UNIVERSIDAD DE OVIEDO

ESCUELA DE INGENIERÍA INFORMÁTICA
UNIVERSIDAD DE OVIEDO

PROYECTO FIN DE MÁSTER

"Tecnologías Semánticas para la recepción, edición y distribución de vídeo digital"

Vº Bº del Director del Proyecto	**AUTOR: Alejandro Montes García** **DIRECTOR: Jose Emilio Labra Gayo** **CODIRECTOR: Jose María Álvarez Rodríguez**

Agradecimientos

- A Chema, Labra y el grupo WESO en general, por confiar en mí y ayudarme a dar mis primeros pasos como investigador.
- A mis compañeros de clase por estos dos fantásticos años.
- Al equipo de QueYeHo, por QueYeHosear y frikear.
- Y por supuesto, y más que a nadie a mi familia y amigos.

Resumen

La red ha transformado los procesos de búsqueda de información e investigación documental en la elaboración de contenidos periodísticos pero empiezan a denotar la necesidad de una tecnología más precisa ya que la sobrecarga de información, también conocida como *information pollution*, se ha puesto de manifiesto como un grave problema.

La búsqueda de información en la red empezó como un complemento de las fuentes de información tradicionales (contactos, hemerotecas, consultas a expertos, etc.) La transformación de los procesos de producción de los medios de información durante los últimos años ha terminado con ese equilibrio para situar Internet como la fuente principal -e incluso única en la mayoría de los casos- de la búsqueda de información en la elaboración de contenidos periodísticos. Un dominio aplastante que ha terminado por descubrir la imprecisión de las herramientas de búsqueda disponibles actualmente. Para solucionar este problema se propone Freews.

Freews es una arquitectura en web semántica para la recepción, edición y distribución de vídeos. Su objetivo es proporcionar un sitio web en el que periodistas amateur o profesionales puedan intercambiar vídeos con los que ilustrar sus noticiarios.

Freews abre cuatro líneas de investigación, sobre recomendación de noticias y de etiquetas, búsqueda y etiquetado de contenido. Dado que realizar un trabajo intensivo en estas cuatro líneas es inabordable para un proyecto fin de máster, se ha limitado el trabajo completo de investigación a la recomendación de noticias, dejando las otras tres en un trabajo de investigación en el que solo se recoge el estado del arte, sin aportar ninguna contribución.

Palabras Clave

Filtrado colaborativo, Procesamiento del Lenguaje Natural, Recomendación basada en contenido, Recomendación de etiquetas, Recomendación de noticias, Web Semántica.

Abstract

The Internet has transformed the processes of finding information and documentary research as far as journalistic content development is concerned, but it needs to move a step forward and the need of filtering all this information has shown up.

The documentary research on the Internet started out as a complement of the traditional sources of information (contacts, press archives, consultations with experts, etc). The metamorphosis of the production processes in the media during the last years has ended with this equilibrium, making the Internet the main, and sometimes the only, source of information as far as journalistic content search is concerned.

Freews is a semantic web architecture for receiving, editing and distributing videos. It aims to provide a website in which amateur or professional journalists can exchange videos that illustrate their news.

Freews opens four research lines: news and tag recommendation, search and semantic content. Since intensive work in these four lines is unapproachable for a Master Thesis, full research work has been targeted to the first research line, news recommendation. The other three research lines has been limited to state of the art research.

Keywords

Content-Based recommendation, Collaborative Filtering, Natural Language Processing, News recommendation, Semantic Web, Tag recommendation.

Índice general

Índice de figuras

Capítulo 1

Introducción

Freews es una arquitectura en web semántica para la recepción, edición y distribución de vídeos. Su objetivo es proporcionar un sitio web en el que periodistas amateur o profesionales puedan intercambiar vídeos con los que ilustrar sus noticiarios. Freews también recomendará nuevos vídeos a sus usuarios que puedan recomendarle.

Un equipo de periodistas profesionales estará trabajando en Freews para moderar contenido y comprobar que cada vídeo contiene la información correcta, de modo que se facilite la búsqueda de material.

1.1. Líneas de investigación

Freews abre cuatro líneas de investigación, sobre recomendación de noticias y de etiquetas, búsqueda y etiquetado de contenido. Dado que realizar un trabajo intensivo en estas cuatro líneas es inabordable para un proyecto fin de máster, se ha limitado el trabajo completo de investigación a la recomendación de noticias, dejando las otras tres en un trabajo de investigación en el que solo se recoge el estado del arte, sin aportar ninguna contribución.

1.1.1. Recomendación

Dado que cada vídeo en Freews representa una noticia, la recomendación de nuevo contenido se hace en base a esa noticia en lugar de recomendar contenido multimedia.

Tradicionalmente, esta recomendación de contenido se basa en técnicas de filtrado colaborativo Filtrado colaborativo, donde solo se tiene en cuenta un histórico de puntuaciones. Esto es válido para contenido multimedia como películas, música, etc... La recomendación de noticias, sin embargo, es distinta debido a la naturaleza de las noticias.

Una noticia deja de ser relevante si ha pasado un tiempo desde que sucedió, pero quizás tenga algo más de relevancia, aunque sea antigua, si el usuario está leyendo sobre el tema. Recomendadores como PENETRATE [74] y SCENE [45] son más específicos para recomendar noticias, pero no tienen en cuenta factores como la credibilidad de las fuentes de información.

El recomendador desarrollado para Freews, llamado Wesomender es un sistema de recomendación sensible al contexto para facilitar a los usuarios de Freews (periodistas amateurs o profesionales) la identificación de temas similares entre distintas fuentes.

Las características contextuales de las noticias son tenidas en cuenta debido a esta naturaleza especial que se mencionaba anteriormente: fecha, intereses actuales del usuario, localización y credibilidad de la fuente son combinadas junto con las técnicas utilizadas por sistemas de recomendación tradicionales para construir un framework adaptativo que provee un sistema de filtrado colaborativo potenciado.

1.1.2. Búsqueda

Freews necesita un buscador potente de modo que se pueda explotar aún más la presencia de un grupo de profesionales que modera el contenido y se asegura de que el mismo sea correcto. Para ello se pueden aplicar técnicas de Procesamiento del Lenguaje Natural y combinarlas con enriquecimiento de búsquedas utilizando datos enlazados de un modo similar al descrito en WESONet [28].

En combinación con el sistema de recomendación, debería ser muy sencillo encontrar el contenido deseado y ordenado de modo que el usuario verá primero las noticias que más posibilidades tienen de interesarle.

1.1.3. Semántica

Dado que Freews es una arquitectura en web semántica, es necesaria la descripción de las noticias usando datos enlazados. Existen varias formas de hacer esto, usando microdatos, RDFa, microformatos, etc... En Freews se ha optado por el uso de microdatos, puesto que equilibra la limpieza del código con las posibilidades semánticas del mismo.

El uso de este tipo de tecnologías permitirá a cualquier dispositivo conocer mejor la naturaleza del contenido ofertado y actuar en consecuencia.

1.1.4. Recomendación de etiquetas

Para ayudar al equipo de profesionales a etiquetar noticias correctamente, es necesario aplicar un sistema de recomendación de etiquetas en los tres idiomas en los que estará disponible Freews, español, inglés y portugués. También es necesario que estas etiquetas hagan referencia al mismo concepto independientemente del idioma en el que estén. Por ejemplo, si el sistema recomienda la etiqueta *"Deportes"* para la versión en español, deberá recomendar *"Sports"* y *"Esportes"* para inglés y portugués respectivamente, de modo que se eviten inconsistencias entre idiomas.

1.2. Motivación

La red ha transformado los procesos de búsqueda de información e investigación documental en la elaboración de contenidos periodísticos pero empiezan a denotar la necesidad de una tecnología más precisa ya que la sobrecarga de información, también conocida como *information pollution*, [51] se ha puesto de manifiesto como un grave problema.

La búsqueda de información en la red empezó como un complemento de las fuentes de información tradicionales (contactos, hemerotecas, consultas a expertos, etc.) La transformación de los procesos de producción de los medios de información durante los últimos años ha terminado con ese equilibrio para situar Internet como la fuente principal -e incluso única en la mayoría de los casos- de la búsqueda de información en la elaboración de contenidos periodísticos. Un dominio aplastante que ha terminado por descubrir la imprecisión de las herramientas de búsqueda disponibles actualmente.

La herramientas de búsqueda actuales son imprecisas, por lo que no ofrecen la variedad informativa deseada o son incapaces de permitir acotaciones más precisas al profesional que busca una información concreta que permita seguir el recorrido histórico de una noticia.

Portales empleados mayoritariamente como GoogleNews[1] o GoogleAlerts[2] garantizan objetivos estimables como la inmediatez pero sin ofrecer el rigor, ni la precisión necesarios en la búsqueda profesional de información. De hecho, sus motores de búsqueda no tienen en cuenta el rigor periodístico, ni la fuente que ha creado la información buscada. En un alto porcentaje de los casos la información que ofrecen estos portales en sus primeras páginas proceden de diferentes medios pero tienen el mismo contenido porque están publicando la

[1] `https://news.google.es/`
[2] `http://www.google.es/alerts`

nota de prensa o el teletipo de la misma empresa o agencia de noticias. La aparente variedad informativa ofertada esconde, en realidad, una inmensidad de contenidos redundantes e idénticos.

La otra fuente mayoritaria de información actualmente son los motores de búsqueda de los propios medios de información pero en la mayoría de los casos la tecnología empleada no ofrece la precisión deseada. Aparecen centenares de contenidos sin ningún orden más allá de la fecha y, sobre todo, sin el etiquetado necesario, ni la aplicación de herramientas semánticas que facilitarían esas búsquedas. Este proyecto trabaja en la creación de un sistema de búsqueda semántico que pretende corregir la mayoría de los defectos citados (imprecisión, repetición, falta de rigor, etc.).

El empleo de tecnologías semánticas en la búsqueda de información permitirá una mayor interacción del usuario con el motor de búsqueda y entre el motor de búsqueda y la fuente noticias, de modo que se mejorará la precisión del rastreo de la información deseada.

Etiquetados sobre localizaciones geográficas, intereses, rigor informativo, personajes, etc. garantizarán una mayor eficiencia alejada de la implantación de motores de búsqueda inspirados en fórmulas documentales de la era analógica que hoy dominan el mercado digital de los medios de información.

Freews busca solventar los problemas de la sobrecarga de información mediante las técnicas de búsqueda y recomendación que se explicarán en este documento, e introducir semántica en los contenidos ofertados mediante el uso de datos enlazados.

Capítulo 2

Fijación de objetivos

Para la elaboración de este proyecto fin de máster, se fijarán dos tipos de objetivos, objetivos de investigación y requisitos funcionales.

2.1. Objetivos de investigación

Para la explotación de las líneas de investigación descritas en la sección 1.1 se fijan los siguientes objetivos:

1. Recomendación de noticias:
   a) Investigación de las técnicas de recomendación habituales y de frameworks o librerías que faciliten el trabajo.
   b) Investigación acerca de los sistemas de recomendación actuales, tanto de noticias como otros sistemas más genéricos.
   c) Propuesta y desarrollo de un sistema de recomendación de noticias adaptado a las necesidades de periodistas amateurs o profesionales, este sistema deberá ser configurable automáticamente.
   d) Evaluación con un equipo de profesionales del sistema de recomendación propuesto.
   e) Publicación de los resultados de esta investigación en una revista de impacto.
2. Búsqueda:
   a) Investigación de las técnicas de búsqueda habituales y de frameworks o librerías que faciliten el trabajo.
   b) Desarrollo de un sistema de búsqueda que implemente las técnicas anteriormente descritas.
3. Semántica:
   a) Investigación acerca de las técnicas empleadas para añadir semántica a los sitios web actualmente.
   b) Adición de semántica en los vídeos de Freews.
4. Recomendación de etiquetas
   a) Investigación de las técnicas de recomendación de etiquetas habituales y de frameworks o librerías que faciliten el trabajo.
   b) Implementación de un sistema de recomendación de etiquetas para Freews.

2.2. Requisitos funcionales

Será necesario elaborar un prototipo de Freews que incluya los resultados de la investigación propuesta anteriormente. En este prototipo primará esta inclusión de resultados antes que otros aspectos como la seguridad, la apariencia estética o la eficiencia.

El prototipo elaborado será utilizado para buscar financiación con la que elaborar un proyecto totalmente funcional y listo para poner en producción, no obstante, este prototipo deberá de cumplir los siguientes objetivos funcionales para poder hacer demostraciones a los posibles inversores.

1. Subida de vídeos en crudo por parte de los usuarios. Estos vídeos solo los podrán ver los administradores para hacer clips a partir de ellos.

2. Subida de clips por parte de los administradores que estarán disponibles para todos los usuarios.

3. Subida de infografías (cabeceras, ráfagas y créditos) por parte de los administradores.

4. Elaboración de noticiarios en los que se mezclen clips e infografías. Los noticiarios tendrán la siguiente estructura: *Cabecera* $(Clip\ Ráfaga)^{+}$ *Clip Créditos*

2.3. Ámbitos de aplicación

El ámbito más directo de los objetivos de investigación descritos en la sección 2.1 es el propio sitio web de Freews.

El recomendador de noticias propuesto en Freews, no obstante puede ser exportado a otros sitios web con funcionalidades de agregadores de noticias como Digg (`http://digg.com/`, Reddit (`http://www.reddit.com/`), o Menéame (`http://www.meneame.net/`). También puede aplicarse a lectores RSS como Feedly (`http://feedly.com/`), o GoodNoows (`http://goodnoows.com/`). También podría aplicarse a periódicos convencionales, aunque en este ámbito el parámetro de la credibilidad de la fuente de información perdería sentido, ya que la fuente es la misma, excepto en el caso de noticias distribuidas por agencias.

Los sistemas y técnicas de búsqueda descritos en este documento se aplican en importantes sitios web como el de la Casa Blanca (`http://whitehouse.gov/`), Netflix (`http://netflix.com/`), o la web del Smithsoninan (`http://siris-collections.si.edu/search/`)

En cuanto a las técnicas de adición de semántica a sitios web, sitios como eBay (`www.ebay.com`), IMDb (`www.imdb.com`) o Ticketmaster (`http://www.ticketmaster.es/`) lo utilizan para facilitar la labor a los buscadores.

Por último, la recomendación de etiquetas es aplicable a cualquier sitio web en el que se publiquen noticias, ya sean periódicos, blogs, redes sociales, etc... Además, también existen sistemas de etiquetado automático de imágenes, no solo de texto, en redes sociales como Google+ (`https://plus.google.com/`) o aplicaciones como Picasa (`http://picasa.google.com/`).

Capítulo 3

Estado Actual de los Conocimientos Científico-Técnicos

3.1. Recomendación de contenido

Un sistema de recomendación se define como un sistema que, dado un conjunto de objetos, selecciona un subconjunto del mismo de acuerdo a las preferencias del usuario. Debido a la forma en la que se realiza esta selección no puede ser expresada como una consulta común a una base de datos [71]. Durante la última década, se han desarrollado varios sistemas de recomendación utilizados en diferentes áreas de negocio [2], y han probado ser efectivos en cuanto al filtrado de información de forma personalizada [50].

Los sistemas de recomendación han sido clasificados por Balabanovic [4], en:

- Recomendadores basados en contenido (*Content-Based recommenders*): Las recomendaciones se basan en un conjunto de propiedades comunes a los objetos que el usuario prefirió en el pasado.
- Recomendadores de filtrado colaborativo (*Collaborative-filtering recommenders*): La recomendación de contenido a un usuario X se basa en los objetos preferidos por los usuarios con gustos similares a los de X.
- Recomendadores híbridos (*Hybrid recommenders*): Las recomendaciones se basan en un combinación de los dos anteriores.

La recomendación basada en contenido no es usada comúnmente por sí sola, pero sí se emplea con frecuencia como parte de un sistema de recomendación híbrido. Estos sistemas híbridos han sido desarrollados para una amplia gama de productos como programas de televisión [6], películas [42], música [73], tiendas de centros comerciales [23], viajes [62], etc.

Los recomendadores puramente basados en algoritmos de filtrado colaborativo tienen su raíz en métodos estadísticos que se describirán en la sección 4.2. Algunos ejemplos de esos recomendadores son los basados en técnicas de factorización de matrices (como SVD) [75], o [31], basados en la similitud entre ítems o usuarios [7], o basados en técnicas de *Slope-One* [43], los cuales son sencillos de implementar y muy eficientes, pero no tan precisos como otros sistemas más complejos.

En la propuesta expuesta en este trabajo, se utilizarán conceptos para recomendar noticias que ya han sido utilizados en otros contextos, por ejemplo, la recomendación en base a temas es similar a la recomendación en base a géneros de películas descrita en [13]. El recomendador expuesto en [14], también utiliza la temática para producir recomendaciones de una plataforma de venta electrónica. Por otro lado, la recomendación basada en proximidad también se ha utilizado en el recomendador mostrado en [39].

Para cumplir el objetivo de configuración automática se pueden utilizar técnicas relacionadas con algoritmos genéticos para aprender como en [61], sin embargo, este trabajo

no permite ser ampliado indefinidamente ya que es una ampliación de [60], que no fue diseñado inicialmente con tal fin.

3.1.1. Recomendación de noticias

La recomendación basada en contenido de noticias tipicamente se basaba en recoger las preferencias del usuario categorizando temas como en [30], [40] y [11] o utilizando modelos de espacios vectoriales como [35] para extraer un conjunto de palabras de su historial de lectura y buscar esas mismas palabras en contenidos publicados más recientemente. Algunos ejemplos de esos sistemas son News Dude [10] y YourNews [3]. Una herramienta algo más sofisticada es Newsjunkie [26], que también permite filtrar los contenidos más recientes.

También se han desarrollado recomendadores de noticias híbridos. Algunos ejemplos representativos son SCENE [45] y PENETRATE [74]. El primero presenta un sistema de recomendación de noticias escalable y de dos etapas distribuidas en dos niveles jerárquicos. En la primera etapa se realiza un filtrado preliminar en base a un breve resumen de las categorías que el usuario prefiera con más alta probabilidad, mientras que en el segundo nivel se lleva a cabo la recomendación de noticias específicas. Por otra parte, PENETRATE utiliza *clústers* de usuarios, que construye a partir de su historial de lectura. En este sistema, cada usuario puede pertenecer a más de un *clúster*. Después, la recomendación se realiza de la forma tradicional cuando se agrupan *clústers*, una misma recomendación para todos los usuarios.

Otro recomendador de noticias es MONERS [41], el cual estima las preferencias del usuario en base a la agregación de la importancia de cada artículo y la proximidad temporal del mismo. MONERS está orientado a la recomendación de noticias en dispositivos móviles.

En cualquier caso, las propuestas mencionadas anteriormente hacen referencia a recomendadores para usuarios, mientras que el recomendador aquí propuesto trata de generar recomendaciones orientadas profesionales del mundo de la información, en un escenario que representa las necesidades laborales diarias de un periodista.

3.1.2. Recomendación de etiquetas

En la actualidad, la recomendación de etiquetas se lleva a cabo mediante el uso de *folksonomías*. En primer lugar, conviene explicar por tanto lo que es una *folksonomía*. En cualquier sistema en el que un usuario pueda subir contenido y etiquetarlo, al conjunto de asignaciones de etiquetas que un usuario asigna a su contenido se denomina *personomía*. Al conjunto de las *personomías* de todos los usuarios se le denomina *folksonomía* [33]. El uso de folksonomías, por tanto, está muy relacionado con la recomendación basada en algoritmos de filtrado colaborativo expuesta anteriormente, ya que se comprueba el tipo de etiquetados que hacen los usuarios ante diversos documentos para recomendarles etiquetas.

Otra forma de recomendar etiquetas cuando el contenido a etiquetar es textual, es basándose en la extracción de palabras clave. Esto está relacionado con las técnicas de Procesamiento del Lenguaje Natural que se verán más adelante.

El trabajo relacionado a la recomendación de etiquetas está orientado a temáticas diversas como blogs [46], marcadores [8] o imágenes [63]. Para el dominio de Freews, sistemas como Zemanta [66] u Open Calais [68] son igualmente válidos.

3.2. Sistemas de búsqueda

En cuanto a la búsqueda de texto, una de las técnicas empleadas más comúnmente es el Procesamiento del Lenguaje Natural .

El Procesamiento del Lenguaje Natural es una técnica que trata de la aplicación de modelos computacionales a texto o discurso hablado. Las áreas de aplicación del Procesa-

miento del Lenguaje Natural incluyen la traducción automática entre lenguajes, sistemas de diálogo que permiten a un humano interactuar con una máquina utilizando lenguaje natural, y la extracción de información, donde el objetivo es transformar texto no estructurado en representaciones estructuradas sobre las que buscar y navegar de forma más flexible. Esta última área es la que nos ocupa en este proyecto.

Para la búsqueda de contenido en Freews se aplicarán técnicas de Procesamiento del Lenguaje Natural utilizando Apache Solr[1]. Apache Solr es un servidor que permite indexar contenido aplicándole técnicas de Procesamiento de Lenguaje natural para posteriormente ejecutar búsquedas textuales sobre él. A estas búsquedas también se le aplicarán esas mismas técnicas. Apache Solr usa Apache Lucene[2] y lo extiende, proporcionando, entre muchas otras, características de búsqueda por geoposicionamiento. Las técnicas de Procesamiento del Lenguaje Natural aplicadas por Solr se recogen en la sección 3.2.1.

Junto con el Procesamiento del Lenguaje Natural, Solr incorpora la posibilidad de añadir sinónimos, de modo que permite enriquecer aún más las búsquedas, aplicando expansión de consulta. Esto es similar al algoritmo descrito en WESONet [28]. Para la construcción de este diccionario de sinónimos se pueden utilizar diccionarios como WordNet [24] o BabelNet [48].

3.2.1. Técnicas de Procesamiento del Lenguaje Natural

Freews incorporará las siguientes técnicas de Procesamiento del Lenguaje Natural para realizar búsquedas.

Eliminación de palabras vacías

Las palabras vacías, o *stopwords*, son palabras que carecen de significado. Dentro de esta categoría se pueden incluir artículos, conjunciones, pronombres, etc... La eliminación de estas palabras, por lo general mejora la relevancia de los resultados obtenidos, ya que al no tenerlas en cuenta, no se encuentran falsas coincidencias, además esta eliminación de *stopwords* incrementa el rendimiento al aligerar las consultas y eliminar datos irrelevantes del índice, ahorrando a su vez espacio de disco.

Extracción de n-Gramas

Un n-grama es una subsecuencia de caracteres de longitud n que forman parte de una palabra o frase. Por ejemplo, para la secuencia de caracteres "*Lorem ipsum*" se tienen los siguientes n-gramas de longitud 4 (o 4-gramas): {"Lore", "orem", "rem ", "em i", "m ip", " ips", "ipsu", "psum"}. Analizando estos n-gramas, se podrían encontrar exitosamente coincidencias que contuviesen erratas o faltas de ortografía. Solr incorpora esto parcialmente. Su forma de trabajo es partir del primer o último n-grama de la secuencia e ir añadiendo la siguiente o la anterior letra respectivamente hasta que el n-grama alcance la longitud deseada. Por ejemplo, en la secuencia de caracteres anterior, para Solr los 4-gramas hasta una longitud máxima de ocho elementos serían: {"Lore", "Lorem", "Lorem ", "Lorem i", "Lorem ip"}.

Stemming

El *stemming* es una técnica para reducir una palabra a su raíz, de modo que el género o tiempo verbal utilizado para indexar un texto no afecte a la búsqueda. Uno de los algoritmos más conocidos de *stemming* es el algoritmo de Porter [54] para el inglés, sin embargo, uno de los requisitos para Freews es que soporte varios idiomas, por lo que es más adecuado el uso de Snowball [55], un lenguaje de programación para crear algoritmos de *stemming*. Solr contiene algoritmos de *stemming* implementados con Snowball para

[1] `http://lucene.apache.org/solr/`
[2] `http://lucene.apache.org/core/`

varios lenguajes como son el castellano, el portugués y el inglés (para el inglés implementa el algoritmo de Porter).

Otras técnicas

Solr incorpora además otras técnicas más triviales para mejorar la búsqueda y el indexado, como puede ser la eliminación de tildes, de mayúsculas, guiones entre palabras, espacios en blanco, descomposición de palabras compuestas, etc...

3.2.2. Técnicas de expansión de consulta

Las técnicas de expansión de consulta, o *Query Expansion* consiste en reformular una consulta, expandiéndola, de modo que puedan encontrarse más coincidencias [69]. El proceso para ello está descrito en el artículo *"WESONet: Applying semantic web technologies and collaborative tagging to multimedia web information systems"* [28] y consiste en lo siguiente:

1. Sea α la consulta inicial formada por un conjunto de palabra $\{w_1, ..., w_n\}$. El primer paso es la obtención de un conjunto de términos normalizados $\tau = \{t_1, ..., t_n\}$ aplicando los procesos de tratamiento del lenguaje natural descritos en la sección 3.2.1.

2. El siguiente paso es obtener un conjunto de conceptos Ω que coincidan con los términos en τ.

3. Tras ello, se aplica *spreading activation*[16] a Ω, de modo que se consigue un nuevo conjunto Ω' de conceptos relacionados con los iniciales. Cada concepto c_i va acompañado de un peso w_i que indica cómo de relacionado está con los términos de Ω.

4. Por ultimo, se crea una nueva búsqueda β formada por los conceptos de Ω' junto con sus pesos. También se incluyen los términos t_j para los que no se han encontrado conceptos relacionados tras aplicar *spreading activation*. La consulta β será la que se ejecute.

Solr soporta *query expansion* de manera parcial, ya que permite agregar un diccionario de sinónimos para cada lenguaje que soporta, sin embargo, no se pueden añadir pesos.

3.3. Inserción de semántica en HTML

La web tradicional cumple sus funciones de manera correcta cuando el consumidor de datos es una persona física, sin embargo, cuando otras máquinas quieren consumir esos datos, aparecen los problemas, ya que la web fue diseñada para ser consumida por personas. Para solventar esos problemas existen dos soluciones, la recuperación de información y la Web Semántica.

La recuperación de información (o *Information Retrieval*) consiste en utilizar los recursos de la web tradicional tal y como están para intentar extraer información de ellos. No implica cambios en la web, pero hace muy costoso el desarrollo de aplicaciones que quieran consumir esos datos.

La Web Semántica [9] propone un enfoque radicalmente opuesto. Lo que la Web Semántica propone es una visión de la web en la que los datos se publican de manera estructurada y se enlazan con otros datos, además, cada dato está identificado con una URI. De este modo, existe una parte de la web que consumirán las personas, y otra parte que consumirán las máquinas. Tim Berners Lee, acuñó el término de Web Semántica e ilustró la importancia de los datos estructurados con el ejemplo de la bolsa de patatas de la Figura 3.1, en la que por un lado de la bolsa aparece la imagen corporativa de la empresa,

como forma de representación para las personas, pero por la parte trasera, aparecen los datos nutricionales de la bolsa de una forma estandarizada y por tanto, más fácilmente entendible por las máquinas. Además, la parte trasera incluye un código de barras que identifica ese producto globalmente.

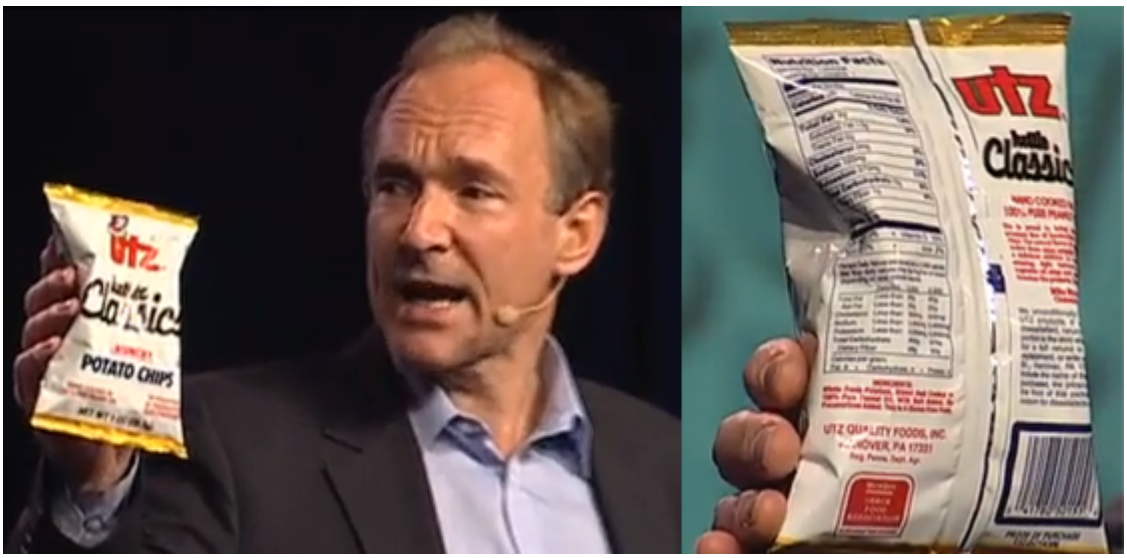

Figura 3.1: Dos formas de representar el mismo objeto

Siendo RDF[3] el formato estándar para el intercambio de datos en la web [70], uno de los problemas a los que ha tenido que enfrentarse han sido cómo insertar RDF en las webs actuales de un modo que se permita la validación del HTML. A continuación se citan alguna de las soluciones propuestas [27] que no han triunfado.

Una solución propuesta ha sido incluir el RDF como comentarios (Listing 1), sin embargo esto no es para nada ortodoxo, ya que los comentarios no han sido ideados para incluir nada que deba ser comprobado por una máquina, sino que están pensados para ser consumidos por las personas.

```
<html>
  <head>
    <title>Spain</title>
  </head>
  <body>
<!-- <rdf:RDF
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:geo="http://www.w3.org/2003/01/geo/wgs84_pos#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:wi-onto="http://data.webfoundation.org/webindex/ontology/">
  <wi-onto:Country rdf:about="http://data.webfoundation.org/ESP">
    <rdfs:label>Spain</rdfs:label>
    <geo:long>-3.74922</geo:long>
    <geo:lat>40.46366700000001</geo:lat>
  </wi-onto:Country>
</rdf:RDF> -->
    <h1>Spain</h1>
    <p>Spain is a country...</p>
  </body>
</html>
```

Listing 1: RDF como comentario

Otra posible solución sería incluir el RDF en el HTML sin más (Listing 2), el problema

[3]Resource Description Framework

es que esto produciría un código HTML no válido, para lo cual sería necesario ampliar la DTD[4] de XHTML, lo cual sería demasiado costoso.

```
<html>
  <head>
    <title>Spain</title>
    <rdf:RDF xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
      xmlns:geo="http://www.w3.org/2003/01/geo/wgs84_pos#"
      xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
      xmlns:wi.onto="http://data.webfoundation.org/webindex/ontology/">
      <wi-onto:Country rdf:about="http://data.webfoundation.org/ESP">
        <rdfs:label>Spain</rdfs:label>
        <geo:long>-3.74922</geo:long>
        <geo:lat>40.46366700000001</geo:lat>
      </wi-onto:Country>
    </rdf:RDF>
  </head>
  <body>
    <h1>Spain</h1>
    <p>Spain is a country...</p>
  </body>
</html>
```

Listing 2: RDF en HTML

Por ultimo, también se propuso utilizar <link> para enlazar desde HTML a otro fichero (Listing 3). Esta solución no es para nada invasiva con el código HTML actual, sin embargo esto significaría el mantenimiento de dos ficheros independientes que representan la misma información.

```
<html>
  <head>
    <title>Spain</title>
    <link rel="meta" type="application/rdf+xml" href="./ESP.rdf" />
  </head>
  <body>
    <h1>Spain</h1>
    <p>Spain is a country...</p>
  </body>
</html>
```

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:geo="http://www.w3.org/2003/01/geo/wgs84_pos#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:wi-onto="http://data.webfoundation.org/webindex/ontology/">
  <wi-onto:Country rdf:about="http://data.webfoundation.org/ESP">
    <rdfs:label>Spain</rdfs:label>
    <geo:long>-3.74922</geo:long>
    <geo:lat>40.46366700000001</geo:lat>
  </wi-onto:Country>
</rdf:RDF>
```

Listing 3: RDF enlazado con link

A continuación se citarán algunas soluciones que han tenido más éxito que las anteriormente mencionadas.

[4]Document Type Definition

3.3.1. Microdatos

Los microdatos (Listing 4) son una propuesta para HTML5 cuyo objetivo es crear un esquema de datos estructurados que sea soportado por los motores de búsqueda más importantes [22]. Los microdatos estructuran la información del siguiente modo:

- Se identifican nuevas entidades mediante `itemscope`.
- Las propiedades se identifican con `itemprop`.
- Los tipos se especifican con `itemtype`.

```
<html>
  <head>
    <title>Spain</title>
  </head>
  <body>
    <h1>Spain</h1>
    <p itemscope itemtype="http://schema.org/Place">
      <span itemprop="name">Spain</span> is a country located at
      <span intemprop="geo" itemscope itemtype="http://schema.org/GeoCoordinates">
        40.463667, -3.74922
        <meta itemprop="latitude" content="40.463667" />
        <meta itemprop="longitude" content="-3.74922" />
      </span>
    </p>
  </body>
</html>
```

Listing 4: Microdatos

Los microdatos se encuentran un punto intermedio entre complejidad y expresividad, ya que tiene un vocabulario extenso y en proceso de ampliación y no añade demasiada verbosidad al HTML.

3.3.2. Microformatos

Los microformatos (Listing 5) son un conjunto de formatos simples de datos abiertos no estándar construidos sobre estándares. Han sido diseñados anteponiendo la legibilidad por parte de los humanos antes que las máquinas [12]. Alguno de los estándares en los que se basa son vCard[5] o iCalendar[6].

```
<html>
  <head>
    <title>Spain</title>
  </head>
  <body>
    <h1>Spain</h1>
    <p>Spain is a country located at <span class="geo">
      <span class="latitude">40.463667</span>, <span class="longitude">-3.74922</span>
    </span></p>
  </body>
</html>
```

Listing 5: Microformatos

[5] http://www.ietf.org/rfc/rfc2426.txt
[6] http://www.ietf.org/rfc/rfc2445.txt

Uno de los problemas que presentan los microformatos es el uso y abuso del atributo `class`. Este atributo fue pensado para usar como selector en las hojas de estilo, además, frameworks como Struts 2 lo utilizan para sus propios usos, por lo que se sobrecargaría este atributo en exceso.

3.3.3. RDFa

RDFa (Listing 6) está inspirado en los microformatos y fue recomendado en 2008 por el W3C. La versión actual, RDFa 1.1 fue desarrollada en 2011 y recomendada en 2012 por el W3C [1]. La propuesta de RDFa consiste en codificar tripeltas RDF dentro de HTML de este modo:

- El sujeto se especifica con el atributo `about`.
- El predicado se especifica mediante `property`, `rel`, o `rev`.
- Las URIs se especifican mediante `href`, `resource`, o `src`.
- Los literales se especifican mediante `content` y opcionalmente se añade el atributo `datatype` para especificar el tipo de dato.

```
<html>
  <head>
    <title>Spain</title>
  </head>
  <body>
    <h1>Spain</h1>
    <p xmlns:geo="http://www.w3.org/2003/01/geo/wgs84_pos#"
     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
     about="http://data.webfoundation.org/ESP">
      <span property="rdfs:label">Spain</span> is a country located at
      <span property="geo:lat" content="40.463667" datatype="xsd:decimal">
        40.463667
      </span>, <span property="geo:long" content="-3.74922" datatype="xsd:decimal">
        -3.74922
      </span>
    </p>
  </body>
</html>
```

Listing 6: RDFa

RDFa es probablemente el formato más flexible, extensible y que más expresividad proporciona. Sin embargo no ha sido ampliamente adoptado por los desarrolladores debido a su complejidad y verbosidad.

3.3.4. Otros

Es tal la importancia de la adición de datos estructurados al código HTML, que redes sociales ampliamente usadas como Twitter o Facebook han creado sus propios formatos, las Twitter Cards[7] y Open Graph[8] respectivamente. Ambas propuestas son muy similares, ya las dos consisten en introducir etiquetas `meta` dentro del nodo `<head>` y ambas sirven para indicar a Twitter y Facebook como previsualizar una web o una acción ejecutada en una web por un usuario (Figura 3.2).

[7] `https://dev.twitter.com/docs/cards`
[8] `https://developers.facebook.com/docs/opengraph/`

Figura 3.2: Visualización con Twitter Cards y Open Graph

Capítulo 4

Descripción del Sistema

Se ha diseñado un recomendador de noticias híbrido basado en el contexto del usuario y en algoritmos de filtrado colaborativo con el objetivo de hacer frente a los problemas existentes en los actuales sistemas de recomendación.

La elección de un sistema híbrido frente a un sistema basado en contenido o en filtrado colaborativo únicamente es que, pese a que los sistemas de filtrado colaborativo producen recomendaciones más precisas [15], también presentan los problemas conocidos como arranque en frío (o *cold start*) y el problema del nuevo elemento (o *new item problem*).

El arranque en frío consiste en que cuando el sistema de recomendación no ha recopilado datos suficientes, no podrá producir ninguna recomendación [59], mientras que el problema del nuevo elemento hace referencia a que si un elemento no ha sido puntuado nunca, este no será recomendado [2]. Este último problema es especialmente dañino cuando se trata de recomendar noticias ya que las últimas noticias son especialmente relevantes, con lo cual el componente de recomendación basada en contenido se antoja como una pieza básica en un sistema de recomendación de noticias.

Para paliar los efectos nocivos de ambos tipos de recomendadores, se ha optado por el diseño de un sistema de recomendador híbrido con dos componentes, uno que produzca recomendaciones basadas en el contenido y el otro utilizando algoritmos de filtrado colaborativo. Cada componente evalúa las noticias que el usuario no ha visto o puntuado y produce recomendaciones de forma independiente, de modo que para cada noticia, cada componente producirá un número entre cero y uno. Estos números se llamarán CB (para el componente basado en contenido) y CF para el de filtrado colaborativo. Cuanto más próximos a uno estén CB y CF, más posibilidades habrá de que al usuario le interese la noticia asociada a esa recomendación. Una vez calculados CB y CF, estos números se utilizarán como parámetros de la fórmula (4.1) para producir una única recomendación.

$$r = w \cdot CB + (1 - w) \cdot CF \tag{4.1}$$

Donde w es la ponderación del componente basado en contenido, siendo esta un número comprendido en el intervalo $(0 - 1)$. Las próximas secciones explicarán cómo se calcula el valor de CB y CF para cada noticia.

Como resumen de los requisitos que ha de satisfacer el recomendador para cumplir con las expectativas de esta investigación, se presenta la tabla 4

4.1. Componente basado en contenido

Existe una regla no escrita, pero muy conocida, en el mundo del periodismo llamada las cinco W's (*The Five W's*)[65] que pone de manifiesto la importancia que tiene para una noticia contestar a las preguntas ¿quien?, ¿cómo?, ¿por qué?, ¿cuando? y ¿dónde? (el nombre de la regla viene de la traducción al inglés de estas preguntas: *who?*, *what?*, *why?*, *when?* y *where?*). Tomando en cuenta las preguntas de la regla de las cinco W's en

Cuadro 4.1: Requisitos del recomendador

Requisito	**Descripción**
Proximidad	Las noticias que ocurren cerca del usuario tienen más posibilidad de interesarle, por lo tanto, esta variable ha de ser considerada.
Tiempo	Es más probable que las noticias más recientes levanten más interés en el usuario, por lo que esto debe ser tenido en cuenta.
Historial de usuario	Los últimos temas que el usuario ha comprobado, son los que han captado su atención en el momento de la recomendación, por lo que esto es un factor importante.
Veracidad	La veracidad de una fuente de información es un factor importante que debe ser considerado para producir recomendaciones.
Aprendizaje off-line	El recomendador ha de evaluar los resultados producidos por un conjunto de algoritmos de filtrado colaborativo siendo ejecutados en el dataset actual. Esto ha de hacerse de forma off-line mientras el sistema funciona online
Cambio en caliente	El recomendador debe ser capaz de cambiar el algoritmo de filtrado colaborativo mientras está produciendo recomendaciones. Debe ser posible también añadir nuevos algoritmos en tiempo de ejecución.

el campo del periodismo, los algoritmos de Wesomender están basados en las siguientes hipótesis:

1. Una noticia es más interesante cuando acaba de ocurrir que un tiempo después.

2. Una noticia que tiene lugar cerca del usuario es más relevante para él.

3. Si un usuario está leyendo noticias que impliquen ciertas entidades o conceptos específicos, o cubran un tema en concreto, entonces es que está mostrando interés en esas entidades, conceptos o temas, incluso aunque las noticias que está leyendo sean viejas, el usuario puede estar buscando referencias sobre un tema en concreto.

Además, un nuevo parámetro llamado credibilidad es añadido para dar valor a la reputación de cada fuente de información. Por ejemplo, una noticia publicada por un medio como la BBC siempre será más creíble que algo que un desconocido escribe en un blog.

Por ello, una fórmula considerando el tiempo, la localización, el contenido y la credibilidad (4.2) ha sido diseñada y su salida será el valor que CB tomará en la fórmula (4.1).

$$CB = \alpha \cdot ER + \beta \cdot T + \gamma \cdot P + \delta \cdot R \tag{4.2}$$

Donde todos los operandos son números entre 1 y 0 y:

- ER se refiere a las etiquetas relacionadas a una noticia. Para obtener este valor se comprobarán las últimas noticias que el usuario ha consultado. Tras ello, se calculará el porcentaje de ocurrencias de esas etiquetas en cada noticia. Por último, para cada etiqueta en la noticia que se está recomendado, se añadirá el porcentaje que le corresponda a la variable ER. Por ejemplo, si la etiqueta "Linked Data" aparece en la noticia que se está recomendando y representa un 10 % de las etiquetas de las noticias consultadas recientemente, se añadirá un 10 % al valor de ER y se procederá de la misma forma para el resto de etiquetas. La forma de extracción de etiquetas se explica en la sección 7.4.2.

- R indica la proximidad temporal de una noticia y será calculada como se muestra en la formula (4.3). Siempre producirá un número entre 1 y 0 ya que siempre se cumplirá que $tiempoactual \geq tiempodepublicacion$.

- P hace referencia a la proximidad y será calculada como se muestra en la fórmula (4.5). Al igual que la fórmula (4.3), esta siempre producirá un número entre 1 y 0, ya que $maxdistancia \geq distancia$, siendo $maxdistancia$ la mayor distancia entre dos puntos de la Tierra, dada por la mitad de la circunferencia de la tierra en el ecuador, y $distancia$ la distancia entre el usuario y el lugar en el que ocurrió la noticia. Para calcular la $distancia$ se utilizará la fórmula de Haversine (4.4), que utiliza la latitud (ϕ_1 y ϕ_2) y la longitud (λ_1 y λ_2) de ambos puntos geográficos.

- T será un porcentaje indicado por un equipo profesional de periodistas y que hará referencia a la credibilidad de cada fuente de información.

La relevancia de cada variable es un porcentaje, calculado en base a una encuesta con un equipo profesional de periodistas en el que determinaban la importancia de cada factor. Tras dicha encuesta se obtuvieron las siguientes conclusiones.

- La proximidad temporal es el factor más importante, de modo que $\delta = 0{,}5$.

- La proximidad espacial sigue a la proximidad temporal en cuanto a relevancia, siendo $\gamma = 0{,}2$.

- La credibilidad y las etiquetas son igualmente importantes, por lo que $\alpha = \beta = 0{,}15$.

$$R = 1 - \frac{tiempoactual - tiempodepublicacion}{tiempoactual} \tag{4.3}$$

$$distancia = 2 \cdot radioterrestre \cdot \arcsin(\sqrt{\sin^2(\frac{\Delta\phi}{2}) + \cos(\phi_1) \cdot \cos(\phi_2) \cdot \sin^2(\frac{\Delta\lambda}{2})}) \tag{4.4}$$

$$P = \frac{maxdistancia - distancia}{maxdistancia} \tag{4.5}$$

4.2. Componente de filtrado colaborativo

Con el objetivo de diseñar un sistema de recomendación adaptable e híbrido, se ha implementado un componente que evalúa el conjunto de datos existente y elige el algoritmo de filtrado colaborativo que mejor se adapta a dicho dataset. Este componente puede ser usado solo o reutilizado en otros sistemas de recomendación. Además, debe cumplir los siguientes requisitos iniciales:

1. Dado que se espera que la evaluación de distintos algoritmos sea lenta, esta debe llevarse a cabo offline, al mismo tiempo que otro sistema de recomendación está trabajando online. En cuanto esta evaluación termine, el nuevo algoritmo de filtrado colaborativo puede ser seleccionado en tiempo de ejecución.

2. A pesar de que los algoritmos a ser evaluados en un origen puede limitarse a un conjunto pequeño de ellos, se debería de poder añadir nuevos algoritmos a esta evaluación de forma sencilla, lo que permitirá a los desarrolladores y a los investigadores añadir nuevos algoritmos a la evaluación, potenciando así el sistema.

3. Para hacer uso de un conjunto inicial de algoritmos sin necesidad de reimplementarlos, se puede utilizar una librería que ya implementa dichos algoritmos. En este caso, el sistema deberá ser implementado como una capa sobre dicha librería.

Tan pronto como se seleccione el mejor algoritmo, el sistema será caapaz de calcular el valor de CF en la fórmula (4.1).

En Wesomender, este componente ha sido implementado sobre Apache Mahout[1]. Se escogió esta librería porque ya implementa un número considerable de algoritmos que pueden ser evaluados, además de un *framework* de evaluación para medir la precisión de esos algoritmos. La alternativa más clara a Apache Mahout es Weka[2], que también incorpora algoritmos de Machine Learning.

Los evaluadores de Mahout toman un porcentaje del dataset a evaluar, a este porcentaje se le llama *training data*. Mahout trata de aprender del *training data* y predecir cómo será la otra parte del conjunto de datos. Uno de los evaluadores implementados por Mahout es la media cuadrática. Se ha escogido este evaluador por ser ampliamente aplicable y siempre definido y finito, excepto en el caso en el que el dataset predicho y el real sean iguales [32]. El resultado de esta evaluación puede ser serializado junto con el recomendador que produjo el resultado, de modo que si se apaga la máquina no se tendrá que repetir la evaluación.

Los siguientes recomendadores implementados por Mahout fueron evaluados como muestra, pero dichos evaluadores son cargados en tiempo de ejecución usado Spring[3], por lo que nuevos recomendadores y suss evaluadores pueden ser añadidos de forma dinámica sin la necesidad de re compilar el sistema completo.

4.2.1. Recomendador basado en el usuario

Este recomendador produce recomendaciones para un usuario teniendo en cuenta qué noticias son las más valoradas por usuarios con gustos similares. Para ello se ha parametrizado con todas las posibles combinaciones de las interfaces `User Similarity` y `User Neighbourhood`.

Las clase que implementan `User Similarity` definen el concepto de similaridad entre usuarios, más concretamente, entre los gustos de los usuarios. Existen varias implementaciones en Mahout para este concepto. El coeficiente de correlación de Pearson [53], la Distancia Euclídea [18], el coseno [67], la Geometría Taxicab [38], el logaritmo de la función de verosimilitud [20], el coeficiente de correlación de Spearman [64] y el índice de Jaccard [57].

Por otra parte, el concepto de *neighbourhood* o vecindario determina cuantos usuarios han de ser tomados en consideración para producir recomendaciones. Una vez que la distancia entre usuarios está calculada, las recomendaciones pueden producirse tomando en cuenta los N usuarios más cercanos o los usuarios dentro de un umbral de similitud. Ambos tipos de vecindario han sido parametrizados y combinados con las formas de calcular la similaridad descritas anteriormente para buscar la mejor dupla.

4.2.2. Recomendador basado en el item

Los recomendadores basados en el item son similares a los basados en el usuario descritos en la sección anterior. Al igual que ellos, utilizan el concepto de vecindario y similitud, pero en lugar de aplicarlo a usuarios, lo aplica a items, en este caso, noticias. Por ello, este algoritmo no busca usuarios similares a un usuario dado, en lugar de ello, busca noticias similares a aquellas que le interesan al usuario. La implementación de este recomendador en Apache Mahout está basada en el sistema descrito en [7].

4.2.3. Recomendador basado en clústers

Este recomendador forma un número específico de clústers o agrupaciones de usuarios, donde cada clúster está compuesto por los usuarios más similares entre ellos. Una vez

[1] http://mahout.apache.org/
[2] http://www.cs.waikato.ac.nz/ml/weka/index.html
[3] http://www.springsource.org/

formadas estas agrupaciones, el comportamiento de este recomendador es el mismo que el recomendador basado en el usuario, tratando a cada clúster como un único usuario. Por ello, el sistema producirá las mismas recomendaciones a todos los usuarios que forman parte de un mismo clúster.

Para medir la distancia entre clústers, tradicionalmente existen dos posibilidades. En primer lugar, dos clústers pueden ser considerados similares si **alguna** pareja de miembros tiene un alto grado de similitud. Por otro lado, se pueden endurecer las condiciones de similitud entre clústers y decir que dos clústers son similares si y solo si **todas** las parejas de miembros tienen una similitud significativa. Para Wesomender se ha hecho uso de ambas combinaciones.

4.2.4. Recomendadores basados en técnicas de factorización de matrices

SVD (del inglés *Single Value Descomposition*) es una técnica de factorización de matrices que puede ser utilizada para producir recomendaciones [37]. Mahout allows using *factorizers* that define how to factorize the rating matrix. Three of this factorizers were used in this system. The first one, uses the Alternating-Least-Squares with Weighted-λ-Regularization and is parametrised as described in [75], the second one uses the Expectation-Maximization algorithm [17] and the last one uses the model described in [31].

4.2.5. Recomendadores Slope One

Los recomendadores *Slope One* [43] tratan de predecir la puntuación que un usuario dará a otro ítem y produce recomendaciones en base a ello. Para predecir las puntuaciones, este sistema de recomendación calcula la diferencia media entre las puntuaciones otorgadas por dos usuarios a los ítems que han votado ambos, tras ello trata de predecir las votaciones que uno de esos dos usuarios dará a un elemento que solo ha votado el segundo usuario. Para clarificar esto se propone el siguiente ejemplo simplificado. Los usuarios A y B han votado los elementos i, j y k. De media el usuario A les ha dado un 7 y el usuario B un 7.5. El algoritmo detecta que B está dando medio punto más de media que A, por lo que si el usuario A ha votado también el elemento l con un 6, el algoritmo predice que B le dará un 6.5.

Capítulo 5

Evaluación

5.1. Metodología

Dado que este sistema de recomendación ha sido diseñado para ser utilizado por profesionales en el dominio periodístico más que para usuarios normales, el objetivo de este estudio es comparar un conjunto de noticias sugeridas por el recomendador con los resultados esperados por un equipo de expertos. Con lo cual, el objetivo es comprobar si las recomendaciones generadas por Wesomender pueden cumplir con las expectaciones y requisitos de los profesionales del sector.

Para llevar a cabo este experimento, un equipo de periodistas ha seleccionado de forma manual un dataset que contiene 30 noticias[1]. Dichas noticias pertenecen a distintos medios de comunicación, lugares y periodos de tiempo.

Para cada noticia se ha extraído el titular y el cuerpo de la misma, así como las tags utilizando Zemanta, la localización y la marca de tiempo de cada una de ellas. La veracidad de las fuentes de información han sido proveídas por el equipo de profesionales. Todas las noticias proceden de medios que publican en inglés, ya que Zemanta parece funcionar mejor extrayendo etiquetas de textos en inglés, además esto permitirá que el dataset sea más legible y extensible por la comunidad científica. Se ha utilizado Zemanta en el experimento en lugar de tags proporcionadas por los periodistas para tratar de hacer el experimento más objetivo y que las opiniones periodísticas no influyan lo más mínimo en el minado del dataset.

Una vez construido el dataset, el equipo de periodistas profesionales evaluó la relevancia de cada noticia con un porcentaje. Considerando las noticias con una relevancia por debajo del 50 % como irrelevantes, se obtuvo que 20 noticias eran relevante y las otras 10 no lo eran.

En esta evaluación, se han medido tres variables, relevancia, precisión y *recall*. En primer lugar, la relevancia de las noticias devueltas por el recomendador. Esta variable fue proveída por los periodistas antes de empezar la evaluación, pero es desconocida para el Wesomender. La precisión ha sido medida como qué porcentaje de las noticias devueltas por el recomendador son relevantes, mientras que el *recall* indica cuantas noticias relevantes fueron devueltas del total de noticias relevantes del dataset.

En este experimento cuantitativo, el equipo de profesionales ha utilizado el recomendador por medio de Freews (caso de uso explicado en el capítulo 7). Se han evaluado las tres variables mencionadas anteriormente después de que el equipo de periodistas evaluase cero, cinco y diez noticias aleatorias, para el primero, los dos primeros y los cinco primeros elementos devueltos por el recomendador. Una vez que un periodista evaluaba una noticia, esta no se le recomendaba de nuevo.

[1] El dataset puede descargarse en: `http://alejandro-montes.appspot.com/research/wesomender/dataset.zip`

Cuadro 5.1: Resultados de la evaluación

Evaluadas	Noticias	Relevancia	Precisión	Recall
0	1º	95 %	100 %	1/20
	1º y 2º	92 %	100 %	2/20
	1º a 5º	70.6 %	80 %	4/20
5	1º	88 %	100 %	1/16
	1º y 2º	90.3 %	100 %	2/16
	1º a 5º	78.1 %	70 %	2,5/16
10	1º	82 %	100 %	1/13,5
	1º y 2º	84 %	100 %	2/13,5
	1º a 5º	87.1 %	100 %	5/13,5

5.2. Resultados y Discusión

Se muestran los resultados del experimento en la tabla 5.1. La columna *"Evaluadas"* hace referencia a la cantidad de noticias evaluadas por los profesionales antes de la recogida de resultados parciales, mientras que la columna *"Noticias"* se refiere a los elementos considerados para madir relevancia, precisión y *recall*.

Por ejemplo, en una fila en la que la columna *"evaluadas"* toma el valor 5 y la columna *"noticias"* es "*1º y 2º*", se está midiendo relevancia, precisión y *recall* para las dos primeras noticias devueltas por el recomendador después de que los periodistas hubiesen votado cinco noticias aleatorias. Una forma más gráfica de observar la tabla es la Figura 5.1.

Los resultados muestran un excelente rendimiento para un usuario nuevo en los dos primeros elementos, peor es significativamente peor para las cinco primeras noticias. En cualquier caso, después de un poco de interacción entre los usuarios y el sistema, hay una mejoría significativa en los resultados para cinco recomendaciones, sin haber ninguna ganancia significativa para los resultados de una y dos recomendaciones, de hecho, estos resultados son ligeramente peores que antes de votar ninguna noticia, probablemente debido al hecho de que después de puntuar una noticia, esta no será recomendada de nuevo. Por lo tanto, en el dataset existente hay una alta probabilidad de que se eliminen de la recomendación noticias relevantes, ya que 2/3 de noticias lo son, mientras que el resto no.

También es importante reparar en que los bajos valores de *recall* se explican en el hecho de que cuando se mide el *recall* en el primer elemento, el mayor número de elementos relevantes que se puede recomendar es uno.

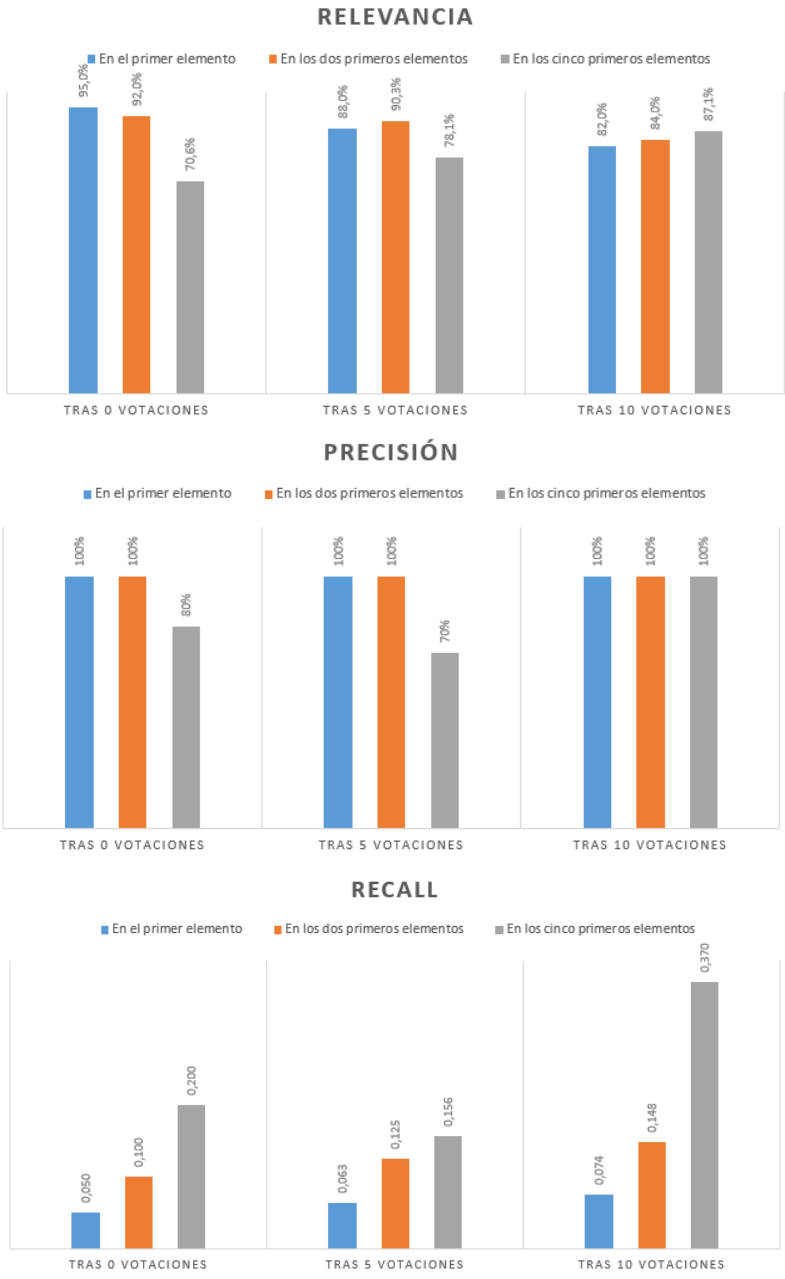

Figura 5.1: Relevancia, precisión y *recall*

Capítulo 6

Conclusiones y trabajo futuro

Se ha presentado un sistema de recomendación híbrido y adaptable. Este sistema está formado por dos componentes, el primero produce recomendaciones basadas en las cinco W's del periodismo [65] y la veracidad de la fuente de información. El otro componente produce recomendaciones basándose en algoritmos de filtrado colaborativo. Este segundo componente evalúa un conjunto de algoritmos y selecciona automáticamente el que mejor se ajusta al dataset actual. Dicha evaluación se puede hacer *off-line* mientras un algoritmo está produciendo recomendaciones para los usuarios. Por último, este componente permite cambiar de algoritmo y la adición de nuevos algoritmos para ser evaluados y usados en tiempo de ejecución, de modo que es fácilmente ampliable. Los primeros resultados de la evaluación de este sistema de recomendación sugieren que este no se ve afectado por el problema de arranque en frío.

Como trabajo futuro que puede ser llevado a cabo para la mejora de este sistema se contemplan varias vías. Por un lado, es recomendable la mejora del rendimiento en ambos componentes. El componente basado en algoritmos de filtrado colaborativo puede ser mejorado utilizando heurísticos que puedan predecir el resultado de la evaluación de un recomendador, buscando correlaciones entre algoritmos de modo que se pueda saltar la evaluación de algunos. La carga de trabajo del otro componente puede ser reducida también utilizando heurísticos como eliminar noticias demasiado antiguas o que tuvieron lugar demasiado lejos como para ser relevantes.

Además de la mejora en el rendimiento, se considera la adición de nuevas variables para mejorar y extender las funciones de Wesomender. Por ejemplo, se pueden aplicar técnicas de análisis de sentimiento como [47] de modo que se habilita la recomendación de noticias basada en sentimiento. Por otro lado, es posible refinar el componente basado en contenido con opiniones de más expertos. También se puede hacer un estudio para que este componente sea aplicable a otras áreas, como la recomendación de contenido bibliográfico. Por último, otra característica interesante para incorporar en este sistema consiste en la implementación de técnicas de *feedback* implícito como la expuesta en [49] de modo que no sería necesario la intervención de los usuarios para recoger votos de noticias, sino que se podrían calcular en base a su intervención con el sistema.

6.1. Difusión del trabajo científico

Se contemplan cuatro vías para la difusión del trabajo científico. En primer lugar, la difusión del código del recomendador a través de un repositorio popular de software, como puede ser github[1]. De este modo el recomendador podrá ser reutilizado en otros proyectos.

Una segunda vía de difusión, con poco alcance son los seminarios internos en el grupo de investigación WESO, en el que los miembros del grupo exponen sus investigaciones.

Las dos vías con más alcance son la publicación de dos artículos en el que se describe el sistema.

[1] `https://github.com/alexmontesg/Wesomender`

En primer lugar, se ha enviado el artículo *Towards an Adaptive and Hybrid News Recommendation System for Journalists: The Wesomender Approach* a la octava edición de las conferencias de doctorado del SEERC (*South-East European Research Centre*)[2] el día 13/05/2013. En dicho artículo se describía el modelo teórico del sistema, junto con unos resultados preliminares del mismo.

Por último, un nuevo artículo titulado *Towards a Journalist-Based News Recommendation System: The Wesomender Approach* fue enviado a la revista Expert Systems with Applications[3]. Se escogió esta revista por dos motivos. En primer lugar, publican una cantidad significativa de artículos sobre sistemas de recomendación, por lo que este trabajo parece encajar a la perfección. Por otro lado, su factor de impacto es bastante alto (2.203) según los datos del *Journal Citations Reports*.

[2] `http://www.seerc.org/dsc2013/index.html`
[3] `http://ees.elsevier.com/eswa/`

Capítulo 7

Caso de uso

La investigación realizada, tanto la de recomendación de contenidos como la de búsqueda y la de etiquetado de vídeos son aplicables a Freews. En el desarrollo de Freews se ha pensado única y exclusivamente en la inclusión de la investigación realizada, dejando de lado otras características importantes como son el diseño del mismo o la seguridad.

7.1. Tecnologías empleadas

7.1.1. Apache Mahout

Mahout es una librería de código abierto de Apache que implementa algoritmos de aprendizaje, o *machine learning*. En la actualidad implementa principalmente, algoritmos de recomendación (filtrado colaborativo), *clustering* y clasificación [52].

Mahout ha sido diseñada para ser escalable, de modo que su uso es apropiado cuando la cantidad de datos con la que se trabaja es muy grande.

En el mundo de la recomendación con Mahout, principalmente se utiliza recomendación de contenidos en base al usuario. La recomendación en base al usuario no se basa en la similitud entre dos usuarios en términos de edad, sexo... sino que se basa en la similitud entre sus gustos. Típicamente lo que se hace es buscar a sus vecinos (usuarios con gustos más similares) y de cada vecino se busca el vídeo que pueda interesar al usuario inicial. Mahout permite especificar la cantidad de vecinos o que lo calcule él mismo en base a un umbral de similitud. Mahout permite además de definir el modo de calcular esa similitud.

Además de la recomendación en base al usuario, Mahout implementa otros tipos de recomendación, como recomendación en base al ítem (vídeo en este caso), Slope One, SVD, interpolación lineal basada en objetos, y recomendación basada en clústers.

Mahout proporciona herramientas para saber qué algoritmo recomienda mejor en cuanto se recopilen datos suficientes. Para hacer esto Mahout toma un porcentaje significativo de esos datos y recomienda en base a ello, después toma el resto de datos y comprueba si ha acertado o no.

Los datos con los que trabaja Mahout para recomendar son bastante simples, tan solo consiste en asociar usuarios e ítems, y opcionalmente, a esta asociación se le da un valor de preferencia. En el caso de Freews se podrían almacenar estas asociaciones sin valor de preferencia en base a los vídeos que un usuario visita (feedback implícito), se podría asignar un valor de preferencia si el usuario puntúa un vídeo (feedback explícito) o se podrían combinar feedback implícito y explícito. Ignorar el valor de preferencia puede ser bueno ya que se optimiza el consumo de memoria, y se libera al usuario de puntuar vídeos, sin embargo, produce recomendaciones menos precisas.

Internamente Mahout reinventa la forma de guardar las preferencias en memoria para ocupar menos, utilizando clases como `PreferenceArray` y sus implementaciones.

Dentro de este proyecto se utilizará Apache Mahout como soporte al sistema de recomendación de vídeos.

7.1.2. Apache Maven

Maven es una herramienta de gestión de proyectos que abarca un *Project Object Model* (POM), un conjunto de estándars, un ciclo de vida del proyecto, un sistema de gestión de dependencias y lógica para ejecutar plugins en distintas fases del ciclo de vida.

En este proyecto, se utilizará Maven meramente como una herramienta de gestión de dependencias, compilación y despliegue del proyecto.

7.1.3. Apache Solr

Solr es un motor de búsqueda escalable y listo para desplegar, que está optimizado para buscar entre gran cantidad de datos (principalmente texto en lenguaje natural), y devolver los resultados ordenados por relevancia [29].

Se utilizará Solr en Freews para implementar el buscador de vídeos.

7.1.4. Apache Struts

Apache Struts es un framework que permite el desarrollo de aplicaciones web robustas y mantenibles de forma sencilla [19]. Struts permite hacer mucho con relativamente poco esfuerzo en el aprendizaje y provee una infraestructura sólida sobre la que construir aplicaciones web. Además, está basada en el patrón Modelo-Vista-Controlador (*Model-View-Controller*), lo que permite el desacoplamiento entre funcionalidades de la aplicación, haciendo el código más mantenible.

7.1.5. FFmpeg

FFmpeg es una plataforma para grabar, convertir y reproducir vídeos. FFmpeg es software libre distribuido bajo licencia GPL [25]. FFmpeg se usará en Freews para la generación de noticiarios a partir de clips de vídeo junto con cabeceras y ráfagas.

7.1.6. MediaElement.js

Con la llegada de HTML5 se ha facilitado la inclusión de contenido multimedia en los sitios web. Sin embargo, el reproductor de vídeo estándar presenta algunos problemas. El más grave, la incompatibilidad entre navegadores y formatos de vídeo. Dado que la preferencia del cliente era almacenar los vídeos únicamente en MP4, este formato solo es soportado por Internet Explorer 9+, Safari, iOS y Android (Chrome lo soporta pero lo abandonará en breve [34]). Por ello nacen otros reproductores como MediaElement.js [21], que implementan un fallback a flash, de modo que si el navegador no es compatible, se transforma el vídeo a Flash al vuelo sin necesidad de configurar nada por parte del cliente ni del servidor.

7.1.7. MongoDB

MongoDB es una base de datos NoSQL orientada a documentos. Está diseñada para aplicaciones web y su modelo de datos está orientado a conseguir un alto rendimiento en operaciones de lectura y escritura. También es altamente escalable. [5] [56].

MongoDB no es solo atractiva por el rendimiento, sino por la sencillez de su modelo de datos. Al permitir guardar documentos anidados de manera jerárquica muy similar a JSON, se evitan en la gran mayoría de ocasiones los típicos `JOIN` de SQL. Un ejemplo de documento almacenado en MongoDB puede verse en el Listing 7.

Como puede observarse en el Listing 7, la ficha de un vídeo puede representarse fácilmente como un documento, de hecho, se asemeja más a un documento que a una entrada en una o varias tablas o a un nodo de un grafo, por eso para Freews se ha escogido MongoDB en lugar de otras bases de datos SQL o NoSQL como Neo4j.

```
{
        "_id" : ObjectID("513f61d86f2027e27cb3be46"),
        "_class" : "com.model.Video",
        "headline" : { "pt" : "As exportacoes alemas caem",
                "en" : "German exports fall",
                "es" : "Caen las exportaciones alemanas"
        }, "tags" : { "pt" : ["Exportacoes", "Alemanha", "Economia"],
                "en" : ["Exports", "Germany", "Economy"],
                "es" : ["Exportaciones", "Alemania", "Economoa"]
        }, "date" : ISODate("2013-03-12T17:11:52.257Z"),
        "lat" : 21.29,
        "lon" : 33.75
}
```

Listing 7: Ficha de vídeo almacenada en MongoDB

7.1.8. Spring

Spring es un framework ligero para el desarrollo de aplicaciones Java basado en la inyección de dependencias (también llamado *Dependency Injection* o *Inversion of Control Container*). La inyección de dependencias permite un desacoplamiento total entre capas ya que las dependencias entre capas se cargan en tiempo de ejecución. Spring toma el control de la ejecución antes de que la aplicación comience, inyecta las dependencias y luego ejecuta el programa. [36]

Además, Spring implementa gran cantidad de utilidades para facilitar la persistencia de datos en multitud de bases de datos, entre ellas MongoDB, por lo que no solo se usará para la inyección de dependencias, sino que también simplificará el código en las capas de persistencia.

7.1.9. Twitter Bootstrap

Bootstrap es un framework sencillo para el desarrollo de frontends de aplicaciones web que promueve el diseño responsable (o *Responsive Design*). Además de ayudar en la apariencia externa del frontend, Bootstrap permite que este se adapte al dispositivo en el que se esta visualizando, ya sea ordenador, móvil o tablet.

7.2. Descripción de Freews

Freews es un portal en el que los usuarios compartirán vídeos de noticias con otros periodistas. Los vídeos serán subidos por los usuarios. Un equipo de expertos verificarán que estos son correctos, podrán ampliar la información que proporciona el usuario que subió el vídeo y seleccionar las mejores tomas. Una vez completado este proceso, será puesto a disposición de los usuarios para que puedan realizar un noticiario con él. Este ciclo de vida de la subida de un vídeo puede apreciarse de forma más gráfica en la Figura 7.1.

Con los vídeos que están publicados, los usuarios podrán realizar noticiarios, que podrán insertar en sus página web, sus blogs, o descargar libremente. En estos noticiarios, además de los vídeos de noticias se incluirán ráfagas, cabeceras y créditos con el logotipo corporativo de Freews. Las ráfagas son vídeos de uno o dos segundos que irán entre dos noticias, mientras que las cabeceras son algo más largas y van al principio del noticiario. Por último los créditos se encuentran al final del mismo. En función de la categoría en la que se cree el noticiario, estas infografías tendrán un *look-and-feel* diferente. Este proceso de creación de noticiarios puede verse en la Figura 7.2.

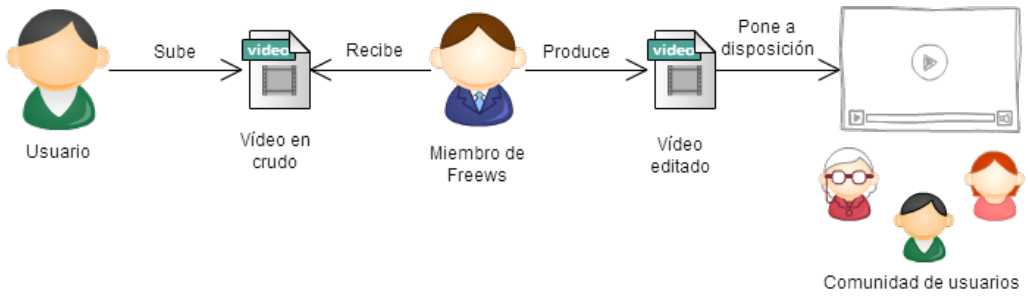

Figura 7.1: Subida de un vídeo

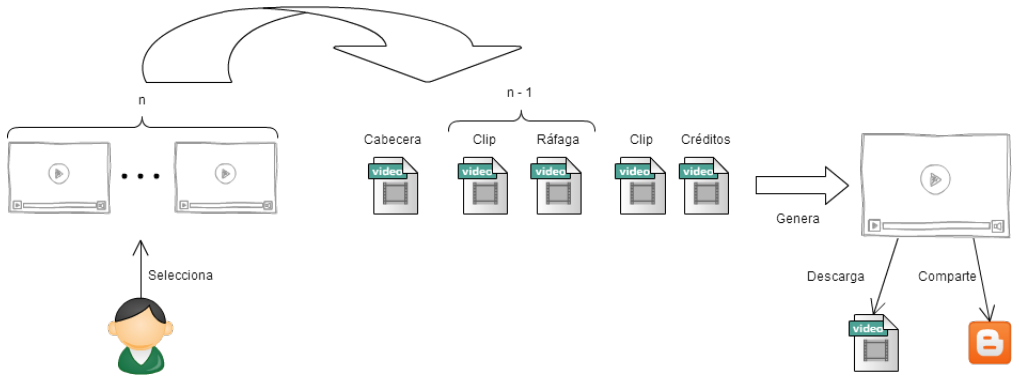

Figura 7.2: Creación de un noticiario

7.3. Arquitectura del sistema

Freews se encuentra dividido en dos sistemas bien diferenciados, cliente y servidor. A su vez, el servidor sigue una arquitectura de N-Capas, en el que se distinguen seis tipos de entidades para los cuales hay una capa de persistencia, otra de negocio y otra de presentación totalmente desacopladas, además de una capa de utilidades que podrán ser usadas por cualquiera de las otras capas y la capa del modelo de datos.

Por motivos de infraestructura, las N-Capas se encuentran desplegadas en una sola máquina, sin embargo, es técnicamente posible distribuirlas en n máquinas. Para explicar mejor esto es necesario distinguir los conceptos *layers* y *tiers*. Las *layers* hacen referencia a la estructura lógica del código, mientras que las *tiers* están relacionadas con dónde está desplegada cada *layer*, es decir, una *tier* es la ubicación física de una o más *layers* [44]. Freews es *n-layers*, y por motivos de infraestructura *1-tier*.

La arquitectura de Freews puede verse de forma simplificada en la Figura 7.3. Por un lado, en el cliente, además del modelo de datos estarían las `Action` de Struts, junto con el fichero `struts.xml` forman el controlador. Por otro lado está la vista, los ficheros jsp que se transforman en html y se sirven al navegador del usuario. Por último, el modelo de datos, compartido por cliente y servidor.

En el servidor estarían las capas de persistencia, negocio y presentación. Para estos grandes grupos existe una capa por cada tipo de dato distinto que use Freews. Todas las capas están desacopladas mediante el uso de Spring.

Las capas de presentación consisten en un conjunto de servicios web RESTful que se ofrecen al cliente. Una posible mejora relativa a la seguridad en el servidor es llevar a cabo la autenticación en estos servicios mediante OAuth[1] o protocolos similares.

Las capas de negocio llevan a cabo operaciones relativas a la lógica de negocio de Freews, mientras que las de persistencia son las encargadas de realizar operaciones CRUD sobre MongoDB y Apache Solr.

[1] http://oauth.net/

Por último, las clases de utilidades, transversales a la arquitectura del servidor y entre las que destaca una, encargada de realizar operaciones con los ficheros de vídeo. Para ello, hace uso del ejecutable de FFmpeg.

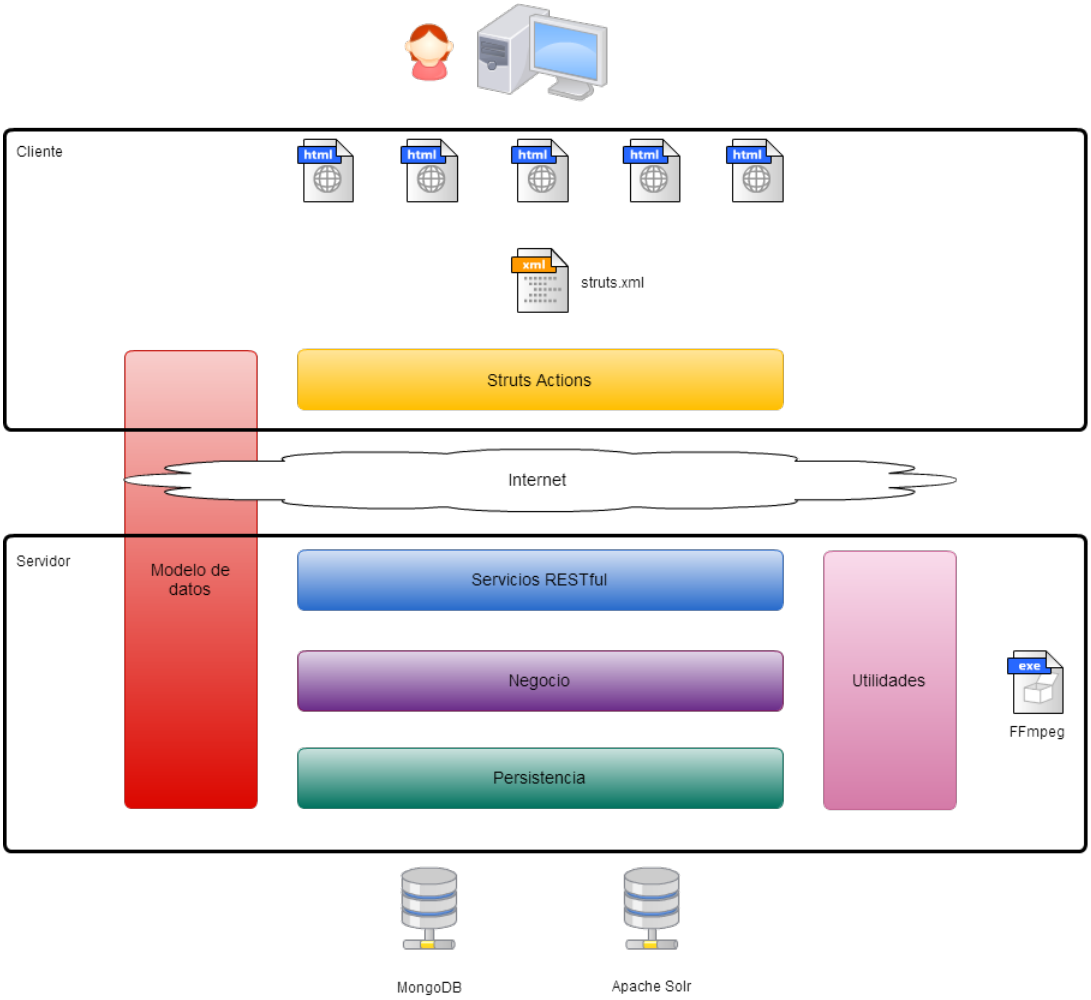

Figura 7.3: Arquitectura simplificada de Freews

7.4. Integración de las líneas de investigación

7.4.1. Recomendación de contenido

El sistema anteriormente descrito es directamente aplicable en Freews, por lo que su integración es sencilla. Mahout permite recoger datos directamente de MongoDB por lo que toda la información que necesita la obtiene de la base de datos de Freews. Por otro lado, en el cliente web se ha incluido un sistema tradicional de voto a través de estrellas utilizando el plugin de JQuery, Raty[2], de modo que los usuarios puedan añadir votos a las noticias.

Para la obtención de los parámetros necesarios para la recomendación, el tiempo actual puede calcularse utilizando código Java, al igual que el porcentaje de entidades relacionadas. Para la obtención de la posición del usuario, se ha utilizado la API de geoposicionamiento de HTML5.

La forma más natural de incorporar la recomendación es variar el orden en el que se presentan las noticias, de modo que se presenten primero las que más posibilidades tienen de ser relevantes para el usuario, con lo cual, la recomendación de contenido en Freews

[2]http://wbotelhos.com/raty/

consiste en cambiar el orden en el que aparecen las noticias en función al resultado de la recomendación.

7.4.2. Recomendación de etiquetas

Para realizar la recomendación de etiquetas se ha optado por utilizar los servicios web de Zemanta, porque esta tecnología ya ha sido utilizada previamente en trabajos de enriquecimiento semántico de texto como [58] y [72].

Un problema de utilizar Zemanta es que esta tecnología no cumple al 100 % los requisitos necesarios para Freews en recomendación de etiquetas, ya que no garantiza que dado un mismo texto en distintos idiomas, las etiquetas que recomiende para cada uno se refieran al mismo concepto. Para paliar este efecto se ha optado por usar Zemanta solo con el texto en inglés, de modo que tan solo recomiende etiquetas en este idioma. Una vez obtenidas las etiquetas en inglés, para traducirlas, se realiza la consulta SPARQL mostrada en el Listing 8 a la DBpedia[3], de este modo se garantiza que el concepto señalado por cada etiqueta sea el mismo.

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT DISTINCT (str(?name) as ?keyword) (lang(?name) as ?lang) WHERE {
        { ?entity rdfs:label "Keyword1"@en .
        ?entity rdfs:label ?name .} UNION
        { ?entity rdfs:label "Keyword2"@en .
        ?entity rdfs:label ?name .} UNION
        ...
        FILTER (langMatches(lang(?name), "pt") ||
          FILTER (langMatches(lang(?name), "es")
}
```

Listing 8: Consulta SPARQL para traducir etiquetas

7.4.3. Búsqueda

Para incluir técnicas de procesamiento de lenguaje natural en Freews se ha optado por utilizar Apache Solr. Este permite configurar un esquema en un fichero XML en el que se configuran:

- Modelo de datos en lo referente a Solr.
- Tipos de datos del modelo.
- Técnicas de procesamiento de lenguaje natural que se aplicarán.

Para una búsqueda textual sobre Freews se ejecutará el proceso representado en la Figura 7.4.

Como puede observarse, en primer lugar se envía la consulta al servidor Solr que aplicará técnicas de procesamiento de lenguaje natural a la misma. Tras ello, el servidor Solr consultará el índice y devolverá, de manera ordenada, los identificadores de los vídeos encajan mejor con la búsqueda. Con estos identificadores, se buscarán los vídeos en la base de datos MongoDB.

En el esquema de Solr es necesario definir qué tipo de datos simples se utilizarán, como se puede observar en el Listing 9, la definición de los mismos es trivial.

Más interesante y compleja es la definición de datos más complejos y que deban ser tratados por Solr, es decir, tipos de datos en los que se almacene sea lenguaje natural. Para ello es necesario no solo especificar el tipo de dato, sino el tipo de analizadores de

[3] http://dbpedia.org/About

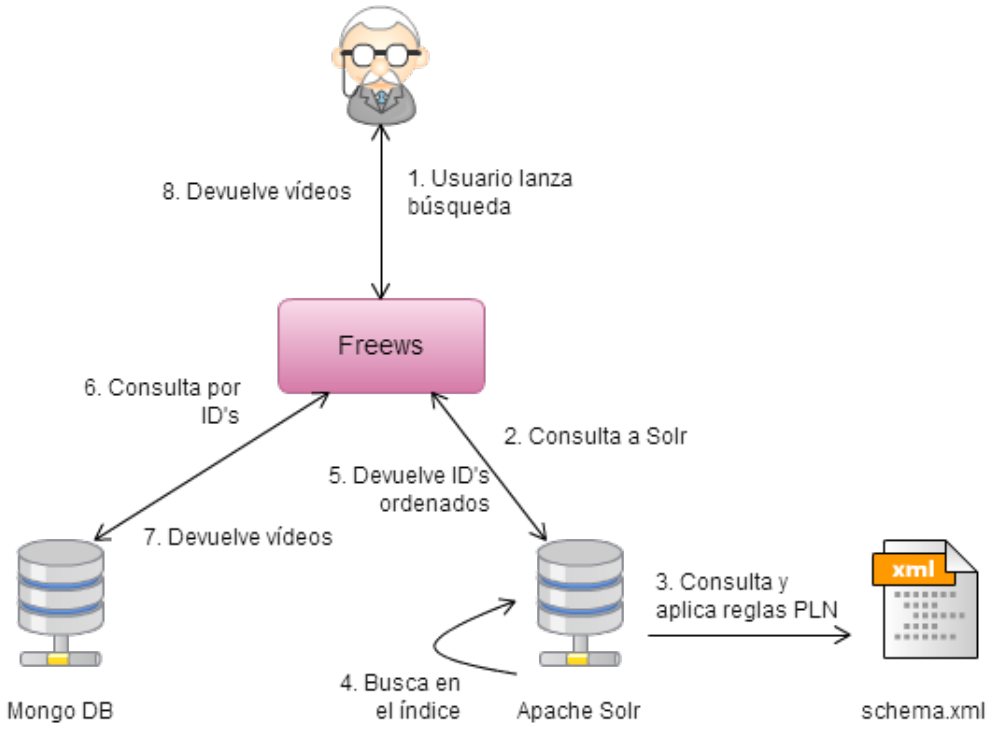

Figura 7.4: Proceso de búsqueda de un vídeo

```
<types>
        <fieldType name="string" class="solr.StrField" sortMissingLast="true" />
        <fieldType name="date" class="solr.TrieDateField" precisionStep="0"
          positionIncrementGap="0"/>
        <fieldType name="location" class="solr.LatLonType" subFieldSuffix="_coordinate"/>
        ...
</types>
```

Listing 9: Tipos de datos simples Solr

Solr que se utilizarán. Un ejemplo de ello para lenguaje natural en inglés puede verse en el Listing 10. De forma análoga se añadirán campos para tratar el texto en castellano y en portugués.

Como puede verse en el Listing 10, al lenguaje natural se le aplican dos analizadores, por un lado en tiempo de indexado, es decir, cuando se guardan datos del tipo especificado en Solr y por otro en tiempo de búsqueda, este segundo analizador se aplica a las consultas que se hagan sobre campos de ese tipo.

Para indexar datos, en primer lugar se eliminan tabulaciones, dobles espacios en blanco, saltos de línea, etc. Tras ello, se eliminan palabras vacías. Seguidamente se procede a la división de palabras que llevan guiones en medio u otros símbolos no alfanuméricos, o palabras escritas siguiendo el estilo CamelCase. Después, cada *token* generado se pasa a minúsculas y por último, se aplicará *Stemming* a cada *token* excepto aquellos marcados como protegidos.

El análisis en tiempo de búsqueda es análogo, con la excepción de que antes de eliminar palabras vacías se aplicará *Query expansion* indexando Wordnet en un formato apropiado para Solr.

Por último, es necesario especificar que datos se guardarán en Solr y el tipo de los mismos. Como se puede ver en el Listing 11, en primer lugar se definen los tipos básicos de los vídeos, seguidamente, se crean unos campos de texto que no serán almacenados, sino que, mediante los nodos `copyField` harán referencia a todos los campos textuales en un idioma concreto. Esto se hace para simplificar las consultas, ya que se puede consultar

```
<types>
        ...
        <fieldType name="text_en" class="solr.TextField"
          positionIncrementGap="100" autoGeneratePhraseQueries="true">
                <analyzer type="index">
                        <tokenizer class="solr.WhitespaceTokenizerFactory"/>
                        <filter class="solr.StopFilterFactory" ignoreCase="true"
                          words="lang/stopwords_en.txt" enablePositionIncrements="true"/>
                        <filter class="solr.WordDelimiterFilterFactory"
                          generateWordParts="1" generateNumberParts="1"
                          catenateWords="1" catenateNumbers="1" catenateAll="0"
                          splitOnCaseChange="1"/>
                        <filter class="solr.LowerCaseFilterFactory"/>
                        <filter class="solr.KeywordMarkerFilterFactory"
                          protected="protwords.txt"/>
                        <filter class="solr.PorterStemFilterFactory"/>
                </analyzer>
                <analyzer type="query">
                        <tokenizer class="solr.WhitespaceTokenizerFactory"/>
                        <filter class="solr.SynonymFilterFactory" synonyms="synonyms.txt"
                          ignoreCase="true" expand="true"/>
                        ...
                </analyzer>
        </fieldType>
        ...
</types>
```

Listing 10: Procesamiento del lenguaje natural para texto en inglés

directamente sobre ese campo general y Solr encontrará los vídeos que encajen, ya sea por su titular, su descripción o sus etiquetas.

7.4.4. Semántica

En Freews, y tras lo expuesto en la sección 3.3 se ha optado por el uso de Microdatos para añadir semántica al HTML. Esta elección se debe a la simplicidad de uso del mismo y que esta simplicidad está equilibrada con su expresividad y extensión.

Existen dos esquemas que representan todo lo que necesita Freews en `schema.org`. Por un lado, Freews necesita representar noticias, para lo que se usará el esquema `NewsArticle`[4]. Por el otro, el componente principal de las noticias publicadas en Freews son vídeos, por lo que se necesita añadir semántica a los vídeos, para ello está el esquema `VideoObject`[5].

En el Listing 13 puede observarse el código HTML de una noticia de Freews etiquetada con microdatos y compararla con el Listing 12 que representa la misma noticia sin microdatos. Como pueden observarse, las diferencias no son excesivas.

Sin embargo, aunque el Listing 13 sea más verboso, puede ser analizado más fácilmente por las máquinas, como muestra de ello se ha utilizado la Structured Data Testing Tool de Google[6] y en la Figura 7.5. En ella puede observarse que Google entiende qué hay en la página. Sabe que se trata de un artículo de noticias, conoce su titular y su cuerpo, sabe cuando se ha creado y cuando se ha publicado y quien lo ha hecho, donde sucedió la noticia qué vídeo lo ilustra e incluso conoce algunos metadatos del vídeo, con lo cual, podrá actuar en consecuencia y mostrar con mejores visualizaciones los datos de esta página.

[4] http://schema.org/NewsArticle
[5] http://schema.org/VideoObject
[6] http://support.google.com/webmasters/bin/answer.py?hl=en&answer=173839&topic=1088473

```
...
<fields>
        <field name="id" type="string" indexed="true" stored="true"
          required="true" multiValued="false" />
        <field name="geo" type="location" indexed="true" stored="true"/>
        <field name="date" type="date" indexed="true" stored="true"/>
        <field name="headline_en" type="text_en" indexed="true" stored="true"
          required="true" multiValued="false"/>
        <field name="description_en" type="text_en" indexed="true" stored="true"
          required="true" multiValued="false"/>
        <field name="tags_en" type="text_en" indexed="true" stored="true"
          required="true" multiValued="true"/>
        <field name="text_en" type="text_en" indexed="true" stored="false"
          multiValued="true"/>
        ...
</fields>
<uniqueKey>id</uniqueKey>
<copyField source="headline_en" dest="text_en"/>
<copyField source="description_en" dest="text_en"/>
<copyField source="tags_en" dest="text_en"/>
...
```

Listing 11: Datos almacenados en Solr

```
<div>
 <header>
   <div>
     <a href="http://156.35.82.101:9006/Freews/SeekVideos">
       <img src="./logo_freews.png" itemprop="logo" />
     </a>
   </div>
   <div>
     <h1>Freews</h1>
   </div>
 </header>
 <h2>
   ECB cuts eurozone interest rate to new record low of 0.5%
 </h2>
 <aside>
   <div>
     <span>
       <a href="http://156.35.82.101:9006/Freews/Search=query=Economy">Economy</a>
     </span>,
     <span>
       <a href="http://156.35.82.101:9006/Freews/Search=query=Eurozone">Eurozone</a>
     </span>
   </div>
   <div id="map">
   </div>
 </aside>
 <section>
   <div>
     <video controls="controls" src="media/videoECB.mp4" type="video/mp4"
      poster="media/posters/videoECB.jpeg">
     </video>
   </div>
   <p>
     The European Central Bank (ECB) has cut its benchmark interest rate to a new ...
   </p>
   <p>
     Grabado el 02/05/2013. Publicado el 02/05/2013
   </p>
 </section>
</div>
```

Listing 12: Noticia de Freews sin microdatos

```
<div itemscope itemtype="http://schema.org/NewsArticle">
 <header itemprop="publisher" itemscope itemtype="http://schema.org/Organization">
   <div>
     <a href="http://156.35.82.101:9006/Freews/SeekVideos">
       <img src="./logo_freews.png" itemprop="logo" />
     </a>
   </div>
   <div>
     <h1 itemprop="name">Freews</h1>
   </div>
 </header>
 <h2 itemprop="headline">
   ECB cuts eurozone interest rate to new record low of 0.5%
 </h2>
 <aside>
   <div itemprop="keywords">
     <span>
       <a href="http://156.35.82.101:9006/Freews/Search=query=Economy">Economy</a>
     </span>,
     <span>
       <a href="http://156.35.82.101:9006/Freews/Search=query=Eurozone">Eurozone</a>
     </span>
   </div>
   <div id="map" itemprop="contentLocation" itemscope itemtype="http://schema.org/Place">
     <div itemprop="geo" itemscope itemtype="http://schema.org/GeoCoordinates">
       <meta itemprop="latitude" content="50.109756" />
       <meta itemprop="longitude" content="8.673395" />
     </div>
   </div>
 </aside>
 <section>
   <div itemprop="video" itemscope itemtype="http://schema.org/VideoObject">
     <video controls="controls" src="media/videoECB.mp4" type="video/mp4"
      poster="media/posters/videoECB.jpeg">
       <meta itemprop="thumbnail" content="media/posters/videoECB.jpeg" />
       <meta itemprop="encodingFormat" content="mpeg4" />
       <meta itemprop="url" content="media/videoECB.mp4" />
       <meta itemprop="videoFrameSize" content="1280x720" />
       <meta itemprop="videoQuality" content="2400" />
     </video>
   </div>
   <p itemprop="articleBody">
     The European Central Bank (ECB) has cut its benchmark interest rate to a new ...
   </p>
   <p>
     Grabado el
       <time itemprop="dateCreated" datetime="2013/05/02" format="yyyy/MM/dd" >
         02/05/2013
       </time>. Publicado el
       <time itemprop="datePublished" datetime="2013/05/02" format="yyyy/MM/dd" >
         02/05/2013
       </time>
   </p>
 </section>
</div>
```

Listing 13: Noticia de Freews etiquetada con microdatos

Datos estructurados extraídos

Item

type:	http://schema.org/newsarticle
property:	
publisher:	*Item 1*
headline:	ECB cuts eurozone interest rate to new record low of 0.5%
keywords:	Economy, Eurozone
contentlocation:	*Item 2*
video:	*Item 3*
articlebody:	The European Central Bank (ECB) has cut its benchmark interest rate to a new record low amid ongoing worries about the eurozone's economic health.
datecreated:	2013/05/02
datepublished:	2013/05/02

Item 1

type:	http://schema.org/organization
property:	
logo:	logo_freews.png
name:	Freews

Item 2

type:	http://schema.org/place
property:	
geo:	*Item 4*

Item 4

type:	http://schema.org/geocoordinates
property:	
latitude:	50.109756
longitude:	8.673395

Item 3

type:	http://schema.org/videoobject
property:	
thumbnail:	media/posters/videoECB.jpeg
encodingformat:	mpeg4
url:	media/videoECB.mp4
videoframesize:	1280x720
videoquality:	2400

Figura 7.5: Análisis con la Structured Data Testing Tool de una noticia de Freews

Capítulo 8

Planificación

La realización del proyecto se ha dividido en dos grandes fases debido a las necesidades especiales del mismo. Ha sido necesario desarrollar un prototipo funcional de Freews con urgencia con el que buscar inversores, dejando a un lado la labor de investigación. Una vez realizado el prototipo, se llevaron a cabo las investigaciones descritas en este documento y la integración de las mismas.

La primera parte fue llevada a cabo entre 22/05/2012 y el 24/09/2012 trabajando a tiempo parcial. Se desarrollaron cuatro iteraciones en las que al comienzo de cada iteración se realizaba una reunión con el director del proyecto y el cliente en las que se fijaban las tareas a realizar, con lo que en cada reunión se realizaba un análisis y un diseño superficial del trabajo a realizar durante el siguiente mes, además, antes de cada reunión era necesario desplegar los avances de modo que el cliente pudiese probar la nueva funcionalidad y señalar fallos durante la siguiente reunión. Las funcionalidades introducidas en cada iteración pueden verse en la Figura 8.1.

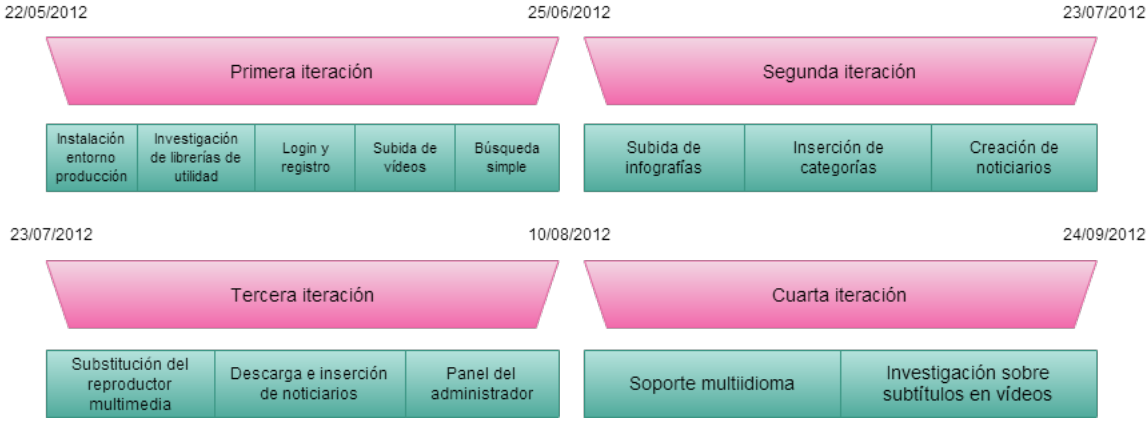

Figura 8.1: Planificación de las funcionalidades de Freews

Tras ello se llevaron a cabo las tareas de investigación dedicando de media una hora al día entre el 24/09/2012 y el 21/05/2013, fecha en la que se entregó el artículo en la revista *Expert Systems with Applications*.

8.1. Presupuesto

8.1.1. Aplicaciones informáticas

Concepto	Unidades	Precio unitario	Cantidad	Total
Sistema operativo Linux Mint	Uds.	0 €	1	0 €
Sistema operativo Windows 7 (Licencia Académica)	Uds.	0 €	1	0 €

Concepto	Unidades	Precio unitario	Cantidad	Total
Sistema operativo Windows 8 (Licencia Académica)	Uds.	0 €	1	0 €
Sistema operativo Windows Server 2008 (Licencia Académica)	Uds.	0 €	1	0 €
Microsoft Project 2010 (Licencia Académica)	Uds.	0 €	1	0 €
JDK7	Uds.	0 €	1	0 €
MikTeX (implementación LaTeX)	Uds.	0 €	1	0 €
Eclipse IDE	Uds.	0 €	1	0 €
Git (cliente)	Uds.	0 €	1	0 €
Maven	Uds.	0 €	1	0 €
Apache Mahout	Uds.	0 €	1	0 €
Apache Solr	Uds.	0 €	1	0 €
FFmpeg	Uds.	0 €	1	0 €

Total de este capítulo: 0 €

8.1.2. Bienes, equipos y servicios informáticos

Concepto	Unidades	Precio unitario	Cantidad	Periodo amort.	Amort. proyecto	Total
Ordenador portátil	Uds.	350 €	1	3 años	116,67 €	116,67 €
Ordenador sobremesa	Uds.	600 €	1	3 años	200 €	200 €
Acceso a Internet	Meses	39 €	14	N/A	N/A	546 €

Total de este capítulo: 862,67 €

8.1.3. Consumibles

Concepto	Unidades	Precio unitario	Cantidad	Total
Papel	Uds.	0,06 €	41	2,46 €
Libreta	Uds.	2,20 €	1	2,20 €
Abono CTA (2 zonas)	Mes.	42,80 €	13	599,20 €

Total de este capítulo: 603,86 €

8.1.4. Desarrollo del prototipo

Concepto	Unidades	Precio unitario	Cantidad	Total
Hospedaje proyecto (Servidor WESO)	Meses	0,905 €	14	12,67 €
Analista programador	Horas	10 €	320	3200 €

Total de este capítulo: 3212,67 €

8.1.5. Entrega del proyecto

Concepto	Unidades	Precio unitario	Cantidad	Total
DVDs	Uds.	0,79 €	3	2,37 €

Total de este capítulo: 2,37 €

8.1.6. Investigación

Concepto	Unidades	Precio unitario	Cantidad	Total
Director de proyecto	Horas	25 €	80	2000 €
Investigador junior	Horas	15 €	275	4125 €
Obtención bibliografía	Fuentes	0 €	75	0 €

Total de este capítulo: 6125 €

8.1.7. Resumen del presupuesto

Capítulo	Subtotal
Aplicaciones informáticas	0 €
Bienes, equipos y servicios informáticos	862,67 €
Consumibles	603,86 €
Desarrollo del prototipo	3212,67 €
Entrega del proyecto	2,38 €
Investigación	6125 €

8.1.8. Total

Total (sin imp.)	Impuestos (IVA: 21 %)	Total (con imp.)
10806,58 €	2269,38 €	13075, 96 €

Índice alfabético

Bibliografía

[1] B. Adida, M. Birbeck, S. McCarron, and I. Herman. Microformats. `http://www.w3.org/TR/rdfa-core/`, 2012. Consultado: 22/04/2013.

[2] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering*, 17(6):734–749, 2005.

[3] J.-W. Ahn, P. Brusilovsky, J. Grady, D. He, and S. Y. Syn. Open user profiles for adaptive news systems: help or harm? In *Proceedings of the 16th international conference on World Wide Web*, WWW '07, pages 11–20, New York, NY, USA, 2007. ACM.

[4] M. Balabanovic and Y. Shoham. Content-based, collaborative recommendation. *Communications of the ACM*, 40:66–72, 1997.

[5] K. Banker. *MongoDB in action*. Manning Pubs Co Series. Manning Publications Company, 2011.

[6] A. B. Barragáns-Martínez, E. Costa-Montenegro, J. C. Burguillo, M. Rey-L'opez, F. A. Mikic-Fonte, and A. M. Peleteiro. A hybrid content-based and item-based collaborative filtering approach to recommend tv programs enhanced with singular value decomposition. *Information Sciences*, 180(22):4290–4311, 2010.

[7] R. M. Bell and Y. Koren. Scalable collaborative filtering with jointly derived neighborhood interpolation weights. pages 43–52, 2007.

[8] D. Benz, K. H. L. Tso, and L. Schmidt-Thieme. Automatic bookmark classification: A collaborative approach. In *Proceedings of the Second Workshop on Innovations in Web Infrastructure*, IWI 2006, 2006.

[9] T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. *Scientific American*, 284(5):34–43, May 2001.

[10] D. Billsus and M. J. Pazzani. A personal news agent that talks, learns and explains. In *Proceedings of the third annual conference on Autonomous Agents*, AGENTS '99, pages 268–275, New York, NY, USA, 1999. ACM.

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

[12] D. Cederholm and R. King. Microformats. `http://microformats.org/`, 2005. Consultado: 22/04/2013.

[13] S.-M. Choi, S.-K. Ko, and Y.-S. Han. A movie recommendation algorithm based on genre correlations. *Expert Systems with Applications*, 39(9):8079 – 8085, 2012.

[14] K. Christidis and G. Mentzas. A topic-based recommender system for electronic marketplace platforms. *Expert Systems with Applications*, 40(11):4370 – 4379, 2013.

[15] P. Cremonesi, R. Turrin, and F. Airoldi. Hybrid algorithms for recommending new items. In *Proceedings of the 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems*, HetRec '11, pages 33–40, New York, NY, USA, 2011. ACM.

[16] F. Crestani. Application of spreading activation techniques in information retrieval. *Artificial Intelligence Review*, 11(6):453–482, 1997.

[17] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society Series B Methodological*, 39(1):1–38, 1977.

[18] M. M. Deza and E. Deza. *Encyclopedia of Distances*, volume 2006. Springer Berlin Heidelberg, 2009.

[19] A. Doray. *Beginning Apache Struts: From Novice To Professional*. Springer (India) Pvt. Limited, 2006.

[20] T. Dunning. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19(1):61–74, 1993.

[21] J. Dyer. Mediaelement.js - html5 video player and audio player with flash and silverlight shims. `http://mediaelementjs.com/`, 2010. Consultado: 02/05/2013.

[22] Equipo de desarrollo de schema.org. schema.org. `http://schema.org/`, 2011. Consultado: 22/04/2013.

[23] B. Fang, S. Liao, K. Xu, H. Cheng, C. Zhu, and H. Chen. A novel mobile recommender system for indoor shopping. *Expert Systems with Applications*, 39(15):11992 – 12000, 2012.

[24] C. Fellbaum, editor. *WordNet: an electronic lexical database*. MIT Press, 1998.

[25] FFmpeg developers. Ffmpeg. `http://ffmpeg.org/`, 2007. Consultado: 30/04/2013.

[26] E. Gabrilovich, S. Dumais, and E. Horvitz. Newsjunkie: Providing personalized newsfeeds via analysis of information novelty. In *In WWW2004*, pages 482–490. ACM Press, 2004.

[27] J. E. L. Gayo. Aplicaciones RDF. Transparencias de la asignatura "Nuevos avances en web semántica", 2012.

[28] J. E. L. Gayo, P. O. de Pablos, and J. M. C. Lovelle. Wesonet: Applying semantic web technologies and collaborative tagging to multimedia web information systems. *Computers in Human Behavior*, 26(2):205–209, Mar. 2010.

[29] T. Grainger and T. Potter. *Solr in Action*. Manning Publications Co., early access edition, 2013.

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

[31] Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. pages 263–272, 2008.

[32] R. J. Hyndman and A. B. Koehler. Another look at measures of forecast accuracy. *International Journal of Forecasting*, 22(4):679 – 688, 2006.

[33] R. Jäschke, L. Marinho, A. Hotho, L. Schmidt-Thieme, and G. Stumme. Tag recommendations in folksonomies. In *Proceedings of the 11th European conference on Principles and Practice of Knowledge Discovery in Databases*, PKDD 2007, pages 506–514, Berlin, Heidelberg, 2007. Springer-Verlag.

[34] M. Jazayeri. Html video codec support in chrome. `http://blog.chromium.org/2011/01/html-video-codec-support-in-chrome.html`, 2011. Consultado: 02/05/2013.

[35] D. Jurafsky and J. H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1st edition, 2000.

[36] M. Konda. *Just Spring*. Real Time Bks. O'Reilly Media, 2011.

[37] Y. Koren, R. M. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30 –37, aug. 2009.

[38] E. F. Krause. *Taxicab Geometry: An Adventure in Non-Euclidean Geometry*, volume 72. Courier Dover Publications, 1987.

[39] M.-H. Kuo, L.-C. Chen, and C.-W. Liang. Building and evaluating a location-based service recommendation system with a preference adjustment mechanism. *Expert Systems with Applications*, 36(2, Part 2):3543 – 3554, 2009.

[40] K. Lang. Newsweeder: Learning to filter netnews. In *in Proceedings of the 12th International Machine Learning Conference (ML95*, 1995.

[41] H. Lee and S. J. Park. Moners: A news recommender for the mobile web. *Expert Systems with Applications*, 32(1):143 – 150, 2007.

[42] G. D. Lekakos and P. Caravelas. A hybrid approach for movie recommendation. *Multimedia Tools and Applications*, 36(1-2):55–70, 2008.

[43] D. Lemire and A. Maclachlan. Slope one predictors for online rating-based collaborative filtering. In *Proceedings of SIAM Data Mining (SDM'05)*, 2005.

[44] R. Lhotka. Should all apps be n-tier? `http://www.lhotka.net/weblog/ShouldAllAppsBeNtier.aspx`, 2005. Consultado: 30/04/2013.

[45] L. Li, D. Wang, T. Li, D. Knox, and B. Padmanabhan. Scene: a scalable two-stage personalized news recommendation system. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, SIGIR '11, pages 125–134, New York, NY, USA, 2011. ACM.

[46] G. Mishne. Autotag: a collaborative approach to automated tag assignment for weblog posts. In *Proceedings of the 15th international conference on World Wide Web*, WWW '06, pages 953–954, New York, NY, USA, 2006. ACM.

[47] A. Moreo, M. Romero, J. L. Castro, and J. M. Zurita. Lexicon-based comments-oriented news sentiment analyzer system. *Expert Systems with Applications*, 39(10):9166 – 9180, 2012.

[48] R. Navigli and S. P. Ponzetto. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250, 2012.

[49] E. R. Núñez-Valdéz, J. M. C. Lovelle, O. S. Martínez, V. García-Díaz, P. O. nez de Pablos, and C. E. M. Marín. Implicit feedback techniques on recommender systems applied to electronic books. *Computers in Human Behavior*, 28(4):1186–1193, July 2012.

[50] J. O'Donovan and B. Smyth. Trust in recommender systems. pages 167–174, 2005.

[51] L. Orman. Fighting information pollution with decision support systems. *J. of Management Information Systems*, 1(2):64–71, 1984.

[52] S. Owen, R. Anil, T. Dunning, and E. Friedman. *Mahout in Action*. Manning Publications Co., first edition, 2011.

[53] K. Pearson. Notes on the history of correlation. *Biometrika*, 13(1):pp. 25–45, 1920.

[54] M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, July 1980.

[55] M. F. Porter. Snowball: A language for stemming algorithms. `http://snowball.tartarus.org/texts/introduction.html`, 2001. Consultado: 18/04/2013.

[56] E. Redmond and J. R. Wilson. *Seven Databases in Seven Weeks*. O'Reilly Media, 2012.

[57] D. J. Rogers and T. T. Tanimoto. A computer program for classifying plants. *Science*, 132(3434):1115–1118, 1960.

[58] M. Rowe and M. Stankovic. Aligning tweets with events: automation via semantics. *Semantic Web Journal*, 3(2):115–130, 2011.

[59] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '02, pages 253–260, New York, NY, USA, 2002. ACM.

[60] C. Shahabi, F. Banaei-Kashani, Y.-S. Chen, and D. McLeod. Yoda: An Accurate and Scalable Web-Based Recommendation System. In C. Batini, F. Giunchiglia, P. Giorgini, and M. Mecella, editors, *Cooperative Information Systems*, volume 2172 of *Lecture Notes in Computer Science*, chapter 31, pages 418–432. Springer Berlin / Heidelberg, Berlin, Heidelberg, Sept. 2001.

[61] C. Shahabi and Y.-S. Chen. An adaptive recommendation system without explicit acquisition of user relevance feedback. *Distributed and Parallel Databases*, 14(2):173–192, 2003.

[62] D.-H. Shih, D. C. Yen, H.-C. Lin, and M.-H. Shih. An implementation and evaluation of recommender systems for traveling abroad. *Expert Systems with Applications*, 38(12):15344 – 15355, 2011.

[63] B. Sigurbjörnsson and R. van Zwol. Flickr tag recommendation based on collective knowledge. In *Proceedings of the 17th international conference on World Wide Web*, WWW '08, pages 327–336, New York, NY, USA, 2008. ACM.

[64] C. E. Spearman. The proof and measurement of association between two things. *The American Journal of Psychology*, 15(1):72–101, 1904.

[65] O. Spencer-Thomas. Writing a Press Release. `http://www.owenspencer-thomas.com/journalism/media-tips/writing-a-press-release`, 2012. URL retrieved on the 2nd March 2013.

[66] B. Spetic and A. Tori. Zemanta. `http://www.zemanta.com`, 2012. Consultado: 10/05/2013.

[67] P.-N. Tan, M. Steinbach, and V. Kumar. Introduction to data mining. *Journal of School Psychology*, 19(1):500, 2005.

[68] Thomsom Reuters. OpenCalais. `http://www.opencalais.com/`, 2008. Consultado: 10/05/2013.

[69] O. Vechtomova and Y. Wang. A study of the effect of term proximity on query expansion. *Journal of Information Science*, 32(4):324–333, 2006.

[70] W3C. RDF Working Group. Resource Description Framework (RDF). `http://www.w3.org/RDF/`, 1999. Consultado: 21/04/2013.

[71] P. Wang. Why recommendation is special? In *15th national conference on artificial intelligence*, pages 111–113, 1998.

[72] J. Weaver and P. Tarjan. Facebook linked data via the graph api. *Semantic Web Journal*, 2012.

[73] K. Yoshii, M. Goto, K. Komatani, T. Ogata, and H. G. Okuno. An efficient hybrid music recommender system using an incrementally trainable probabilistic generative model. *IEEE Transactions on Audio, Speech and Language Processing*, 16(2):435–447, 2008.

[74] L. Zheng, L. Li, W. Hong, and T. Li. Penetrate: Personalized news recommendation using ensemble hierarchical clustering. *Expert Systems with Applications*, 40(6):2127–2136, May 2013.

[75] Y. Zhou, D. Wilkinson, R. Schreiber, and R. Pan. Large-scale parallel collaborative filtering for the netflix prize. In *Proc. 4th International Conference in Algorithmic Aspects in Information and Management, LNCS 5034*, pages 337–348. Springer, 2008.

Apéndice A

Towards an Adaptive and Hybrid News Recommendation System for Journalists: The Wesomender Approach

Towards an Adaptive and Hybrid News Recommendation System for Journalists: The Wesomender Approach

Alejandro Montes-García[1] * and Jose María Álvarez-Rodríguez[2] and Jose Emilio Labra-Gayo[1] and Marcos Martínez-Merino[3]

[1] WESO Research group. Department of Computer Science. University of Oviedo, 33005, Oviedo, Spain
[2] South East European Research Center 54622, Thessaloniki, Greece
[3] Periodismo Humano 33201, Gijón, Spain

Abstract. In this paper a journalist-based recommendation system is presented to automatically configure and exploit news according to expert preferences. News contextual features are also taken into account due to the their special nature: time, current user interests, location or existing trends are combined with traditional recommendation techniques to provide an adaptive framework that deals with data heterogeneities providing an enhanced collaborative filtering system. Since the Wesomender approach is able to suggest context-aware recommendations in the journalism field, a quantitative evaluation with the aim of comparing Wesomender results with the expectations of a team of experts is also performed to show that a context-aware adaptive recommendation engine can fulfil the needs of journalists daily work when retrieving timely and primary information is required.

1 Introduction

A recommender system is defined as a system that selects a set of objects from a superset, according to user's preferences. The way to select this subset of objects cannot be expressed as a common database query [28]. Over the last decade, several recommender systems have been developed so as to be used in different business areas [1], and have proven to be effective at making a personalized filtering of information [23].

Recommender systems were classified by Balabanovic [3], into:

- Content-Based Recommenders: The recommendations are based on a set of properties of the items the user preferred.
- Collaborative Recommenders: The recommendations are based on what items people with similar tastes liked.
- Hybrid Recommenders: The recommendations are based on a combination of the previous approaches.

* Corresponding author. E-mail: alejandro.montes@weso.es.

When dealing with news, there are at least two issues that have to be taken into account, and both are related to time features. First of all, users tend to be more interested in the latest news rather than something that happened some time ago so, the less recent a piece of news is, the less interest a user shows and vice versa. Also, the latest news a user has visited and/or rated are very important to produce recommendations, because the user is explicitly showing interest on a topic or a set of topics.

Other factors that affect news recommendation are the proximity of the place where the action, described in a piece of news, took place. The closer a user is to a place, the events that occur there affect him more. The last factor, is one of the most novel contributions of this paper. Wesomender introduces a new variable called trustworthiness that refers to the credibility given to the mass media that publishes a piece of news.

The documentary research on the Internet started out as a complement of the traditional sources of information. The metamorphosis of the production processes in the media during the last years has ended with this equilibrium, making the Internet the main source of journalistic information.

The Internet has transformed the processes of finding information and documentary research as far as journalistic content development is concerned, but it needs to move a step forward and the need of filtering all this information has showed up.

Websites like GoogleNews and GoogleAlerts provide immediacy, which is quite important for a journalist, but they do not offer the required rigour in the information transmitted. Other systems commonly used by journalists are the search engines of the media itself, but in most cases, these systems produce an information overload and the only possibility of ordering this content by recency.

To deliver proper news recommendations in a way that fulfils the requirements established by professional journalists, an adaptive hybrid recommender system is proposed in this paper. The term *adaptive* hereby refers to the fact that the system supports more than one collaborative recommender algorithm. Those algorithms can be plugged easily to adapt the recommendation engine to the current data model without having to re-compile the whole system.

The remainder of this paper is structured as follows. In section 2 the related work is presented. In section 3 the Wesomender approach is introduced, moreover, its implementation is explained. In section 4 an experiment to measure the benefits of using this recommender is proposed, and finally, in section 5 the conclusions of this paper are exposed.

2 Related Work

Content-based recommenders are not commonly used solely nowadays but as a part of a hybrid recommender. Those hybrid systems have been developed for a wide range of items, involving TV programs [4], films [20], music [29], etc.

Pure collaborative-filtering recommenders have their root in statistical methods like the Pearson correlation coefficient [24] or the Spearman coefficient [27].

Currently, more complex approaches are being used, like the singular value decomposition (or SVD) recommenders such as [31], or [13], that use matrix factorization techniques to deliver recommendations, those recommenders allow "*the incorporation of additional information such as implicit feedback, temporal effects, and confidence levels*" [16]. Other approaches are the Slope-One recommenders like [21], those are easy to implement and quite efficient but not as accurate as the most complex approaches.

2.1 News Recommendation

Content-based news recommenders typically collect the user's reading preferences by using topic distributions such as [12], [18] and [7] or vector space models like [15] to extract a bag of words of their reading history and match those words with newly-published news articles. Some examples of those systems are News Dude [6] and YourNews [2]. Another content-based approach is Newsjunkie [11], which also filters news by novelty.

In this approach, the *"Topics"* requisite is somehow similar to [8], as their system categorises films while the Wesomender categorises news to produce recommendations. The recommender exposed in [9], also uses topics to deliver recommendations in an electronic marketplace environment. The *"Proximity"* variable has the same objective that the location-based service shown in [17].

The collaborative-filtering component of the system presented in this paper is related to [26], but instead of using genetic algorithms as a learning mechanism, the Wesomender tries to learn from the actual and current data.

Hybrid approaches for recommending news have also been developed. Some representative examples are SCENE [22] and PENETRATE [30]. The first one presents a scalable two-stage personalised news recommendation system. It has a two-level recommendation hierarchy, where the first level contains a brief summary for each topic category the user might prefer, and the second level includes specific news articles. On the other hand, PENETRATE, builds clusters of users based on their reading histories, where each user can belong to more than one cluster. Then the system produces recommendations according to the cluster the user belongs to. Another approach is MONERS [19], where the user preferences are estimated by aggregating news article importance and recency.

However, the aforementioned approaches are user-based recommenders rather than journalist-based recommenders, Wesomender tries to deal with recommendation of news in the specific scenario of a journalist daily work.

3 The Wesomender approach

A hybrid recommender has been designed to tackle common problems in existing recommender engines. Content-Based recommenders do not produce recommendations as accurate as collaborative recommenders [10], however, those suffer from the *new item problem*, that causes new items, which may have not been rated by any user yet, will not be considered by the system [1]. As far as news

is concerned, the lack of recommending systems that take into account latest news can be harmful. Collaborative recommenders present also another problem, known as *cold-start* [25]. This happens when many users have not rated enough data, so the system cannot produce recommendations for them.

Therefore, Wesomender is comprised of two main components, a collaborative-filtering component and a content-based component. Each component evaluates the news the user has not seen or rated yet, and produces independent recommendations, the output of those evaluations will be two numbers called CB (for the content-based component) and CF(for the collaborative-filtering component), both between 1 and 0, for each piece of news. The higher those numbers are, the more likely that piece of news will interest the user. Then, both recommendations will be mixed to produce a unique value by using the formula (1).

$$r = w \cdot CB + (1 - w) \cdot CF \tag{1}$$

Where w is the weighting of the content-based component in the recommender and $1 \geq w \geq 0$. The next subsections will explain how the value of CB and CF is calculated for each piece of news.

3.1 Content-Based Component

There is a non-written rule in the world of journalism called *The Five W's* [1] that shows how important it is for a piece of news to answer the questions *who?*, *what?*, *why?*, *when?* and *where?*. Taking into account the aforementioned questions in the news domain, Wesomender algorithms are based on the following hypothesis:

1. A piece of news is more interesting when it has just happened than some time before.
2. A piece of news that happens near a user, is more relevant to him.
3. If a user reads news involving some specific entities or covering some specific topics, he is interested in those entities or topics right now, even if the pieces of news he is reading are old, he might be searching references.

Moreover, a new parameter, called reliability, is also added to enable a right reputation value for each data source. For example, when writing a paper, a reference to another author is more trustworthy than a reference to a blog.

Therefore, a formula considering time, location, content and reliability is outlined in (2) and its output is the value that CB will take in formula (1).

$$CB = \alpha \cdot ER + \beta \cdot T + \gamma \cdot P + \delta \cdot R \tag{2}$$

Where all the operands are numbers between 1 and 0 and:

[1] http://www.owenspencer-thomas.com/journalism/media-tips/writing-a-press-release# GettingFactsStraight URL retrieved on the 2nd March 2013

- ER refers to the entities related to the piece of news. Latest news the user has recently shown interest for will be checked. Then a topic percentage is calculated as the number of occurrences of a topic in those pieces of news. For each entity on the ongoing recommendation, the corresponding percentage is added to the ER variable.
- R stands for recency and will be calculated as shown in formula (3). It will always produce a number between 1 and 0, since $currenttime \geq timeofpublication$.
- P stands for proximity and will be calculated as shown in formula (4). It will always produce a number between 1 and 0, since $maxdistance \geq distance$, being $maxdistance$ the maximum distance between two points on earth, and $distance$ the distance between the user and the piece of news.
- T stands for trustworthiness and will be provided by a team of journalists for each media.

The relevance of each variable is a percentage and it was given by a team of professional journalists who determined how important each factor was. After a survey it was stated that:

- Recency is the most important factor, being $\delta = 0.5$.
- Proximity is the second one in importance, with $\gamma = 0.2$.
- Trustworthiness and the related entities are equally relevant, being $\alpha = \beta = 0.15$.

$$R = 1 - \frac{currenttime - timeofpublication}{currenttime} \tag{3}$$

$$P = \frac{maxdistance - distance}{maxdistance} \tag{4}$$

3.2 Collaborative-Filtering Component

In order to design an adaptive and hybrid recommender, a component that evaluates the existing dataset and chooses the best collaborative-filtering algorithm has been implemented. This component can be used alone or reused in other recommenders. Thus the next initial requirements shall be fulfilled:

1. Since the evaluation of different algorithms is expected to be slow, it must be done off-line, while a recommender engine is working on-line. Once the evaluation is done, the new algorithm can be automatically plugged into the application in runtime.
2. Even though the algorithms to be evaluated can be limited to a small collection of them, new algorithms should be easily added to the system.

As soon as the best algorithm is selected, the system will be able to calculate the CF value in formula (1).

To evaluate the accuracy of the algorithms, the CF component takes a significant part of the dataset and tries to predict the rest, then the differences between the prediction and the real data is measured by using the classical *root*

mean squared. This evaluator has been chosen because it is widely applicable, and is always defined and finite except in the case where all the predicted values and the actual dataset are equal [14].

In this proposal, recommenders like [5], [16],[31] and [21], were evaluated as a sample, but the evaluators are loaded in runtime, so new recommenders and their evaluators can be added dynamically without having to re-compile the whole system.

4 Research Study

Since this recommender is designed to be used by professionals journalism domain rather than for normal users, a team of 2 professionals have evaluated the system under the following conditions.

A small dataset with 30 pieces of news[2] has been mined. It contains news from different places of the world, and from different periods of time.

For each piece of news, the headline and the textual description have been extracted, as well as the topics, localization and UNIX timestamp of every piece of news.

Once the dataset has been built, the team of professionals was asked to separate between relevant and irrelevant news.

In this evaluation, the classical precision and recall variables have been measured. Precision has been measured as how many of the returned elements were relevant, while recall has been measured as how many relevant elements were returned from the total of relevant news.

In this quantitative experiment, the team of professionals has been asked to use the recommender via the website shown in Figure 1, so that precision and recall could be evaluated after rating zero, five and ten random pieces of news, for the first, first two, and first five recommended items. After rating one piece of news, this was not recommended anymore. The results are shown in Table 1.

Fig. 1. Website to interact with the Wesomender

[2] The dataset can be downloaded at: `http://alejandro-montes.appspot.com/research/wesomender/dataset.zip`

The *"Rated"* column refers to the number of elements rated by the professionals before running the evaluation, while the *"i"* column refers to the elements that are being considered to measure precision and recall. Results showed almost a

Table 1. Results of the evaluation

Rated	i	Precision	Recall
	1st	100%	1/20
0	1 - 2	100%	2/20
	1 - 5	80%	4/20
	1st	100%	1/16
5	1 - 2	100%	2/16
	1 - 5	70%	2.5/16
	1st	100%	1/13.5
10	1 - and 2	100%	2/13.5
	1 - to 5	100%	5/13.5

perfect performance for a new user. However, after some interaction between the users and the system, there is not new relevant gain in the results. This is probably because, when a piece of news is rated it is not recommended any longer. This is likely to happen in the mined dataset because 2/3 of the pieces of news are relevant, so there is a higher probability of deleting relevant pieces of news after the first ratings rather than an irrelevant one.

Finally, after ten ratings, the Wesomender got a perfect score in the experiment proposed, probably with a bigger team of journalists that would not have happened because there would have been a higher chance of failing at least one recommendation.

5 Conclusions

An adaptive and hybrid news recommender system has been presented. This system has two components, one produces recommendations based on the *Five W's of journalism* and the trustworthiness of the information source. The other one produces a collaborative-filtering based recommendation, and it can be automatically configured. The first results suggest that the system does not suffer from the cold start problem.

Future work that can be carried out in this system lies in the performance improvement of both components. In the collaborative-filtering component, the evaluation runtime can be improved by using heuristics that can predict the result of a recommender looking for correlations between algorithms. The workload of the other component can be reduced by using heuristics like eliminating very old news or those where the action occurred too far away to affect the user.

As another part of the future work, the experiment should also be improved by adding more pieces of news extracted from RSS feeds of online newspapers as

well as conducting the same experiment again with a higher number of expert in order to get more diverse results. To do so a bigger ratings dataset is needed, a website for the interchange of news called Freews is being developed so as to get more critical mass and mine a news dataset.

6 Acknowledgements

This work has been supported by the project FUO-EM-338-10 ORIGIN of the University of Oviedo Foundation and performed in 2012.

References

1. G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering*, 17(6):734–749, 2005.
2. J.-W. Ahn, P. Brusilovsky, J. Grady, D. He, and S. Y. Syn. Open user profiles for adaptive news systems: help or harm? In *Proceedings of the 16th international conference on World Wide Web*, WWW '07, pages 11–20, New York, NY, USA, 2007. ACM.
3. M. Balabanovic and Y. Shoham. Content-based, collaborative recommendation. *Communications of the ACM*, 40:66–72, 1997.
4. A. B. Barragáns-Martínez, E. Costa-Montenegro, J. C. Burguillo, M. Rey-L'opez, F. A. Mikic-Fonte, and A. M. Peleteiro. A hybrid content-based and item-based collaborative filtering approach to recommend tv programs enhanced with singular value decomposition. *Information Sciences*, 180(22):4290–4311, 2010.
5. R. M. Bell and Y. Koren. Scalable collaborative filtering with jointly derived neighborhood interpolation weights. pages 43–52, 2007.
6. D. Billsus and M. J. Pazzani. A personal news agent that talks, learns and explains. In *Proceedings of the third annual conference on Autonomous Agents*, AGENTS '99, pages 268–275, New York, NY, USA, 1999. ACM.
7. D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, Mar. 2003.
8. S.-M. Choi, S.-K. Ko, and Y.-S. Han. A movie recommendation algorithm based on genre correlations. *Expert Systems with Applications*, 39(9):8079 – 8085, 2012.
9. K. Christidis and G. Mentzas. A topic-based recommender system for electronic marketplace platforms. *Expert Systems with Applications*, 40(11):4370 – 4379, 2013.
10. P. Cremonesi, R. Turrin, and F. Airoldi. Hybrid algorithms for recommending new items. In *Proceedings of the 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems*, HetRec '11, pages 33–40, New York, NY, USA, 2011. ACM.
11. E. Gabrilovich, S. Dumais, and E. Horvitz. Newsjunkie: Providing personalized newsfeeds via analysis of information novelty. In *In WWW2004*, pages 482–490. ACM Press, 2004.
12. T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '99, pages 50–57, New York, NY, USA, 1999. ACM.
13. Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. pages 263–272, 2008.

14. R. J. Hyndman and A. B. Koehler. Another look at measures of forecast accuracy. *International Journal of Forecasting*, 22(4):679 – 688, 2006.
15. D. Jurafsky and J. H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1st edition, 2000.
16. Y. Koren, R. M. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30 –37, aug. 2009.
17. M.-H. Kuo, L.-C. Chen, and C.-W. Liang. Building and evaluating a location-based service recommendation system with a preference adjustment mechanism. *Expert Systems with Applications*, 36(2, Part 2):3543 – 3554, 2009.
18. K. Lang. Newsweeder: Learning to filter netnews. In *in Proceedings of the 12th International Machine Learning Conference (ML95*, 1995.
19. H. Lee and S. J. Park. Moners: A news recommender for the mobile web. *Expert Systems with Applications*, 32(1):143 – 150, 2007.
20. G. D. Lekakos and P. Caravelas. A hybrid approach for movie recommendation. *Multimedia Tools and Applications*, 36(1-2):55–70, 2008.
21. D. Lemire and A. Maclachlan. Slope one predictors for online rating-based collaborative filtering. In *Proceedings of SIAM Data Mining (SDM'05)*, 2005.
22. L. Li, D. Wang, T. Li, D. Knox, and B. Padmanabhan. Scene: a scalable two-stage personalized news recommendation system. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, SIGIR '11, pages 125–134, New York, NY, USA, 2011. ACM.
23. J. O'Donovan and B. Smyth. Trust in recommender systems. pages 167–174, 2005.
24. K. Pearson. Notes on the history of correlation. *Biometrika*, 13(1):pp. 25–45, 1920.
25. A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '02, pages 253–260, New York, NY, USA, 2002. ACM.
26. C. Shahabi and Y.-S. Chen. An adaptive recommendation system without explicit acquisition of user relevance feedback. *Distributed and Parallel Databases*, 14(2):173–192, 2003.
27. C. E. Spearman. The proof and measurement of association between two things. *The American Journal of Psychology*, 15(1):72–101, 1904.
28. P. Wang. Why recommendation is special? In *15th national conference on artificial intelligence*, pages 111–113, 1998.
29. K. Yoshii, M. Goto, K. Komatani, T. Ogata, and H. G. Okuno. An efficient hybrid music recommender system using an incrementally trainable probabilistic generative model. *IEEE Transactions on Audio, Speech and Language Processing*, 16(2):435–447, 2008.
30. L. Zheng, L. Li, W. Hong, and T. Li. Penetrate: Personalized news recommendation using ensemble hierarchical clustering. *Expert Systems with Applications*, 40(6):2127–2136, May 2013.
31. Y. Zhou, D. Wilkinson, R. Schreiber, and R. Pan. Large-scale parallel collaborative filtering for the netflix prize. In *Proc. 4th Int'l Conf. Algorithmic Aspects in Information and Management, LNCS 5034*, pages 337–348. Springer, 2008.

Apéndice B

Towards a Journalist-Based News Recommendation System: The Wesomender Approach

Towards a Journalist-Based News Recommendation System: The Wesomender Approach

Alejandro Montes-García[a,*], Jose María Álvarez-Rodríguez[b], Jose Emilio Labra-Gayo[a], Marcos Martínez-Merino[c]

[a] *WESO Research group.*
Department of Computer Science. University of Oviedo, 33005, Oviedo, Spain
Email: {alejandro.montes, jelabra}@weso.es
[b] *South East European Research Center,*
54622, Thessaloniki, Greece
Email: jmalvarez@seerc.org
[c] *Periodismo Humano,*
33201, Gijón, Spain

Abstract

The present paper introduces a context-aware recommendation system for journalists to enable the identification of similar topics across different sources. More specifically a journalist-based recommendation system that can be automatically configured is presented to exploit news according to expert preferences. News contextual features are also taken into account due to the their special nature: time, current user interests, location or existing trends are combined with traditional recommendation techniques to provide an adaptive framework that deals with heterogeneous data providing an enhanced collaborative filtering system. Since the *Wesomender* approach is able to generate context-aware recommendations in the journalism field, a quantitative evaluation with the aim of comparing *Wesomender* results with the expectations of a team of experts is also performed to show that a context-aware adaptive recommendation engine can fulfil the needs of journalists daily work when retrieving timely and primary information is required.

Keywords: Hybrid Recommender systems, News Recommendation, Collaborative expert filtering, Adaptive systems

*Corresponding author

1. Introduction

A recommender system is defined as a system that selects a set of objects from a superset according to user's preferences. The way to select this subset of objects cannot be expressed as a common database query [44]. Over the last decade several recommender systems have been developed which can be used in different domains [1] and have proven to be effective at making a personalized filtering of information [33].

Recommender systems were classified by Balabanovic [3], into:

- Content-Based Recommenders: Recommendations are based on a set of properties of the items that the user prefers.
- Collaborative Recommenders: Recommendations are based on what items people with similar tastes like.
- Hybrid Recommenders: These systems present a combination of the previous approaches.

When dealing with news recommendation there are some specific factors that have to be taken into account.

- Novelty. Users generally tend to be more interested in the latest news rather than something that happened some time ago so, the less recent a piece of news is, the less interest a user shows and vice versa.
- User history. The latest news a user has visited and/or rated are very important to produce recommendations, because the user is explicitly showing interest on a topic or a set of topics.
- Proximity. Users are more interested in news that happen in nearby places. In general, the closer a user is to a place, the events that occur there affect him more so a news recommender system has to take into account the location of the place where the action described in the piece of news took place.
- Trustworthiness. The credibility of the media that publishes a piece of news. This last factor is one of the most novel contributions of this paper.

The documentary research on the Internet started out as a complement of the traditional sources of information (contacts, press archives, consultations with experts, etc). The metamorphosis of the production processes in the media during the last years has ended with this equilibrium, making the Internet the main, and sometimes the only, source of information as far as journalistic content search is concerned.

The Internet has transformed the processes of finding information and documentary research as far as journalistic content development is concerned, but it needs to move a step forward and the need of filtering all this information has shown up.

Current search tools are based on traditional techniques that are unable to offer a wide variety of news or allow precise bounds to the professional that is following the history of a piece of news. In this sense, information quality and filtering are becoming big challenges due to the vast amount of data that is continuously generated in Internet [8].

Websites like *GoogleNews* and *GoogleAlerts* provide immediacy, which is quite important for a journalist, but they do not offer the required rigour in the information transmitted. Other systems commonly used by journalists are the search engines of the media itself, but in most cases, these systems produce information overload and the only possibility of ordering this content by recency.

To deliver proper news recommendations in a way that fulfils the requirements established by professional journalists, an adaptive hybrid recommender system called *Wesomender* is proposed. The term *adaptive* hereby refers to the fact that the system supports more than one collaborative recommender algorithm. Those algorithms can be plugged easily to adapt the recommendation engine to the current data model without having to recompile the whole system.

The requisites of the system have been summarised in Table 1.

The remainder of this paper is structured as follows. In section 2 the related work is presented. In section 3 the *Wesomender* approach is introduced, moreover, its implementation is explained. In section 4 a use case for the *Wesomender* is presented. In section 5 an experiment to measure the benefits of using this recommender is proposed, and finally, in section 6 the conclusions of this paper are exposed.

Table 1: Requisites for the Wesomender

Short name	Description
Proximity	News that take place near the user are more likely to interest him, and this variable must be considered.
Time	It is more likely that recent news will arouse more interest in the user and therefore this must be taken into account.
User history	The last topics a user checks, are the ones that caught his attention in the moment of the delivery of the recommendations.
Trustworthiness	The credibility of a source of information is an important factor and therefore it must be considered when producing recommendations.
Hot-plug	The recommender must be able to plug in new collaborative-filtering algorithms while the system is running, these algorithms can be plugged to both being evaluated or produce recommendations.
Off-line learning	The recommender must evaluate the results produced by a set of collaborative-filtering algorithms and the current dataset. This must be done while the system is running.

2. Related Work

Content-based recommenders are not commonly used solely nowadays but as a part of a hybrid recommender. Those hybrid systems have been developed for a wide range of items, involving TV programs [4], films [28], music [45], stores [15], travels [40], multimedia content [25], etc.

Pure collaborative-filtering recommenders have their root in statistical methods that are exposed in section 3.2. Some examples of those recommenders are the Scalable Collaborative Filtering with Jointly Derived Neighborhood Interpolation Weights, which improved the prediction accuracy by improving the interpolation precision and deriving simultaneously the interpolation weights for all nearest neighbours[5]. The singular value decomposition (or SVD) recommenders such as [47], or [19] use matrix factorization techniques to deliver recommendations, those recommenders allow the incorporation of additional information such as implicit feedback, temporal effects, and confidence levels [22]. Other approaches are the SlopeOne recommenders like [29], those are easy to implement and quite efficient but not as accurate as the more complex approaches.

In this approach, the *User history* requisite is somehow similar to [9], as their system categorises films while the *Wesomender* categorises news to produce recommendations. The recommender exposed in [10], also uses topics to deliver recommendations in an electronic marketplace environment. The *"Proximity"* variable has the same objective that the location-based service shown in [24].

The collaborative-filtering component of the system presented in this paper is related to [39], but instead of using genetic algorithms as a learning mechanism, the *Wesomender* tries to learn from the actual and current data, this is more inefficient, but it has been used because this is the tool that Mahout provides for evaluating an algorithm. Another difference is that in [39], the work is based on the algorithm presented by [38], while in this work, potentially any algorithm can be plugged-in to enhance the system.

2.1. News Recommendation

Content-based news recommenders typically collect the user's reading preferences by using topic distributions such as [18], [26] and [7] or vector space models like [21] to extract a bag of words of their reading history and match those words with newly-published news articles. Some examples of

those systems are News Dude [6] and YourNews [2]. Another content-based approach is Newsjunkie [16], which also filters news by novelty.

Hybrid approaches for recommending news have also been developed. Some representative examples are SCENE [30] and PENETRATE [46]. The first one presents a scalable two-stage personalised news recommendation system. It has a two-level recommendation hierarchy, where the first level contains a brief summary for each topic category the user might prefer, and the second level includes specific news articles. On the other hand, PENETRATE, builds clusters of users based on their reading histories, where each user can belong to more than one cluster. Then the system produces recommendations according to the cluster the user belongs to. Another approach is MONERS [27], where the user preferences are estimated by aggregating news article importance and recency.

However, the aforementioned approaches are user-based recommenders rather than journalist-based recommenders, *Wesomender* tries to deal with recommendation of news in the specific scenario of a journalist daily work.

3. The Wesomender Approach

A hybrid recommender based on both, user content and collaborative filtering, has been designed to tackle common problems in existing recommender engines. The first ones, do not produce recommendations as accurate as the later ones [11]. However, collaborative recommenders suffer from the *new item problem* described by Adomavicius in [1] where new items, which may have not been rated by any user yet, are not considered by the system. In the case of journalist news, recommending systems must take into account latest news Collaborative recommenders present also another problem, known as *cold-start* [36]. This happens when many users have not rated enough data, so the system cannot produce recommendations for them.

Therefore, *Wesomender* is comprised of two main components, a collaborative-filtering component and a content-based component. Each component evaluates the news the user has not seen or rated yet, and produces independent recommendations, the output of those evaluations will be two numbers called CB (for the content-based component) and CF(for the collaborative-filtering component), both between 0 and 1, for each piece of news. The higher those numbers are, the more likely that piece of news will interest the user. Then, both recommendations will be mixed to produce a unique value by using the formula (1).

$$r = w \cdot CB + (1 - w) \cdot CF \tag{1}$$

Where w is the weighting of the content-based component in the recommender and $0 \leq w \leq 1$. The next subsections will explain how the value of CB and CF is calculated for each piece of news.

3.1. Content-Based Component

There is a non-written rule in the world of journalism called *The Five W's* [42] that shows how important it is for a piece of news to answer the questions *who?*, *what?*, *why?*, *when?* and *where?*. Taking into account the aforementioned questions in the news domain, Wesomender algorithms are based on the following hypothesis:

1. A piece of news is more interesting when it has just happened than some time before.
2. A piece of news that happens near a user, is more relevant to him.
3. If a user reads news involving some specific entities or covering some specific topics, he is interested in those entities or topics right now, even if the pieces of news he is reading are old, he might be searching references.

Moreover, a new parameter, called trustworthiness, is also added to enable a right reputation value for each data source. For example, when writing a paper, a reference to another author is usually more trustworthy than a reference to a blog.

Therefore, a formula considering time, location, content and reliability is outlined in (2) and its output is the value that CB will take in formula (1).

$$CB = \alpha \cdot ER + \beta \cdot T + \gamma \cdot P + \delta \cdot R \tag{2}$$

Where all the operands are numbers between 0 and 1 and:

- ER refers to the entities related to the piece of news. The latest news the user has recently shown interest for will be checked. Then a topic percentage is calculated as the number of occurrences of a topic in those pieces of news. For each entity on the ongoing recommendation, the corresponding percentage is added to the ER variable. Finally the extraction of entities is carried out using an external library that is able

to process and extract entities from text-based contents, in particular Zemanta[1] has been used.

- R stands for recency and will be calculated as shown in formula (3). It will always produce a number between 0 and 1, since $currentTime \geq timeOfPublication$.

- P stands for proximity and will be calculated as shown in formula (5). It will always produce a number between 0 and 1, since $maxDistance \geq distance$, being $maxdistance$ the maximum distance between two points on earth given by half of the length of the Earth's circumference on the equator, and $distance$ the distance between the user and the piece of news. The $distance$ is given by the latitude (ϕ_1 and ϕ_2) and longitude (λ_1 and λ_2) of the two points and calculated using the Haversine formula (4).

- T stands for trustworthiness and will be provided by a team of journalists for each media.

For all these operands, 0 means poor relevance for the user whereas 1 means maximum significance.

The importance of each variable is a percentage and it was given by a team of professional journalists who determined how important each factor was. After a survey it was stated that:

- Recency is the most important factor, being $\delta = 0.5$.

- Proximity is the second one in importance, with $\gamma = 0.2$.

- Trustworthiness and the related entities are equally relevant, being $\alpha = \beta = 0.15$.

$$R = 1 - \frac{currentTime - timeOfPublication}{currentTime} \tag{3}$$

$$distance = 2 \cdot earthradius \cdot \arcsin(\sqrt{\sin^2(\frac{\Delta\phi}{2}) + \cos(\phi_1) \cdot \cos(\phi_2) \cdot \sin^2(\frac{\Delta\lambda}{2})}) \tag{4}$$

$$P = \frac{maxDistance - distance}{maxDistance} \tag{5}$$

[1] http://www.zemanta.com/

3.2. Collaborative-Filtering Component

In order to design an adaptive and hybrid recommender, a component that evaluates the existing dataset and chooses the best collaborative-filtering algorithm has been implemented. This component can be used alone or reused in other recommenders. Thus the next initial requirements shall be fulfilled:

1. Since the evaluation of different algorithms is expected to be slow, it must be done off-line, while a recommender engine is working on-line. Once the evaluation is done, the new algorithm can be automatically plugged into the application at runtime.
2. Even though the algorithms to be evaluated can be limited to a small collection of them, new algorithms should be easily added to the system. This will allow developers and researchers to add new algorithms to be taken into account for the evaluation, enhancing the system.
3. In order to make use of an initial set of existing algorithms without having to re-implement them, a library that already implements those algorithms can be used. If so, the system can be implemented as a layer over the existing library.

As soon as the best algorithm is selected, the system will be able to calculate the CF value in formula (1).

In our approach, this component is implemented on the top of Apache Mahout [37]. It has been selected because it already implements a considerable set of techniques that can be easily evaluated through an API to test the accuracy of algorithms. Nevertheless other existing data mining frameworks such as Weka [17] are being also studied to implement the *Wesomender* approach and comparing the different capabilities and results.

The evaluators take a defined percentage of the dataset to be evaluated, called training data, and use this training data to learn from it and predict what the other part of the dataset says. One of the evaluators implemented by Apache Mahout is the classical *root mean squared*, this evaluator has been used because it is widely applicable, and is always defined and finite except in the case where all the predicted values and the actual dataset are equal [20]. The result of this evaluation can be serialized with the recommender that produced the result.

The following recommenders implemented by Mahout were evaluated as a sample, but the evaluators are loaded in runtime using reflection, so new

recommenders and their evaluators can be added dynamically without having to re-compile the whole system.

3.2.1. Generic User Based Recommender

This recommender produces recommendations to a user, taking into account what other users with similar tastes like. To do so, it was parametrised with all the possible combinations of a User Similarity and a User Neighbourhood.

The user similarity defines a notion of similarity between two users, it determines how similar their tastes are. There are several implementations in Mahout for this concept. For this systems the following ones were used: Pearson correlation coefficient [34], Euclidean distance [13], Cosine similarity [43], Manhattan distance [23], Log-Likelihood [14], Spearman's rank correlation coefficient [41] and Tanimoto Similarity [35].

On the other hand, the neighbourhood concept determines how many users have to be taken into account to produce recommendations, once the distance between users is calculated, the recommendations can be either produced taking into account the nearest N users or the users within a threshold. Those two approaches of the neighbourhood concept were parametrised and combined with the similarities described before to look for the best duo.

3.2.2. Item Based Recommender

The item based recommender is similar to the user based recommender described in the previous section. It also uses the concept of neighbourhood and similarity, but this recommender applies them to items instead of users. So this algorithm does not look for similar users to another one, but for items similar to those the user likes. The implementation of this recommender in Apache Mahout is based in [5].

3.2.3. Tree Clustering Recommender

This recommender builds a specific number of clusters of users, each cluster is composed by the most similar users. Then the behaviour of this recommender is the same as the User Based Recommender, but treating the clusters as a single user, therefore, the recommendations produced will be the same to all users within a specific cluster.

To measure the distances between clusters, there are two possibilities, on the one hand two clusters can be considered similar if some pair of mem-

bers has high similarity, or they can be considered similar if all the pairs of members have high similarity. Both approaches were tested.

3.2.4. SVD Recommender

SVD is a matrix factorization technique that can be used to produce recommendations [22]. Mahout allows using *factorizers* that define how to factorize the rating matrix. Three of this factorizers were used in this system. The first one, uses the Alternating-Least-Squares with Weighted-λ-Regularization and is parametrised as described in [47], the second one uses the Expectation-Maximization algorithm [12] and the last one uses the model described in [19].

3.2.5. Slope One Recommender

The Slope One Recommender [29] recommends items by predicting the rating that a user would give to an item. To do so, it calculates the average difference between the ratings of one item and another for users who rated both, and then predicts the rating that one of those users would give to an item that was rated only by the other one.

4. Use Case: A Video News Management System for Journalists

The use of information technologies on the Internet era has completely changed the view, mission and concepts in which traditional journalism was based. The transition from paper-based systems to a real digital journalism is not a mere question that can be tackled with the deployment of a news website. Like other domains, the journalism web revolution is now happening and existing media companies are facing problems to adapt themselves to the requirements of this dynamic environment: way of working, information and data filtering or crowd needs (such as time, location or trustworthiness) to name a few. In particular the distribution of news around the world was, and is still, controlled by major media corporates and the reuse of contents in television, radio or newspapers is, often, simply focused on the literal translation of the original source after passing a strict and bureaucratic work flow. Thus news about catastrophes, sports, politics, etc. are always similar regardless the channel or the media company.

On the other hand both social aspects and technical advances are providing the proper context to enable users participation and create a real collective intelligence environment in which journalism can take advantage

of. In this sense contributions in social networks are turning the web in the wisdom of crowds platform where new information resources such as pictures or videos are being continuously generated. Existing major multimedia sites such as Flickr, Picasa, Youtube or Dailymotion are now providing the mechanisms to manage this vast amount of data. Nevertheless the fully deployment of domain-based services is still on-going work and in the case of journalists, searching information resources can be tedious and time-consuming due to the lack of customized tools for filtering information and data. For instance the PhotoMerchant[2] site has been deployed to afford a platform where photographers can share, organize and make business with their own pictures.

In this context Freews emerges as an effort to provide a platform for video management in which both users and professionals such as journalists can easily manage their contents through advanced information management techniques. Thus services such as advanced search, faceted-browsing, tagging, video edition or context-aware recommendations are delivered with the aim of assisting users in video cataloguing and management and partially fulfilling the requirements of the new journalism realm.

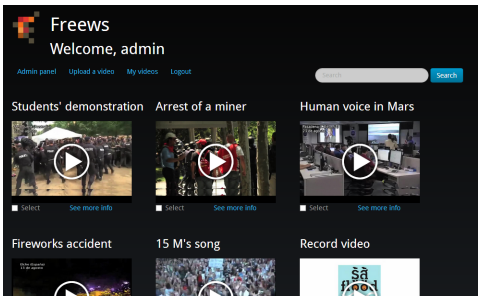

Figure 1: Freews homepage

In particular, Freews is a a website for the interchange of news that offers professional journalists the chance of sharing small clips that contain pieces of news and mix them together in order to make news bulletins and share or broadcast them. The pieces of news uploaded to Freews will be supervised by an expert team so that all the information published in Freews will be reliable even though some users will be more trustworthy than others.

In order to ease the identification of pieces of news that are more likely

[2] http://www.photomerchant.net/

to interest a user, the Wesomender and a search engine that uses NLP techniques are integrated in Freews.

Although Wesomender has been applied to build a video recommendation system, it is also applicable to news aggregation websites such us *The Huffington Post*[3] or *Google News*.

5. Evaluation

5.1. Research design

Since this recommender engine is designed to be used by professionals in the journalism domain rather than regular users, the purpose of this study is to compare a set of news suggested by the recommendation engine with the expected results of a panel of experts. Thus, the objective is to asses if recommendations provided by *Wesomender* can fulfil the expectations and requirements of professionals in this sector.

5.2. Sample

In order to perform this experiment a dataset containing 30 pieces of news has been manually selected by a panel of two journalists[4]. It contains news from different locations and periods of time.

For each piece of news, the headline and the textual description have been extracted, as well as the tags using Zemanta, and the localization and UNIX timestamp of every piece of news. The reliability of the sources of the news was provided by the team of professionals. All the pieces of news were written in English, due to the fact that Zemanta works better with English texts and, thus the experiment can be easily extend to other existing datasets. Zemanta is used here so that human opinions do not affect to the tag extraction process and therefore the experiment is more objective.

Once the dataset has been built, the team of professionals was asked to evaluate the relevance of each piece of news in a percentage. Considering all news with a relevance under 50% as irrelevant, 10 pieces of news were marked as irrelevant and 20 as relevant.

[3] www.huffingtonpost.com/

[4] The dataset can be downloaded at: `http://alejandro-montes.appspot.com/research/wesomender/dataset.zip`

Table 2: Results of the evaluation

Rated	i	Relevance	Precision	Recall
0	1st	95%	100%	1/20
	1st and 2nd	92%	100%	2/20
	1st to 5th	70.6%	80%	4/20
5	1st	88%	100%	1/16
	1st and 2nd	90.3%	100%	2/16
	1st to 5th	78.1%	70%	2.5/16
10	1st	82%	100%	1/13.5
	1st and 2nd	84%	100%	2/13.5
	1st to 5th	87.1%	100%	5/13.5

In this evaluation, three variables have been measured. First of all, the relevance of the news returned, this variable was given by the journalists before starting the evaluation but it is unknown for the Wesomender. Precision has been also measured as how many of the returned elements were relevant, while recall has been measured as how many relevant elements were returned from the total of relevant news.

In this quantitative experiment, the team of professionals has been asked to use the recommender via Freews, so that relevance, precision and recall could be evaluated after rating zero, five and ten random pieces of news, for the first, first two, and first five recommended items. After rating one piece of news, this was not recommended anymore.

5.3. Results and Discussion

The results are shown in Table 2. The *"Rated"* column refers to the number of elements rated by the professionals before running the evaluation, while the *"i"* column refers to the elements that are being considered to measure relevance, precision and recall. So, for example in a row where *"Rated"* takes the value five and *"i"* takes the value *1st and 2nd*, relevance, precision and recall for the first two elements was measured after the journalists had rated five random elements. Figure 2 also shows the relevance values obtained after the evaluation.

Results showed an excellent performance for a new user for the first two elements, but it is significantly worst for the first five elements. However, after some interaction between the users and the system, there is a meaningful improvement in the results for five recommendations, with no new relevant

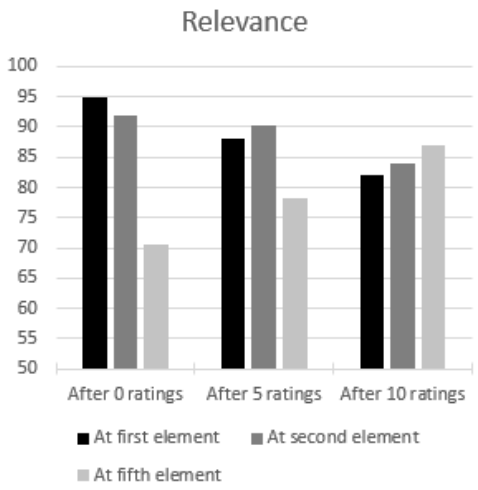

Figure 2: Relevance

gain in the results for one and two recommendations. In fact, these results are slightly worse than before rating any element because when a relevant piece of news is rated, it is not recommended any more. This is likely to happen in the mined dataset because 2/3 of the pieces of news are relevant, there is a higher probability of deleting relevant pieces of news after the first ratings rather than an irrelevant one.

It is also important to notice, that those low values of recall are explained due to the fact that when measuring recall for the first recommendation, the maximum number of relevant elements returned cannot be higher than one.

6. Conclusions

An adaptive and hybrid news recommender system has been presented. This system has two components, one produces recommendations based on the *Five W's of journalism* [42] and the trustworthiness of the information source. The other one produces a collaborative-filtering based recommendation. This component evaluates a collection of recommendation algorithms and plugs-in the best one for the current dataset. This evaluation can be done off-line, while another recommender is plugged, producing recommendations for the users. This component allows the *hot-plug* of new algorithms to be evaluated and used. The first results suggest that the system does not suffer from the cold start problem.

Future work that can be carried out in this system lies in the performance improvement of both components. In the collaborative-filtering component, the evaluation runtime can be improved by using heuristics that can predict the result of a recommender looking for correlations between algorithms. The workload of the other component can be reduced by using heuristics like eliminating very old news or those where the action occurred too far away to affect the user.

Furthermore new variables are being considered to improve and extend the capabilities of *Wesomender*. For instance, sentiment analysis techniques like [31] can be applied to enable sentiment-based recommendation. Moreover, the content based component can be refined with the opinions of more experts and applied to other areas with similar features such us bibliographic information retrieval. Another interesting task that is also planned lies in the implementation of implicit feedback techniques such as [32] so that there is no need of explicitly collecting news ratings.

7. Acknowledgements

This work has been partially funded by the project FUO-EM-134-12 developed in cooperation between the WESO Research Group and *Periodismo Humano* and by the Spanish project MICINN-12-TIN2011-27871 ROCAS (Reasoning on the Cloud by Applying Semantics).

References

[1] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering*, 17(6):734–749, 2005.

[2] J.-W. Ahn, P. Brusilovsky, J. Grady, D. He, and S. Y. Syn. Open user profiles for adaptive news systems: help or harm? In *Proceedings of the 16th international conference on World Wide Web*, WWW '07, pages 11–20, New York, NY, USA, 2007. ACM.

[3] M. Balabanovic and Y. Shoham. Content-based, collaborative recommendation. *Communications of the ACM*, 40:66–72, 1997.

[4] A. B. Barragáns-Martínez, E. Costa-Montenegro, J. C. Burguillo, M. Rey-L'opez, F. A. Mikic-Fonte, and A. M. Peleteiro. A hybrid content-based and item-based collaborative filtering approach to recommend tv programs enhanced with singular value decomposition. *Information Sciences*, 180(22):4290–4311, 2010.

[5] R. M. Bell and Y. Koren. Scalable collaborative filtering with jointly derived neighborhood interpolation weights. pages 43–52, 2007.

[6] D. Billsus and M. J. Pazzani. A personal news agent that talks, learns and explains. In *Proceedings of the third annual conference on Autonomous Agents*, AGENTS '99, pages 268–275, New York, NY, USA, 1999. ACM.

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

[8] Y.-I. Chang, L.-W. Huang, J.-H. Shen, and Y.-S. Wang. An efficient signature-based strategy for supporting inexact filtering in information filtering systems. *Expert Systems with Applications*, 36(4):8431 – 8442, 2009.

[9] S.-M. Choi, S.-K. Ko, and Y.-S. Han. A movie recommendation algorithm based on genre correlations. *Expert Systems with Applications*, 39(9):8079 – 8085, 2012.

[10] K. Christidis and G. Mentzas. A topic-based recommender system for electronic marketplace platforms. *Expert Systems with Applications*, 40(11):4370 – 4379, 2013.

[11] P. Cremonesi, R. Turrin, and F. Airoldi. Hybrid algorithms for recommending new items. In *Proceedings of the 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems*, HetRec '11, pages 33–40, New York, NY, USA, 2011. ACM.

[12] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society Series B Methodological*, 39(1):1–38, 1977.

[13] M. M. Deza and E. Deza. *Encyclopedia of Distances*, volume 2006. Springer Berlin Heidelberg, 2009.

[14] T. Dunning. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19(1):61–74, 1993.

[15] B. Fang, S. Liao, K. Xu, H. Cheng, C. Zhu, and H. Chen. A novel mobile recommender system for indoor shopping. *Expert Systems with Applications*, 39(15):11992 – 12000, 2012.

[16] E. Gabrilovich, S. Dumais, and E. Horvitz. Newsjunkie: Providing personalized newsfeeds via analysis of information novelty. In *In WWW2004*, pages 482–490. ACM Press, 2004.

[17] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: an update. *SIGKDD Explor. Newsl.*, 11(1):10–18, Nov. 2009.

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

[19] Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. pages 263–272, 2008.

[20] R. J. Hyndman and A. B. Koehler. Another look at measures of forecast accuracy. *International Journal of Forecasting*, 22(4):679 – 688, 2006.

[21] D. Jurafsky and J. H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1st edition, 2000.

[22] Y. Koren, R. M. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30 –37, aug. 2009.

[23] E. F. Krause. *Taxicab Geometry: An Adventure in Non-Euclidean Geometry*, volume 72. Courier Dover Publications, 1987.

[24] M.-H. Kuo, L.-C. Chen, and C.-W. Liang. Building and evaluating a location-based service recommendation system with a preference adjustment mechanism. *Expert Systems with Applications*, 36(2, Part 2):3543 – 3554, 2009.

[25] J. E. Labra, P. Ordonez, and J. M. Cueva. WESONET: Applying semantic web technologies and collaborative tagging to multimedia web information system. *Computers in Human Behavior*, 26(2), 2011.

[26] K. Lang. Newsweeder: Learning to filter netnews. In *in Proceedings of the 12th International Machine Learning Conference (ML95)*, 1995.

[27] H. Lee and S. J. Park. Moners: A news recommender for the mobile web. *Expert Systems with Applications*, 32(1):143 – 150, 2007.

[28] G. D. Lekakos and P. Caravelas. A hybrid approach for movie recommendation. *Multimedia Tools and Applications*, 36(1-2):55–70, 2008.

[29] D. Lemire and A. Maclachlan. Slope one predictors for online rating-based collaborative filtering. In *Proceedings of SIAM Data Mining (SDM'05)*, 2005.

[30] L. Li, D. Wang, T. Li, D. Knox, and B. Padmanabhan. Scene: a scalable two-stage personalized news recommendation system. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, SIGIR '11, pages 125–134, New York, NY, USA, 2011. ACM.

[31] A. Moreo, M. Romero, J. L. Castro, and J. M. Zurita. Lexicon-based comments-oriented news sentiment analyzer system. *Expert Systems with Applications*, 39(10):9166 – 9180, 2012.

[32] E. R. Núñez-Valdéz, J. M. C. Lovelle, O. S. Martínez, V. García-Díaz, P. O. nez de Pablos, and C. E. M. Marín. Implicit feedback techniques on recommender systems applied to electronic books. *Computers in Human Behavior*, 28(4):1186–1193, July 2012.

[33] J. O'Donovan and B. Smyth. Trust in recommender systems. pages 167–174, 2005.

[34] K. Pearson. Notes on the history of correlation. *Biometrika*, 13(1):pp. 25–45, 1920.

[35] D. J. Rogers and T. T. Tanimoto. A computer program for classifying plants. *Science*, 132(3434):1115–1118, 1960.

[36] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '02, pages 253–260, New York, NY, USA, 2002. ACM.

[37] C. E. Seminario and D. C. Wilson. Case study evaluation of mahout as a recommender platform. In *in Proceedings of the 6th ACM Conference on Recommender Engines (RecSys 2012)*, 2012.

[38] C. Shahabi, F. Banaei-Kashani, Y.-S. Chen, and D. McLeod. Yoda: An Accurate and Scalable Web-Based Recommendation System. In C. Batini, F. Giunchiglia, P. Giorgini, and M. Mecella, editors, *Cooperative Information Systems*, volume 2172 of *Lecture Notes in Computer Science*, chapter 31, pages 418–432. Springer Berlin / Heidelberg, Berlin, Heidelberg, Sept. 2001.

[39] C. Shahabi and Y.-S. Chen. An adaptive recommendation system without explicit acquisition of user relevance feedback. *Distributed and Parallel Databases*, 14(2):173–192, 2003.

[40] D.-H. Shih, D. C. Yen, H.-C. Lin, and M.-H. Shih. An implementation and evaluation of recommender systems for traveling abroad. *Expert Systems with Applications*, 38(12):15344 – 15355, 2011.

[41] C. E. Spearman. The proof and measurement of association between two things. *The American Journal of Psychology*, 15(1):72–101, 1904.

[42] O. Spencer-Thomas. Writing a Press Release. `http://www.owenspencer-thomas.com/journalism/media-tips/writing-a-press-release`, 2012. URL retrieved on the 2nd March 2013.

[43] P.-N. Tan, M. Steinbach, and V. Kumar. Introduction to data mining. *Journal of School Psychology*, 19(1):500, 2005.

[44] P. Wang. Why recommendation is special? In *15th national conference on artificial intelligence*, pages 111–113, 1998.

[45] K. Yoshii, M. Goto, K. Komatani, T. Ogata, and H. G. Okuno. An efficient hybrid music recommender system using an incrementally trainable

probabilistic generative model. *IEEE Transactions on Audio, Speech and Language Processing*, 16(2):435–447, 2008.

[46] L. Zheng, L. Li, W. Hong, and T. Li. Penetrate: Personalized news recommendation using ensemble hierarchical clustering. *Expert Systems with Applications*, 40(6):2127–2136, May 2013.

[47] Y. Zhou, D. Wilkinson, R. Schreiber, and R. Pan. Large-scale parallel collaborative filtering for the netflix prize. In *Proc. 4th Int'l Conf. Algorithmic Aspects in Information and Management, LNCS 5034*, pages 337–348. Springer, 2008.