustan al usuario y el segundo es otro conjunto que contiene los usuarios a los que les gusta la película dada. A cada tupla se le coloca una etiqueta indicando si la película le gusta o no al usuario.

- Función que representa el contenido: Estas funciones están disponibles para realizar aprendizaje, con información semiestructurada, en fuentes como la de IMDb, usada por EachMovie. Las funciones de contenido se obtienen de esta base de datos: Actores, Actrices, Directores, Géneros, Idiomas, Países, etc.

- Funciones Híbridas: Buscan crear funciones adecuadas que generen un aprendizaje exitoso. En lugar de tener las funciones colaborativas y por contenido separadas, el sistema define nuevas funciones colaborativas influenciadas por el contenido, que son las funciones híbridas. Por ejemplo, se aislan grupos de usuarios a los que les gustan ciertos géneros de películas (como “comedias” o “dramas”), en la función “*usuarios a los que les gusta comedias*”. Estas funciones combinan el conocimiento sobre los usuarios a quienes les gusta un conjunto de películas con el conocimiento de algunos contenidos asociados con las películas en el conjunto.

Algunos ejemplos de funciones que combinan colaborativo y por contenido, relativas a los géneros de las películas son (usando los géneros más populares):

“Comedias que le gustan al usuario”
“Dramas que le gustan al usuario”
“Películas de acción que le gustan al usuario”

Una ventaja de esta propuesta inductiva en relación a otros métodos de filtrado social, es su flexibilidad, ya que se puede codificar información colaborativa y por contenido como parte de la representación del problema sin ninguna modificación

algorítmica, y posibilita agregar más representaciones usando contenido para mejorar el rendimiento.

3.2.3. Filtrado de información colaborativo basado en contenido, con el uso de agentes cooperativos. Un grupo de investigación [28], del Departamento de Inteligencia y Ciencias Computacionales, del Instituto de Tecnología de Nagoya, Japón, realizaron una propuesta de filtrado de información colaborativo y basado en contenido, como una manera de aprendizaje activo para clasificar y recomendar documentos, con la utilización de agentes cooperativos que juegan un papel fundamental en la búsqueda y localización de información relevante, en beneficio de los usuarios; donde la relevancia depende de cada usuario y bajo un tópico o dominio particular. Por esto, el desarrollo de sistemas inteligentes, personalizados, basados en contenido, de clasificación de documentos, son cada día más atractivos. La representación de las preferencias de los usuarios para formar su perfil, usado en el filtrado basado en contenido, es una tarea compleja, especialmente porque dependiendo del usuario, los atributos para cada clase pueden cambiar con el tiempo.

Los documentos y los perfiles se representan comúnmente mediante un vector de palabras claves. Tomando un conjunto de ítems calificados correctamente como relevantes o irrelevantes, se pueden identificar las palabras claves importantes para un dominio. Estas palabras o características relevantes (prototipos) se usan para aprender y clasificar documentos.

Por otra parte, los sistemas cooperativos multiagentes comparten información "social" implícitamente, que puede potencialmente ser usada para mejorar las tareas de recuperación de información relevante, así como para refinar el conocimiento particular de cada agente.

El sistema multiagente RAAP (Research Assistant Agent Project) combina las ventajas de los dos modelos de filtrado, por contenido y colaborativo. En RAAP los agentes personales ayudan al usuario (un investigador) a clasificar información de WWW en un dominio específico, y a la vez recomienda estos URLs a otros investigadores con intereses similares.

El sistema fue desarrollado no solo para mostrar que se pueden combinar las dos técnicas de filtrado, sino especialmente, para probar que ambas son necesarias para mejorar los resultados de un proceso completo de aprendizaje. Estos agentes aprenden tanto de recursos locales como compartidos. Su estrategia incluye un proceso de retroalimentación "clasificación-recomendación", en donde el agente sugiere la clasificación de los documentos almacenados y los ratings se extractan de las acciones del usuario tales como aceptación/rechazo de una recomendación, o la revisión de la sugerencia del agente. Por tanto, entre mejor sea la clasificación que hace el agente, mejores serán las recomendaciones. De la misma forma, entre más usuarios utilicen el sistema, habrá más posibilidades de

actualizar el prototipo del usuario para cada dominio, dándole al agente la oportunidad de ser más efectivo en el siguiente paso de la clasificación.

En el siguiente numeral se describen brevemente algunos sistemas existentes y de los que hay literatura disponible.

3.3 EJEMPLOS DE SISTEMAS BASADOS EN CONTENIDO, COLABORATIVOS O HÍBRIDOS

Los siguientes son ejemplos de Sistemas de Recomendación que aplican filtrado colaborativo, o filtrado por contenido o una combinación de los dos.

WebWatcher. [17] Proyecto desarrollado en Carnegie Mellon, consistente en la creación de un agente "guía de recorrido" para el world wide web. Una vez el usuario le comunica al agente que clase de información busca, el asistente lo acompaña de página en página, mientras consulta en la web, resaltando hipervínculos que considera de interés. Esta estrategia para dar recomendaciones es aprendida de retroalimentaciones de recorridos anteriores, basado solo en contenido. Los objetivos de búsqueda están restringidos a reportes técnicos y las palabras claves pueden ser autor, título, etc.

Letizia. [18] Otro sistema basado en contenido para la recomendación de documentos web. El usuario no necesita introducir información acerca de sus intereses, Letizia supervisa sus acciones y usa algunas heurísticas para determinar qué es interesante para ese usuario. Por ejemplo, si un usuario hace un "bookmark" de un documento (guarda la dirección electrónica del documento), esto significa que es de su interés. Otra heurística menos fuerte es que si el usuario analiza los "links" (enlaces) del documento, muy probablemente está relacionado con sus necesidades de información. Los documentos se representan como listas de palabras claves.

Syskill & Webert. [19] Usando recomendaciones basadas en contenido, este sistema predefine algunos tópicos que pueden ser objetivos posibles de un usuario. El sistema posee un índice para cada tópico que se crea manualmente, así cuando el usuario evalúe algunos documentos de este índice, el sistema puede recomendar las páginas más relacionadas con las ya evaluadas. El algoritmo para seleccionar documentos relevantes es un clasificador Bayesiano.

FAB. [14] Sistema de recomendación adaptativo colaborativo para la web. Tiene diferentes clases de agentes: Agentes de colección (buscan nueva información de un número de tópicos limitado), agentes de selección (uno por cada usuario que tenga su modelo para recomendarle los documentos más interesantes) y un enrutador central (envía páginas que obtiene de los agentes de colección y selección de usuarios con un perfil similar al contenido de las páginas). El usuario recibe regularmente un listado de páginas para evaluar. Esta información se

utiliza para actualizar el agente de colección original (que no está ligado al usuario) y su agente de selección. Este agente usa TF-IDF (term frequency – inverse document frequency) para obtener las palabras claves del documento y la medida del coseno, para calcular la similaridad entre el perfil del usuario y un documento. Los documentos mejor evaluados se envían a otros usuarios con perfiles similares.

PTV. [21] Este sistema recomienda programas de televisión a través de WWW y WAP. Guarda un perfil de usuario compuesto de canales, palabras claves, programas, etc., que son de su interés. El sistema selecciona los k usuarios más parecidos al usuario actual y le ofrece los r mejores programas. Cuando un usuario solicita una recomendación, el sistema le muestra algunos de los seleccionados de los r programas y otros encontrados de recomendaciones por contenido.

MOVIELENS. [22] Recomienda películas usando información de otros usuarios con preferencias similares y basado en evaluaciones previas hechas por otros usuarios. Movielens usa diferentes agentes para recolectar información usando diferentes métodos y combinándolos para obtener mejores resultados. Los desarrolladores del sistema comparan los resultados al usar solo información basada en contenido o usando combinación de varios agentes, y concluyen que la mejor solución es la mezcla de varios agentes y la información basada en la retroalimentación del usuario.

Casper. [23] Es una herramienta que ayuda a encontrar empleo, funciona haciendo razonamiento basado en casos. El sistema evalúa cada nuevo trabajo, comparándolo con otros trabajos ya evaluados y lo propone si es el más similar a alguno que le haya interesado al usuario. Utiliza una métrica de suma ponderada para calcular la similaridad y usa como características, la clase de trabajo, el salario, experiencia, etc. Casper es también colaborativo ya que hace recomendaciones de usuarios similares, donde el cálculo de la similaridad se hace evaluando el número de diferentes trabajos que han evaluado en común.

WebCobra. [24] Es también un sistema de recomendación donde inicialmente el usuario evalúa un conjunto de documentos de donde se extrae un vector de palabras claves que será usado para identificar a este usuario. Este vector es enviado a un servidor que usa el método del coseno simple para calcular la similaridad y asigna un usuario a un grupo. Cuando el usuario evalúa otros documentos, selecciona cuales de ellos son los mejores para enviárselos a sus compañeros de grupo. Los temas para los grupos se concentran en tópicos muy específicos. El usuario puede pedir recomendaciones y recibirá los documentos marcados como interesantes por otros miembros de grupo.

4. IMPLEMENTACIÓN DE MODELOS DE INTEGRACIÓN

4.1 HERRAMIENTAS UTILIZADAS.

4.1.1 Base de datos de EachMovie [4]. El Centro de Investigación de Compaq recopiló una serie de información de películas, y corrió el servicio de recomendación por espacio de 18 meses, para experimentar con el sistema de filtrado colaborativo. Durante este tiempo, 72.916 usuarios dieron 2'811.983 calificaciones (ratings numéricos) para 1.628 películas diferentes (filmes y videos). Este conjunto de datos está disponible para que pueda ser usado por investigadores en el área de filtrado colaborativo, para probar sus algoritmos.

El conjunto de datos está disponible como eachmoviedata.tar.gz (archivos de texto comprimido). Hay tres tablas, una por archivo:

- Personas (person.txt), opcional, contiene datos dados por las personas participantes.

- Películas (movie.txt), provee información descriptiva sobre cada película, como por ejemplo, ID, que es la llave primaria (numérica), el Nombre (texto), IMDb_URL, que es el URL[14] de la Base de datos de películas de Internet, entre otros. Los URL de IMDb son provistos por cortesía de Internet Movie DataBase.

- Votos (voto.txt), contiene los datos de los ratings dados por los usuarios. Contiene los siguientes datos entre otros: Person_ID (numérico), Movie_ID (numérico), Score (numérico, entre 0 y 1), weigth (numérico, entre 0 y 1), fecha y hora de modificación.

Score es la calificación (rating) dado por una persona a una película en particular. La interfaz del usuario le muestra 0,1,2,3,4 o 5 estrellas, que corresponden a una calificación con los valores 0, 0.2, 0.4, 0.6, 0.8, 1.0 respectivamente.

Para tener acceso a los datos, es necesario aceptar los términos legales de uso y solicitarlos al centro de Investigación en Sistemas de Compaq.

4.1.2 Java y Oracle. Para la programación de los algoritmos y el manejo de los datos, se utilizaron dos herramientas ampliamente usadas, por su reconocida eficiencia y su disponibilidad en la universidad.

[14] Uniform Resource Locator. Un localizador de recursos uniformes es una representación compacta de la localización y del método de acceso a un recurso disponible vía internet.

- **JAVA:** el sistema de Programación de Java, de creciente aceptación mundial, desarrollado por Sun Microsystems, debe su popularidad principalmente a la programación orientada a objetos, la independencia de la plataforma y las posibilidades de trabajo en red. Algunas de las características más importantes son: orientación a objetos, compatibilidad, portabilidad, amigable para el trabajo en red, seguridad, entre otras.

- **ORACLE:** es la base de datos más extensamente usada en el mundo, puede ejecutarse en cualquier tipo de máquina, desde PC hasta grandes mainframes. Para acceder a Oracle desde Java, existen dos APIs[15]: JDBC y SQLJ. JDBC es un conjunto de clases e interfaces escritas en Java para permitir a otros programas Java enviar sentencias SQL a un sistema manejador de bases de datos como Oracle. SQLJ es un nuevo estándar para introducir sentencias SQL dentro de programas Java.

4.2 MÉTRICAS [6]

Se consideran tres dimensiones claves que pueden medir la calidad de un algoritmo de predicción.

1. Cubrimiento [38]: Es una medida del porcentaje de ítems para los cuales el sistema de recomendación puede calcular una predicción. Un valor bajo para el cubrimiento indica que el usuario debe privarse de un gran número de ítems, o evaluarlos basado en otro criterio diferente a la recomendación. Un valor alto de cubrimiento muestra que el sistema de recomendación apoya la selección entre la mayoría de los ítems.

   Una medida básica de cubrimiento es el porcentaje de ítems para los cuales está disponible una predicción. Este valor puede variar dependiendo de los ratings del usuario y del conjunto de usuarios similares (vecindario). Una manera de realizar esta medida es respondiendo al siguiente cuestionamiento: ¿Para todos los ratings dados por cada usuario, estuvo el sistema dispuesto a hacer una recomendación para cada ítem inmediatamente antes de que fuera calificado? Se calcula la medida de cubrimiento como el porcentaje de ítems recomendados por el sistema sobre el total de ítems calificados.

2. Exactitud. Para evaluar la exactitud de un algoritmo de predicción, se consideran dos clases de métricas: Estadísticas y de soporte para decisiones.

   Las métricas estadísticas evalúan la exactitud de un sistema de predicciones comparando el valor dado por este contra el valor dado por el usuario. Como

[15] API Application Progamming Interface. Un API es una serie de funciones que los programas pueden usar para hacer que el sistema operativo realice ciertas tareas, como abrir ventanas, archivos, cajas de mensajes o ejecutar tareas más complejas.

ejemplo de este tipo de métricas está MAE (Mean Absolute Error) o media del error absoluto, que se obtiene como la diferencia del promedio absoluto entre los ratings calculados y los ratings actuales. Otras métricas usadas son: La raíz cuadrada del error medio y la correlación entre ratings y predicciones. Los resultados de estas métricas estadísticas arrojan conclusiones muy similares.

Las métricas de exactitud como soporte para decisiones, miden que tan bien las predicciones ayudan a los usuarios a seleccionar ítems de alta calidad, de un conjunto de ítems. Esta medida se basa en el hecho de que para muchos usuarios, el filtrado es un proceso binario. Un ejemplo es la medida de sensibilidad ROC (Receiver Operating Characteristic), o *ROC sensitivity*, que evalúa el poder de diagnóstico de un sistema de filtrado. Un sistema de predicción opera en forma similar a un filtro, donde una predicción alta es equivalente a aceptar el ítem y una predicción baja a rechazar el ítem.

La sensibilidad ROC está dada por el área bajo la curva ROC. Una curva que grafica sensibilidad y especificidad de la prueba.

- La sensibilidad está definida como la probabilidad de que un buen ítem seleccionado aleatoriamente, sea aceptado por el filtro.

- La especificidad está definida como la probabilidad de que un mal ítem seleccionado aleatoriamente, sea rechazado por el filtro.

La curva ROC grafica la sensibilidad (de 0 a 1) y 1 – especificidad (de 0 a 1), obteniendo un conjunto de puntos, variando el umbral del puntaje de la predicción sobre el cual el ítem es aceptado. Por ejemplo, un punto particular puede corresponder a establecer el filtro en una predicción de exactamente 4, es decir, ver cualquier película recomendada con una predicción de 4 o superior. El área bajo la curva aumenta a medida que el filtro puede encontrar más “buenos ítems”, mientras rechaza más “malos ítems”.

Para esto, se considera un buen ítem, si el usuario le dio una calificación de 4 o superior, y un mal ítem en caso contrario. A esta medida de sensibilidad con umbral de 4, se le conoce como ROC-4.

Una segunda medida de exactitud como soporte para decisiones es la Tasa de Rechazo o *Reversal Rate*, que evalúa con que frecuencia el sistema comete grandes errores, que pueden debilitar la confianza que el usuario tiene en el sistema de recomendación, y la tercera medida de esta clase, es la de Sensibilidad PRC o PRC Sensitivity, que determina el grado en el que el sistema presenta información relevante. Este valor es el área bajo la curva Precision-Recall [38].

Dentro de las métricas empleadas para medir la precisión de un modelo, pueden nombrarse las siguientes: [16]

- Media del error absoluto Normalizado NMAE (Normalized Mean Absolute Error): Es una medida normalizada de la desviación entre los estimados y los verdaderos ratings dados por los usuarios y se calcula como:

$$NMAE = \frac{\sum_{(i,j)\in T} \left| r_{i,j} - P_{i,j} \right|}{ns} \qquad (10)$$

T es el conjunto de prueba, $r_{i,j}$ *es* el rating dado por el usuario i al ítem j, $P_{i,j}$ es la predicción hecha por el sistema, *n* es el número de ítems en *T* y *s* es el número de puntos en la escala.

- Exactitud (Accuracy): Porcentaje de ítems que son clasificados correctamente.

Para las siguientes ecuaciones, S_1 es el conjunto de ítems recomendados y S_2 es el conjunto de ítems aceptados por el usuario (si son de su interés).

$$Exactitud = \frac{\left|S_1 \cap S_2\right| + \left|\overline{S_1} \cap \overline{S_2}\right|}{\left|S_2\right| + \left|\overline{S_2}\right|} \qquad (11)$$

- Precisión (precision): Porcentaje de ítems recomendados que son relevantes.

$$Precision = \frac{\left|S_1 \cap S_2\right|}{\left|S_1\right|} \qquad (12)$$

- Recall: Porcentaje de ítems relevantes que son recomendados.

$$Recall = \frac{\left|S_1 \cap S_2\right|}{\left|S_2\right|} \qquad (13)$$

- F-measure: Es una combinación ponderada de Recall y Precision.

$$F - measure = \frac{2 * \Pr ecision * \Re call}{\Pr ecision + \Re call} \quad (14)$$

4.3 PROPUESTAS DE INTEGRACIÓN DE MODELOS DE FILTRADO

Como ya se expuso en este documento, los modelos básicos de filtrado de información que se utilizan para los modelos de integración propuestos son: el filtrado colaborativo y el filtrado por contenido. Para medir el comportamiento de la integración, es necesario primero observar como trabajan los modelos en forma separada. A continuación se hace una descripción breve de cada uno de ellos.

4.3.1 Algoritmo de Filtrado Colaborativo. (Ver Numeral 2.2) El punto de partida es la información que se tiene de cada usuario con respecto a las películas (ítems o documentos) que ha calificado. Con estos datos se construye la matriz de usuarios por documentos (UxD), donde cada fila corresponde a las calificaciones de un usuario dado y cada columna a todas las calificaciones dada por los usuarios a una película. Esta matriz es muy dispersa, porque cada usuario en promedio ha calificado un porcentaje bajo de películas. Las calificaciones oscilan entre 1 y 5, siendo 5 el mejor valor.

Para hallar los usuarios con gustos similares, se utiliza el coeficiente de correlación de Pearson. El algoritmo halla la calificación estimada para una película que el usuario activo no ha visto, buscando entre los usuarios más altamente correlacionados con él y que hayan calificado el ítem para el que se desea hallar el estimado.

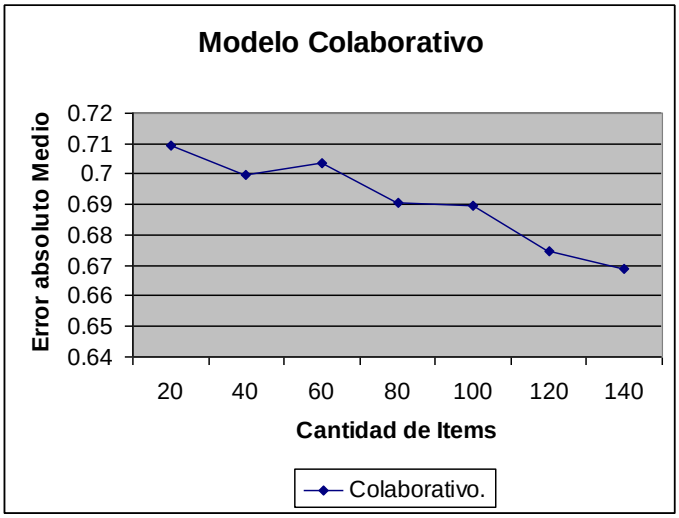
Para probar la bondad de los resultados, se utiliza la media del error absoluto, comparando los valores estimados con los valores reales que cada usuario le ha dado a cada ítem. En la figura 12 se muestran los resultados obtenidos, probando con conjuntos de pruebas de 20, 40, 60,80, 100,120 y 140 ítems y un número constante de usuarios igual a 50.

4.3.2 Algoritmo de Filtrado por Contenido. (Ver numeral 2.1) Para este modelo se necesita conocer el perfil de cada usuario (esto es, la colección de palabras claves que describen sus intereses), que se representa en un vector de términos para ser comparado con los documentos (descripción de las películas) que han sido representados de forma similar.

La base de datos de EachMovie, no tiene creados los perfiles, por cuanto se requiere realizar un proceso para construirlos. Esta tarea busca las películas

mejor calificadas por cada usuario (calificadas con 5, y si no tiene, se buscan las calificadas con 4) y extrae las palabras claves de dichas películas, teniendo en cuenta tomar películas de diferentes géneros. Estas palabras se almacenan en un vector que constituye su perfil. Para hallar el grado de correspondencia entre un perfil de usuario y un documento se utiliza el método del coseno entre los ángulos de los vectores que representan al ítem y al perfil.

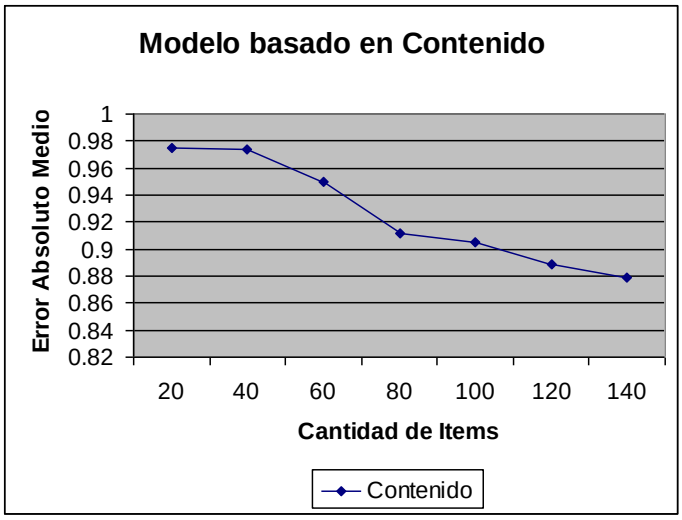
Figura 12. Variación de la Media del error absoluto, con el incremento en el número de ítems, para el filtrado colaborativo.



De manera semejante a las pruebas mencionadas en el numeral 4.3.1, se utiliza la media del error absoluto para determinar que tan cercano está el valor que predice el algoritmo al valor real dado por el usuario. La figura 13 ilustra los resultados.

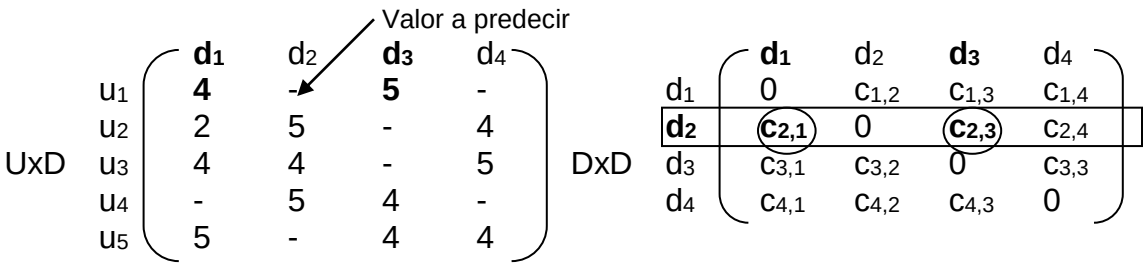
Para los modelos que se explican en los numerales 4.3.3 y 4.3.4, se probó el análisis por contenido, trabajando con dos matrices, la de usuarios por documentos (UxD), que se obtiene de la información de EachMovie y la de correlación documento por documento (DxD)que se debe construir. Para predecir un valor de calificación de un usuario a un ítem particular, se buscan todos los documentos calificados por el usuario activo y en la matriz DxD se localiza cual de ellos tiene un mayor valor de correlación con el ítem que se quiere calificar y el valor de calificación que posea se asignará al ítem que se busca predecir.

Figura 13. Variación de la Media del error absoluto, con el incremento en el número de ítems, para el filtrado por contenido.



Con el siguiente ejemplo se muestra la operación de las dos matrices. Se desea predecir la calificación que el usuario 1 (u_1) le daría al documento 2 (d_2). Este usuario ha calificado los documentos 1 y 3, con valores de 4 y 5 respectivamente. En la matriz DxD, se toma la fila 2 que corresponde a la correlación del documento 2 con todos los demás documentos y se busca el mayor valor entre los documentos que el usuario 1 ya calificó, que para el ejemplo son el d_1 y d_3. Se toma el mayor valor de correlación, por ejemplo $c_{2,3}$ que corresponde a la correlación del documento 2 con el documento 3; paso seguido, se busca la calificación que el usuario le dio al documento 3, (para este caso es 5) y este valor sería la predicción para el documento 2. La figura 14 muestra los elementos de las matrices que se tendrían en cuenta para esta operación.

Figura 14. Operación de las matrices UxD y DxD.

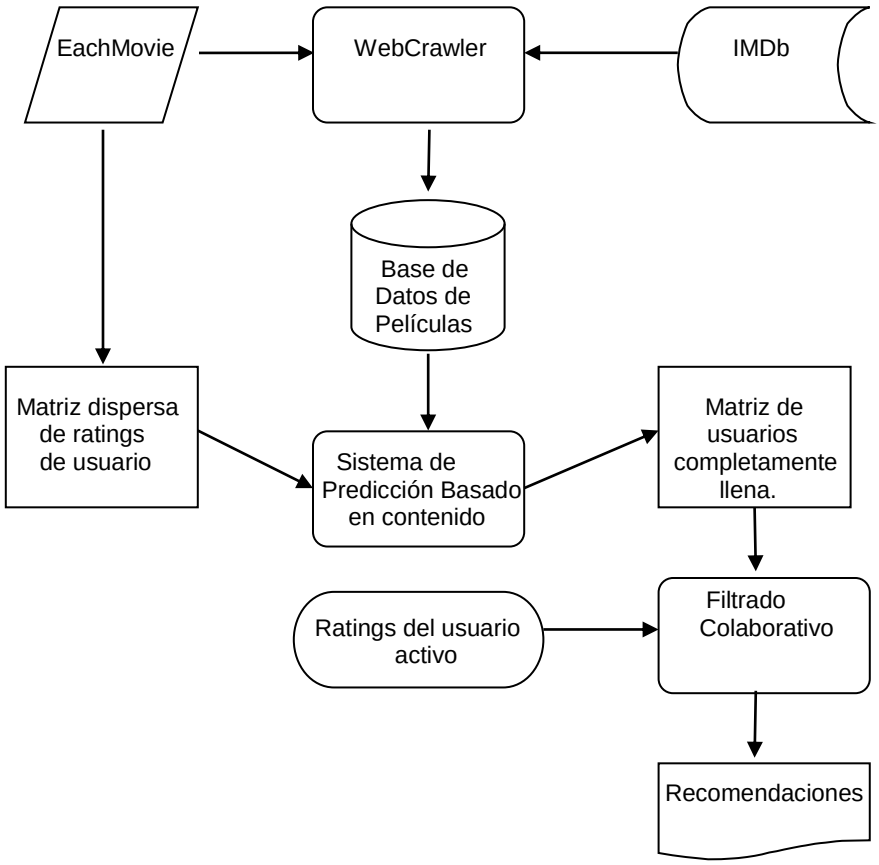


Los numerales 4.3.3, 4.3.4 y 4.3.5 presentan las tres propuestas de integración utilizando estos modelos.

4.3.3 Modelo híbrido de recomendación de información: Filtrado Colaborativo Mejorado por Contenido[16] [5]. La propuesta de integración de las dos técnicas más ampliamente probadas y difundidas para el filtrado de información, como son filtrado colaborativo y basado en contenido, intenta minimizar las debilidades fundamentales del filtrado colaborativo, que ya se expusieron en el numeral 2.2. La descripción general del sistema se observa en la figura 15.

El Web Crawler usa los URLs del conjunto de datos de EachMovie para descargar el contenido de las películas de IMDb (Internet Movie Database). Estos contenidos son almacenados en la Base de Datos de Contenidos y de Películas. De EachMovie también se obtiene la matriz de usuarios-ítems donde cada celda es el rating dado por un usuario a un ítem. Cada fila de esta matriz es un vector de ratings de usuario. Esta matriz es muy dispersa, ya que la mayoría de los ítems no han sido calificados por todos los usuarios.

Figura 15. Descripción general del sistema.



[16] Este modelo se encuentra documentado con el nombre de "Content-Boosted Collaborative Filtering". Prem Melville, Raymond J. Mooney and Ramadass Nagarajan.

El sistema de predicción basado en contenido, toma los datos del vector de ratings de usuario y crea un pseudo vector de ratings, que está compuesto de las calificaciones que ha dado el usuario actual y de las predicciones hechas por el sistema basado en contenido para los ítems no calificados. Todos estos pseudo vectores se colocan juntos para formar la pseudo matriz de ratings, que ahora está totalmente llena. Aplicando el modelo de filtrado colaborativo a la pseudo matriz de ratings, se calcula la predicción para un nuevo ítem (uno que el usuario no ha calificado).

4.3.3.1 Descripción del Sistema. La propuesta involucra el desarrollo de dos procesos, el basado en contenido y el colaborativo:

1. Proceso basado en contenido. Ya teniendo los contenidos de las películas, se ejecuta el algoritmo de filtrado por contenido (descrito en el numeral 2.1), comparando ítems con ítems, esto es, los ítems no calificados, con los que ya poseen un rating y mediante el cálculo del coseno entre los ángulos de los vectores que representan los dos ítems que se están comparando, se determina que tanto un ítem se parece a otro. El rating del ítem más semejante es asignado al que no ha sido evaluado.

   Dada la naturaleza tan variada de la temática de los documentos, esto es, la descripción de cada una de las películas de la base de datos, se tomó como heurística para hallar la similitud entre documentos, asignar un peso especial a la palabra clave que identifica el género al que pertenece la película. En muchos casos, una película pertenece a varios géneros, por ejemplo es de Humor y Ciencia Ficción a la vez. Algunas han sido clasificadas en 3, 4, 5 o 6 géneros. Por tanto, si una pareja de documentos (uno calificado por el usuario y otro no), presenta coincidencias en los géneros, se considera al documento rankeado, como un buen candidato a ser recomendado. Para esto, se agrega un valor dado k, por cada coincidencia en los géneros; de esta forma, se le da a la palabra clave género un mayor peso, por ser el término que más información aporta sobre el ítem (descripción de la película).

   El propósito es usar la información del contenido de los ítems que ya han sido calificados, y empleando las predicciones del algoritmo de filtrado por contenido, convertir la matriz dispersa de usuarios e ítems, en una matriz totalmente llena, para usarla en el sistema de filtrado colaborativo y obtener las recomendaciones que finalmente serán entregadas al usuario.

   Para cada usuario *u*, se debe crear un *pseudo vector de ratings*, en la base de datos. Este *pseudo vector* v_u, está formado por los ítems calificados por el usuario *u*, cuando se conoce este dato, y por los ratings generados por el sistema de predicciones basado en contenido, descrito anteriormente.

$$V_{u,i} = \begin{cases} r_{u,i} & \text{Si el usuario u calificó al ítem i} \\ c_{u,i} & \text{En caso contrario.} \end{cases}$$

$c_{u,i}$ es la predicción hecha por el sistema basado en contenido puro.

2. Proceso colaborativo. Los *pseudo vectores* de todos los usuarios (con los datos calculados por contenido) se colocan juntos en una *pseudo matriz V* de los ratings. Luego se ejecuta el filtrado colaborativo, con el proceso descrito en el numeral 2.2.1.

   La similaridad entre el usuario activo *a* y otro usuario *u* se calcula usando el coeficiente de correlación de Pearson. En lugar de usar los votos originales, se usan los votos provistos por los pseudo vectores de ratings de usuarios v_a y v_u.

3. Factores de ponderación. La precisión del pseudo vector de ratings calculada para cada usuario depende del número de películas que él o ella haya calificado. Si se han calificado muchas películas, las predicciones basadas en contenido son buenas y por lo tanto, los pseudo vectores de ratings son bastante precisos. Lo contrario ocurre con pocas películas rankeadas.

Las imprecisiones en estos vectores pueden producir altas correlaciones engañosas, por lo tanto, para introducir confianza en estos valores, se introducen los siguientes factores de ponderación:

Factor Ponderado de Significancia: Este factor busca devaluar las correlaciones basadas en pocos ítems rankeados en común.

$$sg_{a,u} = \begin{cases} n/50 & \text{para } n<50 \\ 1 & \text{En caso contrario} \end{cases} \qquad \text{(a)}$$

Donde n es el número de ítems calificados en común por el usuario a y el usuario u, el valor 50, se basa en la curva de aprendizaje del modelo de predicción basado en contenido, como se explica en [5], puesto que a mayor número de ítems, el rendimiento mejora, pero, por encima de 50 ítems, el sistema tiende a nivelarse.

Factor de Ponderación Armónico Medio: De forma similar al anterior, el propósito es dar mayor peso a las correlaciones de aquellos pseudo rankings de usuario, que tengan más de 50 ítems calificados por el usuario.

$$m_i = \begin{cases} n_i/50 & \text{si } n_i < 50 \\ 1 & \text{En caso contrario} \end{cases}$$

n_i es el número de ítems calificados por el usuario *i*.

$$hm_{i,j} = \frac{2m_i m_j}{m_i + m_j} \quad \text{(b)}$$

Así, la correlación entre los usuarios de la matriz densa V, al menos con 50 ítems calificado en común, reciben el mayor peso.

Peso de Correlación híbrido: Este factor se obtiene de la suma de (a) y (b).

$$hw_{a,u} = hm_{a,u} + sg_{a,u} \quad \text{(c)}$$

Factor de Ponderación Propio: Pretende dar mayor importancia a los datos del usuario activo (pseudo vector de usuario), que a los datos de los demás usuarios correlacionados. En otras palabras, busca incrementar la confianza en el sistema de predicción basado en contenido, para el usuario activo.

$$sw_a = \begin{cases} n_a/50 * max & \text{Si } n_a < 50 \\ max & \text{En caso contrario} \end{cases} \quad \text{(d)}$$

Donde n_a es el número de ítems rankeados por el usuario activo.
Max, es un indicador de la confianza total en el sistema de predicción basado en contenido. Los autores [5], proponen utilizar un valor Max igual a 2.

Combinando los factores expuestos, la predicción final del modelo de Filtrado Colaborativo mejorado por contenido se calcula como:

$$p_{a,i} = \bar{v}_a + \frac{sw_a (c_{a,i} - \bar{v}_a) + \sum_{\substack{u=1 \\ u \neq a}}^{n} hw_{a,u} P_{a,u} (v_{u,i} - \bar{v}_u)}{sw_a + \sum_{\substack{u=1 \\ u \neq a}}^{n} hw_{a,u} P_{a,u}} \quad (15)$$

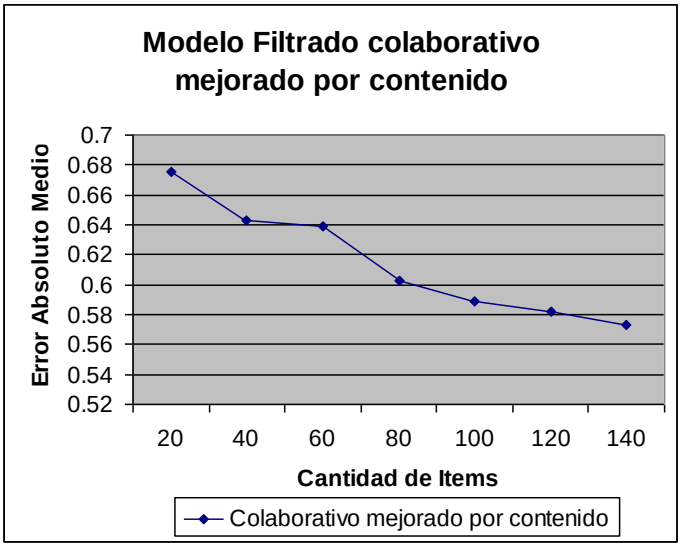
$p_{a,i}$ = Predicción para el ítem *i* del usuario *a*.
$\bar{v}_a$ = Media de los votos del usuario *a*.
$c_{a,i}$ = Predicción por contenido para el ítem *i* del usuario *a*.
$v_{u,i}$ = Pseudo rating de usuario *u* para el ítem *i*.
$\bar{v}_u$ = es la media de todos los ítems del usuario *u*.
sw_a = Factor de ponderación propio, descrito en (d), para el usuario *a*.
$hw_{a,u}$ = Peso de correlación híbrido, descrito en (c), entre el usuario *a* y *u*.
$P_{a,u}$ = Coeficiente de Correlación de Pearson entre el usuario *a* y *u*.
n es el tamaño del vecindario (número de usuarios con los que se correlaciona el usuario activo).

4.3.3.2 Resultados obtenidos. Una vez realizada la integración de los dos modelos, hallando las predicciones para un usuario dado y un ítem específico, con la fórmula descrita en (15), es necesario determinar la bondad de los resultados. Para conocer el grado de precisión de la propuesta, se utiliza como métrica, el cálculo de la media del error absoluto (MAE), dado que se conocen los ratings que los usuarios efectivamente dieron a las películas vistas.

Como conjunto de prueba se tomaron 50 usuarios y diferentes grupos de películas para observar como variaban las respuestas al incrementar el número de ítems. Se tomó inicialmente un subconjunto de 20 ítems, luego 40, 60, 80, 100, 120 y finalmente 140.

Efectivamente, como era de esperarse, el algoritmo arroja valores más cercanos a los ratings de usuario, a medida que el número de ítems crece, como se observa en la figura 16.

Figura 16. Variación de la Media del error absoluto, con el incremento en el número de ítems, para el modelo híbrido de filtrado colaborativo, mejorado por contenido.



Análisis de los Resultados. La disminución en el valor de la Media del error absoluto, comparado con los modelos de filtrado colaborativo y filtrado por contenido puros, obedece a varios aspectos:

- Por contar con una matriz de usuarios completamente llena, es posible hallar valores de correlación entre usuarios que tengan muy pocos ítems calificados, utilizando el llenado de la matriz por contenido, para suplir la debilidad del filtrado colaborativo para los usuarios nuevos.

- La incorporación del Factor de Ponderación propio, le da al sistema un mayor grado de confianza, al asignarle mayor peso a las calificaciones dadas por el usuario activo, que no requieren ningún cálculo del sistema y que son la fuente para que el modelo de filtrado por contenido halle los valores de las predicciones de los ítems no rankeados.

- El uso de una matriz usuarios/ítems completamente llena, posibilita al sistema para encontrar usuarios similares y hallar un vecindario mejor y más representativo. Aun cuando dos usuarios no tengan ítems calificados en común, pueden tener gustos muy similares, y este modelo si refleja estas similitudes, lo que no ocurriría en un sistema de filtrado colaborativo puro.

4.3.4 Modelo Combinado para Recomendación de Información, usando Filtrado basado en Contenido y Colaborativo[17]. [16] Se propone un método para combinar los dos modelos de filtrado, basado en contenido y colaborativo, utilizando el conjunto de datos de EachMovie, como en la propuesta descrita anteriormente.

4.3.4.1 Descripción del Sistema.

Enfoque basado en contenido: el sistema analiza el contenido de los ítems que ha calificado un usuario para crear su perfil que es la representación de sus intereses (palabras claves, frases o características). Luego se estudia el contenido de los ítems desconocidos para el usuario, y los compara con el perfil y estima cuáles de aquellos ítems son de su interés. Este método es efectivo para recomendar documentos textuales, ya que es necesario que los ítems sean entendibles por el computador.

Para representar ítems y perfiles se emplea el vector espacial. Para medir qué tanto un ítem corresponde al perfil, se usa el coseno entre los ángulos de los vectores que representan al ítem y al perfil, que es la medida más comúnmente utilizada por los sistemas de recomendación basados en contenido. El rango de valores oscila entre –1 y 1, y se divide en subintervalos, tantos, como valores tenga la escala de ratings utilizada, el subintervalo más bajo se asigna al valor más bajo de la escala y el subintervalo más alto, al valor más alto de la escala. Para cada ítem se le asigna un estimado de acuerdo al subintervalo al cual pertenezca el valor de su peso. Los ítems con un valor superior a T_1 (umbral escogido), se le recomiendan al usuario. Por ejemplo, si se utiliza una escala con valores de 0 a 5, siendo 0 el valor más bajo y 5 el más alto, se escoge un valor $T_1 = 4$, por ser el valor más común, así solo se le recomiendan al usuario los ítems con rating estimado igual o superior a 4.

Enfoque Colaborativo: el sistema trabaja coleccionando los ratings de los usuarios y comparando estos ratings se localiza a quienes comparten intereses comunes, para hacer recomendaciones personalizadas basado en las opiniones de usuarios con gustos similares.

Se utiliza el método de correlación, como el más ampliamente conocido para filtrado colaborativo. Este método mantiene una matriz $R = \{ r_{ij} \}$, donde r_{ij} es el rating del usuario i para el ítem j. Las filas de la matriz R representan los perfiles de los usuarios. Esta matriz se utiliza para encontrar los usuarios con gustos similares aplicando el coeficiente de correlación de Pearson. Para el cálculo del estimado se usan no solo los usuarios con intereses similares (correlación positiva) sino también usuarios con intereses opuestos (correlación negativa).

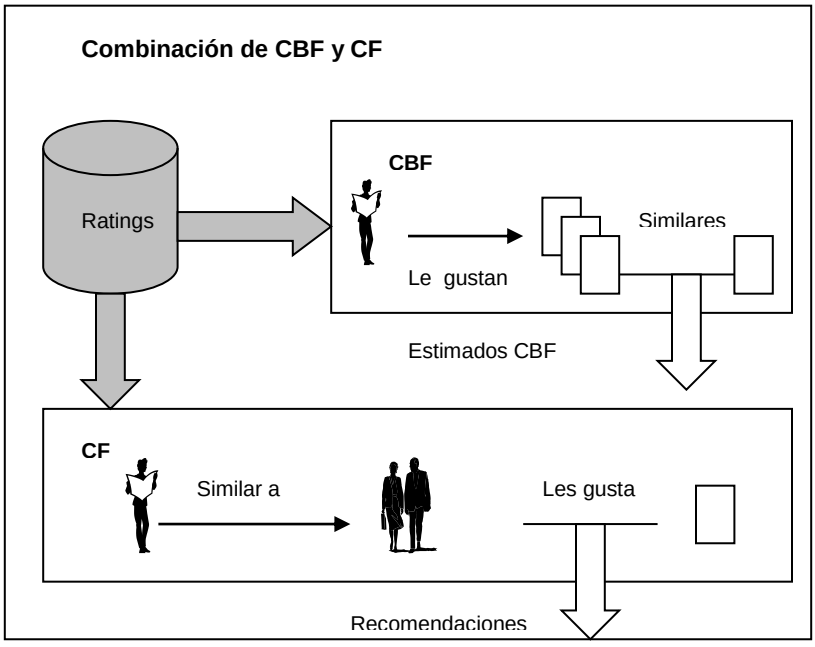
[17] Modelo propuesto por Gabriela Polcicová y Pavol Návrat. Combining Content-based and Collaborative Filtering.

Para hallar un nuevo estimado, se emplean los ratings de usuario *i*, tal *que* | $coor_i$ | > T_2 (Umbral dado). Este valor T_2 se establece de acuerdo al comportamiento de los datos, y se emplea para descartar aquellos usuarios con muy baja correlación con el usuario activo.

Propuesta Combinada Basada en Contendido y Colaborativa. El filtrado basado en contenido se utiliza para llenar los ratings que no se conocen para el método colaborativo. Se necesitan ambos perfiles, el basado en contenido y el basado en ratings. Este método trata de minimizar la principal debilidad del filtrado colaborativo, la falta de datos y el problema de la precisión de las estimaciones y recomendaciones del basado en contenido.

Para medir la similaridad entre usuarios , se usan dos clases de coeficientes. El primero es el Coeficiente de Correlación de Pearson, *corr(c,i)*, que utiliza los ratings de los usuarios *c* e *i*. El segundo es el Coeficiente de Correlación de Pearson, *corr'(c,i)*, que se calcula usando tanto los ratings como el estimado basado en contenido puro.

Figura 17. Esquema del método combinado CF (Filtrado Colaborativo) y CBF (Filtrado basado en contenido).



Para hallar *corr'(c,i)* se toman en cuenta solo aquellos ítems que han sido calificados al menos por uno de los usuarios comparados. Se consideran solo estas dos situaciones:

- Ambos usuarios calificaron un ítem.
- Solo uno de los dos usuarios calificó el ítem y se toma la estimación hecha por el filtrado por contenido como el rating del otro usuario.

Los ítems que no han sido calificados por ninguno de los dos usuarios, no se usan en el cálculo de *corr'(c,i).*

El estimado se calcula como sigue:

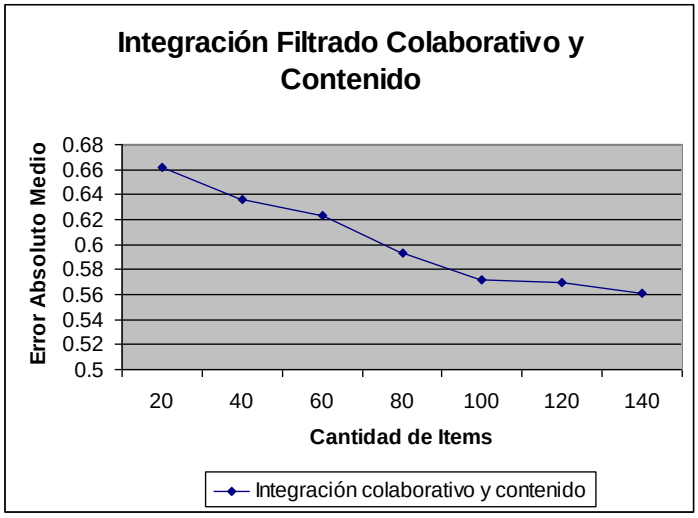
$$Predicción = \bar{u} + \frac{\sum_{i=1}^{n} corr_i * (rating_i - \bar{i}) + \sum_{i=1}^{n} corr'i * (rating_i^{CBF} - \bar{i})}{\sum_{i=1}^{n} |corr_i| + \sum_{i=1}^{n} |corr'_i|} \quad (16)$$

donde $\bar{u}$ es la media de los ratings dados por el usuario al que se le está calculando la predicción, $corr_i$ es el Coeficiente de Correlación de Pearson entre el usuario *i* y el usuario *u*, $rating_i$ es el rating dado por el usuario *i* para el ítem al que se le está calculando la predicción, $\bar{i}$ es el promedio de los ratings dados por el usuario *i* para los artículos en común, *n* es el número total de usuarios en el sistema, que tienen alguna correlación con el usuario y que han calificado el ítem, $rating_i^{CBF}$ es el rating dado por el usuario i, si está disponible, o la estimación calculada por el método basado en contenido.

Para reducir los requerimientos de memoria, se escogió un conjunto de prueba de 100 usuarios. El umbral de recomendación se establece en $T_1 = 4$, así, solo los ítems con calificación ≥ a T_1 se recomiendan a un usuario. El umbral T_2 se establece teniendo en cuenta las características de los ítems de la base de datos de EachMovie, lo que significa que solo los ratings de usuarios con $(corr_i) \geq T_2$ se utilizan para el cálculo del estimado.

4.3.4.2 Resultados obtenidos. De forma similar al modelo anterior, se prueba la efectividad del algoritmo hallando la media del error absoluto, entre los valores dados por el usuario y las predicciones hechas por el sistema. Como se observa en la figura 18, el error se va haciendo menor al ingresar más ítems a la base de prueba.

Figura 18. Variación de la Media del error absoluto, con el incremento en el número de ítems, para el modelo de combinación de filtrado colaborativo y por contenido.



Análisis de Resultados. Como en la propuesta descrita anteriormente, el valor de la media del error absoluto es más bajo que al considerar los dos modelos de filtrado separadamente, por las siguientes razones:

- En forma similar al modelo descrito en el numeral 4.3.3, la incorporación de filtrado por contenido, minimiza las debilidades del filtrado colaborativo (numeral 2.2).

- Aquellos ítems que no han sido calificados por ninguno de los dos usuarios para los que se está calculando su correlación, no son tenidos en cuenta, puesto que no se tiene una medida real del agrado o desagrado que el o los usuarios tengan por ese ítem.

- Para calcular el estimado final, utiliza las dos predicciones, una usando el filtrado colaborativo puro, y la otra adicionando filtrado por contenido, para aprovechar las fortalezas de cada uno de los modelos. El contenido servirá para los cálculos de usuarios nuevos y el colaborativo, para aquellos ítems cuya vista lógica sea muy pobre.

4.3.5 Combinación de Filtrado basado en Contenido y Colaborativo, manteniendo los dos modelos separados[18]. [20]

La propuesta hecha por [20], combina el filtrado por contenido y colaborativo, generando las predicciones como el promedio ponderado de la predicción basada en contenido y la predicción colaborativa. Los pesos para cada modelo se determinan por usuario, permitiéndole al sistema determinar la mezcla óptima de las dos recomendaciones, ayudando a resolver el problema de la oveja negra. Dado que el número de ratings crece con el tiempo, al filtrado colaborativo se le da mayor peso, aumentando la exactitud de la predicción.

Este trabajo combinado, no es un sistema híbrido, porque las bases para las predicciones por contenido y de filtrado colaborativo se mantienen separadas. Esto permite que el sistema se beneficie de los avances individuales en cada uno de los modelos, ya que no hay interdependencia entre los contenidos de los dos componentes.

El filtrado colaborativo calcula una predicción de la siguiente manera:

$$\textit{Predicción} = \bar{u} + \frac{\sum_{i=1}^{n} corr_i * (rating_i - \bar{i})}{\sum_{i=1}^{n} corr_i} \qquad (17)$$

donde $\bar{u}$ es la media de los ratings dados por el usuario *u* al que se le está calculando la predicción, $corr_i$ es el Coeficiente de Correlación de Pearson entre el usuario *i* y el usuario *u*, $rating_i$ es el rating dado por el usuario *i* para el ítem al que se le está calculando la predicción, $\bar{i}$ es el promedio de los ratings dados por el usuario *i* para los artículos en común, y *n* es el número total de usuarios en el sistema, que tienen alguna correlación con el usuario y que han calificado el ítem.

El algoritmo de filtrado basado en contenido, hace la correspondencia entre las palabras claves del ítem y las palabras claves del perfil de usuario. Estas palabras claves se seleccionan de acuerdo a su frecuencia de ocurrencia en el texto, asumiendo que la ocurrencia de las palabras es una medida útil del grado de importancia de esta palabra.

El objetivo de la combinación es encontrar los pesos adecuados que lleven a una predicción más precisa. [27]

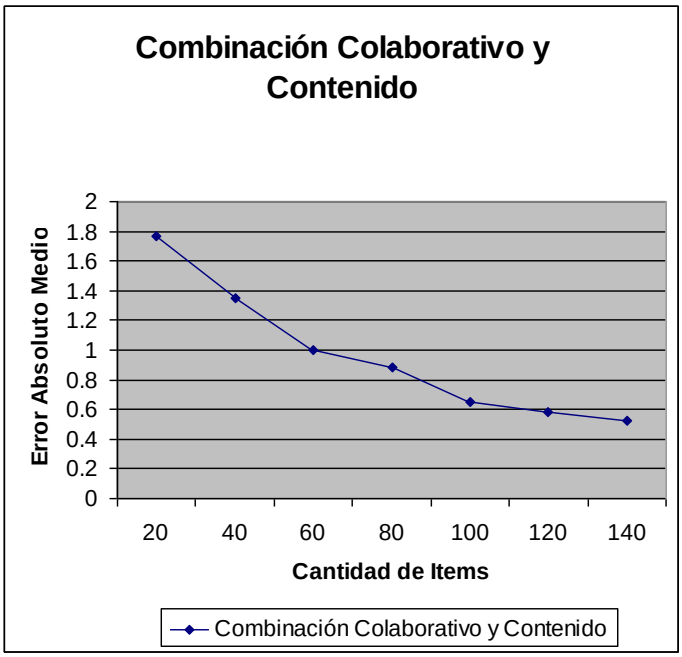
[18] Propuesta presentada por Anuja Gokhale, en su trabajo de Tesis de Maestría en Ciencias Computacionales, "Improvements to Collaborative Filtering Algorithms". Worcester Polytechnic Institute.

En la fase inicial, la predicción se calcula como:

*0.5 * calificación obtenida del filtrado colaborativo + 0.5 * calificación obtenida del filtrado basado en contenido.*

Cada vez que el usuario ingrese un rating para un ítem, este valor se compara con el calculado por los dos modelos. Los nuevos pesos para los modelos cambiarán de acuerdo a cual de los dos valores estuvo más cerca del rating dado por el usuario, así un peso subirá un valor dado *x* y el otro peso bajará el mismo valor. Los valores de los pesos seguirán ajustándose con cada nuevo cálculo que se realice.

Figura 19. Variación de la Media del error absoluto, con el incremento en el número de ítems, combinando filtrado colaborativo y por contenido, manteniendo los dos modelos separados.



5. ANÁLISIS DE RESULTADOS

Uno de los principales objetivos de los sistemas de filtrado y recomendación de información es ayudar a los usuarios a encontrar, de una manera más efectiva los contenidos que realmente necesitan, en este trabajo se intenta medir la utilidad de los modelos propuestos en dos dimensiones: precisión (Medida con la Media del Error Absoluto) y cubrimiento (Coverage). Adicionalmente, se estudia el comportamiento de los modelos propuestos, desde el punto de vista del Filtrado Colaborativo, del Filtrado por contenido y la dificultad en la codificación y ejecución de los mismos.

Media del Error Absoluto.

Los resultados de la Media del Error Absoluto se describen a continuación:

Para realizar la comparación de los resultados obtenidos por los diferentes modelos estudiados, es necesario tener la referencia del comportamiento de los algoritmos de filtrado colaborativo y por contenido separadamente. Los resultados que se van a comparar corresponden a los algoritmos explicados en los numerales 4.3.1, 4.3.2 , 4.3.3, 4.3.4 y 4.3.5. (Ver Anexo A).

En la siguiente tabla se muestran los valores de la Media del Error Absoluto, obtenidos para los diferentes modelos. La base de datos de prueba se trabajó inicialmente con 50 usuarios y 20 ítems, luego con 40, 60, 80, 100, 120 y finalmente con 140 ítems.

Modelo 1: Usando Filtrado colaborativo puro, descrito en el numeral 4.3.1

Modelo 2: Usando Filtrado por contenido puro, descrito en el numeral 4.3.2

Modelo 3: Usando Filtrado colaborativo mejorado por contenido descrito en el numeral 4.3.3

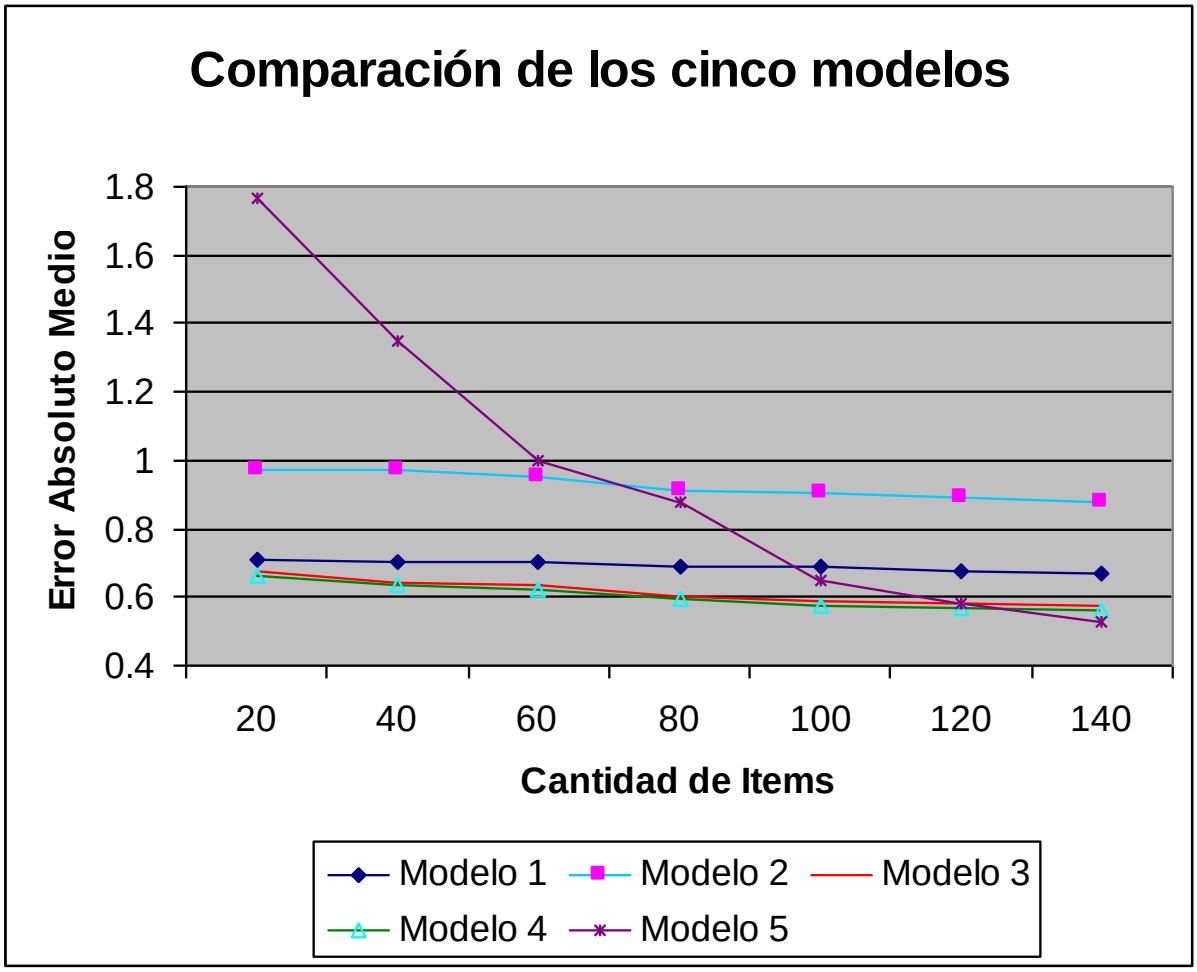
Modelo 4: Usando la integración de Filtrado colaborativo y contenido, propuesta por [16] y descrita en el numeral 4.3.4

Modelo 5: Usando la combinación de Filtrado colaborativo y contenido, propuesta por [20] y descrita en el numeral 4.3.5

Tabla 4. Media del error absoluto, manteniendo el número de usuarios constante.

No. Items	Modelo 1	Modelo 2	Modelo 3	Modelo 4	Modelo 5
20	0.709533	0.97475588	0.6754384	0.66171291	1.764732843
40	0.69979861	0.97342286	0.64278497	0.636511	1.348995638
60	0.703506	0.95027507	0.63871862	0.62303235	1.001345225
80	0.690564	0.91121588	0.60221543	0.59274162	0.878954561
100	0.689473	0.905018018	0.5887334	0.57203862	0.64644969
120	0.6746373	0.88862213	0.58190696	0.56943637	0.584326547
140	0.66878413	0.879192829	0.57337521	0.56112758	0.524698752

Figura 20. Resultados comparativos de los cinco modelos estudiados.



Los modelos integrados (3 y 4) tienen un comportamiento aceptable, aún con muy pocos datos, es decir, pueden aplicarse para conjuntos de datos con pocos ítems,

sin embargo, si se requiere una solución simple, rápida y fácil de implantar, tiene ventaja el filtrado colaborativo puro, sobre los modelos que integran o combinan las dos técnicas de filtrado. Pero en la práctica, las fuentes de información que se requieren filtrar son de gran volumen, al igual que el número de usuarios destino. La matriz de U X D es muy dispersa, porque el número de documentos (ítems) es mucho mayor que la cantidad de información que un usuario puede procesar. Por esto, las propuestas de integración de varias técnicas, se están imponiendo, tratando de incluir las fortalezas de cada una y minimizando sus debilidades individuales.

El módulo 5, que va ajustando sus pesos a medida que se realizan más predicciones, de manera individual para cada usuario, claramente muestra mejora en su desempeño a medida que crece el número de ítems evaluados. Para su aplicación a una base de datos real (no un conjunto de prueba), ofrece una buena opción como se observa en los resultados de precisión de la Media del Error Absoluto.

Cubrimiento.

El análisis del cubrimiento se describe en los siguientes párrafos:

La métrica de Cubrimiento (Coverage), explicada en el numeral 4.2, determina el porcentaje de ítems que serán tenidos en cuenta por el sistema de recomendación en el momento de calcular una predicción.

Adicional a las ventajas observadas en las propuestas que combinan o integran los dos modelos de filtrado, según la Media del error absoluto, también se pueden ver buenos resultados en cuanto al cubrimiento.

Modelo 1 (Filtrado colaborativo): presenta el cubrimiento más bajo, puesto que para ítems nuevos, cuando aún ningún usuario les ha dado una calificación, simplemente no serán tenidos en cuenta para hacer las predicciones y recomendaciones. Para este modelo el nivel de cubrimiento es inversamente proporcional al grado de dispersión de la matriz Usuario X Ítems, es decir, a mayor dispersión menor es el grado de cubrimiento.

Modelo 2 (Filtrado por contenido): siempre que sea posible obtener una representación del ítem (una representación del mismo, por ejemplo un vector de términos), se podrá incluir a dicho ítem en la recomendación. Si se dispone del perfil del usuario, se realizará la función que evalúa el grado de correspondencia entre la representación del ítem y la representación del perfil de usuario. Aún si no se tiene una representación del perfil de usuario, otra posibilidad es hallar la correlación Documento X Documento de los que han sido evaluados por el usuario y tomar como valor probable de calificación para el ítem que se quiere evaluar, la

calificación que tenga el ítem que tenga la más alta correlación con éste. Por lo tanto, el valor del cubrimiento es aproximadamente 1.

Modelo 3 (Filtrado colaborativo mejorado por contenido): debido a que el modelo trabaja con una matriz totalmente llena (que ha sido completada usando filtrado por contenido), ya no tiene las restricciones de cubrimiento del modelo de filtrado colaborativo. Cada uno de los ítems tiene ahora una calificación y podrá ser usado para encontrar los usuarios similares y realizar predicciones.

Modelo 4 (Integración de Filtrado colaborativo y contenido, propuesta por [16]): en este modelo se aumenta la cantidad de ítems que tienen una calificación, hallada mediante filtrado por contenido, pero no llega a alcanzar un nivel de cubrimiento como el del modelo 3. Los ítems nuevos, no serán tomados en cuenta para cálculos de predicción. Por lo tanto su cubrimiento es superior al modelo 1, pero inferior al modelo 2 y 3.

Modelo 5 (Combinación de Filtrado colaborativo y contenido, propuesta por [20]): por trabajar independientemente con los modelos de filtrado colaborativo y por contenido, cada uno de estos componentes tiene un valor de cubrimiento diferente; el cálculo correspondiente al filtrado colaborativo tiene la restricción que se mencionó en el modelo 1. Para realizar una predicción se utiliza:

($peso_1$) * calificación obtenida del filtrado colaborativo + ($peso_2$) * calificación obtenida del filtrado basado en contenido.

Por lo tanto, el cubrimiento del modelo 5 es igual al del modelo 1, y depende de la dispersión de los datos.

Con este análisis, un modelo que trabaje con una matriz de datos completamente llena (como el modelo 3), garantiza un cubrimiento total, incluyendo todos los ítems en sus predicciones. Claro está, sin dejar de lado el grado de imprecisión implícito en los cálculos por contenido.

Desde la perspectiva del Filtrado Colaborativo.

El análisis desde esta perspectiva busca determinar cómo los modelos propuestos pueden ayudar a resolver los problemas que presenta el Filtrado Colaborativo, particularmente el problema del estado inicial (Cold Start Problem) y el de la dispersión de los datos.

Modelo 3 (Filtrado colaborativo mejorado por contenido): Para un ítem nuevo, este modelo calcula un rating estimado por contenido, en el momento en que llena por completo la matriz U X D (usuarios X documentos); y lo mismo ocurre para un usuario nuevo, que aún no ha calificado ningún ítem.

Modelo 4 (Integración de Filtrado colaborativo y contenido, propuesta por [16]): este modelo no aplica para un ítem nuevo, puesto que para calcular el coeficiente de correlación de Pearson, entre dos usuarios, requiere que por lo menos uno de los dos haya dado una calificación al ítem. Por lo anterior, al realizar la sumatoria correspondiente, no tiene en cuenta los ítems sin calificaciones. Para el caso de un usuario nuevo, puede calcularse su correlación con otros usuarios, hallando su estimado mediante análisis de contenido, pero es muy limitado, porque depende de la cantidad de calificaciones que hayan hecho los demás usuarios, por lo tanto, la predicción puede ser muy inexacta si la matriz U X D es muy dispersa.

Modelo 5 (Combinación de Filtrado colaborativo y contenido, propuesta por [20]): por ser un modelo que mantiene separados el filtrado colaborativo y por contenido, tiene la misma debilidad expuesta en el numeral 2.2, el filtrado colaborativo no es funcional para usuarios nuevos o para ítems recién ingresados al sistema. La predicción calculada, sumando los resultados obtenidos de los dos modelos de filtrado, no sería muy cercana en estos casos.

En síntesis, el Modelo 3 es el que se comporta de la mejor manera para afrontar el problema del estado inicial. Así mismo, ofrece mejores posibilidades en el cálculo de las predicciones cuando la base de datos de prueba produce una matriz U X D muy dispersa, generalmente porque el número de ítems es muy grande y los usuarios no tienen la capacidad de evaluar más que una pequeña fracción de los mismos.

Desde la perspectiva del Filtrado por Contenido.

El filtrado basado en contenido es absolutamente dependiente de la calidad de la información que se posea tanto de los ítems como de los perfiles de usuario, puesto que es necesario extraer y representar esta información con términos que sean significativos y generen una predicción aproximada. La base de datos de EachMovie, presenta dos deficiencias fundamentales, la primera es no poseer los perfiles de usuario, por cuanto los modelos propuestos utilizan una aproximación construida utilizando los términos de los ítems mejor evaluados; y la segunda, es la calidad de las descripciones de las películas, que en algunos casos es muy limitada y con un alto grado de subjetividad, debido a que la descripción de la misma, es el comentario enviado por una persona que vio la película, no por un especialista o experto en el tema.

Estas deficiencias se ven reflejadas en los resultados de los modelos 2 y 3. El modelo 2 (Filtrado por contenido), muestra un resultado de la Media del Error Absoluto significativamente mayor al del modelo 1 (Filtrado colaborativo), debido a que este trabaja exclusivamente con datos proporcionados por los usuarios.

De las propuestas de integración (Modelos 3 y 4), el modelo 3 que utiliza en mayor grado el cálculo aproximado de ratings por contenido (al llenar completamente la

matriz de U X D), presenta un resultado inferior al modelo 4, que solo utiliza el filtrado por contenido para el cálculo aproximado de la calificación de un usuario, cuando el otro usuario con el que se ejecuta la comparación si dio una calificación para el mismo.

Desde la perspectiva de la codificación y ejecución de los modelos.

Si la única medida a tener en cuenta fuese la facilidad de codificación y el tiempo de ejecución de lo algoritmos desarrollados, se optaría por trabajar con el de filtrado colaborativo, ya que solo trabaja realizando comparaciones sobre valores numéricos almacenados en una matriz (U X D), pero como ya se mencionó en este documento, este modelo presenta serias debilidades que se intentan cubrir con los modelos que integran también el análisis de contenido.

De los modelos propuestos para integrar o combinar las dos técnicas de filtrado descritas, el Modelo 4 (Integración de Filtrado colaborativo y contenido, propuesta por [16]), requiere mayor tiempo de proceso, porque cada vez que necesita hallar la similitud entre dos usuarios, debe volver a revisar cuáles ítems han sido calificados por al menos uno de los dos usuarios, puesto que no se trata de llenar en su totalidad la matriz U X D, sino dar mayor valor a los ratings dados por los usuarios. El número de accesos a la matriz y de estructuras de almacenamiento temporales necesarias, es mayor que si llenara en su totalidad la matriz U X D una sola vez como lo hace el modelo 3.

En el ejemplo se ilustran los diferentes cálculos que haría el modelo 4, con una matriz de tres usuarios (Ui) y 4 documentos (Dj).

UxD		d_1	d_2	d_3	d_4
	u_1	4	4	-	-
	u_2	-	5	4	-
	u_3	-	4	-	5

Al comparar u_1 y u_2, el modelo debe hallar un estimado por contenido del rating para u_1d_3 y u_2d_1, pero no debe almacenar estos valores en la matriz original.
Al comparar u_1 y u_3, se hallará el valor estimado por contenido para u_3d_1 y para u_1d_4. Como se observa, no se debe tener en cuenta el valor u_1d_3 que se halló por contenido, porque ninguno de los dos usuarios ha calificado el documento 3.

Adicionalmente, el cálculo de la predicción para un valor determinado, utiliza dos sumatorias de orden n en el numerador y dos sumatorias en el denominador, como se describe en la ecuación (16), a diferencia de los modelos 3 y 5 que utilizan solo una sumatoria.

Los modelos 3 y 5, muy similares en cuanto al esfuerzo requerido para la codificación y ejecución de los algoritmos respectivos, más sencillos que el modelo 4 , presentan resultados de precisión con una diferencia significativa, lo que apunta a seleccionar el modelo 5, como el recomendado teniendo en cuenta sus resultados en el análisis de precisión de la métrica de la Media del Error Absoluto (descrita anteriormente) y los requerimientos para su ejecución.

6. CONCLUSIONES

- La incorporación de la información por contenido en un modelo de Filtrado de información, muestra mejoras significativas en las predicciones de un sistema de recomendación, y posibilita al sistema para dar respuestas aun cuando se trate de un usuario nuevo o con gustos particulares. Así también, pueden incluir en las predicciones aquellos ítems de recién ingreso a la base de datos.

- Los métodos combinados mejoran el cubrimiento del filtrado colaborativo, puesto que pueden emplearse aun para aquellos usuarios que no han calificado ítems, y reducen la media del error absoluto del filtrado basado en contenido, especialmente en la fase inicial, donde se cuenta con un número reducido de ratings.

- En los sistemas propuestos, finalmente se trabaja con el modelo más utilizado de Filtrado Colaborativo, por lo tanto, todas las optimizaciones que se hagan al mismo, como por ejemplo en el hallazgo de los mejores vecino o nuevas formas de encontrar la correlación entre usuarios, podrán presentar mejores resultados. De forma similar, mejoras en la correspondencia de contenidos de documentos, o en el análisis semántico, se pueden incorporar en el sistema integrado.

- Los resultados obtenidos en el modelo de filtrado por contenido, evidencian la debilidad en el mismo, por no utilizar un vocabulario controlado, dada la naturaleza de la información de la base de datos que se empleó como conjunto de entrenamiento y de pruebas. Puesto que las descripciones para cada ítem (película), fueron realizadas por usuarios que expresaron su punto de vista sobre una película en particular, en algunos casos son muy cortas, en otros, muy poco ilustrativas, y en general es la apreciación de sólo un usuario, que puede ser muy subjetiva. La situación puede ser muy diferente para otro tipo de documentos, por ejemplo, con el Abstract de un artículo científico, donde el autor intenta mostrar en pocas líneas la esencia del mismo.

- Los tres modelos de integración que se programaron, mostraron mejores resultados que los modelos de filtrado por contenido o colaborativo por separado, y puede observarse que existen otras propuestas como la incorporación de agentes que trabajan por objetivos o el uso de funciones de clasificación, que se encuentran documentadas, y que de igual manera trabajan con los dos enfoques: Colaborativo y por contenido.

- El modelo descrito en 4.3.4 (Modelo Combinado para recomendación de información usando filtrado basado en contenido y colaborativo), arroja mejores

resultados que el modelo descrito en 4.3.3 (Modelo híbrido de recomendación de información: filtrado colaborativo mejorado por contenido), porque para el cálculo de la predicción, solo considera los ítems de los que se conoce por lo menos una calificación dada por alguno de los dos usuarios a los que se les está hallando la correlación, hecho que le agrega mayor confianza a la predicción, al tomar los datos reales ingresados al sistema.

- Los algoritmos fueron probados todos sobre el mismo conjunto de datos de prueba, para que los resultados puedan ser comparados. El tipo de ítem que se utilice, por ejemplo, artículos de revistas de un área de conocimiento particular, resúmenes de libros, chistes, comentarios de películas o programas de televisión, pueden determinar un comportamiento diferente en la aplicación de los algoritmos. Por lo tanto, es importante la selección de la base de datos que se empleará para la realización de las pruebas de los modelos.

- El aspecto más importante de un sistema de filtrado colaborativo es la calidad de la información que se tiene de los usuarios, y de un sistema basado en contenido es la calidad de la descripción que se tiene de cada ítem (y de cada usuario). Con la base de datos utilizada para las pruebas se encuentra la debilidad de una descripción muy pobre de cada una de las películas, lo que limita la funcionalidad del análisis por contenido, agregando el hecho de trabajar con una aproximación del perfil del usuario, que no suministra la base de datos.

- Para una implementación real de los algoritmos, tendría que trabajarse con perfiles dinámicos para los usuarios, adicionando o removiendo términos de los vectores de perfiles, de acuerdo a las preferencias de éstos. Este proceso podría hacerse implícitamente (de acuerdo al comportamiento del usuario) o en forma explícita solicitando al usuario que alimente su propio perfil con el propósito de hacer búsquedas más precisas, agregándole términos claves o removiendo los que ya no sean de interés. En los algoritmos probados, se trabaja con el perfil que se crea a partir de los datos de la base de datos de EachMovie y no se modifica, esto es una limitante para el filtrado colaborativo.

7. TRABAJO FUTURO

Una propuesta de mejoramiento para los sistemas de filtrado de información es la incorporación del Razonamiento Basado en Casos (CBR Case Based Reasoning), concebido como una metodología para modelar el razonamiento humano, resolviendo problemas al adaptar soluciones usadas en problemas anteriores. Cada caso contiene una descripción del problema, más una solución y/o resultado. El proceso de razonamiento usado para resolver el problema no se almacena, pero está implícito en la solución. Para resolver un problema, este se compara con los casos en la base de casos y se recuperan los que sean similares. Los casos recuperados se usan para sugerir una solución que luego es reusada y probada. Finalmente el problema actual y la solución se guardan como parte de un nuevo caso. Cotter y Smyth en su trabajo "PTV: Intelligent personalized tv guides", propone la combinación de tres sistemas de recomendación: Uso del perfil de usuario, CBR y el filtrado colaborativo.

Para continuar trabajando con la base de datos de EachMovie, se presenta la posibilidad de usar otros datos que podrían mejorar la efectividad del filtrado basado en contenido, al manejar más información de cada ítem, por ejemplo, teniendo en cuenta no solo el género al que pertenece la película, sino datos como los actores principales, el director, la fecha de realización, etc., que permitieran encontrar más puntos de coincidencia y mejores vecinos para realizar la predicción.

Se recomienda la utilización de otras métricas para la medición de la eficiencia de los algoritmos, por ejemplo, F-Measure (que combina Recall and Precision) o las curvas ROC.
Otra recomendación es probar la eficiencia de los modelos propuestos con otra base de datos, para comparar el comportamiento de los algoritmos en diferentes escenarios para determinar que tan dependientes de los datos son los modelos o si los resultados encontrados siguen el mismo patrón sin importar la fuente.

En la literatura que se encuentra en el área del filtrado de información y sistemas de recomendación, se presentan diversas alternativas buscando mejorar la efectividad de las predicciones, tal es el caso del trabajo realizado por Soboroff y Nicholas [40], que proponen la combinación de filtrado colaborativo y por contenido usando Indexación Semántica Latente o la incorporación de sistemas multiagentes (MAS Multi-Agent Systems) como lo presentan Delgado e Ishii [41], en el estudio de sistemas de recomendación basados en agentes (ARS Agent-based Recommender Systems), definiendo un agente para cada usuario y como funciones objetivo de aprendizaje las preferencias de los usuarios. Estas y otras

alternativas pueden incorporase a los diferentes modelos combinados así como la búsqueda de nuevas técnicas de integración.

La posibilidad de utilizar el concepto de "*Filterbot*" o Robot de Filtrado [38], como una herramienta para el *rankeo* automático, que se encargue de evaluar nuevos documentos tan pronto como sean publicados, llevarían a tener un sistema con los últimos documentos disponibles, listos para ser utilizados por un filtrado colaborativo. El sistema de filtrado colaborativo trata al *Filterbot* como otro usuario más, muy prolífico, que introduce muchos *ratings*, pero no hace ninguna solicitud de predicción. El motor del sistema no necesita saber si los usuarios son *Filterbots* o humanos.

Este *Filterbot* es un agente que es llamado cada vez que llega un nuevo documento y retorna un valor numérico (entre 1 y 5 o dependiendo de la escala utilizada). Para esto se vale de un análisis de contenido. El sistema de filtrado colaborativo personaliza la asignación de pesos a cada *Filterbot*, así, si un usuario coincide consistentemente con el mismo *filterbot*, este tendrá un peso alto para ese usuario. Los resultados de esta propuesta muestran otra manera de integrar los *ratings* basados en contenido en un sistema de filtrado colaborativo.

8. REFERENCIAS BIBLIOGRÁFICAS

[1] Sarvar, B., Karypis, G., Konstant, J., and Riedl, J. Item-based Collaborative Filtering Recommendation Algorithms. In WWW10 Conference. May, 2001

[2] Breese, J., Heckerman, D., and Kadie, C. Empirical Analysis of Predictive Alforithms for Collaborative Filtering. In Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence. Madison, 1998.

[3] Sarwar, B., Karypis, G., Konstant, J., and Riedl, J. Analysis of Recommendation Algorithms for E-Commerce. In Proceedings of the ACM EC'00 Conference. Minneapolis, MN. 2000

[4] Compaq Systems Research Center. EachMovie Collaborative Filtering data set. http://www.research.compaq.com/SCR/eachmovie/ 1.997 – 2001

[5] Melville, P., Mooney, R., Nagarajan, R. Content-Boosted Collaborative Filtering. In Proceedings of the SIGIR-2001 Workshop on Recommender Systems, New Orleans, LA, 2001.

[6] Herlocker, J., Konstan, J., Borchers, A. and Riedl, J. An algorithmic framework for performing collaborative filtering. In SIGIR'99: Proceedings of the 22nd Anual International ACM SIGIR Conference on Research and Development in Information Retrieval. 1999

[7] Wise, J., Thomas, J., Pennock, K., Lantip, D., Pottier, M., Schur, A., and Crow, V. Visualizing the non-visual: spatial analysis and interaccion with information from text documents. Pacific Northwest Lab, Richland, WA. 1.995

[8] Maes, P.Agents that Reduce Work and Information Overload. Communication of the ACM, 37 (7):30-40. July 1994.

[9] Van Rijsbergen, C. J. Information Retrieval. Computing Science. Computing Science University of Glasgow. http://www.dcs.gla.ac.uk/Keith/Chapter.1/ch.1.html 1979.

[10] Belkin, N.J. and Croft, W. B. Information Filtering and Information Retrieval: Two Sides of the Same Coin. Communicatios of the ACM, 35(12):29-38. December 1992.

[11] Resnick, P., Varian, H.R., and Guest Editors. Recommender Systems. Communicatios of the ACM, 40(3):56-58. March 1997.

[12] Goldberg, D., Nichols, D., Oki, B. M., and Terry, D. Using Collaborative Filtering to weave an Information Tapestry. Communicatios of the ACM, 35(12):61-70. December 1992.

[13] Sánchez Pereira, A. La Diseminación Selectiva de Información a través de Internet. Dirección General de Bibliotecas de la Universidad Nacional Autónoma de México. http://dgbiblio.unam.mx

[14] Balabanovic, M. and Shoham Y. Fab: Content-Based, Collaborative Recommendation. Communicatios of the ACM, 40(3):66-72. March 1997.

[15] Baudisch, P. Joining Collaborative and Content-based filtering. German National Research Center for Information Technolgy GMD. Darmstadt, Germany. http://www.darmstadt.gmd.de/~baudisch/Publications 1999.

[16] Polcicová, G. and Návrat, P. Combining Content-based and Collaborative Filtering. Slovak University of Technology, Department of Computer Science and Engineering, Bratislava, Slovakia. http://www.dcs.elf.stuba.sk 2000.

[17] Armtrong, R., Freitag, D., Joachims, T., and Mitchell, T. WebWatcher: A learning apprentice for the world wide web".
http://www-2.cs.cmu.edu/afs/cs.cmu.edu/project/theo-6/web-agent/www/project-home.html 1995.

[18] Lieberman, H. Letizia: An Agent that Assists Web Browsing. http://lieber.www.media.mit.edu/people/lieber/Lieberary/Letizia/Letizia-AAAI/Letizia.ps 1995.

[19] Pazzani, M., Muramatzu, J., and Billsus, D. Syskill & Webert: Identifying interesting web sites. http://www.ics.uci.edu/~pazzani/Publications/aaai-webert.pdf 1996.

[20] Claypool, M., Gokhale, A., Miranda, T., Murnikov, P., Netes, D., and Sartin, M. Combinig Content-Based and Collaborative Filters in an Online Newspaper. http://citeseer.nj.nec.com/claypool99combining.html 1999.

[21] Cotter, P. And Smyth, B. PTV, Personalised Services for Digital TV http://www.ptvplus.com/ptv/home.jsp 2000.

[22] Good, N., Schafer, J., Konstan, J., Borchers, A., Sarwar, B., Herlocker, J. And Riedl, J. Movielens: Helping you find the right movies.
http://www.movielens.umn.edu/ 1999

[23] Bradley, K., Rafter, R., and Smyth, B. Case-Based User Profiling for Content Personalisation. http://kermit.ucm.ie/casper/ah2000bradley.ps 2000

[24] Vel, O., and Nesbitt, S. A Collaborative filtering Agent System for Dynamic Virtual Communities on the web. http://citeseer.nj.nec.com/de-collaborative.html 1997

[25] Bueno, D., Conejo, R., and David, A. METIOREW: An Objective Oriented Content Based and Collaborative Recommending System. http://wwwis.win.tue.nl/ah2001/papers/METIOREW.pdf 2001.

[26] Herlocker, J., Konstan, J., Borchers, A. and Riedl, J. Explaining Collaborative Filtering Recommendations. http://cs.oregonstate.edu/~herlock/papers/explanations_cscw2000.pdf 2000.

[27] Gokhale, Anuja. Improvements to Collaborative Filtering Algorithms. Worcester Polytechnic Institute. http://www.cs.wpi.edu/~claypool/ms/cf-improve/cf-improve.ps 1999.

[28] Basu, C., Hirsh, H., and Cohen, W. Recommendation as Classification: Using Social and Content-Based Information in Recommendation. http://citeseer.nj.nec.com/basu98recommendation.html 1988.

[29] Delgado, J., Ishii, N., and Ura, T. Content-based Collaborative Information Filtering: Actively Learning to Classify and Recommend Documents. http://citeseer.nj.nec.com/delgado98intelligent.html 1988.

[30] Van Metern, R. and Van Someren, M. Using Content-Based Filtering for Recommendation. http://www.ics.forth.gr/~potamias/mlnia/paper_6.pdf

[31] Baeza-Yates, R. and Ribeiro-Neto, B. Modern Information Retrieval. Capítulos 1 al 5. Hatlow, England. Addison Wesley. 1999.

[32] Porter, M. The Porter Stemming Algorithm. http://www.tartarus.org/~martin/PorterStemmer/

[33] Zazo Rodríguez, A, Figuerola, P., Alonso, J., y Gómez, R. Recuperación de información utilizando el modelo vectorial. Participación en el taller CLEF-2001. Informe Técnico. http://tejo.usal.es/inftec/2002/DPTOIA-IT-2002-006.pdf 2002.

[34] Papadimitriou, C., Ragharan, P., Tamaki, H. and Vempala, S. Latente Semantic Indexing: A probabilistic Analysis. http://citeseer.nj.nec.com/papadimitriou98latent.html 1.997.

[35] Pereyra, A. La diseminación selectiva de información a través de internet: propuesta de boletín electrónico de información bibliográfica especializada en economía, utilizando la base de datos CLASE. Universidad Nacional Autónoma de

México.
http://www.dgbiblio.unam.mx/servicios/dgb/publicdgb/bole/fulltext/volII1/clase.html

[36] Fisher, D., Hildrum, K., Hong, J., Newman, M., Thomas, M. and Vuduc, R. SWAMI: a framework for collaborative filtering algorithm development and evaluation. http://guir.cs.berkeley.edu/projects/swami/swami-paper/paper.html

[37] Wang, P. Java. Con programación orientada a objetos y aplicaciones en la WWW. International Thomson Editores. 2000. Pág. 2-9

[38] Sarwar, B., Konstan, J., Borchers, A., Herlocker, J., Miller, B. y Riedl, J.. Using Filtering Agents to Improve Prediction Quality in the GroupLens Research Collaborative Filtering System. http://www.cs.umn.edu/Research/GroupLens/papers/pdf/filterbot-CSCW98.pdf Copyrigth ACM 1998.

[39] Cotter, P. y Smyth, B. PTV: Intelligent personalized tv guides. http://www.cs.ucd.ie/students/pcotter/default.htm In Twelfth Conference on Innovative Applications of Artificial Intelligence, 2000.

[40] Soboroff, I. y Nicholas, C. Combining content and collaboration in text filtering. http://citeseer.nj.nec.com/soboroff99combining.html 1.999.

[41] Delgado, J. e Ishii, N. Multi-Agent Learning in Recommender Systems for Informaction Filtering on the Internet. http://www.informatik.uni-trier.de/~ley/db/indices/a-tree/i/Ishii:Naohiro.html 2001.

BIBLIOGRAFÍA

Armtrong, R., Freitag, D., Joachims, T., and Mitchell, T. WebWatcher: A learning apprentice for the world wide web".
http://www-2.cs.cmu.edu/afs/cs.cmu.edu/project/theo-6/web-agent/www/project-home.html 1995.

Baeza-Yates, R. and Ribeiro-Neto, B. Modern Information Retrieval. Capítulos 1 al 5. Hatlow, England. Addison Wesley. 1999.

Balabanovic, M. and Shoham Y. Fab: Content-Based, Collaborative Recommendation. Communicatios of the ACM, 40(3):66-72. March 1997.

Basu, C., Hirsh, H., and Cohen, W. Recommendation as Classification: Using Social and Content-Based Information in Recommendation.
http://citeseer.nj.nec.com/basu98recommendation.html 1988.

Baudisch, P. Joining Collaborative and Content-based filtering. German National Research Center for Information Technolgy GMD. Darmstadt, Germany. http://www.darmstadt.gmd.de/~baudisch/Publications 1999.

Belkin, N.J. and Croft, W. B. Information Filtering and Information Retrieval: Two Sides of the Same Coin. Communicatios of the ACM, 35(12):29-38. December 1992.

Bradley, K., Rafter, R., and Smyth, B. Case-Based User Profiling for Content Personalisation. http://kermit.ucm.ie/casper/ah2000bradley.ps 2000.

Breese, J., Heckerman, D., and Kadie, C. Empirical Analysis of Predictive Alforithms for Collaborative Filtering. In Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence. Madison, 1998.

Bueno, D., Conejo, R., and David, A. METIOREW: An Objective Oriented Content Based and Collaborative Recommending System.
http://wwwis.win.tue.nl/ah2001/papers/METIOREW.pdf 2001.

Claypool, M., Gokhale, A., Miranda, T., Murnikov, P., Netes, D., and Sartin, M. Combinig Content-Based and Collaborative Filters in an Online Newspaper. http://citeseer.nj.nec.com/claypool99combining.html 1999.

Compaq Systems Research Center. EachMovie Collaborative Filtering data set. http://www.research.compaq.com/SCR/eachmovie/ 1.997 – 2001

Compendio Tesis y otros Trabajos de Grado. Instituto Colombiano de Normas Técnicas y Certificación (ICONTEC). Bogotá, Colombia. 2002

Cotter, P. y Smyth, B. PTV: Intelligent personalized tv guides. http://www.cs.ucd.ie/students/pcotter/default.htm In Twelfth Conference on Innovative Applications of Artificial Intelligence, 2000.

Cotter, P. y Smyth, B. PTV, Personalised Services for Digital TV http://www.ptvplus.com/ptv/home.jsp 2000.

Delgado, J., Ishii, N., and Ura, T. Content-based Collaborative Information Filtering: Actively Learning to Classify and Recommend Documents. http://citeseer.nj.nec.com/delgado98intelligent.html 1988.

Delgado, J. e Ishii, N. Multi-Agent Learning in Recommender Systems for Informaction Filtering on the Internet. http://www.informatik.uni-trier.de/~ley/db/indices/a-tree/i/Ishii:Naohiro.html 2001.

Fisher, D., Hildrum, K., Hong, J., Newman, M., Thomas, M. and Vuduc, R. SWAMI: a framework for collaborative filtering algorithm development and evaluation. http://guir.cs.berkeley.edu/projects/swami/swami-paper/paper.html

Gokhale, Anuja. Improvements to Collaborative Filtering Algorithms. Worcester Polytechnic Institute. http://www.cs.wpi.edu/~claypool/ms/cf-improve/cf-improve.ps 1999.

Goldberg, D., Nichols, D., Oki, B. M., and Terry, D. Using Collaborative Filtering to weave an Information Tapestry. Communicatios of the ACM, 35(12):61-70. December 1992.

Good, N., Schafer, J., Konstan, J., Borchers, A., Sarwar, B., Herlocker, J. And Riedl, J. Movielens: Helping you find the right movies.
http://www.movielens.umn.edu/ 1999

Herlocker, J., Konstan, J., Borchers, A. and Riedl, J. An algorithmic framework for performing collaborative filtering. In SIGIR'99: Proceedings of the 22nd Anual International ACM SIGIR Conference on Research and Development in Information Retrieval. 1999

-------- Explaining Collaborative Filtering Recommendations.
http://cs.oregonstate.edu/~herlock/papers/explanations_cscw2000.pdf 2000.

Jirón, M. y Olivares, L. DSI Escolar: Un timón para no naufragar en internet. http://www.bibliotecarios.cl/conferencia/documentos/jiron.pdf

Lieberman, H. Letizia: An Agent that Assists Web Browsing. http://lieber.www.media.mit.edu/people/lieber/Lieberary/Letizia/Letizia-AAAI/Letizia.ps 1995.

Maes, P.Agents that Reduce Work and Information Overload. Communication of the ACM, 37 (7):30-40. July 1994.

Melville, P., Mooney, R., Nagarajan, R. Content-Boosted Collaborative Filtering. In Proceedings of the SIGIR-2001 Workshop on Recommender Systems, New Orleans, LA, 2001.

Papadimitriou, C., Ragharan, P., Tamaki, H. and Vempala, S. Latente Semantic Indexing: A probabilistic Analysis. http://citeseer.nj.nec.com/papadimitriou98latent.html 1.997.

Pazzani, M., Muramatzu, J., and Billsus, D. Syskill & Webert: Identifying interesting web sites. http://www.ics.uci.edu/~pazzani/Publications/aaai-webert.pdf 1996.

Polcicová, G. and Návrat, P. Combining Content-based and Collaborative Filtering. Slovak University of Technology, Department of Computer Science and Engineering, Bratislava, Slovakia. http://www.dcs.elf.stuba.sk 2000.

Porter, M. The Porter Stemming Algorithm. http://www.tartarus.org/~martin/PorterStemmer/

Resnick, P., Varian, H.R., and Guest Editors. Recommender Systems. Communicatios of the ACM, 40(3):56-58. March 1997.

Sánchez Pereira, A. La Diseminación Selectiva de Información a través de Internet. Dirección General de Bibliotecas de la Universidad Nacional Autónoma de México. http://dgbiblio.unam.mx

Sarwar, B., Karypis, G., Konstant, J., and Riedl, J. Analysis of Recommendation Algorithms for E-Commerce. In Proceedings of the ACM EC'00 Conference. Minneapolis, MN. 2000

-------- Item-based Collaborative Filtering Recommendation Algorithms. In WWW10 Conference. May, 2001

Sarwar, B., Konstan, J., Borchers, A., Herlocker, J., Miller, B. y Riedl, J.. Using Filtering Agents to Improve Prediction Quality in the GroupLens Research Collaborative Filtering System. http://www.cs.umn.edu/Research/GroupLens/papers/pdf/filterbot-CSCW98.pdf Copyrigth ACM 1998.

Soboroff, I. y Nicholas, C. Combining content and collaboration in text filtering. http://citeseer.nj.nec.com/soboroff99combining.html 1.999.

Van Metern, R. and Van Someren, M. Using Content-Based Filtering for Recommendation. http://www.ics.forth.gr/~potamias/mlnia/paper_6.pdf

Van Rijsbergen, C. J. Information Retrieval. Computing Science. Computing Science University of Glasgow. http://www.dcs.gla.ac.uk/Keith/Chapter.1/ch.1.html 1979.

Vel, O., and Nesbitt, S. A Collaborative filtering Agent System for Dynamic Virtual Communities on the web. http://citeseer.nj.nec.com/de-collaborative.html 1997

Wang, P. Java. Con programación orientada a objetos y aplicaciones en la WWW. International Thomson Editores. 2000. Pág. 2-9

Wise, J., Thomas, J., Pennock, K., Lantip, D., Pottier, M., Schur, A., and Crow, V. Visualizing the non-visual: spatial analysis and interaccion with information from text documents. Pacific Northwest Lab, Richland, WA. 1.995

Zazo Rodríguez, A, Figuerola, P., Alonso, J., y Gómez, R. Recuperación de información utilizando el modelo vectorial. Participación en el taller CLEF-2001. Informe Técnico. http://tejo.usal.es/inftec/2002/DPTOIA-IT-2002-006.pdf 2002.

ANEXO A. ALGORITMOS

MODELO 1. FILTRADO COLABORATIVO. NUMERAL 4.3.1

```
Inicio
  /*  Rating[ ][ ] es la matriz de Usuarios X Items
  /*  K[ ][ ] es la matriz de correlaciones Usuarios * Usuarios
  /*  Promedio[ ] es el vector de promedios de los ratings dados por los usuarios

  Función Llena_Matriz de Usuarios X Items (Datos de EachMovie)

  T2 = constante de umbral de correlación

  para cada usuario u = 1 hasta num_usuarios haga
    cont=0
    suma_rating = 0
    para cada ítem i = 1 hasta num_ítem haga
      si Rating[u][i] ǂ 0 entonces  /* Si el usuario u calificó al ítem i */
        suma_rating = suma_rating + Rating[u][i]
        cont = cont + 1
      finsi
    fin para
    Promedio[u] = suma_rating / cont
  fin para
  para cada usuario c = 1 hasta num_usuarios haga
    para cada usuario u = 1 hasta num_usuarios haga
      si c ǂ u entonces
        K[c][u] = Cálculo Coeficiente de Correlación de Pearson (Rating[c], Rating[u])
      fin si
    fin para
    para cada ítem i = 1 hasta num_ítem haga
      sum1 = 0
      sum2 = 0
      para cada usuario u = 1 hasta num_usuarios haga
        si Rating[u][i] > 0 y K[c][u] > T2
          sum1 = sum1 + (Rating[u][i] – Promedio[u]) * K[c][u]
          sum2 = sum2 + | k[c][u] |
        finsi
      fin para
      Estimado_colaborativo[c][i] = Promedio[c] + (sum1 / sum2)
    fin para
  fin para
fin
```

MODELO 2. FILTRADO BASADO EN CONTENIDO. NUMERAL 4.3.2

```
Inicio
  /* Pesos [ ][ ]  Matriz de pesos de  Items X Palabras

  Función Llena_Matriz de Usuarios X Items (Datos de EachMovie)

  para cada ítem i = 1 hasta num_ítems haga
    para cada palabra w = 1 hasta num_palabras_claves haga
        Pesos[i][w] = Cálculo del peso de la palabra w para el ítem i
    fin para
    Normalizar (Pesos[i])
  fin para
  para cada usuario c = 1 hasta num_usuarios haga
    para cada ítem i = 1 hasta num_ítems haga
      para cada palabra w = 1 hasta num_palabras_claves haga
        Perfil[i][w] = Rating[c][i] * Pesos[i][w]
      fin para
    fin para
    Normalizar (Perfil[i])
    para cada ítem i = 1 hasta num_ítems haga
      para cada palabra w = 1 hasta num_palabras_claves haga
        s = s + perfil[i][w] • items[i][w]
      fin para
      Estimado_contenido [c][i] = Asignar_estimado(s)
    fin para
  fin para
Fin
```

MODELO 3: USANDO FILTRADO COLABORATIVO MEJORADO POR CONTENIDO DESCRITO EN EL NUMERAL 4.3.3

```
Inicio

  Función Llena_Matriz de Usuarios X Items (Datos de EachMovie)
  Max = 2 ( constante )

  /* Llenado de la matriz  Pseudo_rating[u][i] con valores estimados por contenido */
  para cada usuario u = 1 hasta num_usuarios haga
    cont = 0
    para cada ítem i = 1 hasta num_ítems haga
      si Rating[u][i] = 0 entonces
        Pseudo_rating[u][i] = Rating estimado por contenido (Numeral 4.3.2)
      sino
        Pseudo_rating[u][i] = rating[u][i]
        cont = cont +1
      finsi
    fin para
    num[u] = cont
  fin para
  para cada usuario u = 1 hasta num_usuarios haga
    para cada ítem i  = 1 hasta num_ítems haga
      si rating[u][i] = 0 entonces

        /* Algoritmo Colaborativo */
        para cada usuario c = 1 hasta num_usuario
          suma_rating = 0
          para cada ítem j = 1 hasta num_ítems haga
            suma_rating = suma_rating + Pseudo_rating[c][j]
            si rating[u][j] ‡ 0 y rating[c][j] ‡ 0
              n = n + 1
            fin si
          fin para
          Promedio[c] = suma_rating / num_items
        fin para
        para cada usuario c = 1 hasta num_usuario haga
          si c ‡ u entonces
            K[c] = Coef. de Corr. de Pearson (Pseudo_rating[u] , Pseudo_rating[c])
          finsi
        fin para
        para cada ítem j = 1 hasta num_ítems haga
          sum1 = 0
          sum2 = 0
          para cada usuario c = 1 hasta num_usuario haga
            si K[u] > T2 entonces
              /*  Factores de Ponderación */

              /*  Factor Ponderado de Significancia */
```

```
                si n < 50 entonces
                   sg[u][c] = n / 50
                sino
                  sg[u][c] = 1
                finsi

                /* Factor de Ponderación armónico medio */
                hm[u][c] = (2 * num[u] * num[c]) / (num[u] + num[c])

                /* Peso de correlación híbrido */
                hw[u][c] = hm[u][c] + sg[u][c]

                /* Factor de ponderación propio */
                si num[c] < 50 entonces
                  sw[c] = num[c] / 50 * max
                sino
                  sw[c] = max
                finsi
                sum1 = sum1 + (Pseudo_rating[c][j] – Promedio[c]) * K[c] * hw[u][c]
                sum2 = sum2 + | K[c] | * hw[u][c]
              finsi
            fin para
          fin para
        fin si
        Estimado_colaborativo[u][i] = Promedio[u] + (sw[u] * Pseudo_rating[u][i] –
                                                     Promedio[u] +  sum1 / (sw[u] + sum2))
    fin para
  fin para
fin
```

MODELO 4. MODELO COMBINADO PARA RECOMENDACIÓN DE INFORMACIÓN, USANDO FILTRADO BASADO EN CONTENIDO Y COLABORATIVO, DESCRITO EN EL NUMERAL 4.3.4

```
Inicio
  T2 = constante de umbral de correlación
  Función Llena_Matriz de Usuarios X Items (Datos de EachMovie)

  /*  Inicialización */

  /*  k_r[][] Matriz de correlación Usuario X Usuario para los ratings reales. */
  /*  k_e[][] Matriz de correlación Usuario X Usuario para los ratings calculados por */
    contenido.

  /*  Cálculo del estimado colaborativo */
  para cada usuario u = 1 hasta num_usuarios haga
    para cada usuario c = 1 hasta num_usuarios haga
      k_r[c] = Coeficiente de Correlación de Pearson (Rating[u], Rating[c])

      /*La matriz k_e[] se llena con los ratings de los usuarios u y c, y con el estimado por
        contenido, cuando a alguno de los dos usuarios comparados le falte un rating. */

      k_e[c] = Coeficiente de Correlación de Pearson (Rating[u], Rating[c] y los Estimados
                por contenido)
    fin para
    para cada ítem i = 1 hasta num_item haga
      sum1 = 0
      sum2 = 0
      para cada usuario c = 1 hasta num_usuarios haga
        si Rating[c][i] > 0 y k_r[c] > T2 entonces
          sum1 = sum1 + (Rating[c][i] – Promedio[c]) * k_r[c]
          sum2 =  sum2 + |k_p[c]
        finsi
        si Estimado[c][i]  y k_e[c] > T2 entonces
          sum1 = sum1 + (Estimado[c][i] – Promedio[c]) * k_e[c]
          sum2 =  sum2 + |k_e[c]
        finsi
      fin para
      Estimado_colaborativo[u][i] = Promedio[u] + sum1/sum2
    fin para
  fin para
fin.
```

MODELO 5. USANDO LA COMBINACIÓN DE FILTRADO COLABORATIVO Y CONTENIDO, PROPUESTA POR [20] Y DESCRITA EN EL NUMERAL 4.3.5

```
Inicio

  Función Llena_Matriz de Usuarios X Items (Datos de EachMovie)

  */ W1 es el peso inicial para el filtrado colaborativo. */
  */ W2 es el peso inicial para el filtrado por contenido. */

  W1 = 0.5
  W2 = 0.5

  para cada usuario u = 1 hasta num_usuario haga
    para cada ítem i = 1 hasta num_ítems haga
      si Rating[u][i] ‡ 0 entonces
        cont = cont +1
      fin si
    fin para
    Incremento = 0.1 / cont

    /* Para calcular una predicción */
    para cada ítem i = 1 hasta num_ítems haga
      si Rating[u][i] = 0 entonces
        Rating_Estimado=W1*Estimado_Colaborativo + W2*Estimado_Basado_Contenido
      fin si
    fin para

    /* Para cada nuevo rating del usuario, se recalculan los pesos */
    si |Rating[u][i] - Estimado_Colaborativo| < |Rating[u][i] – Estimado_Basado_Contenido|
       W1 = W1 + Incremento
       W2 = W2 - Incremento
    fin si
    si |Rating[u][i] - Estimado_Colaborativo| > |Rating[u][i] – Estimado_Basado_Contenido|
       W1 = W1 - Incremento
       W2 = W2 + Incremento
    fin si
  fin para
fin
```

CÁLCULO DE LA MEDIA DEL ERROR ABSOLUTO

/* Para todos los modelos, el cálculo se realizó sustrayendo de la matriz de Usuarios X Documentos, cada uno de los valores dados por el usuario y hallando el estimado corespondiente. Para el cálculo de la Media del Error Absoluto para cada usuario se encontró la diferencia entre el valor real dado por el usuario y el valor estimado por el modelo. */

```
Inicio
  S2 = 0
  para cada usuario u = 1 hasta num_usuarios haga
    S1 = 0
    Cont = 0
    para cada ítem i = 1 hasta num_ítems haga
      si Rating[u][i ] ǂ 0
        cont = cont + 1
        valor_real = Rating[u][i]
        Rating[u][i] = 0
      fin si

      /*El estimado se calcula de acuerdo al modelo que se esté evaluando (uno de los 5
      modelos explicados */
      valor_estimado = Estimado_Modelo (usuario[u], item[i])
      S1 = S1 + ( |valor_real – valor_estimado|)
    fin para
    S2 = S2 + S1/cont
  fin para
  MAE = S2 / num_usuarios
fin
```