Trabajo Fin de Grado

Desarrollo de un Sistema de Monitorización y Recomendación de Documentos en la Web

Development of a System for Monitoring and Recommendation of Web Documents

Autor

Guillermo Azón Edroso

Director

Sergio Ilarri Artigas

ESCUELA DE INGENIERÍA Y ARQUITECTURA
2018

TRABAJOS DE FIN DE GRADO / FIN DE MÁSTER

Escuela de Ingeniería y Arquitectura Universidad Zaragoza

DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD

(Este documento debe acompañar al Trabajo Fin de Grado (TFG)/Trabajo Fin de Máster (TFM) cuando sea depositado para su evaluación).

D./Dª. GUILLERMO AZÓN EDROSO,

con nº de DNI 73210680-R en aplicación de lo dispuesto en el art. 14 (Derechos de autor) del Acuerdo de 11 de septiembre de 2014, del Consejo de Gobierno, por el que se aprueba el Reglamento de los TFG y TFM de la Universidad de Zaragoza,

Declaro que el presente Trabajo de Fin de (Grado/Máster) GRADO, (Título del Trabajo) DESARROLLO DE UN SISTEMA DE MONITORIZACIÓN Y RECOMENDACIÓN DE DOCUMENTOS EN LA WEB

DEVELOPMENT OF A SYSTEM FOR MONITORING AND RECOMMENDATION OF WEB DOCUMENTS,

es de mi autoría y es original, no habiéndose utilizado fuente sin ser citada debidamente.

Zaragoza, 29 de JUNIO de 2018

Fdo: GUILLERMO AZÓN EDROSO

Desarrollo de un Sistema de Monitorización y Recomendación de Documentos en la Web.

RESUMEN

En la actualidad, el ser humano convive con una sociedad en la que la información crece a pasos agigantados. En numerosas ocasiones puede suponer un gran reto, ya que el usuario puede verse saturado con los datos que recibe constantemente. Emails llegando diariamente, notificaciones de las redes sociales, periódicos y blogs recomendando artículos, etc. Por este problema, es necesario ofrecerle al usuario la capacidad de tener el manejo sobre lo que necesita.

Este trabajo ha sido realizado con el fin de crear una herramienta que pueda servir de ayuda al usuario para tener un control sobre diferentes temas que sean de su interés, centrándose, por el momento, en las noticias de texto. El usuario es capaz de realizar un seguimiento de un tópico de su interés, pudiendo almacenar la información acerca de él a su gusto. En todo momento tiene la posibilidad de valorar sus resultados, para posteriormente poder recibir una recomendación basada en contenido que le permita conocer nuevas noticias que pudieran ser de su agrado.

Este trabajo no sólo destaca por el hecho de poder recomendar noticias, sino que son más las funcionalidades ofrecidas al usuario. Se han combinado técnicas de sistemas de recomendación junto a otras técnicas de minería de textos y de recuperación de información. Se le da la posibilidad al usuario de realizar un análisis más conciso de la noticia, permitiéndole obtener de ella los términos más relevantes. En el caso de las localizaciones, se le ofrece la opción de situarlas en un mapa para poder conocer en todo momento dónde están los lugares de los que habla la noticia.

Como resultado de este trabajo se ha desarrollado un sistema que combina técnicas de sistemas de recomendación con técnicas de recuperación de información, con el fin de proporcionar al usuario una ayuda en el seguimiento de sus temas preferidos.

En relación al futuro, podría recibir extensiones y mejoras por parte del grupo COS2MOS, siendo una potencial base para investigaciones futuras. También podría ser utilizado en el contexto académico en el caso de que pudiese ser de utilidad para la enseñanza de determinados conceptos, bien sea utilizando directamente la interfaz desarrollada o modificando código y añadiendo nuevas funcionalidades.

Índice

Lista de figuras

Lista de tablas

1. Introducción

Hoy en día vivimos en una sociedad en la que se convive con la gran cantidad de información que se genera continuamente. Los emails están continuamente llegando, las notificaciones de las redes sociales son cada vez más frecuentes, los periódicos y blogs, e incluso YouTube, cada día te ofrecen tanto noticias como vídeos que pueden resultarte relevantes, etc. Toda esta información puede llegar a saturar a un determinado usuario, lo que conlleva que acabe sin visitar dichos documentos. Por todo esto, si un usuario quiere realizar el seguimiento de un determinado tópico de manera eficiente, va a necesitar de un filtrado que le permita mantener la revisión de sus ítems manejable, ya que de lo contrario, el usuario se rendirá o realizará lecturas aleatorias.

El gran aumento de la información conlleva también el auge de las técnicas de recuperación de información y de los sistemas de recomendación. Cada vez son más las empresas que incluyen técnicas de análisis de datos para conocer mejor su rendimiento y poder ofrecer lo que el usuario realmente necesita. Los datos son importantes, por lo que un buen análisis de ellos puede suponer un crecimiento del beneficio de la empresa.

Por toda esta gran cantidad de datos que maneja un usuario, la sociedad actual ha llegado a denominarse en muchas ocasiones como la sociedad de la información. Tanto las empresas como las personas manejan continuamente gran cantidad de datos que se mueven a una velocidad vertiginosa por la red, lo que ayuda a que en pocos segundos, la información pueda pasar de un punto del mundo a otro.

Aprovechando la celebración reciente del foro "Big Data to Action 2018" [1], se va a comentar la definición que propuso Luis Ortiz [2], director del Área de Big Data & Analytics de MSMK (Madrid School of Marketing [3]), sobre la realidad actual. La sociedad de hoy en día la define como "mundo VUCA", el cual se caracteriza por su Velocidad (V), Incertidumbre (U), complejidad (C) y ambigüedad (A). La velocidad a la hora de evolucionar la sociedad es un hecho. Crece a pasos agigantados, lo que en muchas ocasiones nos lleva a la incertidumbre sobre cómo afrontar los acontecimientos futuros. Todo está relacionado, por lo que si la velocidad de evolución es vertiginosa y la incertidumbre que ésta genera no nos permite tener ciencia exacta de lo que ocurrirá en un futuro, las organizaciones tienen muy presente una complejidad a la hora de comprender la relación entre los diferentes elementos. Por último, en cuanto a la ambigüedad se refiere, cada vez es más difícil comprender lo que ocurre en la realidad, ya que se proporciona más de un significado sobre la misma, lo que implica que no se sepa conceptualizar correctamente.

1.1. Presentación de DodoAid

Como se ha comentado, la sociedad crece a pasos agigantados, lo que implica que la información y los datos que la forman crezcan a su ritmo. Por tanto, si se quiere que el usuario no acabe sobrecargado, se le deberá permitir tener la capacidad de controlar el seguimiento de la información que desee. Por todo esto, se considera que podría ser de gran utilidad para el usuario tener una herramienta que le permitiese controlar toda la información de interés. De este modo, y con el fin de ayudar al usuario a manejar la gran cantidad de datos que le rodean, nace DodoAid. En el Anexo D, se muestra el proceso que se ha seguido tanto para la elección del nombre como para la elaboración del logo.

DodoAid (Aid for Documents and Other Digital Objects) es un sistema de recomendación de objetos digitales que trata de aliviar la sobrecarga de información del usuario cuando éste quiere recibir información de un cierto tema. Más allá de las técnicas de recuperación de información y de minería de textos que puede realizar, también usa técnicas relacionadas con los sistemas de recomendación para sugerir al usuario determinados ítems que pueden ser de su interés en relación a las preferencias del usuario.

Su propósito ha sido, desde un primer momento, la ayuda al propio usuario, por lo que todas las decisiones que se han tomado han sido en busca de la comodidad de éste. El objetivo marcado consistía en la elaboración de un sistema que permitiese al usuario poder personalizar en todo momento sus búsquedas. De esta manera, a través de la conexión al motor de búsqueda de Google, se le permite la búsqueda de sus noticias preferidas. Aparte de esto, se le proporciona la posibilidad de conocer en todo momento las noticias que ha visto, permitiéndole realizar acciones sobre ellas, como valorarlas, añadirlas a favoritas o borrarlas si no son de su agrado. Es por ello, que la información sobre sus búsquedas es representada como si de un modelo jerárquico se tratase, permitiendo al usuario guardar las noticias (los hijos) separadas por carpetas (los padres) que él mismo puede crear a su gusto. Sobre dicha estructura jerárquica, el usuario es capaz de renombrar los elementos que la forman o borrarlos en cualquier momento.

En segundo lugar, se ha pensado que el usuario podría querer guardar toda la información que ha ido recopilando con el uso de la herramienta. Por ello, se le ofrece la posibilidad de exportar la información correspondiente a sus búsquedas en un determinado formato. De esta manera, se asegura poder guardar la información acerca de sus noticias en todo momento.

A partir de la idea de que el usuario pueda exportar los resultados de sus búsquedas, se llega a la conclusión de que también es posible que quiera importar su propia información para guardarla en la estructura jerárquica que él mismo ha definido. De este modo, se le van a proporcionar dos formas de hacerlo. La primera de ellas consiste en escribir la información relacionada a la noticia de forma manual. En cambio, la segunda de ellas le permite importar un

dataset estructurado de manera correcta por un usuario experto, el cual podrá transferir al conjunto de noticias vistas con el fin de tenerlas almacenadas junto a las que ha ido obteniendo con el uso de DodoAid.

Por otro lado, uno de los aspectos más importantes de la aplicación es la recomendación. Al tratarse inicialmente de una aplicación de escritorio para un solo usuario, se ha realizado una recomendación basada en contenido. Para su correcto funcionamiento, el usuario ha debido de realizar de manera implícita alguna valoración, como por ejemplo, visitar un enlace o valorar una noticia. Dicha recomendación trabaja a partir de la distancia coseno entre vectores de palabras, es decir, estudia la similitud entre los textos a partir de la bolsa de palabras que los forman.

Por último, conforme se ha ido realizando el proyecto, se consideró de interés enviar un artículo corto para su evaluación a la Conferencia Española sobre Recuperación de Información (CERI 2018), que fue aceptado [4]. Como resultado de ello, se ha completado el sistema con algunas nuevas funcionalidades que permiten al usuario aplicar técnicas de minería de textos para poder realizar un procesamiento de la información y obtener nombres de entidades, palabras claves, localizaciones, etc. Estas últimas se permiten ubicar en un mapa, de tal manera que el usuario puede conocer el lugar exacto sobre las localizaciones que habla el documento. Además, dichas localizaciones van relacionadas con información relevante obtenida de la DBpedia, permitiendo en todo momento al usuario poder acceder a la DBpedia para obtener toda la información sobre dicha localización.

1.2. Flujo de trabajo en la herramienta

Una vez conocemos el objetivo del sistema, se va a pasar a analizar más a fondo el sistema desarrollado. En la Fig. 1.1, se puede ver el flujo de trabajo de DodoAid a partir de las interacciones que va realizando el usuario.

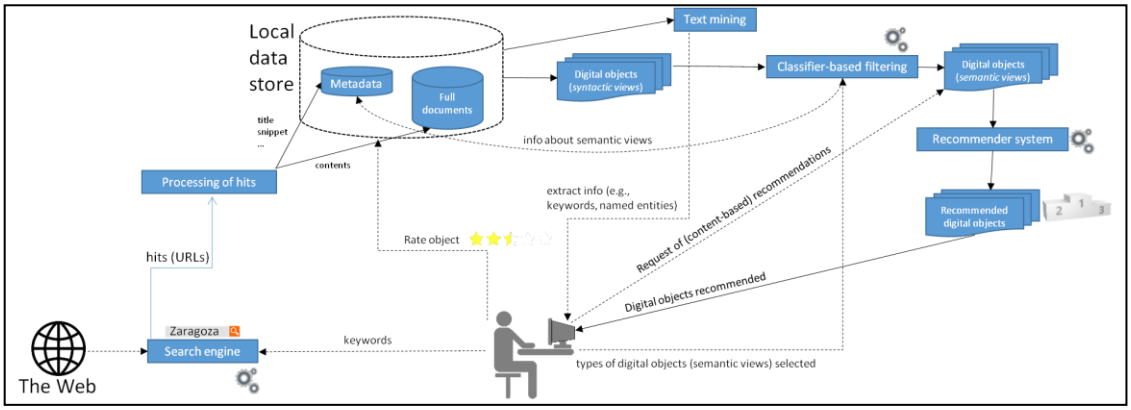

Fig. 1.1.- Flujo de trabajo de DodoAid.

El sistema dispone, en primer lugar, de un motor de búsqueda que va a recuperar los objetos digitales de la Web. El usuario puede realizar un seguimiento de la información que sea de su

interés. Por ejemplo, un determinado usuario realiza una búsqueda sobre el término "Zaragoza" obteniendo los resultados a partir del motor de búsqueda utilizado. Dichos resultados van a ser almacenados en la base de datos de dos formas bien diferenciadas. Por un lado, se van a almacenar aquellos metadatos que puedan resultar de mayor interés, como por ejemplo el título o el *snippet*. Por otro lado, se va a permitir descargar todo el contenido del objeto digital para poder almacenarlo en su totalidad. Este segundo caso se aplica cuando se quiere realizar un procesamiento de la información textual, en el cual se descarga el contenido de la noticia mediante el uso de jsoup [5] y se almacena en la base de datos.

Una vez que el usuario ha realizado búsquedas de interés, surge el término de "syntactic views", las cuales hacen referencia a los objetos digitales obtenidos tras una búsqueda de un usuario. Para poder entenderlo correctamente, se va a explicar mediante un ejemplo. Un usuario realiza una búsqueda sobre el término "Zaragoza". El resultado que obtiene son diez objetos digitales diferentes que hablan sobre dicho término de interés. Por tanto, las "syntactic views" en este ejemplo corresponderían con los objetos digitales que se han devuelto tras la búsqueda realizada.

A partir de la información almacenada en la base de datos, el sistema puede aplicar técnicas de minería de textos. De este modo, el usuario va a poder procesar la información textual de sus objetos digitales y va a obtener información relevante sobre ellos, como por ejemplo el nombre de entidades, palabras clave o localizaciones. A partir de estas últimas, se permite al usuario ubicarlas en un mapa para tener conocimiento de los sitios en los que se habla en un determinado documento digital.

Otra funcionalidad que dispone DodoAid es el hecho de poder clasificar los documentos. Para ello, el usuario debe establecer un conjunto de datos de entrenamiento con los que poder entrenar el clasificador. Una vez entrenado el clasificador, se generan las "semantic views". Éstas corresponden a las categorías en las que podrán clasificarse aquellos ítems que no tengan una clasificación previa. Para ver más clara la diferencia entre "syntactic views" y "semantic views" se va a poner un ejemplo sencillo. El usuario ha definido dos "semantic views" que corresponden con "Programming" y "Football". Al entrenar el clasificador con dichos términos, todos los ítems que se van a clasificar van a pertenecer a una categoría u otra. El usuario ha realizado dos nuevas búsquedas: "Data Mining" y "Real Zaragoza". Los objetos digitales que obtiene el usuario a partir de dichas búsquedas corresponden con las "syntactic views". El usuario quiere poder clasificarlas, ya que dichos objetos digitales están sin categoría. Por ello, tras tener el clasificador entrenado, lo lógico sería que la clasificación nos dijese que "Data Mining" pertenece a la categoría "Programming" y "Real Zaragoza" a la categoría "Football".

Para finalizar, el usuario puede dar una valoración a los diferentes documentos digitales que obtiene. Para que el sistema de recomendación trabaje correctamente, es fundamental que el

usuario haya valorado algunos ítems. De este modo, el sistema de recomendación aprende los gustos del usuario y va a poder ofrecerle ítems que sean similares a los de su interés.

1.3. Objetivo

El objetivo que se trata de abordar en este trabajo es la elaboración de un sistema que permita al usuario buscar y monitorizar de forma automática información de interés en la Web. A partir de sus búsquedas, se aplicarán técnicas de minería de textos que, combinándolo junto a técnicas de sistemas de recomendación que utilizan las valoraciones de los usuarios sobre su interés en los ítems, aprendan los gustos de los usuarios para obtener recomendaciones de objetos digitales relevantes.

El propio usuario puede realizar búsquedas sobre los aspectos que más le interesen, permitiéndole en todo momento tener un acceso a dichas búsquedas para organizarlas según sus gustos. Además, el usuario podrá evaluar las noticias para que el sistema de recomendación pueda aprender de dichas valoraciones y pueda ofertarle noticias de su agrado.

Por último, se han abordado técnicas de recuperación de información, con el objetivo de permitir al usuario conocer con mayor detalle la información que se le muestra en los objetos digitales. De este modo, se permite al usuario conocer los términos más relevantes de un determinado objeto.

1.3. Estructura del documento

En este trabajo se presenta el desarrollo de la herramienta DodoAid. Dentro de este primer capítulo se va a hablar sobre cómo nace DodoAid, cuál es su flujo de trabajo y cuál es el objetivo que trata de abordar. En el capítulo 2, se describen algunos ejemplos de trabajos similares a DodoAid. En el capítulo 3, se va a hablar brevemente de la base de datos utilizada y el esquema entidad/relación que se sigue. En el capítulo 4, se va a profundizar más en las funcionalidades que ofrece DodoAid al usuario. En el capítulo 5, se va a realizar un análisis de las tecnologías que se han utilizado para poder implementar las funcionalidades descritas. Por último, en el capítulo 6, se presentan las conclusiones obtenidas en el trabajo junto a los futuros cambios que se espera realizar en el sistema.

Adicionalmente, se presentan diferentes anexos para complementar la memoria. En el Anexo A, se muestran los diagramas de secuencia para ver la interacción entre los diferentes elementos a la hora de realizar una acción. En el Anexo B, se adjunta un manual de usuario para que sepa cómo acceder a todas las funcionalidades de manera sencilla. En el Anexo C, se muestran diferentes pruebas de evaluación que se han realizado. Por último, en el Anexo D se muestra información relacionada con la elección del nombre y el diseño del logo.

2. Trabajo relacionado

En el siguiente apartado se va a realizar un análisis de diferentes trabajos que tienen una relación similar a DodoAid [6]. Los trabajos seleccionados se caracterizan por ofrecer al usuario la recomendación de ítems. Para finalizar, se hablará brevemente del público objetivo que podría llegar a utilizar la herramienta desarrollada.

2.1. Análisis de trabajos similares

1. Stories around You: Location-based Serendipitous Recommendation of News Articles [7] →Este trabajo se centra en ofrecer al usuario una recomendación de artículos basándose en la localización del usuario, pero centrándose, principalmente, en la serendipia. La serendipia consiste en un hallazgo valioso de algo inesperado, cuando lo que se estaba buscando es algo totalmente diferente. Por tanto, este proyecto trata de ofertar al usuario, en base a su localización, la recomendación de artículos de interés que en un principio para él no deberían serlo. Para entenderlo mejor se va a relatar el ejemplo proporcionado en su artículo. Alicia es una persona que no le gusta la música country. Un día, ella está paseando por un pueblo cerca de las montañas mientras escucha la radio con su teléfono móvil. De repente, en la radio ponen una canción country muy pegadiza y a Alicia le acaba gustando. Este hecho podría ser considerado como una experiencia de serendipia, ya que desde un principio a ella no le gustaba la música country y de casualidad ha descubierto una canción que sí. En resumen, el objetivo de este proyecto es tratar de recomendarle al usuario a partir del estudio de la información personal que proporciona, artículos que puedan ser de su interés de manera casual.

En comparación con DodoAid, no necesita de valoración previa del usuario sobre sus preferencias, sino que la recomendación se realiza en función de la localización más predominante del usuario. Por tanto, en relación con el sistema de recomendación desarrollado, DodoAid va a realizar recomendaciones a los usuarios más acordes a su gusto, por lo que la probabilidad de acertar es mayor. En cambio, con el otro existe la posibilidad de hacer descubrir al usuario algún interés que con DodoAid no hubiese encontrado.

2. Personalized News Recommendation with Context Trees [8] → Este proyecto se centra en la recomendación de noticias en una página web y se basa en realizar un seguimiento de la secuencia de artículos leídos por el usuario dentro de la propia página web. Se centran en el problema de que el usuario muchas veces no proporciona información suficiente para poder ofrecerle una buena recomendación. De esta manera, a partir de utilizar el algoritmo de los árboles de contexto, permiten ofrecer una recomendación a los visitantes anónimos, basada únicamente en cómo navega por el sitio web. Es decir, si el usuario está navegando por las noticias relacionadas con los deportes, el sistema de recomendación le ofrecerá nuevas noticias relevantes acerca de ese tema.

En comparación con DodoAid, no necesita de valoración previa del usuario sobre sus preferencias, sino que la recomendación se realiza en función de la navegación del usuario por la web. La forma en la que trabaja permite ofrecer recomendaciones a todos los usuarios sin necesidad de decir cuáles son sus preferencias. En cambio, en DodoAid si un usuario no realiza la valoración de ninguna noticia no se pueden conocer sus preferencias, por lo que no se puede recomendar ningún ítem. Pero si un usuario ha establecido sus preferencias, entonces DodoAid va a realizar recomendaciones más exactas, ya que con el otro sistema de recomendación es posible que un usuario haya pasado navegando bastante tiempo en un sitio de la web que realmente no le interesa, y las recomendaciones que se le realicen sean acerca de dicho tema.

3. PEN recsys system [9] →Al igual que el anterior, proporciona recomendaciones de documentos de una determinada web, en este caso, es usado sobre una página web de periódicos para evaluar varios algoritmos para la recomendación de noticias. Destaca por la posibilidad de realizar la recomendación con diferentes algoritmos:

- **a.** Cuatro versiones de sistemas de recomendación de árboles de contexto.
- **b.** Filtrado colaborativo simple.
- **c.** Filtrado basado en contenido.
- **d.** Artículos más populares.
- **e.** Artículos aleatorios.

DodoAid únicamente realiza el filtrado basado en contenido, aunque en un futuro se espera añadir el filtrado colaborativo. En relación a la recomendación, PEN ofrece una mayor posibilidad a la hora de cómo realizarla, pero seguramente a un usuario inexperto no le importe el hecho de cómo se ha hecho o qué técnica se ha utilizado. Por esta razón, se considera que DodoAid puede resultar más útil para usuarios inexpertos, ya que ofrece más funcionalidades a parte de la recomendación. Para un usuario experto, cuyo único objetivo sea el estudio los distintos tipos de recomendación, PEN puede ser una buena herramienta para comenzar su investigación.

4. Swissinfo.ch website [10] →De manera muy similar al anterior, este ejemplo es una página web que muestra las noticias de Suiza para todos aquellos que viven en el extranjero. Dependiendo del idioma que se escoja en la página web, va a mostrar unas noticias u otras. Se ha comprobado que si se selecciona el idioma inglés, se obtienen resultados distintos que si se selecciona francés o castellano. Por tanto, swissinfo.ch es un sistema de recomendación de noticias que se basa en función del idioma que decida el usuario, ya que recomienda aquellos documentos que considera más relevantes para cada idioma. A diferencia de DodoAid, no necesita ningún tipo de valoración del usuario sobre sus preferencias, ya que únicamente con saber el idioma que desea se produce la recomendación. Swissinfo.ch, aunque te da la opción de escoger el tema de noticias que quiere leer el usuario, no aprende de los gustos de éste por lo

que existe la posibilidad que haya algún usuario al que no se le muestren noticias de su interés. En cambio, en DodoAid como se necesita valoración previa del usuario, se va a poder recomendar en función a dichas preferencias.

5. Usenet [11,12] →Usenet es un sistema de discusión global en Internet. Cada día crece debido a la participación de los usuarios añadiendo su opinión sobre diversos temas. Se podría definir Usenet como la colaboración de servidores separados que intercambian las noticias. Un determinado usuario escribe un tema sobre el que hablar y gracias al modelo jerárquico que sigue se puede clasificar fácilmente en el tema sobre el que trata y recomendarle más en relación a éste. GroupLens crea un filtrado colaborativo para poder recomendar las lecturas al usuario, en función de las ya realizadas por el propio usuario. Actualmente se usa en mayor medida para compartir o descargar archivos, dejando más de lado su función de intercambiar noticias.

El filtrado es colaborativo, mientras que en DodoAid es basado en contenido. Si un usuario prefiere que se le recomienden noticias en función de otros usuarios con gustos similares, entonces se decantará por Usenet. En cambio, si prefiere una recomendación únicamente a partir de sus búsquedas, su opción es DodoAid.

6. MONERS [13] → Por último, se encuentra este sistema de recomendación de noticias que se centra en los usuarios móviles, como por ejemplo [14,15], subrayando la importancia de considerar la novedad de los artículos. Los móviles deben disponer de conexión inalámbrica y para la recomendación de estas noticias se ha de tener en cuenta aspectos relacionados con el teléfono, ya que, por ejemplo, no todos disponen de la misma capacidad de almacenamiento. Pero, resumiendo, MONERS es un sistema de recomendación de noticias que se centra en las necesidades de los usuarios móviles.

Al igual que DodoAid, necesita que el usuario establezca sus preferencias para poder realizar una recomendación. Ésta se basa en función de la importancia del artículo y de lo reciente que sea. A la hora de escoger entre uno de los dos sistemas de recomendación, va a depender de si el usuario desea una aplicación de escritorio o si lo que desea es una aplicación móvil. Aún así, un ordenador tiene mayores prestaciones que un teléfono móvil, por lo que DodoAid vence a su competidor en este aspecto.

Para finalizar el apartado de análisis de la competencia, se va a adjuntar la tabla comparativa Tabla 2.1, en la que se pueden apreciar las características principales de cada uno de los trabajos descritos.

	Plataforma	Aspecto Clave	Cómo Funciona	Info. previa usuario
Stories around you	Dispositivo móvil	Serendipia	A partir de la localización predominante del usuario, realiza recomendación de ítems que en un principio no tendrían que gustarle al usuario.	No. Aprende de la localización.
PNR Context Trees	Página web	Navegación del usuario	A partir de la navegación del usuario sobre una página web, se puede descubrir cuáles son sus mayores preferencias y realizar la recomendación.	No. Anónimo.
PEN	Página web	Algoritmos de recomendación	Es un sistema de recomendación de noticias que permite la recomendación con diferentes algoritmos: - 4 versiones de sistemas de recomendación de árboles de contexto. - Filtrado colaborativo simple. - Aproximación basada en contenido. - Artículos más populares. - Artículos aleatorios.	Sí. Según el algoritmo que se vaya a usar se pueden necesitar datos del usuario.
Swissinfo	Página web	Recomendación de noticias en función del idioma	Dispone de varios idiomas a la hora de mostrar las noticias. Dependiendo del idioma seleccionado, se van a recomendar aquellas noticias que sean más relevantes para dicho idioma.	No. Solamente el idioma que es seleccionado.
Usenet	Web.	Recomendación a partir de temas de discusión de interés	El usuario escribe sobre un tema de discusión de su interés. El sistema, a partir de su modelo jerárquico, lo clasifica y recomienda al usuario temas relacionados a éste en función de la clasificación jerárquica.	Sí. A partir de los temas en los que va participando.
MONERS	Dispositivo móvil	Recomendación de páginas web en dispositivos móviles	Se basa en la recomendación de noticas para móviles en función de la importancia del artículo o de lo reciente que sea.	Sí.
DodoAid	Aplicación escritorio.	Recomendación de noticias en función de las preferencias del usuario.	Se basa en la recomendación de noticias en función del filtrado basado en contenido, el cual va a recomendar al usuario noticias dependiendo de sus preferencias.	Sí.

Tabla 2.1.- Tabla comparativa con los trabajos relacionados a DodoAid.

Como conclusión, se observa que los trabajos analizados se centran en la recomendación dentro de una sola página web. DodoAid, a diferencia del resto, se centra en la Web pudiendo abarcar la recomendación de todos los objetos digitales que ésta contiene.

2.2. Público objetivo

El sistema que se ha creado ha sido pensado en todo momento para un público que pueda disponer de un ordenador con conexión a la red. Su interfaz es muy intuitiva y no hace falta tener conocimientos avanzados de informática para poder darle un correcto uso. También se ha pensado en que algunas partes del sistema puedan ser más útiles para usuarios más avanzados. Se considera avanzado, al usuario con algún conocimiento sobre cómo funciona un clasificador o sobre la recuperación de información, ya que en esos apartados un usuario sin ningún conocimiento previo no conocerá que estructura seguir para importar un *dataset* o qué opciones utilizar para realizar la minería de textos y recuperar información relevante.

En cambio, un usuario sin conocimientos informáticos, puede disfrutar de muchas funcionalidades. Por ejemplo, además de recibir recomendaciones en función de sus gustos, puede utilizar la herramienta para seguir la información relevante sobre un ítem de su interés, permitiéndole en todo momento almacenar las noticias a su gusto y así acudir a ellas cuándo sea de su necesidad.

3. Base de Datos

Para el almacenamiento de los datos en el sistema generado, se ha utilizado SQLite [16] como sistema gestor de bases de datos. Las tablas usadas se han ido generando en función de las necesidades que iban surgiendo en el desarrollo del proyecto. Inicialmente se disponía de una tabla para los usuarios y otra para las noticias, a partir de las cuales se podía almacenar y consultar todo lo necesario para trabajar. Conforme se ha ido desarrollando el sistema, han surgido mejoras y, consecuentemente, se ha tenido que modificar la estructura de la base de datos. A la hora de mostrar al usuario las noticias que ha visualizado, se le ofrece la posibilidad de almacenarlas en una estructura similar a un modelo de datos jerárquicos teniendo que almacenar en la base de datos la ruta de cada noticia. Por ejemplo, un usuario ha realizado una búsqueda sobre "Zaragoza" y la ha guardado en una carpeta denominada "Aragón". En la base de datos se almacenará la ruta para dicha noticia como "Aragón/Zaragoza". De este modo, no existe la posibilidad de confundir dos búsquedas, ya que el usuario podría haber generado otra búsqueda de Zaragoza, pero quedándose únicamente con las noticias que hablasen del fútbol.

Otro aspecto que se ha tenido en cuenta a medida que se iba programando el sistema es la posibilidad que el usuario pueda establecer su propia configuración a la hora de mostrar la información. De este modo, se ha creado una tabla para la configuración, aunque se podría haber puesto directamente en un atributo para el usuario. Se ha decidido definirlo de esta manera por el hecho de que en un futuro se espera que también pase a ser un sistema de recomendación basado en el filtrado colaborativo y, por tanto, haya más usuarios almacenados.

Por último, se acabó modificando ligeramente la tabla de noticias, permitiendo el almacenamiento completo de la misma con el fin de que cuando un usuario quiera realizar el procesamiento de la información sobre un determinado ítem por segunda vez, éste sea más rápido por el hecho de que ya se había descargado su contenido. Además, a la hora de importar los *datasets* para trabajar con el clasificador, se ha creado una nueva tabla que almacena toda la información correspondiente al *dataset*, para así tenerla diferenciada de las noticias visualizadas por el propio usuario.

A continuación, en la Fig. 3.1 se puede ver el esquema entidad/relación, de manera sencilla y resumida, con el que trabaja el sistema desarrollado. Por simplicidad, se ha decidido no añadir los atributos que forman cada entidad, con el fin de darle mayor relevancia a la estructura que se ha seguido para poder tener toda la información del sistema almacenada. En relación a los atributos, se van a comentar brevemente los de mayor relevancia y cuáles actúan como claves. La entidad "User" va a tener como clave el nombre de usuario, de tal manera que cuando se pase a un filtrado colaborativo, no se pueda dar el caso de tener dos usuarios bajo el mismo nombre. La entidad "Configuration" tiene atributos relacionados con la configuración del sistema, como por ejemplo, la configuración de las columnas a mostrar a la hora de realizar una

búsqueda. La entidad "Noticia" tiene como clave primaria el nombre del usuario, el link de la noticia y el término de la búsqueda, con el fin de que para un determinado usuario no se pueda guardar una misma noticia obtenida dos veces a partir de la misma búsqueda. Su objetivo es el de almacenar las diferentes noticias, por lo que se va a guardar información relacionada a los metadatos (*snippet*, título, etc.) y las valoraciones que ha realizado el usuario sobre ellas. En la entidad "File" la clave primaria corresponde con el nombre del usuario, el nombre del fichero y el nombre de la carpeta y, por último, para la entidad "Ruta" la clave va a ser la ruta establecida para las noticias. Hay que tener en cuenta, que una ruta puede contener N noticias.

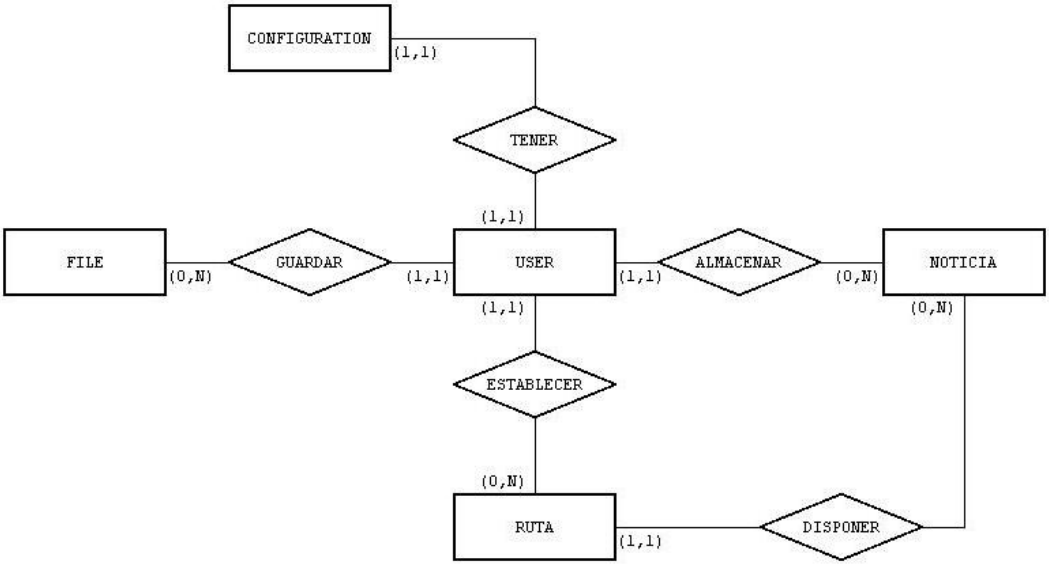

Fig. 3.1.- Esquema Entidad/Relación simplificado del sistema DodoAid.

En relación al posterior esquema relacional, el objetivo ha sido poder tener en tablas bien diferenciadas los distintos elementos que participan.

La tabla "File" almacena los ficheros de los *datasets* importados por el usuario. Un usuario puede almacenar todos los ficheros que desee, pero éstos únicamente corresponden a un determinado usuario.

La tabla "Configuration" corresponde al almacenamiento de la configuración que desea el usuario en el sistema. Inicialmente, todos los usuarios que utilicen DodoAid, disponen de la misma configuración. Una vez dentro, tienen la posibilidad de adaptarla a su gusto, por tanto la configuración es única para el usuario y un usuario sólo puede disponer de una configuración.

La tabla "Ruta" hace referencia a las rutas que tienen las noticias para ser mostradas al usuario. Inicialmente, cuando el usuario realiza una búsqueda, todas se guardan bajo la ruta "All/". Cuando el usuario va creando carpetas y va situando las noticias a su gusto, se modifica la ruta por la correspondiente formada por su padre. Imaginemos que el padre ha creado la carpeta "Fútbol". Dentro de ella ha creado una subcarpeta llamada "Equipos", en la cual almacena la búsqueda que ha realizado sobre "Real Zaragoza". Las noticias correspondientes a

dicha búsqueda son almacenadas con la ruta “Fútbol/Equipos/Real Zaragoza”. Si el usuario vuelve a buscar “Real Zaragoza”, pero no lo sitúa en ninguna carpeta, la ruta de ésta será “All/Real Zaragoza”, siendo sus noticias diferenciadas de las primeras. De este modo, si borra las últimas noticias porque se da cuenta que ya había buscado lo que necesitaba, no afecta a ninguna búsqueda que sea sobre el tema “Real Zaragoza” que no esté almacenada con esa misma ruta.

En la Fig. 3.2, se puede ver la estructura que se ofrece al usuario a la hora de almacenar sus noticias.

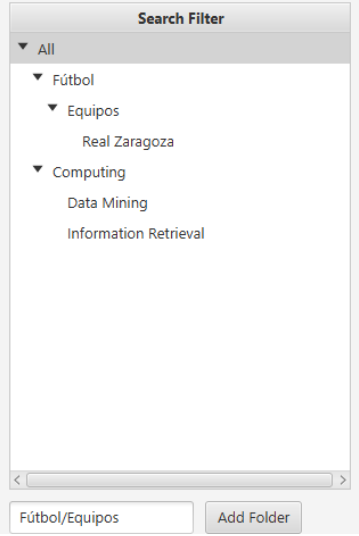

Fig. 3.2.- Ejemplo de la estructura de almacenamiento de las noticias.

A continuación se va a dar una explicación sobre las dos tablas que mayor importancia tienen en el esquema. En primer lugar, la tabla “User”, que corresponde a la información del usuario. Esta tabla ha sido creada con el fin de que en un futuro cobre más protagonismo por el hecho de que para el filtrado colaborativo se espera de la participación de más de un usuario.

Por último, la tabla “Noticia” es la que más información aporta al usuario, ya que en ella se almacena todos los datos correspondientes a las búsquedas de noticias. Por tanto, en relación con la tabla “User”, un usuario va a poder almacenar todas las noticias que desee, mientras que, por el momento, una noticia sólo puede corresponder a un determinado usuario.

Resumiendo, se ha creado una base de datos sencilla, en la que la información está agrupada según lo que representa, que permita acceder a los diferentes datos sin mucha carga de trabajo.

4. Funcionalidades

En este apartado se van a explicar las distintas funcionalidades que tiene DodoAid y cómo se ha realizado cada una de ellas. En el Anexo A y en el Anexo B se complementa esta información mostrando diagramas de secuencia y un manual de usuario respectivamente.

4.1. Realizar búsquedas

Para obtener resultados acerca de una determinada consulta, se ha utilizado como motor de búsqueda Google a partir de JSON Custom Search API [17], el cual nos permite realizar cien búsquedas diarias de manera gratuita.

Con el uso de este API, se ha encontrado una limitación a la hora de ofrecerle al usuario la fecha exacta de publicación de las noticias buscadas, ya que no ofrece ningún metadato que permita obtenerla. En cambio, se permite al usuario realizar búsquedas a partir de una fecha determinada. Simplemente se establece la fecha de inicio y la fecha final de la búsqueda, y los resultados mostrados corresponden a noticias publicadas entre ese periodo de tiempo.

Entre la información relevante que se muestra al usuario se encuentra el título de la noticia, el *snippet* (resumen de la noticia) y la URL.

4.1.1.- Configurar la salida de la tabla de búsqueda

Cuando el usuario recibe las noticias por pantalla, se le da la posibilidad de configurar la salida de las columnas según lo que desee ver. En la Fig. 4.1 se aprecian las columnas que puede visualizar y cómo se le permite estructurarlas.

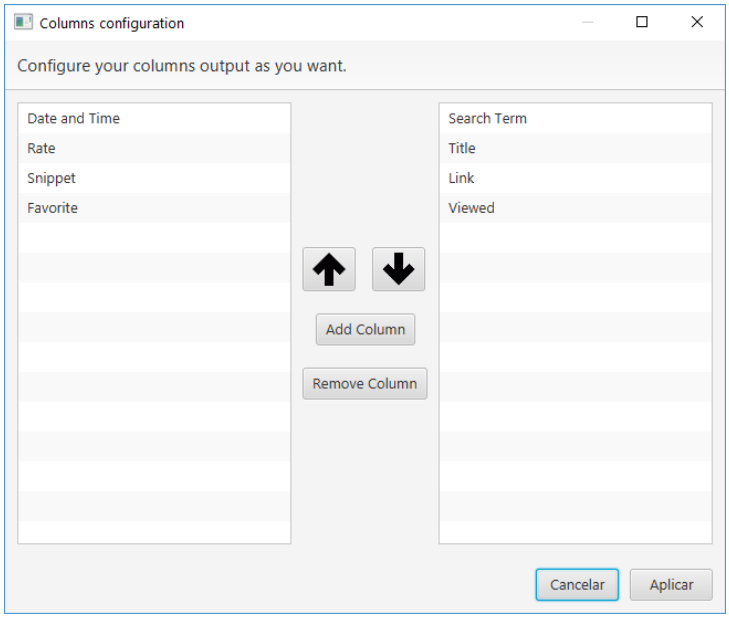

Fig. 4.1.- Configuración de columnas a mostrar.

En la parte derecha se sitúan las columnas que se muestran al usuario. En cambio, en la parte izquierda aparecen aquellas columnas que no se muestran. El usuario puede configurarlo cómo desee para obtener la información a su gusto. A la hora de mostrar la información, algunas columnas son consideradas más relevantes que otras, por lo que su tamaño es mayor que las otras. Las columnas de "Title", "Snippet" y "Link" son las consideradas con una mayor relevancia, por lo que a la hora de mostrarlas en la tabla disponen de un mayor tamaño para su visualización.

4.1.2.- Visualizar una noticia

Si el usuario quiere ver una noticia que se le ha proporcionado, debe acceder a la URL que se le muestra por pantalla. Al pulsar sobre ella, se abre la página web en el navegador por defecto del usuario para que pueda verla sin problema alguno.

4.1.3.- Valorar una noticia

El usuario puede valorar todas las noticias con una nota que va desde el 0 hasta el 10. Inicialmente, una noticia es clasificada como "NE" (Not Evaluated). La valoración de las noticias va a ser fundamental para poder recomendar al usuario noticias similares a sus gustos. Posteriormente se explicará el algoritmo que se sigue para poder realizar la recomendación de las noticias.

4.1.4.- Borrar un ítem

Al igual que se puede valorar una noticia, también puede ser eliminada de las noticias vistas. De esta manera, el usuario puede tener almacenadas únicamente las noticias que le interesen, eliminando aquellas que no le aporten nada. Una vez que el usuario elimina una noticia, no puede recuperarla en un futuro, por lo que ha de estar seguro que quiere hacer dicha acción, ya que para poder recuperarla debe volver a realizar la búsqueda esperando a obtener dicha noticia de nuevo.

4.1.5.- Añadir a favoritos

Por último, en relación a los resultados obtenidos tras una búsqueda, el usuario va a poder añadir o quitar una noticia de su lista de favoritos. Al igual que la valoración, este factor va ser usado también a la hora de la recomendación de las noticias.

4.2. Visualizar los ítems mostrados

Para que el usuario pueda tener un control sobre las noticias que ha ido visualizando, se le ofrece la opción de poder verlas en cualquier momento. Puede filtrarlas por los términos que elija y adaptar la salida con la configuración que desee.

A la hora de filtrarlas, le basta con hacer una pulsación doble sobre un determinado ítem. Éste a su vez puede ser padre y contener otros ítems. En este caso, se mostrará la información de todos sus hijos. Por ejemplo, el usuario tiene una carpeta que se llama "Noticias" y dentro de ellas ha guardado las búsquedas relacionadas a "Economía" y "Política". Al pulsar sobre "Noticias", el usuario recibe información sobre todos los términos que la forman, por lo que se le muestra por pantalla las noticias almacenadas con las búsquedas de "Economía" y "Política".

Al igual que en el punto 4.1, se le permite realizar las mismas tareas sobre las noticias visualizadas.

4.3. Importar URLs

Pensando en el hecho de que el usuario únicamente podía acceder a las noticias a través de las búsquedas realizadas, se llega a la conclusión de que se le tiene que permitir poder añadir cualquier noticia de manera manual. Es por ello que surgen dos opciones de hacerlo. La primera de ellas se va a explicar en este punto y la segunda de ellas va a ser explicada en el punto 4.4.

Al usuario se le permite añadir una noticia de manera manual pidiéndole tres datos básicos sobre ella: la URL, el título y un término que la defina (correspondería al término de búsqueda). De esta manera, el usuario puede añadir todas las noticias que desee a la herramienta.

4.4. Importar datasets

La segunda manera de poder importar sus propias noticias se realiza con la ayuda de Weka [18] y su clase "TextDirectoryLoader". Para este caso, es necesario que el *dataset* haya sido previamente estructurado por un usuario experto, ya que en caso de no haber sido bien formado es probable que el usuario obtenga error a la hora de importar.

La clase "TextDirectoryLoader" recorre todo el directorio que se le ha enviado en busca de subdirectorios para poder obtener los ficheros almacenados en estos últimos. De esta manera, se cargan en el sistema todas las noticias almacenadas en esos directorios para poder posteriormente realizar, si se desea, un entrenamiento del clasificador generado.

En un principio, estos ficheros cargados se guardan en una estructura diferente a las noticias vistas, pero se le da al usuario la posibilidad de poder almacenarlos en éstas.

4.5. Clasificador

Otra funcionalidad que se ha tratado de proporcionar al usuario es el hecho de poder clasificar sus datos bajo categorías. Para ello, se ha utilizado Weka.

Lo primero que se tiene que hacer es definir un conjunto de datos de entrenamiento con los que entrenar el clasificador. A partir de este entrenamiento, el clasificador va a definir las vistas semánticas, las cuales corresponden a las categorías en las que van a poder clasificarse los diferentes objetos digitales. Por ejemplo, un usuario ha realizado las búsquedas de los términos "Java", "Data Mining" y "Big Data", los cuales los agrupa bajo la vista semántica "Programación". El clasificador va a ser entrenado y, posteriormente, los nuevos objetos que estén sin clasificar que sean similares al contenido de dicha vista semántica, serán clasificados dentro de ella.

Para poder entrenar el clasificador se puede realizar de dos formas diferentes. La primera de ellas consiste en importar un *dataset* bien generado. Una vez que esto lo ha hecho, el usuario puede entrenar el clasificador. La segunda de ellas, como se observa en la Fig. 4.2, permite al usuario crear el conjunto de datos de entrenamiento a partir de las búsquedas que ha realizado.

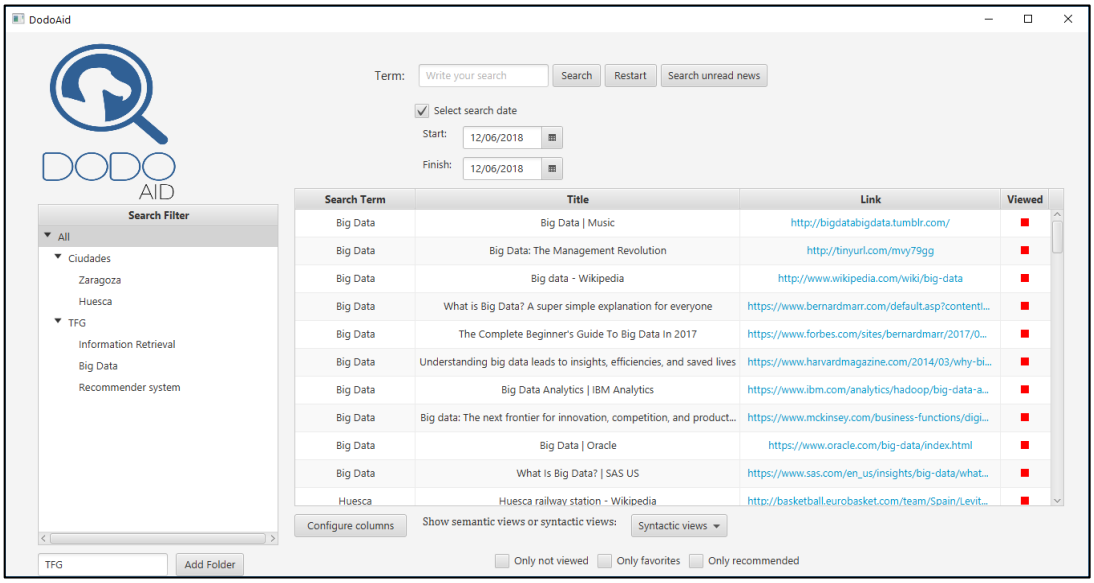

Fig. 4.2.- Estructura definida para entrenar el clasificador.

A la hora de entrenar el clasificador, se van a tener dos categorías bien diferenciadas como son “Ciudades” y “TFG”, las cuales contienen objetos digitales para ser entrenadas. De esta manera, cuando el usuario quiera clasificar un nuevo objeto que no esté clasificado, recibirá como solución una de las dos categorías que han sido definidas.

Además, cuando se entrena el clasificador, se le muestra al usuario una pantalla con la información más destacada. De esta manera, va a poder conocer el porcentaje de documentos bien o mal clasificados junto a una matriz de resultados que le puede permitir saber la relación entre las clasificaciones. También se le ofrecen otras estadísticas de interés como podrían ser la precisión o el *recall*.

Una vez terminada la clasificación, el usuario va a poder ver todos los documentos que han sido clasificados para las distintas categorías, distinguiéndose el hecho de poder visualizar vistas sintácticas, Fig. 4.3, de vistas semánticas, Fig. 4.4.

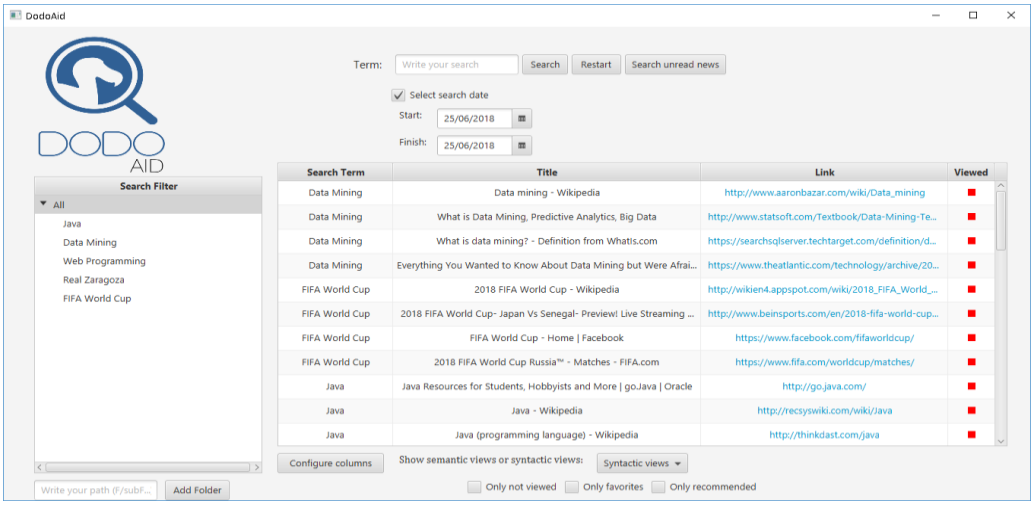

Fig. 4.3.- Visualización de las vistas sintácticas clasificadas.

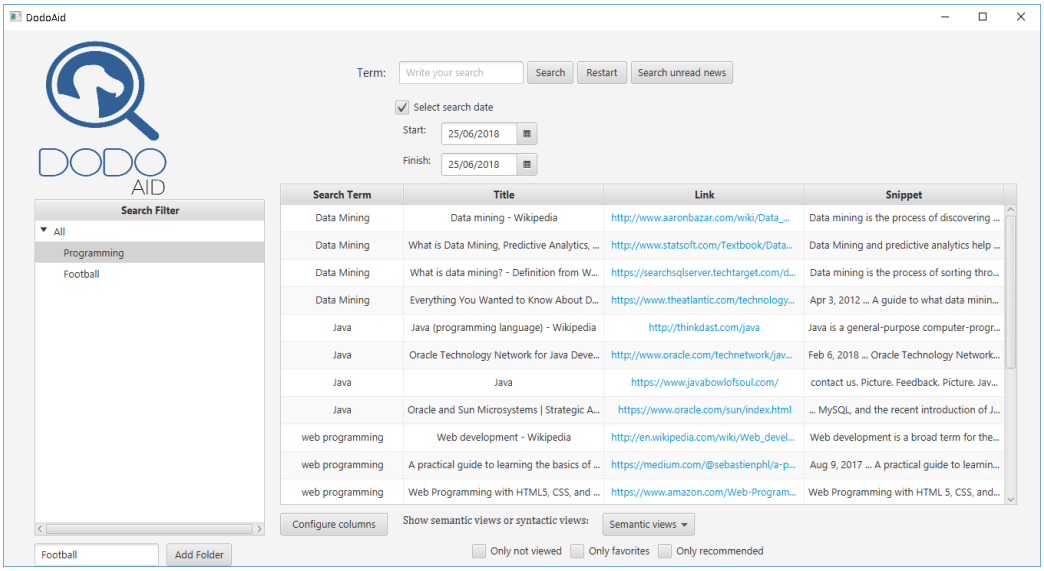

Fig. 4.4.- Visualización de las vistas semánticas clasificadas.

Como resultado de pulsar sobre una determinada vista semántica, se muestran todas las vistas sintácticas que lo componen.

Para el clasificador se ha utilizado el algoritmo J48[18,19]. Se podía haber decidido elegir otro, pero no se han realizado pruebas para comprobar cuál nos podría ofrecer una mejor clasificación. Para la evaluación se ha utilizado una validación cruzada de 10 iteraciones [18].

4.6. Exportar datos

Si el usuario es capaz de poder importar sus datos, es bastante probable que quiera poder descargarlos de manera local. De este modo, se le permite la descarga de las noticias que ha ido visualizando en un fichero con formato .html. La estructura interna del fichero permite guardar los datos siguiendo el siguiente formato:

Link_News ;;; Title_News ;;; Search term ;;; Search date ;;; View ;;; Favorite ;;; Rate

Una de las tareas futuras en el sistema corresponde con la mejora de la exportación de los datos. Se espera que se permita al usuario elegir las noticias que quiera exportar, en lugar de descargar todas las que tiene almacenadas. Además, se espera que el usuario pueda elegir los datos a descargar, en vez de tener un patrón fijo de descarga de los documentos. Por ejemplo, si un usuario quiere únicamente guardar el link y el título, permitir hacerlo. Por último, otra futura modificación consistiría en aumentar la posibilidad de elección de formato, ya que actualmente se exige al usuario la descarga en formato .html.

4.7. Recomendación de objetos digitales

Para la recomendación de noticias se ha utilizado Apache Mahout [20], aunque no disponga de un módulo específico para la recomendación basada en contenido. Uno de los puntos fuertes de Apache Mahout es la recomendación basada en el filtrado colaborativo. En nuestro caso, la intención es poder recomendarle al usuario las noticias en función de sus valoraciones, y no en relación a las valoraciones de otros usuarios, que no tienen por qué estar disponibles.

Apache Mahout no permite la realización directa de una recomendación basada en contenido, por lo que si lo que estás buscando es este tipo de recomendación tienes que crear tu propia función de similitud.

A continuación se va a explicar el algoritmo que se sigue para la recomendación de los ítems. Para ello, la explicación se va a dividir en dos secciones. En la primera de ellas, se va a explicar cómo se calcula la similitud coseno entre los vectores de palabras. En la segunda de ellas, se va a hablar sobre cómo se determina la puntuación que da el usuario a una determinada noticia. Para finalizar, se va a dar una explicación de cómo ambos aspectos se relacionan para obtener la recomendación.

4.7.1. Cálculo de la similitud entre objetos digitales

En primer lugar, las noticias a comparar son transformadas a un vector de palabras. A partir de este vector de palabras se calculan dos aspectos fundamentales en nuestra recomendación: el TF y el TF-IDF.

El término TF (Term Frecuency) corresponde a la repetición de un determinado término a lo largo de todo el documento. Por ejemplo, si tenemos la noticia “El rojo es rojo” se formaría una bolsa de palabras con tres términos: “El”, “rojo” y “es”. Se cuenta el número de repeticiones para cada término y se divide en función del número de términos que forman el documento. De este modo, se llega a un vector en el que se almacena el término junto a su valor de TF: [{term:“El”,TF:0.25},{term:”rojo”,TF:0,5}, {term:”es”,TF:025}].

En cambio, TF-IDF (Term Frecuency – Inverse Document Frecuency) es una medida cuyo objetivo es expresar lo relevante que es una palabra dentro de un documento situado en una colección de documentos.

Imaginemos que tenemos los siguientes documentos:

1.- “El cielo azul”

2.- “El cielo gris”

3.- “La luna amarilla”

En primer lugar, se crea la bolsa de palabras con todos los documentos quedando el siguiente resultado: [El, cielo, azul, gris, La, luna, amarilla]

A continuación, se calcula el valor de TF para cada frase:

1.- [{term: “El”, TF: 0.33}, {term:”cielo”, TF: 0.33}, {term:”azul”, TF: 0.33}]

2.- [{term: “El”, TF: 0.33}, {term:”cielo”, TF: 0.33}, {term:”gris”, TF: 0.33}]

3.- [{term: “La”, TF: 0.33}, {term:”luna”, TF: 0.33}, {term:”amarilla”, TF: 0.33}]

Una vez calculado, se pasa a obtener el valor IDF, el cual se calcula aplicando la siguiente fórmula: IDF(t) = log_e(Total de documentos / Número de documentos que contienen t).

Siguiendo con el ejemplo, el número total de documentos sería tres y habría que calcular el valor de IDF para cada término de la bolsa de palabras:

IDF(“El”) = log_e(3/2) = 0.405

IDF(“cielo”) = log_e(3/2) = 0.405

IDF(“azul”) = log_e(3/1) = 1.098

IDF(“gris”) = log_e(3/1) = 1.098

IDF(“La”) = log_e(3/1) = 1.098

IDF(“luna”) = log_e(3/1) = 1.098

IDF(“amarilla”) = log_e(3/1) = 1.098

Para terminar, únicamente quedaría calcular el valor TF-IDF, que se obtiene a partir de la multiplicación entre el valor de TF y el IDF correspondiente. En el ejemplo que se ha puesto, el resultado del vector con los valores TF-IDF quedaría de la siguiente manera:

1.- [{term:”El”, TF-IDF = 0.33x0.405}, {term:”cielo”, TF-IDF = 0.33x0.405}, {term:”azul”, TF-IDF = 0.33x1.098},{term:”gris”, TF-IDF = 0}, {term:”La”, TF-IDF = 0}, {term:”luna”, TF-IDF = 0}, {term:”amarilla”, TF-IDF=0}]

2.- [{term:”El”, TF-IDF = 0.33x0.405}, {term:”cielo”, TF-IDF = 0.33x0.405}, {term:”azul”, TF-IDF = 0 },{term:”gris”, TF-IDF = 0.33x1.098}, {term:”La”, TF-IDF = 0}, {term:”luna”, TF-IDF = 0}, {term:”amarilla”, TF-IDF=0}]

3.- [{term:”El”, TF-IDF = 0}, {term:”cielo”, TF-IDF = 0}, {term:”azul”, TF-IDF = 0 },{term:”gris”, TF-IDF = 0 }, {term:”La”, TF-IDF = 0.33x1.098}, {term:”luna”, TF-IDF = 0.33x1.098}, {term:”amarilla”, TF-IDF=0.33x1.098}]

Para calcular si son similares entre sí, se va a aplicar el algoritmo de similitud coseno entre los diferentes documentos [21]. La fórmula correspondiente es la siguiente:

A = [A_1, A_2,… A_N]

B = [B_1, B_2,… B_N]

$$\text{Sim\_Cos (A, B)} = \frac{\sum_{i,j}^{N} s_{ij} A_i B_j}{\sqrt{\sum_{i,j}^{N} s_{ij} A_i A_j}\sqrt{\sum_{i,j}^{N} s_{ij} B_i B_j}},$$

donde s_{ij} = similitud(característica$_i$, característica$_j$).

Para entender mejor la fórmula, sigamos de nuevo con el ejemplo que teníamos:

1.- [0.145, 0.145, 0.362, 0, 0, 0, 0]

2.- [0.145, 0.145, 0, 0.362, 0, 0, 0]

3.- [0, 0, 0, 0, 0.362, 0.362, 0.362]

$$\text{Sim\_Cos (1,2)} = \frac{0.145*0.145+0.145*0.145}{\sqrt{0.145*0.145+0.145*0.145+0.362*0.362}\sqrt{0.145*0.145+0.145*0.145+0.362*0.362}} = 0.24$$

$$\text{Sim\_Cos}(1,3) = \frac{0}{\sqrt{0.145*0.145+0.145*0.145+0.362*0.362}\sqrt{0.362*0.362+0.362*0.362+0.362*0.362}} = 0$$

$$\text{Sim\_Cos}(2,3) = \frac{0}{\sqrt{0.145*0.145+0.145*0.145+0.362*0.362}\sqrt{0.362*0.362+0.362*0.362+0.362*0.362}} = 0$$

Por tanto, se observa cómo la frase 1 y 2 tendrían el mayor valor de similitud coseno y por tanto se podrían considerar similares entre sí aplicando la técnica de creación de bolsas de palabras con el valor TF-IDF.

4.7.2. Cálculo de la puntuación de un objeto digital

Para el cálculo de la puntuación de un objeto digital, se ha aplicado un algoritmo que nos permite conocer la satisfacción de un usuario sobre una noticia determinada. El algoritmo se centra en tres aspectos, dándole a cada uno de ellos un valor de importancia diferente. El primero de ellos corresponde a las valoraciones que ha dado el usuario sobre una determinada noticia (α). El segundo de ellos hace referencia a comprobar si la noticia está en la lista de favoritas o no (β) y, el último de los factores se centra en el número de visitas que ha hecho un usuario sobre el enlace de una noticia (γ).

$$\text{score}(\alpha,\beta,\gamma) = \alpha \cdot 0.5 + \beta \cdot 2.5 + \gamma \cdot 0.25$$

La valoración total del algoritmo es de un máximo de 10 puntos y el reparto de los puntos se realiza de la siguiente manera:

La valoración de una noticia va del 0 al 10, por lo que el valor que le haya dado se multiplica por una constante de 0.5. Este aspecto es el que más relevancia tiene en el cálculo.

El hecho de que una noticia esté en favoritas hace que se sume 2.5, mientras que si no está en la lista de favoritas no suma.

Por último, el número de visitas suma de forma similar al hecho de añadirlo a favoritas, pero en el caso de que el usuario visite una noticia 10 o más veces, el máximo que va a poder sumar es de 2.5 (0.25*10). De este modo, una noticia va a poder estar valorada por el usuario hasta con una nota de 10 puntos.

Para ver de forma más clara el funcionamiento del algoritmo se va a explicar con un ejemplo. Un usuario ha valorado con un 7 una noticia, la ha añadido a sus favoritas y la ha visitado 2 veces. El resultado sería: 7*0.5 + 2.5 + 2*0.25 = 6.5.

El mejor de los casos corresponde a que el usuario haya valorado una noticia con un 10, la haya añadido a favoritas y haya visitado la página 10 o más veces. En cambio, el peor de los casos se define con que el usuario haya valorado la noticia con un 0, no haya añadido la noticia a la lista de favoritos y no haya visitado la página ninguna vez.

Con este algoritmo se puede saber si una noticia puede ser de interés o no para el usuario, ya que al calcular el TF-IDF se sabe que una determinada noticia es similar a otra, pero si la noticia

a la que es similar tiene una valoración en este último algoritmo explicado muy bajo, es probable que al usuario no le guste la noticia.

4.7.3. Cálculo de las recomendaciones

A continuación, se va a ver cómo se relacionan los dos factores que se han comentado en las dos secciones anteriores. Para que una noticia sea recomendada al usuario, ésta ha tenido que tener una similitud coseno alta junto a una noticia que haya tenido un resultado alto en la valoración dada por el usuario. Si en cambio ha coincidido con una noticia de mala valoración por parte del usuario, se decidirá no recomendarla ya que es probable que no sea de su agrado. Las noticias que se recomiendan son aquellas que han superado el threshold o umbral de recomendación. Por ejemplo, si sobre una valoración del usuario sobre 10, se establece el threshold con valor 5, únicamente se recomiendan aquellas noticias que el rating predicho haya superado el umbral establecido.

En resumen, el sistema de recomendación va a necesitar las valoraciones dadas por el usuario para poder trabajar. A partir de estas valoraciones, el sistema ya sabe cuáles son los gustos del usuario. A continuación, realiza el cálculo de similitud coseno entre los vectores de palabras de las diferentes noticias. Una determinada noticia se relacionará con la que haya obtenido un valor de similitud mayor. Una vez relacionada, hay que ver si dicha noticia va a pasar el threshold o no. Por tanto, al usuario se le van a recomendar aquellas noticias que sean similares a las que haya valorado previamente con una buena puntuación.

4.8. Procesamiento de la información

El objetivo ha consistido en que el usuario pudiese recibir un análisis más profundo de las noticias que recibía. Este apartado se puede considerar de mayor utilidad para usuarios con una mayor experiencia en este tema, ya que a la hora de configurar el procesamiento de la información se tienen que establecer una serie de opciones a seguir.

En primer lugar, se tienen que elegir las opciones previas al procesamiento. El usuario puede decidir si quiere eliminar o no las “stop words” o palabras vacías. Si el usuario decide no borrarlas, entonces no se realiza ningún tipo de filtrado sobre el documento en relación a este aspecto. En cambio si el usuario escoge eliminarlas, se va a realizar un filtrado del texto mediante el uso de la librería “Exude” [22]. Esta librería funciona de la siguiente manera:

1.- Se filtran las palabras duplicadas eliminándolas del texto.

2.- Se filtran las “stop words” del resultado obtenido del filtrado del paso anterior.

3.- Se filtra el resultado del apartado anterior utilizando el algoritmo de Porter, que trabaja removiendo los sufijos más comunes de diferentes palabras.

A continuación se le pide al usuario que escoja entre tres opciones bien diferenciadas:

1.- Lemmatization → Proceso que consiste en hallar el lema correspondiente de las palabras que forman la noticia. El lema es la forma que por convenio se acepta para representar todas las variables de una misma palabra. Por ejemplo, todas las conjugaciones de un verbo se definen por su infinitivo. Estudié, estudiaba, estudiasteis, etc. son representadas por el lema estudiar.

2.- Porter → Como se ha explicado previamente, corresponde a la eliminación de los sufijos englobando todas esas palabras sobre un mismo término o stem.

3.- No Stemming → Está opción corresponde a que el usuario no quiere que se realice ninguna transformación sobre el texto de entrada.

Una vez que el usuario ha elegido las técnicas previas al procesamiento, debe seleccionar sobre qué documentos quiere realizar la recuperación de información y qué datos quiere analizar. A la hora de la elección de los datos a analizar se le dan distintas opciones:

- *Snippet*
- Título
- Título y *Snippet*
- All Contents → Esta opción corresponde a la descarga de todo el documento de la noticia. Para ello se ha utilizado jsoup, que nos permite recuperar y manipular datos almacenados en los documentos HTML. El contenido descargado es almacenado en la base de datos, para que si posteriormente el usuario quiere volver a realizar un trabajo de procesamiento de la información sobre dichos datos no se tengan que volver a descargar.

Por último, el usuario puede elegir la técnica que desea a la hora de obtener las palabras claves del texto. En este caso, se permite al usuario la elección de las técnicas TF y TF-IDF, las cuales han sido explicadas previamente en la sección 4.7.

Una vez el usuario ya se ha decantado por las opciones a utilizar se produce el proceso de recuperación de información. Para ello, se ha utilizado la librería CoreNLP de Stanford [23], la cual fue diseñada fundamentalmente para el procesamiento de lenguaje natural. De este modo, la librería permite el reconocimiento de entidades dentro de una determinada noticia. A la hora de mostrarle al usuario el resultado obtenido por pantalla, se le muestran tanto las entidades obtenidas por la librería como las que disponen de un mayor TF-IDF o TF.

La librería CoreNLP sigue en crecimiento y cada vez son más los idiomas que domina para realizar el reconocimiento de las entidades. En las pruebas que se han realizado, se han utilizado el modelo que proporcionan para el inglés (su fichero .jar ocupa 1,04 GB) y para el castellano (359,8 MB).

```
List<CoreMap> sentences = document.get(CoreAnnotations.SentencesAnnotation.class);

for (CoreMap sentence : sentences) {
    // traversing the words in the current sentence
    // a CoreLabel is a CoreMap with additional token-specific methods
    for (CoreLabel token : sentence.get(CoreAnnotations.TokensAnnotation.class)) {
        // this is the text of the token
        String word = token.get(CoreAnnotations.TextAnnotation.class);
        // this is the POS tag of the token
        String pos = token.get(CoreAnnotations.PartOfSpeechAnnotation.class);
        // this is the NER label of the token
        String ne = token.get(CoreAnnotations.NamedEntityTagAnnotation.class);
```

Fig. 4.5.- Recuperación de la información con la librería CoreNLP de Stanford.

En la Fig. 4.5, se observa la forma en la que la librería CoreNLP trabaja para obtener la información relevante en la recuperación de información. En la primera variable almacena la palabra que se analiza. En la segunda, se guarda el tipo de palabra que es (nombre, adjetivo, adverbio,…) y en la última de ellas el tipo de entidad que se trata (organización, localización, persona, etc.).

4.9. Mapa de localización

A raíz de las localizaciones obtenidas en el punto anterior, se permite al usuario poder ubicarlas en un mapa y poder conocer los puntos exactos sobre los que hablan los documentos.

Únicamente se están mostrando las noticias que aparecen en el texto, pero otro aspecto que se podía haber tenido en cuenta corresponde con el hecho de haber podido ofrecerle al usuario cuál es la noticia más relevante de un documento. Para ello, se podía haber utilizando tanto el TF como el TF-IDF. Cualquiera de los dos nos podría haber ayudado a conocer cuál es la localización que más peso tiene en una noticia leída. Por ejemplo, en relación al TF, si tenemos una noticia en la que “Huesca” aparece tres veces y “Zaragoza” aparece una vez, se va a saber mediante dicha técnica que “Huesca” tiene mayor relevancia que “Zaragoza”, ya que aparece en tres ocasiones, mientras que para “Zaragoza” únicamente se encuentra una repetición. Por tanto, si en un futuro optamos por ir más allá y se decide poder ofrecerle al usuario cuál es la localización más relevante por noticia, se aplicará alguna técnica que nos permita conocer la relevancia de un determinado término en un documento.

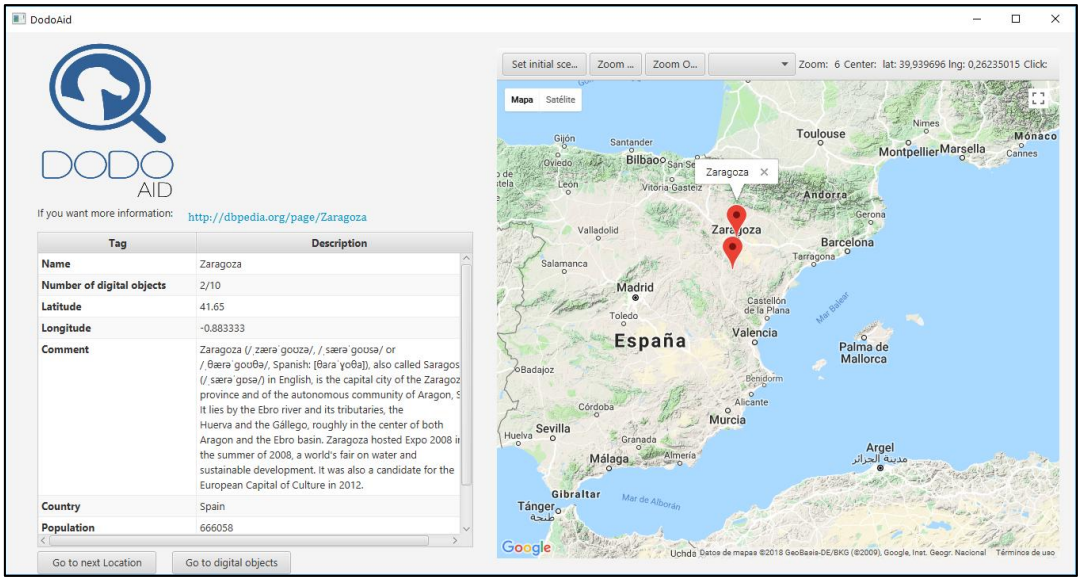

Fig. 4.6.- Mapa de localización sobre la búsqueda "Zaragoza" con el procesamiento realizado sobre el snippet.

En el mapa se muestran los marcadores de las ubicaciones que han sido detectadas en el procesamiento de la información. Al pulsar sobre uno de ellos, como se observa en la Fig. 4.6, se muestra información relevante sobre dicha localización, incluyendo un pequeño resumen acerca de ella o, por ejemplo, la población que dispone. Además, se permite al usuario acceder directamente a la página de la DBpedia [24] sobre la cual se obtiene dicha información.

Con la opción "Go to digital objects", se permite al usuario poder acceder a los documentos que contenían dicho término. En el ejemplo que se observa en la figura adjuntada, la localización "Zaragoza" ha sido detectada en 2/10 objetos digitales, por lo que al pulsar sobre dicha opción se verá la información relacionada con esos dos documentos.

Para poder devolver la información de las localizaciones, se ha usado el lenguaje SPARQL [25] y Apache Jena [26] como motor para realizar dichas consultas. A continuación, en la figura Fig. 4.7, se adjunta el código empleado para estas consultas.

```
"PREFIX geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>"+
"PREFIX dbo: <http://dbpedia.org/ontology/>" +
"PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>" +
"SELECT * WHERE {" +
"  ?s a dbo:Place ." +
"  ?s geo:lat ?lat ." +
"  ?s geo:long ?long ." +
"  ?s rdfs:comment ?com ." +
"  ?s dbo:country ?count ." +
"  ?s dbo:populationTotal ?pop ." +
" FILTER langMatches(lang(?com),'en')" +
"}" );
```

Fig. 4.7.- Consulta SPARQL para obtener la información sobre las localizaciones.

La consulta la procesa el SPARQL endpoint correspondiente y se devuelven los atributos que se consideran de mayor relevancia (comentario, población, país, etc.). Por último, se decide que los resultados sean filtrados y devueltos para el idioma inglés.

4.10. Configuración

Cuando el usuario se conecta por primera vez a la aplicación se le genera una configuración por defecto.

En cualquier momento va a poder modificar el nombre de usuario que asignó junto a su primera ejecución. En el futuro, cuando haya más usuarios y se realice el filtrado colaborativo, se comprobará que el nombre de usuario no ha sido asignado a otro previamente.

En segundo lugar, se le permite la modificación de los datos a mostrar sobre las noticias. De esta manera, se evita que el usuario tenga que configurar las columnas cada vez que realice una búsqueda, ya que realizando el cambio desde la pestaña de configuraciones se va a mantener dicho formato de salida para todas las veces que entre al sistema. Por defecto se muestra la información correspondiente al término de búsqueda, el título, el link y si ha sido visualizada previamente o no.

Por último, a la hora de realizar la recuperación de información, el usuario debe elegir unas opciones previas. Por defecto las opciones que se le establecen al usuario son: “Do not remove stop words” y “Porter” para el procesamiento previo; “All” haciendo referencia a que el procesado se realiza sobre todos los documentos almacenados en la base de datos; “All contents”, significando que se descarga el contenido de todos los ficheros y la técnica “TF-IDF” para obtener las palabras de mayor relevancia. En cualquier momento puede modificar dichas opciones y definirlas según sus necesidades.

5. Tecnologías usadas

A continuación se va a realizar un resumen de las tecnologías empleadas para la programación de las funcionalidades explicadas en el punto 4.

- Se ha decidido usar el lenguaje de programación Java (http://www.oracle.com/ technetwork/java/javase/downloads) debido a que ha sido con el que más se ha trabajado durante toda la carrera, por lo que se tiene un mayor conocimiento de él sobre cualquier otro. Se ha usado la última versión disponible (JDK 8u171).

- A la hora de la programación de la interfaz gráfica se tuvieron en cuenta dos opciones: Swing y JavaFX. Previamente no se había trabajado con ninguna de las dos, por lo que ambas suponían un reto. A partir de las declaraciones de Oracle acerca de que el futuro de la programación de interfaces gráficas está en JavaFX, se decide ver qué nos proporciona que Swing no haga. Se observa que JavaFX ha mejorado en algunos aspectos a Swing, como por ejemplo permitiendo el uso de hojas de estilo para realizar la apariencia de la interfaz. Aunque para el trabajo que se ha realizado hubiese servido el uso de Swing, se decide usar JavaFX por el hecho de que Oracle vaya a seguir trabajando sobre ella, dejando más de lado Swing.

- A la hora de escoger el sistema gestor de la base de datos también se han tenido en mente varias posibilidades, entre las que destacan MySQL y SQLite, ya que es con las que se había trabajado con más énfasis en anteriores ocasiones. Ambas utilizan el lenguaje SQL, por lo que hay que ver qué nos proporciona cada una. MySQL es más usada en la actualidad y nos proporciona mejores prestaciones, como por ejemplo un mayor espacio de almacenamiento o una mayor seguridad a la hora de acceder a los datos de la base de datos. En cambio, SQLite nos permite asociar de manera sencilla a la aplicación la base de datos y utilizarla de una forma sencilla, eficaz, potente y rápida. SQLite realiza operaciones de forma eficiente y es más rápido que MySQL. SQLite es independiente, por lo que a la hora de trabajar simplemente se realizan llamadas a subrutinas o funciones de las propias librerías de SQLite, lo que reduce la latencia en el acceso a la base de datos, la cual es almacenada como un fichero en un único ordenador. Primando la velocidad de acceso a los datos, se decido usar SQLite, ya que para el propósito actual de nuestro sistema es más que suficiente.

- Como motor de búsqueda, nuestro prototipo accede a Google a partir de utilizar JSON Custom Search API, proporcionado por el mismo Google (https://developers. google.com/custom-search/). Al ser utilizada la versión gratuita, se encuentran limitaciones en su uso, ya que se pueden realizar cien consultas como máximo al día. Una dificultad encontrada al usar este API, es que aunque se permita el filtrado de las búsquedas por fecha, no se puede acceder a la fecha de los documentos debido a que no hay ningún metadato que nos proporcione dicha información. A la hora de elegir el motor de búsqueda se han tenido presentes otras opciones como Bing o Yahoo.

- Para la representación de las localizaciones en un mapa se ha usado GMapsFX (https://rterp.github.io/GMapsFX/), el cual es un wrapper de Java sobre Google Map's JavaScript API (https://developers.google.com/maps) que permite trabajar sobre un escenario JavaFX sin la necesidad de utilizar JavaScript para la conexión con el mapa de Google. Se han buscado otras posibilidades a la hora de la representación del mapa, como por ejemplo OpenStreetMap (https://www.openstreetmap.org/), la cual en avances futuros podría sustituir a GMapsFX, pero para la necesidad que se tenía actualmente y la facilidad con la que GMapsFX permite trabajar con JavaFX, se ha acabado decantando por su uso.
- Se ha utilizado DBpedia (http://es.dbpedia.org/) para acceder a información sobre las localizaciones que se obtenían en el procesamiento de la información.
- Para obtener la información de DBpedia se ha utilizado el lenguaje SPARQL (http://es.dbpedia.org/sparql).
- Como soporte para la ejecución de las consultas SPARQL se ha utilizado Jena (https://jena.apache.org/) y su motor de consultas ARQ, el cual permite la ejecución de consultas SPARQL.
- Para el procesamiento de textos se ha utilizado Stanford CoreNLP (https://stanfordnlp.github.io/CoreNLP/). Se han desarrollado pruebas tanto para el idioma castellano como el inglés. Para el procesamiento previo de los textos y, en concreto, para la eliminación de stopwords (palabras no relevantes) se ha utilizado Exude (https://exude. herokuapp.com, https://github.com/uttesh/exude)
- Para la recomendación se ha usado Apache Mahout (https://mahout.apache.org/), que es una librería de Java para el aprendizaje automático que incorpora funcionalidades para sistemas de recomendación.
- Para la clasificación se han utilizado los algoritmos disponibles en Weka (https://www.cs.waikato.ac.nz/ml/weka/). Además, aprovechando la clase de Weka "TextDirectoryLoader" se permite al usuario cargar *datasets*, de manera bien estructurada, para poder entrenarlo y posteriormente cargar nuevos *datasets* para clasificarlos.

6. Conclusiones y Trabajo Futuro

En este proyecto se ha descrito el desarrollo de DodoAid, una herramienta que trata de ayudar al usuario en el seguimiento de información de interés sobre documentos web. Combina el uso de técnicas relacionadas con el campo de la recuperación de información, minería y procesamiento de textos y los sistemas de recomendación. En el Anexo C, se puede apreciar una evaluación preliminar que se ha realizado sobre el sistema de recomendación.

Es un trabajo que en el futuro podría recibir extensiones y mejoras por parte del grupo COS2MOS, siendo una potencial base para investigaciones futuras. También podría ser utilizado en el contexto académico en el caso de que pudiese ser de utilidad para la enseñanza de determinados conceptos, bien sea utilizando directamente la interfaz desarrollada o modificando código y añadiendo nuevas funcionalidades.

Como resultado de este proyecto, se ha podido contribuir en la Conferencia Española sobre Recuperación de Información (CERI 2018) incluyendo un artículo corto [27].

En relación a los esfuerzos para el desarrollo del proyecto, se han superado las horas que se debían invertir, ya que prácticamente todos los conceptos han sido nuevos. Estos esfuerzos invertidos se van a mostrar en distintos grupos que engloban todo el trabajo realizado.

- Elección del tema, investigación inicial y redacción de la propuesta del proyecto. (10 horas aprox.)
- Investigación sobre el tema escogido. (20 h. aprox.)
- Aprendizaje de las tecnologías usadas en la elaboración del proyecto. (65 h. aprox.)
- Programación del sistema. (260 h. aprox.)
- Redacción de la memoria final (50 h. aprox.)
- Total: 405 horas invertidas aproximadamente.

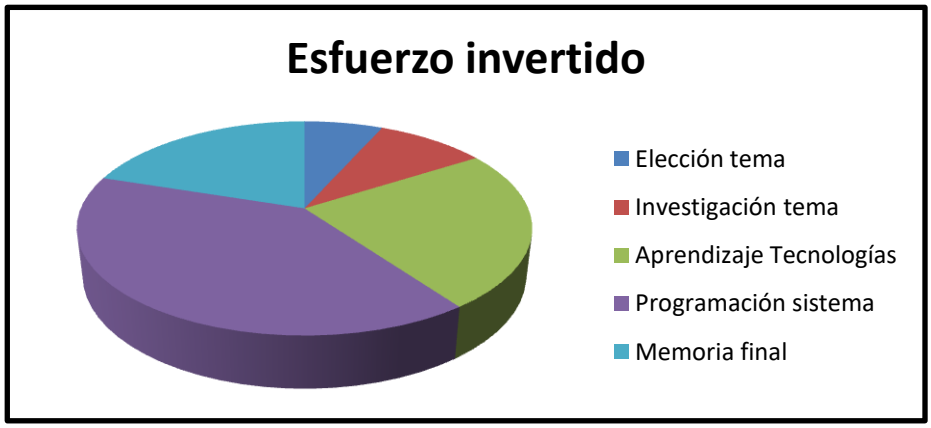

Fig. 6.1.- Diagrama circular de los esfuerzos invertidos.

En primer lugar, en la Fig. 6.1, se adjunta una gráfica circular con la representación del esfuerzo invertido para la elaboración de este proyecto.

En la tabla Tabla 6.1 se adjunta el diagrama de Gantt del desarrollo de este proyecto. Cada recuadro de cada mes corresponde a una semana de trabajo en el área especificada.

La tarea que ha llevado un mayor tiempo ha sido la programación del sistema, ya que prácticamente entre dos y tres meses han sido invertidos en la programación de la aplicación. La segunda tarea que ha llevado una mayor carga de trabajo ha sido la relacionada al aprendizaje de las tecnologías a utilizar. Al ser prácticamente todo nuevo, se ha tenido que estudiar cómo funcionan las diferentes tecnologías y qué nos podrían aportar.

Desde que se decidió el tema sobre el que trabajar, se produjo un periodo de inactividad que duró hasta el comienzo del mes de febrero, con el que se retomó la actividad del proyecto.

Actividades	**Noviembre**				**Febrero**				**Marzo**				**Abril**				**Mayo**				**Junio**			
	1	**2**	**3**	**4**	**1**	**2**	**3**	**4**	**1**	**2**	**3**	**4**	**1**	**2**	**3**	**4**	**1**	**2**	**3**	**4**	**1**	**2**	**3**	**4**
Elaboración propuesta																								
Investigación sobre el tema																								
Aprendizaje JavaFX																								
Base de Datos																								
Programación del sistema																								
Aprendizaje técnicas IR																								
Programación necesidades IR																								
Aprendizaje Apache Mahout																								
Sistema de Recomendación																								
Evaluación Sist. Recomendación																								
Redacción de la memoria final																								

Tabla 6.1.- Diagrama de Gantt de los esfuerzos invertidos.

En relación a la opinión personal acerca de la elaboración de este proyecto, se considera que ha sido de gran interés y de gran utilidad para la formación académica en temas que me gustan. Al inicio del proyecto, no se tenía apenas conocimiento del tema abordado, por lo que prácticamente todo lo que se ha hecho en este proyecto se ha tenido que aprender. Únicamente, el lenguaje de programación Java y el sistema gestor de bases de datos SQLite se dominaban en mayor medida. Por este motivo, las horas invertidas en el proyecto han superado las establecidas para su realización, ya que el hecho de aprender nuevas tecnologías a usar ha hecho que se tuviera que invertir bastante tiempo en el estudio de las tecnologías y en saber cómo aplicarlas en la programación del sistema.

A pesar de todo esto, la valoración personal sobre este proyecto es muy positiva, ya que se ha invertido el tiempo en aprender aspectos que no se conocían sobre temas que son de mi agrado.

Para finalizar, se van a comentar todos los aspectos que se han pensado para una futura evolución y mejora del sistema.

En primer lugar, la interfaz que se ha generado es muy simple y su objetivo ha sido que contuviese lo mínimo que necesita el usuario. Para el futuro, cabe la posibilidad de mejorarla en relación a los aspectos visuales, ya que únicamente se ha centrado en que realice las funcionalidades que se quieren aportar al usuario, sin darle relevancia al diseño. En relación al usuario, también se espera realizar una mejora en cuanto a la usabilidad, modificando todos aquellos aspectos que le supongan una dificultad en el manejo del sistema.

Siguiendo con el usuario, se espera poder realizar una evaluación de la satisfacción con la herramienta con el fin de poder añadir nuevas mejoras y poder adaptar en mayor medida la herramienta a sus gustos y necesidades.

En segundo lugar, se va a seguir tratando de mejorar el sistema de recomendación basado en contenido con el fin de disminuir el porcentaje de error que se obtiene tratando de mejorar el algoritmo de recomendación con el que se trabaja. Relacionado con esto, uno de los principales objetivos de futuro consiste en añadir la posibilidad de realizar un filtrado colaborativo. Esto, en un principio, supondría tener que modificar la base de datos para poder almacenar información sobre más de un usuario. Como desde un principio se tenía en mente la ampliación del sistema a un filtrado colaborativo, la información es almacenada a partir del identificador del usuario y todas las restricciones ya han sido tenidas en cuenta para que dos usuarios no puedan tener mismo identificador, permitiendo así la posibilidad de almacenar información de más de un usuario. Otro aspecto que habría que estudiar en relación a esto sería el hecho de cambiar el gestor de base de datos a usar.

En tercer lugar, existen diversas funcionalidades dentro del sistema que podrían mejorarse. Este es el caso de la exportación de los datos. Actualmente, sólo se pueden exportar los datos en un único formato, y el usuario no puede decidir sobre qué datos quiere exportar. Las mejoras en relación a este apartado consisten en permitir que el usuario pueda decidir qué datos quiere exportar sobre las noticias y que pueda guardarlos en el formato que quiera.

Siguiendo con las mejoras de las funcionalidades que se ofrecen al usuario, otro cambio consistiría en modificar la forma de mostrar las localizaciones, pasando a usar OpenStreeMap.

Por último, el cambio más ambicioso y menos trivial consistiría en permitir que el sistema trabajase sobre otros objetos digitales. Por el momento, se trabaja únicamente sobre ficheros de textos, pero en un futuro se espera que se pueda dar soporte a otros objetos digitales, como serían las imágenes o los vídeos.

Referencias

[1] Big Data to Action 2018 - Un evento de MSMK. Big Data to Action 2018 - "The Big Jump" |Un evento de MSMK. [online] Available at: http://bigdatatoaction.com/ [Accessed 4 Jun. 2018].

[2] MSMK - Madrid School Marketing. Luis Ortiz | Madrid School of Marketing. [online] Available at: https://madridschoolofmarketing.es/msmk/faculty/luis-ortiz [Accessed 4 Jun. 2018].

[3] MSMK - Madrid School Marketing. Madrid School of Marketing | Escuela de Marketing y Negocios. [online] Available at: https://madridschoolofmarketing.es/ [Accessed 4 Jun. 2018].

[4] Ceri2018.unizar.es. (2018). CERI 2018. 5th Spanish Conference on Information Retrieval 25-27th June 2018, Zaragoza, Spain. [online] Available at: http://ceri2018.unizar.es/ [Accessed 26 Jun. 2018].

[5] Jsoup. [online] Available at: https://jsoup.org/ [Accessed 21 May. 2018]

[6] Research-paper recommender systems: a literature survey", Beel, J., Gipp, B., Langer, S. et al. International Journal on Digital Libraries, November 2016, Volume 17, Issue 4, pp 305–338.

[7] Yonata Andrelo Asikin and Wolfgang Wörndl. 2014. Stories Around You: Locationbased Serendipitous Recommendation of News Articles. In Second International Workshop on News Recommendation and Analytics (NRA), Vol. 1181. CEUR Workshop Proceedings, 1–8

[8] Florent Garcin, Christos Dimitrakakis, and Boi Faltings. 2013. Personalized News Recommendation with Context Trees. In Seventh ACM Conference on Recommender Systems (RecSys). ACM, 105–112.

[9] Florent Garcin and Boi Faltings. 2013. PEN recsys: A Personalized News Recommender Systems Framework. In International News Recommender Systems Workshop (NRS). ACM, 3–9.

[10] Florent Garcin, Boi Faltings, Olivier Donatsch, Ayar Alazzawi, Christophe Bruttin, and Amr Huber. 2014. Offline and Online Evaluation of News Recommender Systems at swissinfo.ch. In Eighth ACM Conference on Recommender Systems (RecSys). ACM, 169–176.

[11] Joseph A. Konstan, Bradley N. Miller, David Maltz, Jonathan L. Herlocker, Lee R. Gordon, and John Riedl. 1997. GroupLens: Applying Collaborative Filtering to Usenet News. Commun. ACM 40, 3 (March 1997), 77–87.

[12] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. 1994. GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In ACM Conference on Computer Supported Cooperative Work (CSCW). ACM, 175–186.

[13] H.J. Lee and Sung Joo Park. 2007. MONERS: A news recommender for the mobile web. Expert Systems with Applications 32, 1 (January 2007), 143 – 150.

[14] Alisa Sotsenko, Marc Jansen, and Marcelo Milrad. 2014. Using a Rich Context Model for a News Recommender System for Mobile Users. In 22nd Conference on User Modeling, Adaptation, and Personalization (UMAP), Vol. 1181. CEUR Workshop Proceedings, 1–4.

[15] Kam Fung Yeung and Yanyan Yang. 2010. A Proactive Personalized Mobile News Recommendation System. In Developments in E-systems Engineering (DESE). IEEE, 207–212.

[16] SQLite. [online] Available at: https://www.sqlite.org/index.html [Accessed 5 May. 2018]

[17] Google Custom Search JSON API. [online] Available at: https://developers.google.com/custom-search/json-api/v1/overview [Accessed 20 Feb. 2018]

[18] Eibe Frank, Mark A. Hall, and Ian H. Witten (2016). The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques", Morgan Kaufmann, Fourth Edition, 2016.

[19] Weka. J48. [online] Available at: http://weka.sourceforge.net/doc.dev/weka/classifiers/trees/J48.html [Accessed 23 Jun. 2018].

[20] Apache Mahout. [online] Available at: https://mahout.apache.org/ [Accesed 22 May. 2018]

[21] Adomavicius, Gediminas, and Alexander Tuzhilin. 2005. "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.

[22] GitHub. uttesh/exude. [online] Available at: https://github.com/uttesh/exude [Accessed 13 Jun. 2018].

[23] Manning, Christopher D., Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 55-60.

[24] DBpedia [online] Available at: http://es.dbpedia.org/ [Accessed 23May. 2018]

[25] DuCharme, Bob. 2013. *Learning SPARQL: querying and updating with SPARQL 1.1*. " O'Reilly Media, Inc.".

[26] Apache Jena. [online] Available at: https://jena.apache.org/ [Accesed 23 May. 2018]

[27] "Towards the Development of a Tool to Keep Track of Interesting Information in a Sea of Digital Documents", Sergio Ilarri and Guillermo Azón, Fifth Spanish Conference on Information Retrieval (CERI 2018), Zaragoza, June 2018, ACM Press, ISBN 978-1-4503-6543-7/18/06, 4 pages, 2018. DOI: 10.1145/3230599.3230610.