Anexos

ÍNDICE

Lista de figuras

Lista de tablas

A. Diagramas de secuencia

A continuación se van a facilitar los diagramas de secuencia correspondientes a cada una de las funcionalidades que dispone el sistema generado.

1.- Conectarse a la aplicación

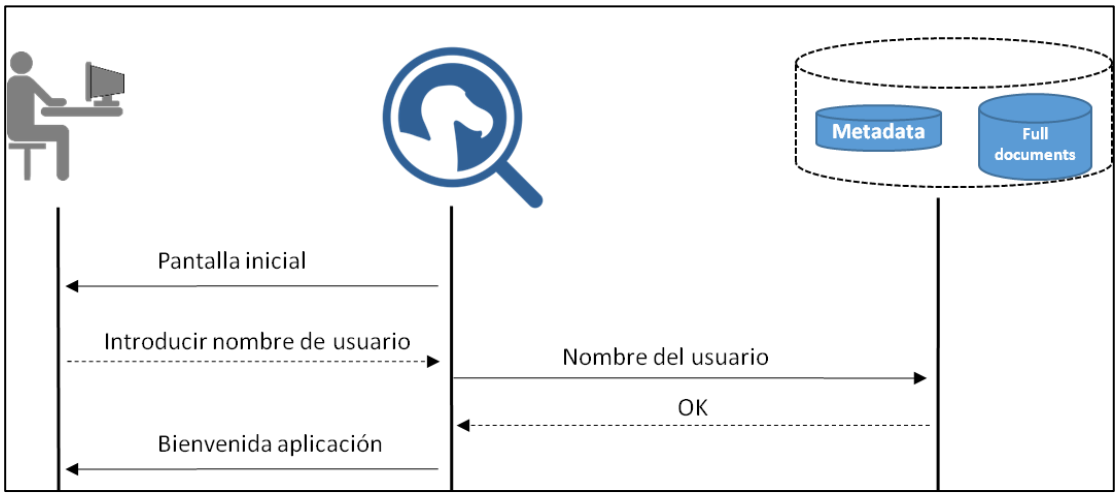

Fig. A.1.- Conexión a la aplicación.

En primer lugar, el usuario accede a la aplicación y se le va a mostrar una pantalla en la que escribir su nombre. Una vez lo haya escrito, siempre que acceda a la aplicación se le reconocerá con dicho nombre.

2.- Realizar una búsqueda

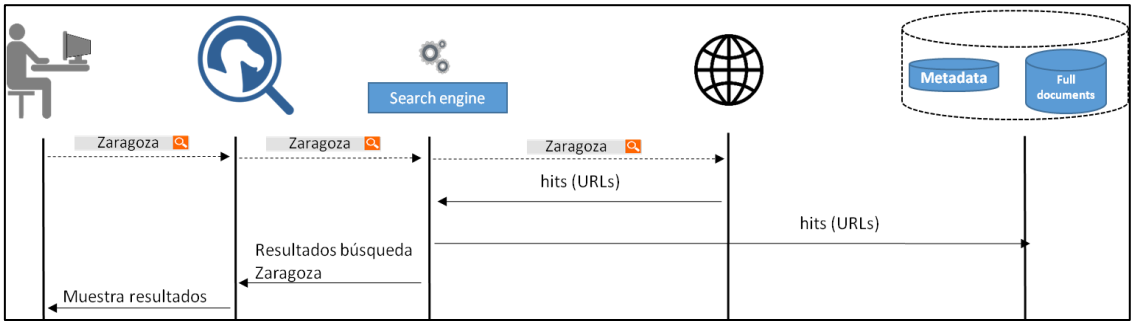

Fig. A.2.- Búsqueda del término "Zaragoza" en la aplicación.

A la hora de realizar una búsqueda, el usuario debe escribir en el buscador que se proporciona el término que le interesa. De este modo, la aplicación se conecta con el motor de búsqueda y se le devuelven objetos digitales, situados en la Web, relacionados a dicho término.

3.- Borrar un objeto digital

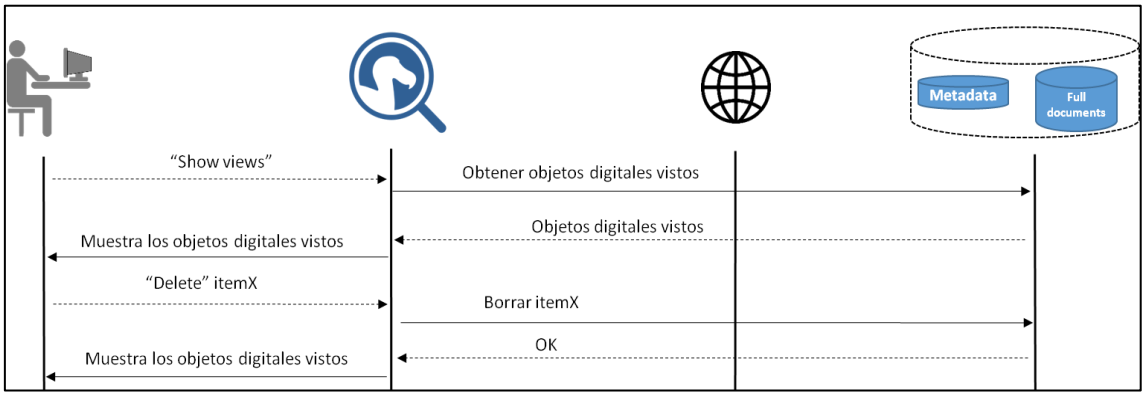

Fig. A.3.- Borrar un objeto digital.

Cuando el usuario quiera borrar un determinado objeto digital, se le permitirá hacerlo de dos formas diferentes. En la Fig. A.3, únicamente se adjunta una de ellas. En este caso, el usuario debe acceder a la opción "Show views", dónde se muestran todos los objetos digitales vistos por el usuario. Haciendo click derecho sobre uno de ellos, le aparece la opción de borrarlo. El segundo caso es realizar dicha acción desde la pantalla en la que se muestran los objetos digitales relacionados al término buscado. Haciendo click derecho en esta última pantalla sobre un ítem, el usuario también recibe la opción de poder borrarlo.

4.- Añadir a favoritos

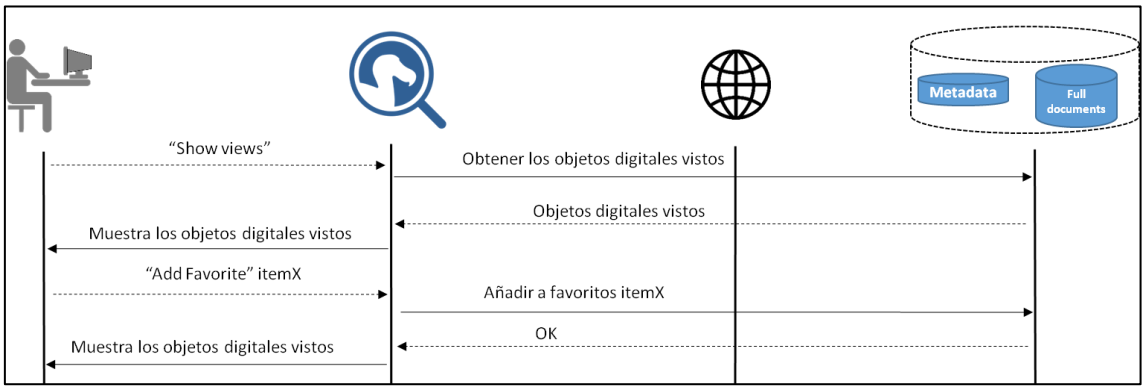

Fig. A.4.- Añadir a favoritos un objeto digital.

Cuando un usuario quiera añadir un objeto digital a la lista de favoritos debe seguir el mismo proceso realizado para borrarlo. Al hacer click derecho sobre uno de ellos, se le permite al usuario poder añadirlo/quitarlo de favoritos. Al igual que el anterior punto, también puede realizarse esta tarea desde los dos casos comentados.

5.- Valorar un objeto digital

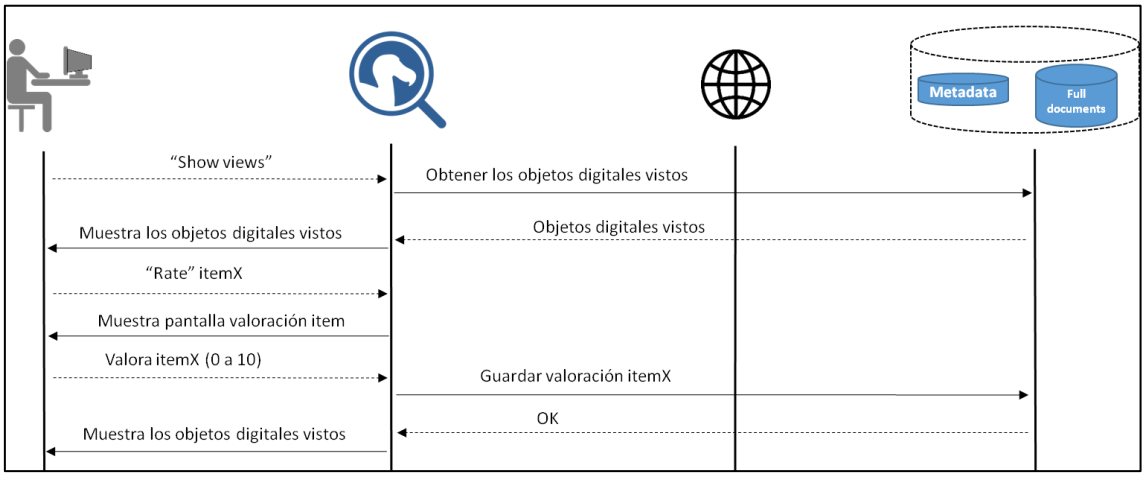

Fig. A.5.- Valoración de un objeto digital.

A la hora de valorar un objeto digital el usuario debe seguir el mismo proceso llevado a cabo tanto para borrar como para añadir a favoritos. En este caso, al hacer click derecho sobre un determinado objeto digital y elegir la opción "Rate ítem", el usuario tiene la posibilidad de valorarlo con un rango del 0 al 10. Inicialmente, todos los ítems estarán valorados como "NE" (Not Evaluated).

Si un usuario quiere modificar una valoración, debe seguir el mismo proceso que seguiría para valorarla por primera vez.

6.- Añadir a vistos

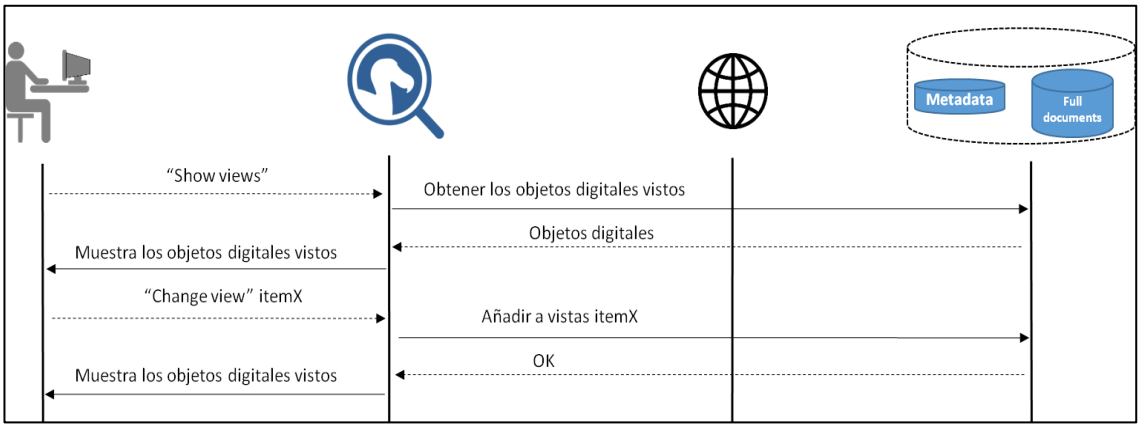

Fig. A.6.- Añadir un objeto digital a vistas.

Para añadir un determinado objeto digital a la lista de vistos, el usuario puede hacerlo de dos maneras diferentes. La primera de ellas es la que se puede ver en la Fig. A.6, que corresponde al caso de cambiar el ítem a vistos a partir de hacer click derecho sobre él y elegir la opción "Change viewed". La segunda opción corresponde a visitar la URL que se proporciona. De este modo, al usuario se le abre en el navegador web y se considera directamente como objeto digital visualizado.

7.- Acceder a los objetos digitales vistos

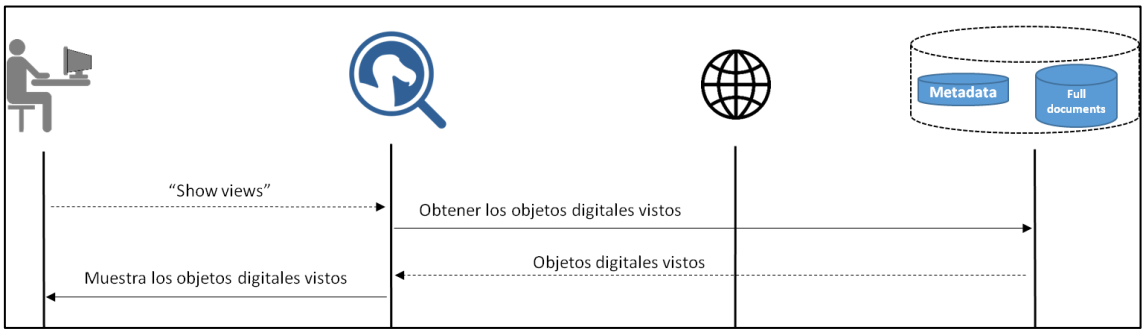

Fig. A.7.- Acceder a objetos digitales vistos.

Para acceder a los objetos digitales que ha visitado, el usuario debe acceder a la pestaña "Show views". En dicha pantalla se muestra, a partir de la configuración establecida por el propio usuario, los ítems que ha visitado.

8.- Visualizar objetos digitales favoritos

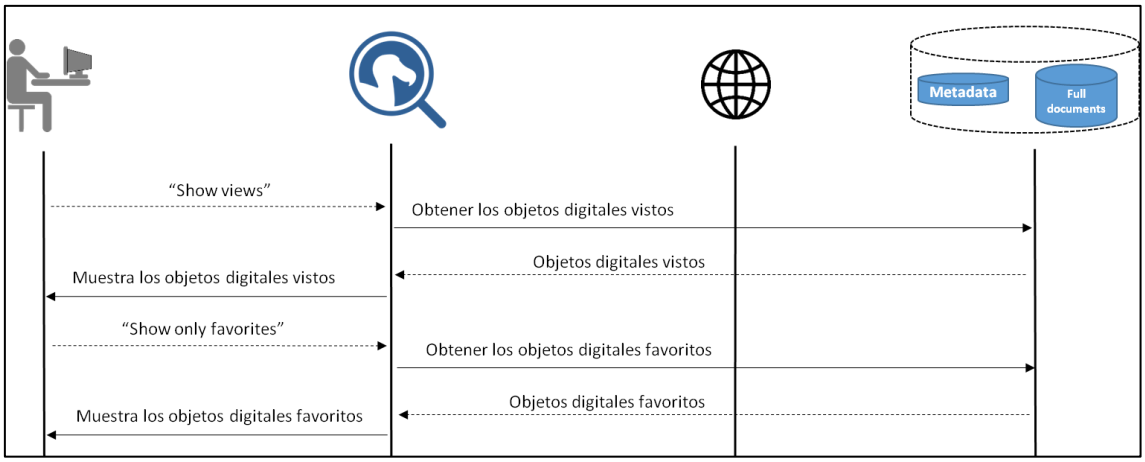

Fig. A.8.- Ver los objetos digitales favoritos.

Para acceder a los objetos digitales que han sido marcados como favoritos por el usuario, debe acceder a la pestaña "Show views" y, dentro de ésta, marcar la opción "Show only favorites".

9.- Visualizar objetos digitales no vistos

Para acceder a los objetos digitales que no han sido vistos por el usuario, debe acceder a la pestaña "Show views" y, dentro de ésta, marcar la opción "Show only not viewed".

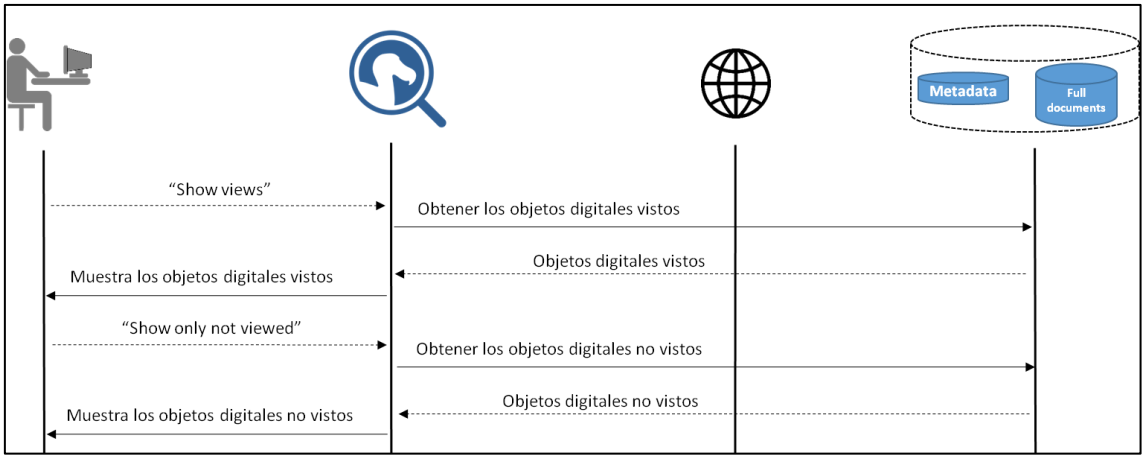

Fig. A.9.- Mostrar los objetos digitales que no han sido visualizados por el usuario.

10.- Visualizar semantic views

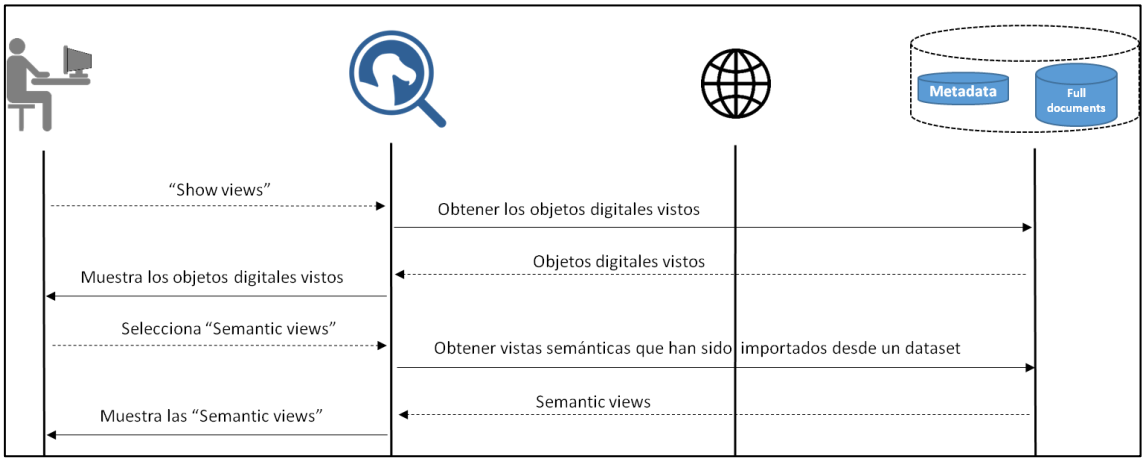

Fig. A.10.- Visualizar semantic views.

Las "semantic views" corresponden a las categorías en las que puede clasificarse un determinado ítem a partir del clasificador. Para poder ver las "semantic views" el usuario habrá tenido que importar un *dataset* previamente (léase el punto 14 para saber cómo hacerlo) y entrenado el clasificador (léase el punto 15). Una vez haya completado ambas tareas, el usuario debe acceder a la pestaña "Show Views" y seleccionar la opción "Semantic Views". En el caso de que no se haya entrenado el clasificador, al marcar dicha opción se muestra una tabla sin contenido.

11.- Configuración de las columnas

Una columna corresponde con una característica sobre un objeto digital. A la hora de seleccionar las columnas a mostrar, el usuario puede elegir entre varias opciones: "Snippet", "Title", "Date and Time", "Search Term", "Rate", "Favorite", "Link" y "Viewed".

Para acceder a la configuración de la columna, el usuario debe ir a la pestaña “Show Views” y posteriormente seleccionar la opción “Configure Columns”. Si el usuario modifica la configuración y sale de la pestaña, los cambios no se guardan y la próxima vez que acceda a ver los objetos digitales vistos, la configuración de las columnas será la inicial. Si el usuario quiere que se guarde la configuración para futuras tareas, debe seguir la explicación del punto 21.

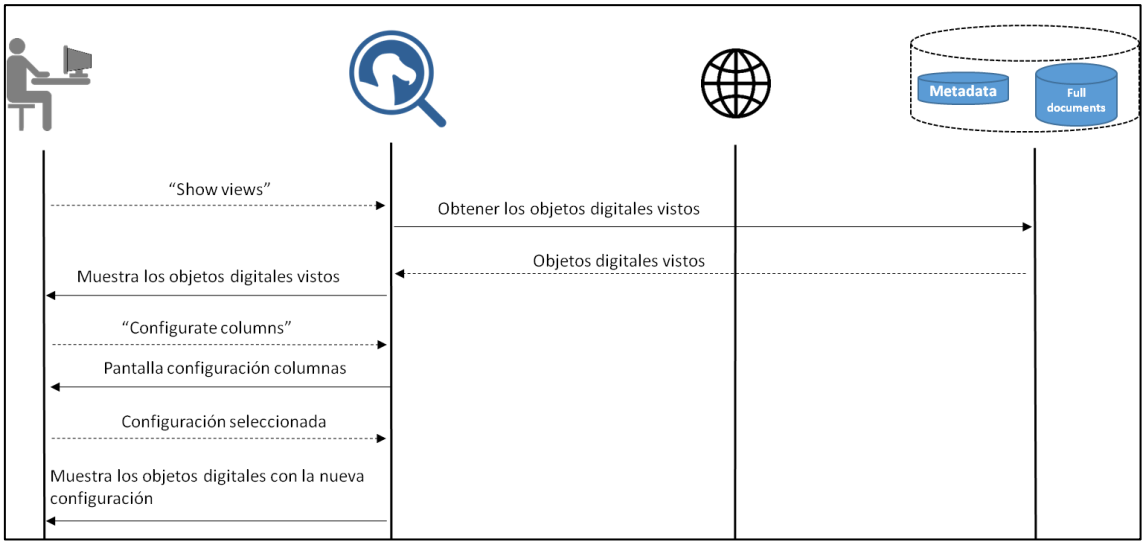

Fig. A.11.- Configuración de las columnas a mostrar.

12.- Obtener recomendaciones

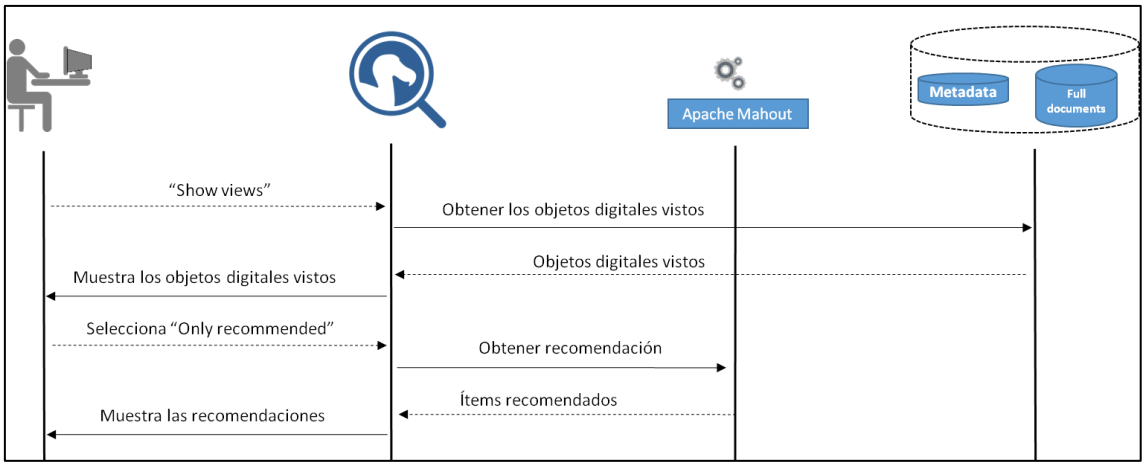

Fig. A.12.- Obtener recomendaciones.

Cuando un usuario quiere recibir recomendaciones sobre los objetos digitales que podrían ser de su interés, debe acceder a la pestaña de “Show Views” y seleccionar la opción “Only Recommended”. De este modo, se muestran los objetos digitales que pueden resultar de interés para el usuario.

13.- Importar objetos digitales

Al usuario se le permite importar los objetos digitales que desee para así poder almacenarlos junto a los obtenidos desde la aplicación. Para poder acceder a esta funcionalidad, debe ir a la pestaña “Import data”.

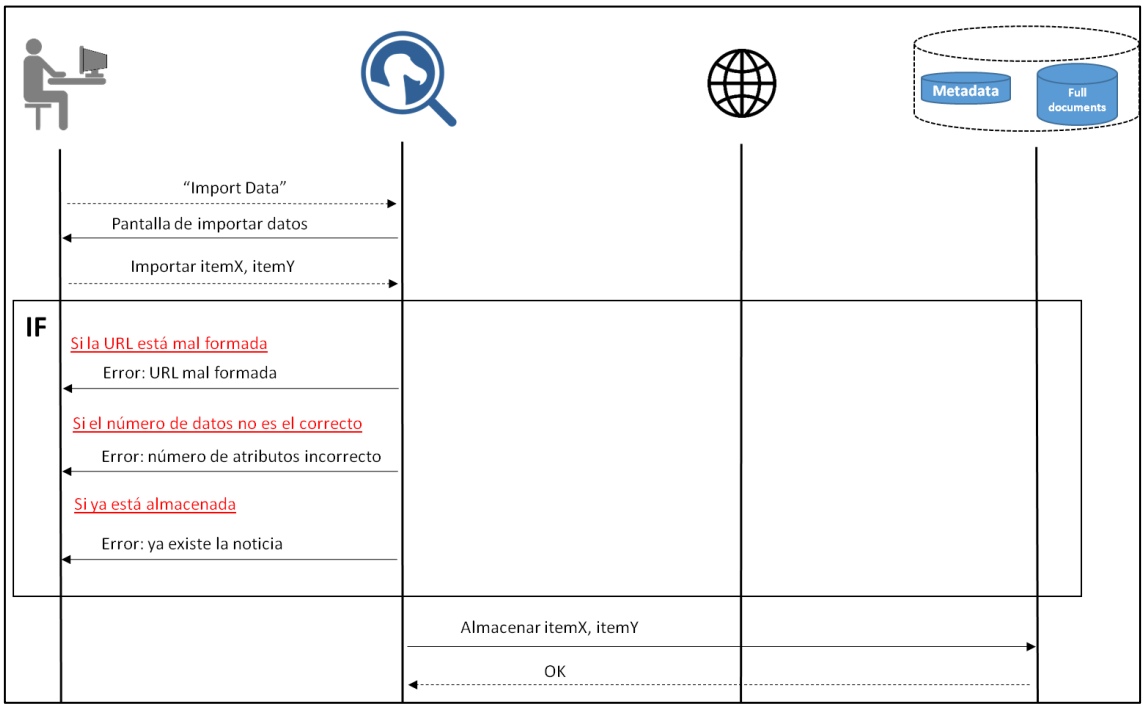

Fig. A.13.- Importar objetos digitales.

14.- Importar datasets

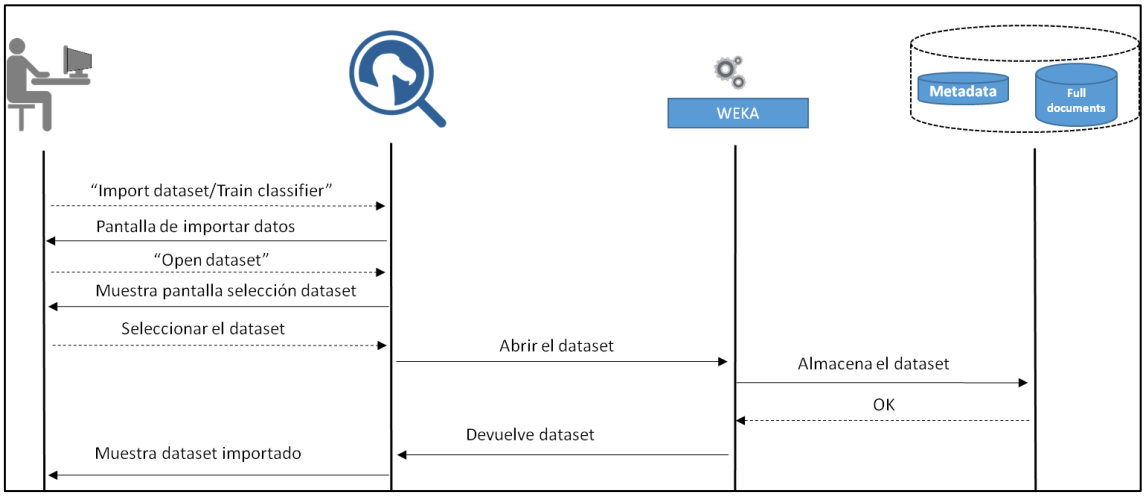

Fig. A.14.- Importar dataset.

Al igual que se pueden insertar los objetos digitales de manera manual, se permite al usuario poder añadirlos a partir de *datasets* bien estructurados. Para ello, el usuario debe acceder a la pestaña “Import dataset/Train Classifier” y seleccionar la opción “Open dataset”. Si el *dataset* que se pretende importar no sigue una estructura de carpetas correcta, no se permite importarlo.

15.- Entrenar clasificador

A la hora de entrenar un clasificador, el usuario debe acceder a la pestaña "Import dataset/Train Classifier". Para poder entrenarlo, es necesario que previamente se haya importado un *dataset* (léase el punto 14 para saber cómo hacerlo) Una vez que se disponga un *dataset* importado en la aplicación, el usuario debe seleccionar la opción "Train classifier" para comenzar con el entrenamiento.

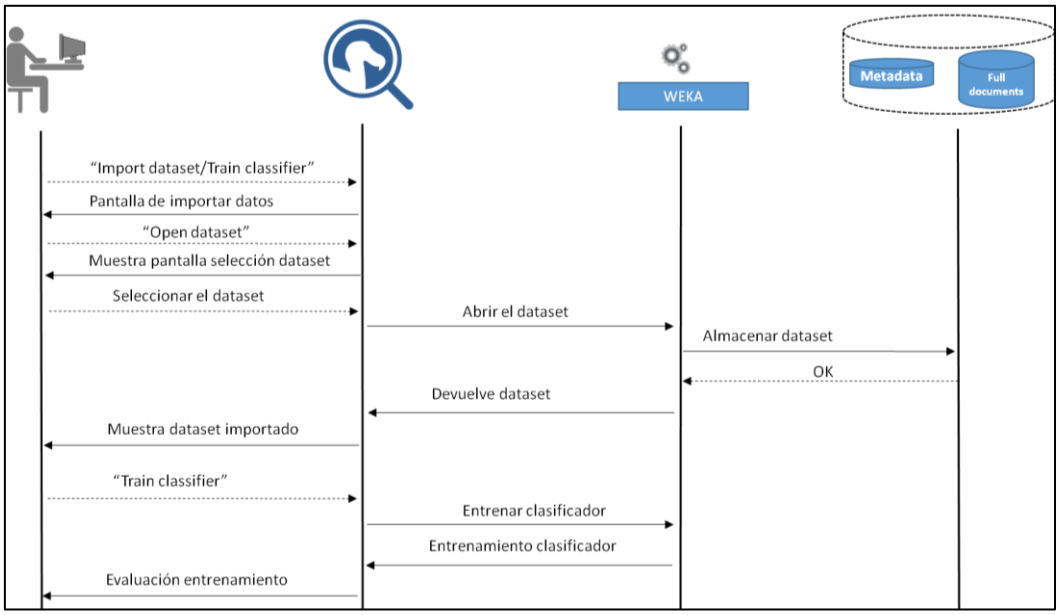

Fig. A.15.- Entrenar clasificador.

16.- Clasificar dataset

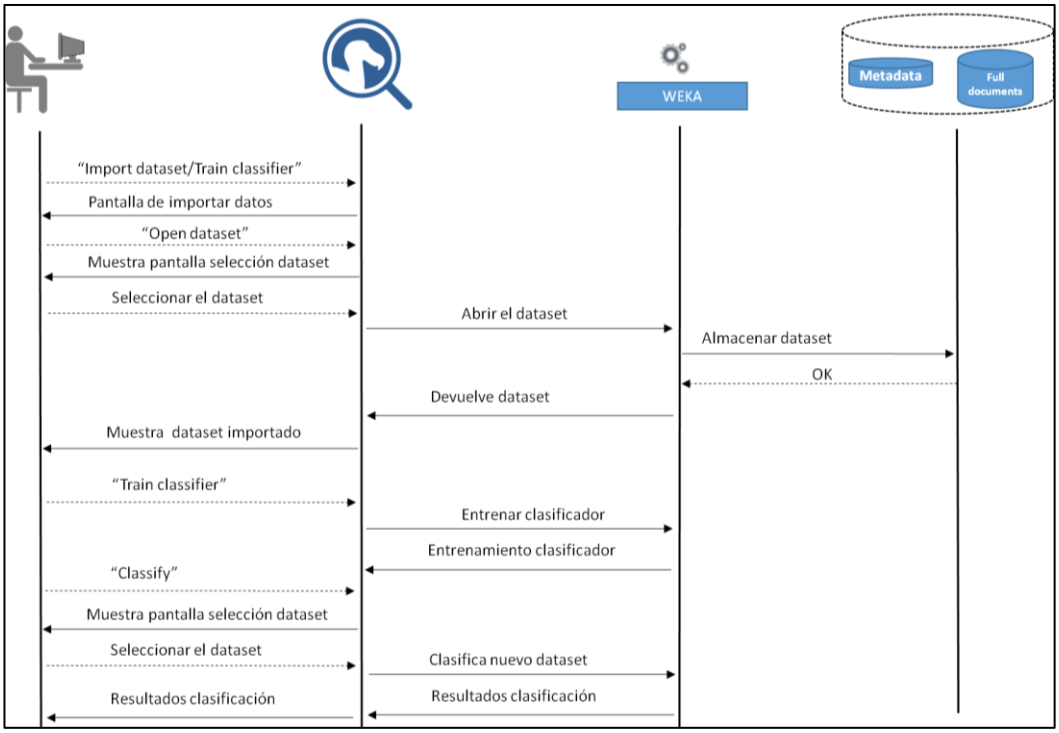

Fig. A.16.- Clasificar dataset.

Cuando el usuario quiera clasificar un objeto digital, se debe acceder a la pestaña “Import dataset/Train Classifier” y seleccionar la opción “Classify”. Previamente, el usuario ha tenido que cargar un *dataset* en la aplicación (léase el punto 14) y entrenar el clasificador (léase el punto 15). Una vez estos dos pasos hayan sido concluidos, el usuario deberá elegir un nuevo *dataset* para proceder a clasificarlo.

17.- Transferir dataset a objetos digitales vistos

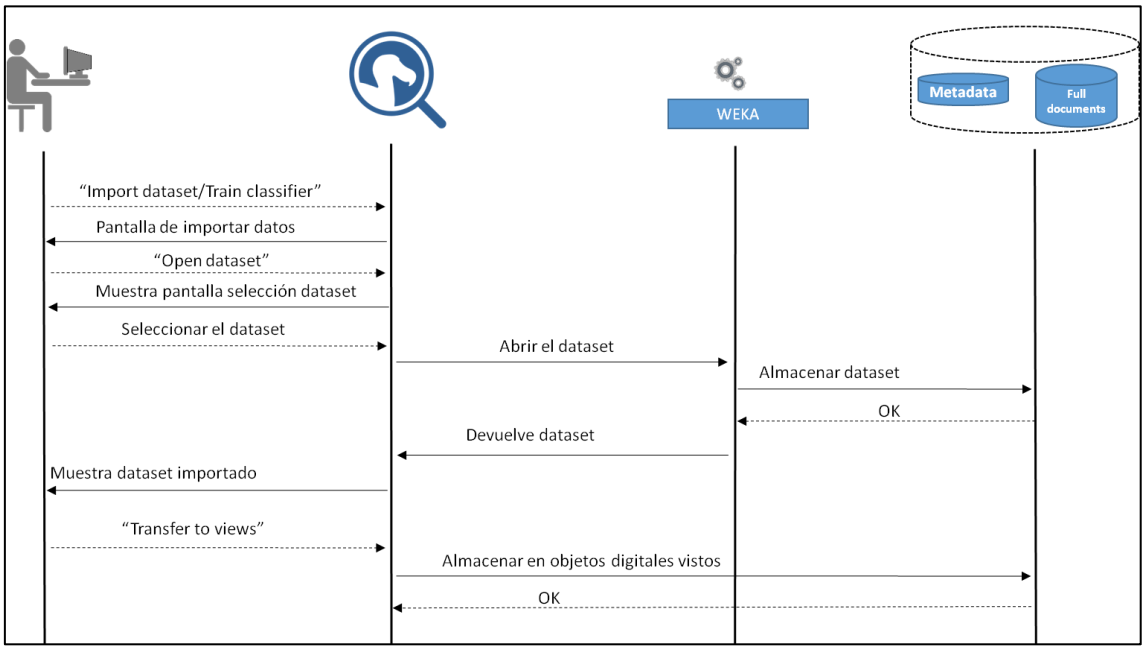

Fig. A.17.- Transferir dataset a los objetos digitales vistos de la aplicación.

El usuario puede transferir los objetos digitales al conjunto de objetos digitales que ha visualizado desde la herramienta. Para ello, debe acceder a la pestaña “Import dataset/Train Classifier”, importar un *dataset* (léase el punto 14) y, una vez se haya cargado, seleccionar la opción “Transfer to views”.

18.-Exportar datos

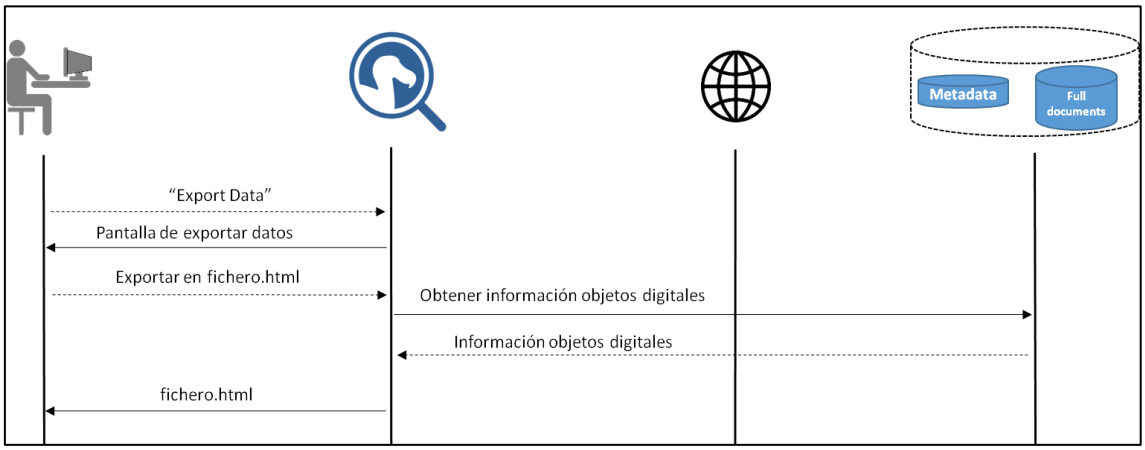

Fig. A.18.- Exportar datos.

Para exportar los datos obtenidos en la aplicación, el usuario debe pulsar en la pestaña "Export Data" y, a continuación, seleccionar la ruta dónde se va a guardar el fichero, con formato .html, que contiene toda la información relacionada a los objetos digitales vistos.

19.- Procesamiento de la información

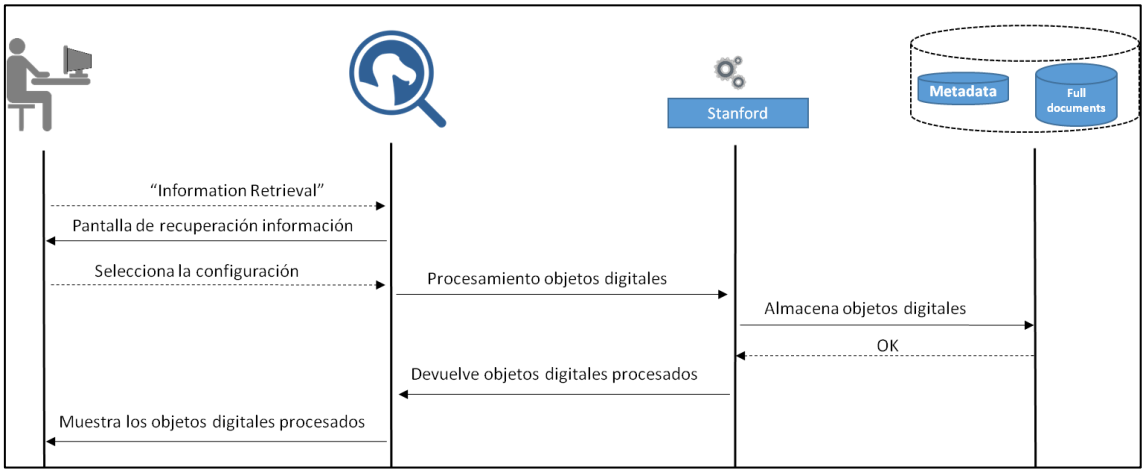

Fig. A.19.- Procesamiento de la información.

Cuando el usuario quiera realizar una tarea de procesamiento de la información, debe acceder a la pestaña "Information Retrieval". A continuación, tiene que seleccionar la configuración previa que desee para realizar la tarea y, una vez ya la ha elegido, se le va a mostrar información relevante sobre el contenido de los objetos digitales, como podría ser el nombre de entidades, personas, palabras claves, localizaciones, etc.

20.- Ubicación en el mapa de las localizaciones

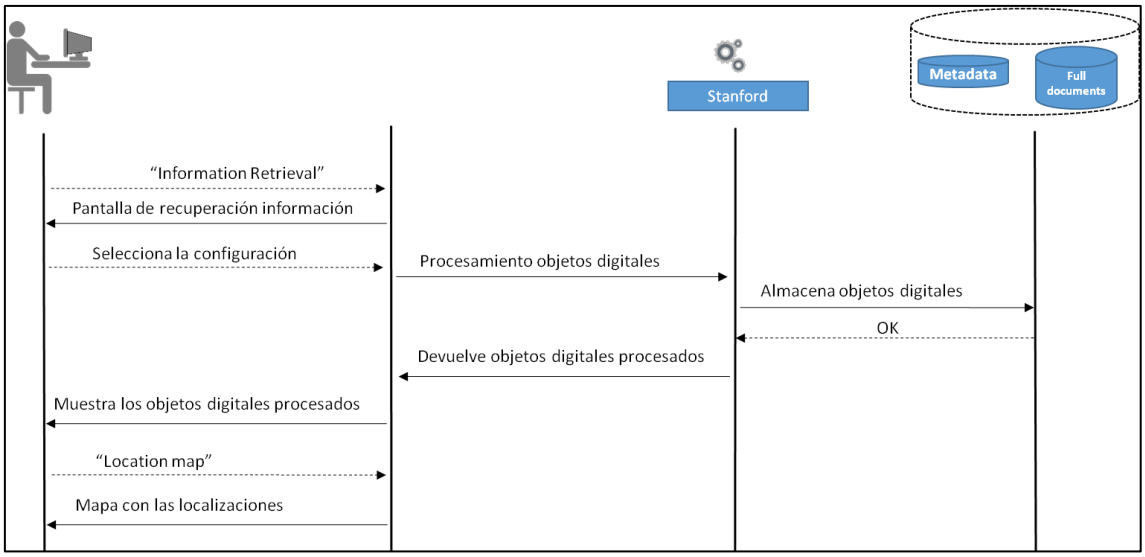

Fig. A.20.- Ubicación en el mapa de las localizaciones obtenidas en el procesamiento de la información.

Para poder acceder a la ubicación en el mapa de las localizaciones, es necesario que previamente se haya hecho un trabajo de procesamiento de información sobre los objetos

digitales que se desee (léase el punto 19). Cuando éste haya finalizado, el usuario debe seleccionar la opción “Location Map” que le mostrará un mapa en el que se sitúan las localizaciones obtenidas del proceso de recuperación de la información realizado.

21.- Modificar configuración de usuario.

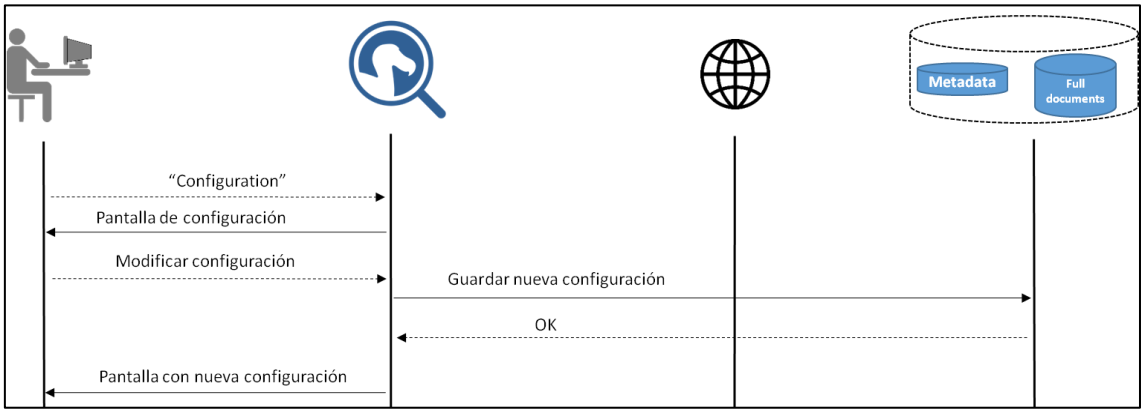

Fig. A.21.- Modificar la configuración del usuario.

Cuando el usuario quiera cambiar la configuración inicial del sistema, debe acceder a la pestaña “Configuration”, en la que se le permite modificar su nombre, definir las columnas a mostrar para las noticias y definir las opciones a seguir para el procesamiento de la información (léase el punto 19).

B. Manual de usuario

En un futuro, se espera que DodoAid siga añadiendo funcionalidades para mejorar progresivamente. Por el momento, se dispone de un prototipo funcional que se va a presentar a continuación a través de capturas del mismo. Mediante el seguimiento de este apartado, el usuario va a poder conocer cómo ejecutar las diferentes funcionalidades que ofrece el sistema. Como por ahora los objetos digitales que permite procesar DodoAid corresponden con aquellos que disponen de información textual, se va a referir durante todo el manual a noticias textuales. Si en un futuro se permite el soporte de otro tipo de objetos digitales, las funcionalidades serían las mismas para todos ellos.

Fig. B.1.- Pantalla de Registro de Usuario tras la primera ejecución de la aplicación.

En primer lugar, el usuario accede a la aplicación y se le va a mostrar una pantalla, como la que se adjunta en la Fig. B.1, en la que escribir su nombre. Una vez lo haya escrito, siempre que acceda a la aplicación se le recibirá con un mensaje de bienvenida junto al nombre escogido por el usuario, como se puede apreciar en la Fig. B.2.

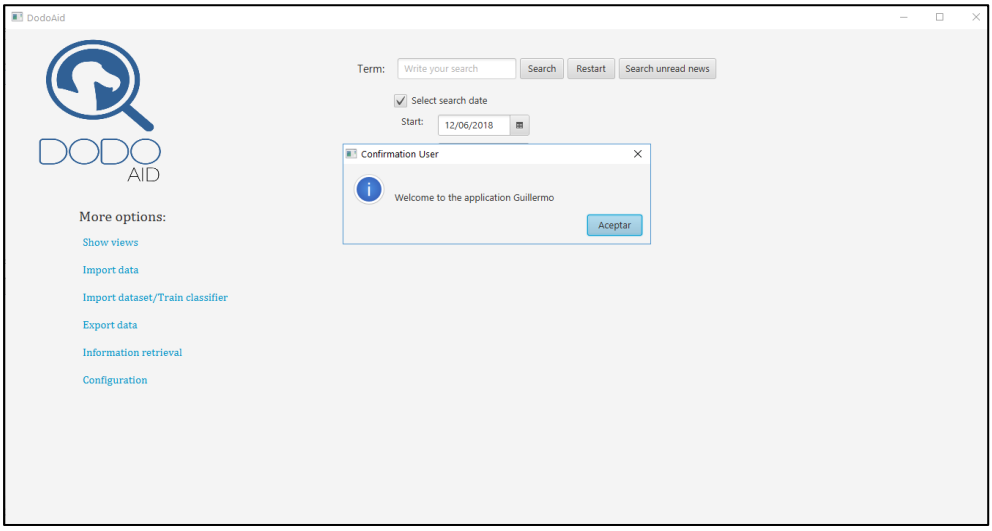

Fig. B.2.- Pantalla de bienvenida tras registro.

Una vez recibido el mensaje de bienvenida, el usuario ya está listo para la navegación por la aplicación.

Lo primero que puede hacer es buscar información sobre alguno de sus temas de interés. Cuando desee buscar una información acotada entre dos fechas, el usuario debe seleccionar la casilla “Select search date” y seleccionar una fecha de inicio y una fecha de final. En la Fig. B.3, se puede ver cómo el usuario ha realizado una búsqueda sobre el término “Zaragoza” entre las fechas seleccionadas.

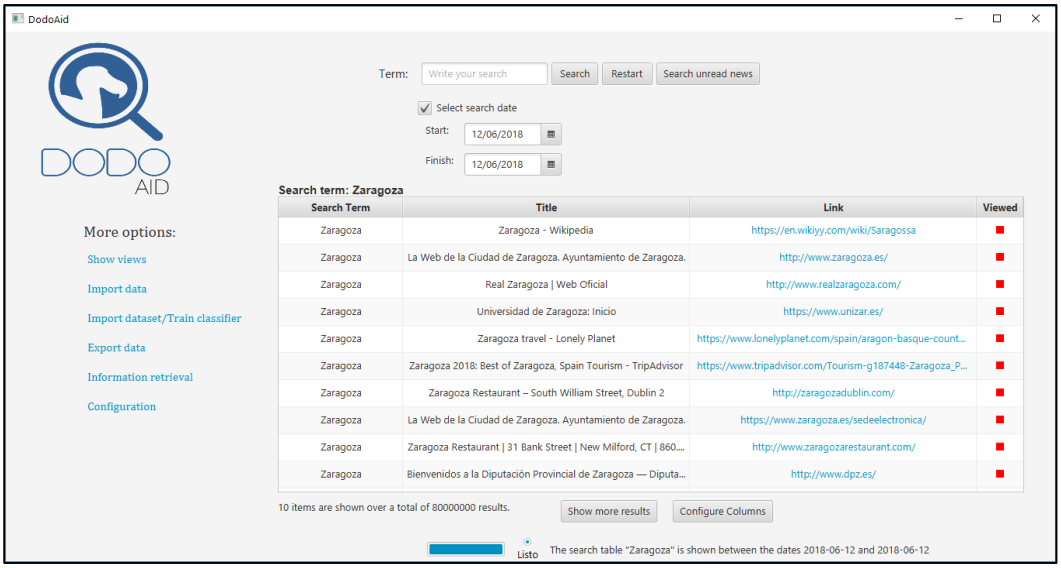

Fig. B.3.- Búsqueda de un término con fecha introducida.

Si por el contrario un usuario desea realizar una determinada búsqueda sin especificar una acotación de las fechas, debe desmarcar la casilla “Select search date”. En la Fig. B.4, se enseña como un usuario realiza una búsqueda para el término “Huesca” sin seleccionar la casilla de elección de fechas.

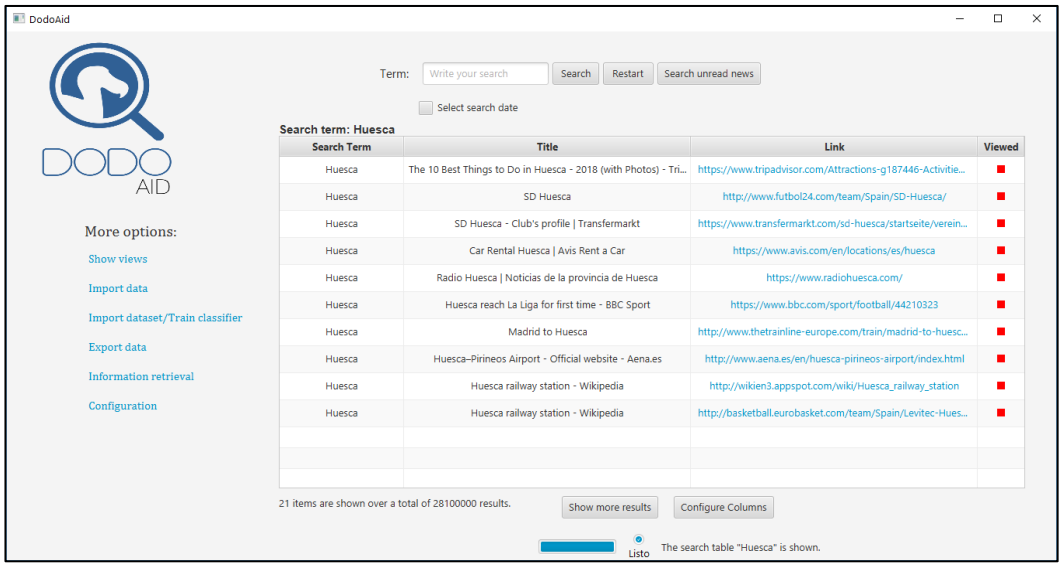

Fig. B.4.- Búsqueda de un término sin fecha introducida.

El número de noticias que va a recibir el usuario es de diez por búsqueda, pero si sobre un determinado tema está interesado en obtener más, existe la posibilidad de seguir recibiendo nuevas. El botón “Show more results” va a permitir al usuario recibir diez noticias más cada vez que sea pulsado. Si el usuario ha estado buscando sobre un término en varias ocasiones, es probable que reciba noticias que ya haya visualizado. Para evitar esto, el usuario puede buscar únicamente aquellas que no se le hayan mostrado previamente pulsando sobre la opción “Search unread news”. En la Fig. B.5, se aprecia cómo se ha realizado una búsqueda sobre el término “Huesca”, pero al pulsar sobre la búsqueda de noticias no vistas, los resultados son distintos a los obtenidos previamente para la misma búsqueda (véase Fig. B.4).

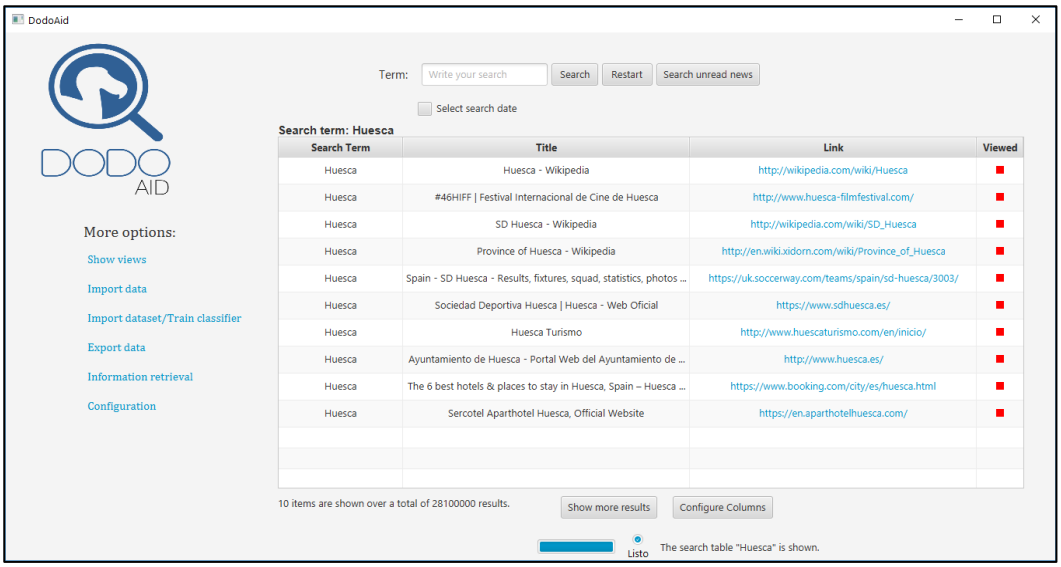

Fig. B.5.- Búsqueda de las novedades sobre un término sin introducir la fecha.

En DodoAid el objetivo es que el usuario pueda tener un control continuo sobre las noticias que ha ido visualizando. Es por ello, que en todo momento se le permite valorar aquellas noticias que ha recibido. Como puede apreciarse en la Fig. B.6, al hacer click derecho sobre un determinado ítem, se le permite realizar distintas acciones sobre él:

- “Change view” → Permite establecer una noticia como vista. El usuario puede saber que esa noticia ya la ha leído.
- “Delete item” → Si no está interesado en una de las noticias recibidas, la puede borrar para así no almacenarla. Esta función es de gran utilidad para aquellos usuarios que quieran tener en la aplicación únicamente las noticias de su interés.
- “Rate item” → Se puede valorar una determinada noticia con una nota del 0 al 10. Inicialmente se considera “NE” (Not Evaluated).
- “Add Favorite” → El usuario puede añadir o quitar una noticia de su lista de favoritas en cualquier momento.

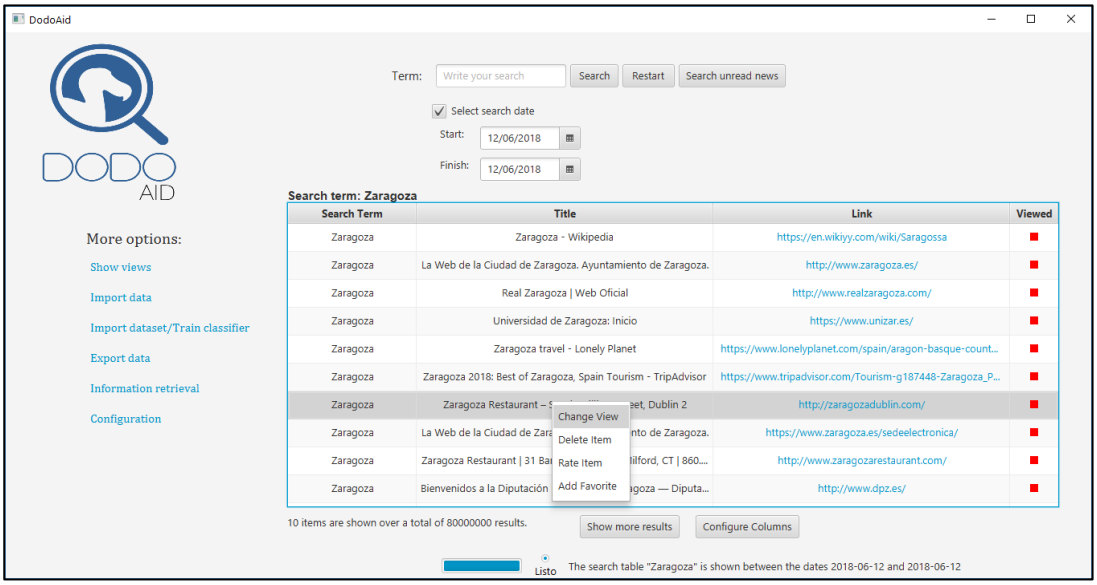

Fig. B.6.- Opciones a realizar sobre una noticia obtenida.

Si no está conforme con la configuración de las columnas que se le proporciona, existe la posibilidad de seleccionar la que desee con el botón "Configure Columns". Se le abrirá una pantalla como la que se muestra en la Fig. B.7, dónde puede añadir o quitar las columnas que quiera. De este modo, el usuario puede definir la información que desea recibir sobre las distintas noticias.

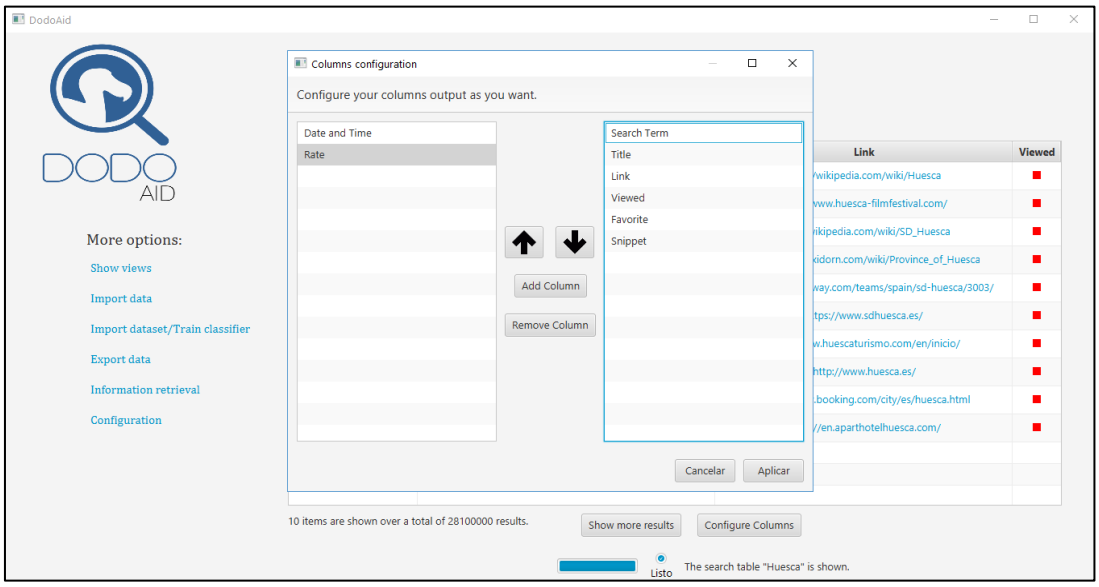

Fig. B.7.- Configuración de columnas a la hora de realizar una búsqueda.

Si por algún casual la información que recibe en alguna de las columnas es superior a su tamaño y no la puede ver completa, detenga su ratón sobre ella y podrá verla en su totalidad como se aprecia en la Fig. B.8.

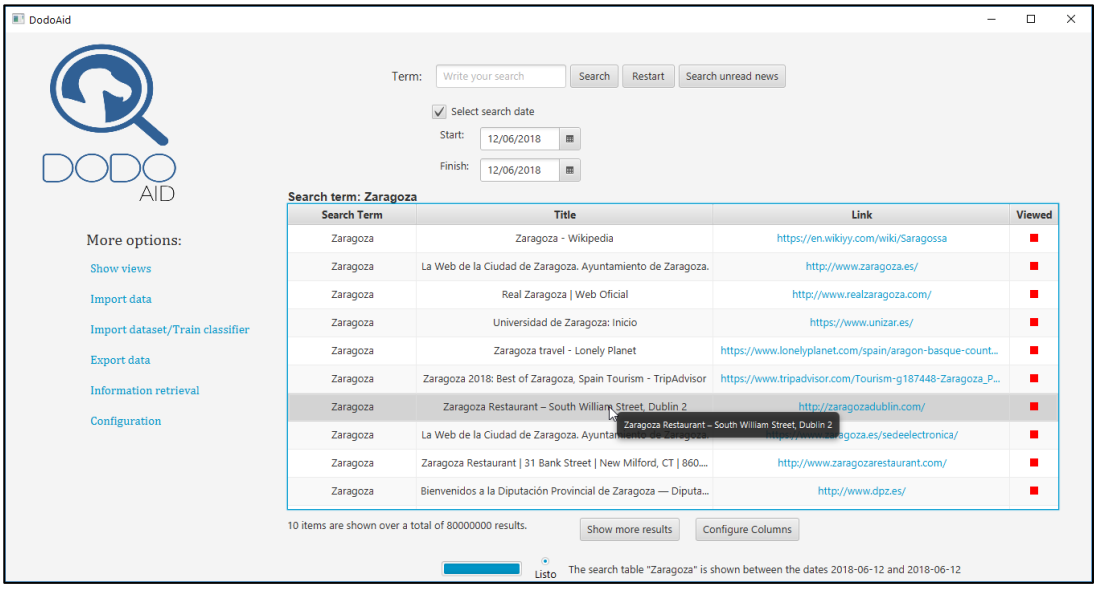

Fig. B.8.- Tooltip con la información correspondiente a los campos de las noticias devueltos.

En el momento que haya realizado alguna búsqueda, podrá volver a acceder a ella siempre que quiera pulsando sobre la opción "Show views". Se obtendrá una ventana similar a la de la Fig. B.9, en la que el usuario puede tener el control sobre sus noticias.

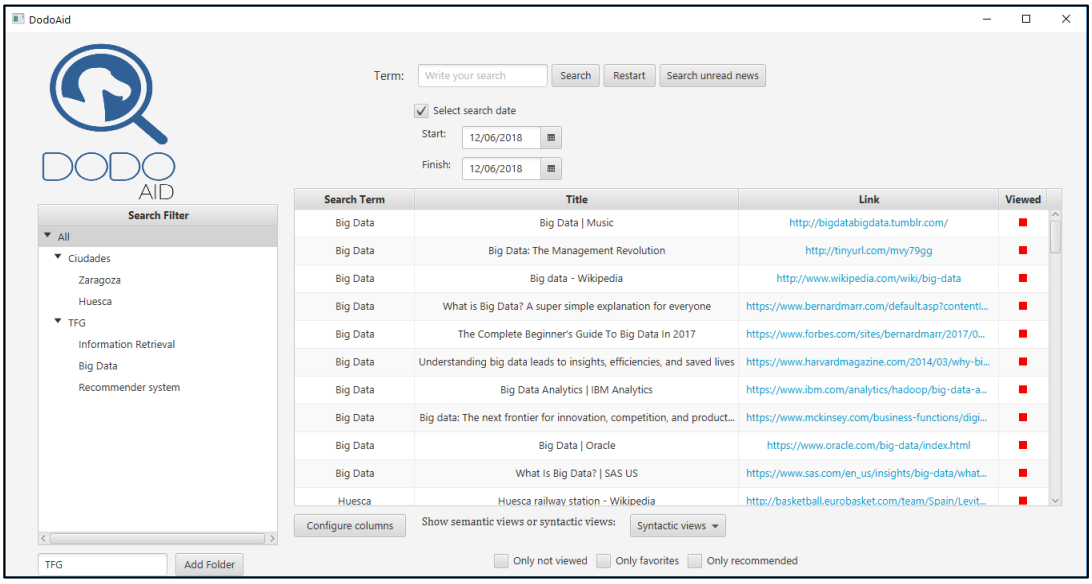

Fig. B.9.- Visualización de las noticias vistas por el usuario.

Al igual que a la hora de realizar una búsqueda, si el usuario no está satisfecho con la configuración de las columnas a mostrar, tiene el botón "Configure columns" que al ser pulsado (véase Fig. B.7) permitirá al usuario seleccionar las columnas que desee. Una vez haya terminado la configuración, véase un ejemplo en la Fig. B.10, podrá disfrutar de la información de sus noticias de la forma que le sea más cómoda y que más le agrade.

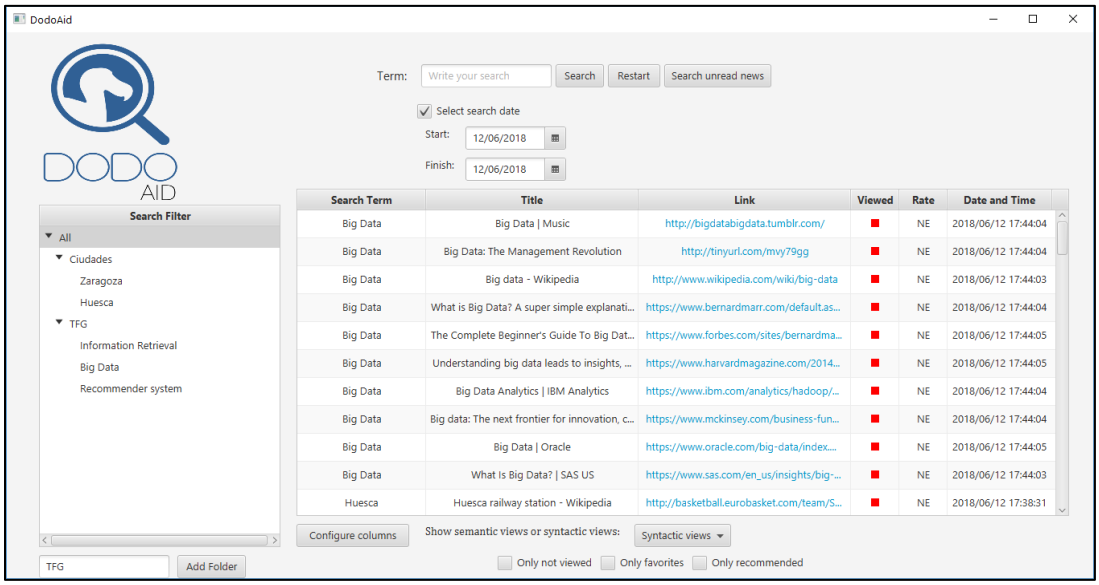

Fig. B.10.- Visualización de las noticias tras configurar las columnas a mostrar.

A la hora de mostrar la información sobre las noticias vistas, DodoAid actúa como si de un sistema jerárquico se tratase, ya que permite al usuario almacenar las diferentes búsquedas bajo "carpetas" que él mismo va creando según sus necesidades. Si lo que desea es crear una carpeta que englobe un conjunto de búsquedas, debe pulsar sobre "Add Folder" para crearla. El usuario puede arrastrar sus búsquedas y depositarlas sobre las carpetas generadas. Por ejemplo, en la Fig. B.10, se acaba de crear la carpeta "TFG" en la que se han almacenado los términos de búsqueda "Information Retrieval", "Big Data" y "Recommender system". De este modo, el usuario tiene el control sobre sus noticias, ya que puede ir distribuyéndolas bajo nuevos conceptos que le faciliten posteriormente el acceso ellas.

Si por un casual se ha equivocado al crear una, se le da la posibilidad tanto de borrarla como renombrarla. Para ello, como se ve en la Fig. B.11, debe hacer click derecho sobre ella.

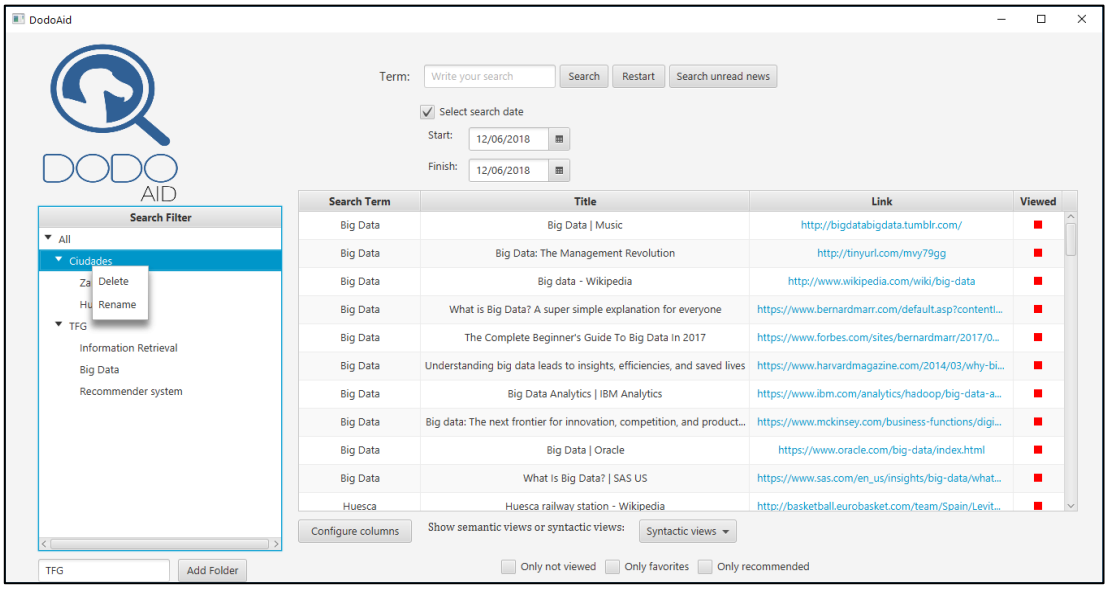

Fig. B.11.- Borrar o renombrar una noticia a partir del filtrado de búsquedas.

Una vez ya conoce como tener almacenadas a su gusto las noticias, es posible que pueda querer añadir alguna noticia que ha visto desde fuera de la aplicación. Para ello, desde la pantalla principal (véase Fig. B.2) puede acceder a la opción “Import data”. En la Fig. B.12, puede apreciarse cómo se deben escribir las noticias para poder ser importadas. En primer lugar, ha de escribir la url; en segundo lugar, el título de la noticia y, en último lugar, el término de búsqueda.

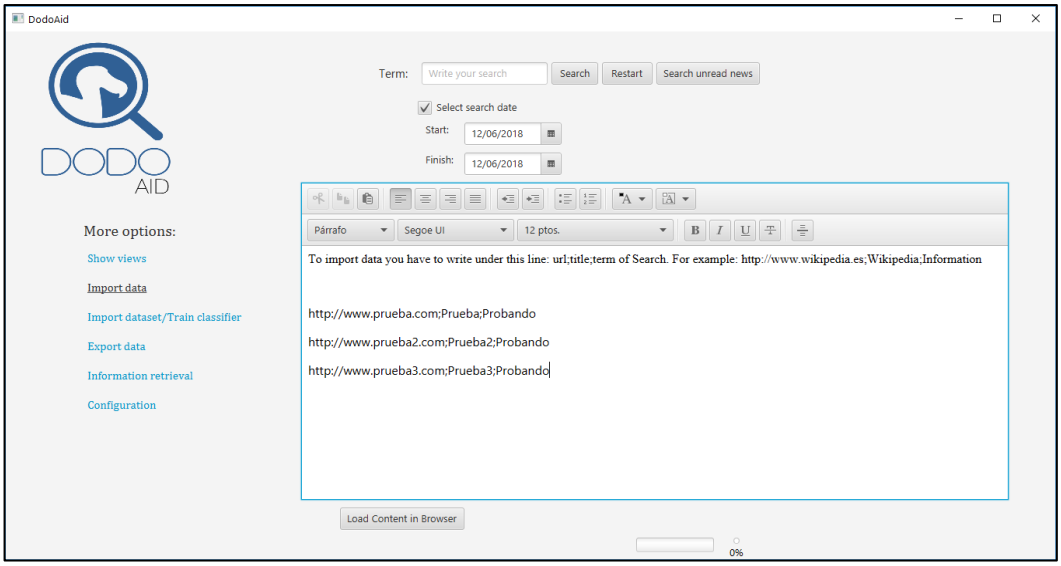

Fig. B.12.- Importar datos.

Si no sigue la estructura indicada para importar noticias, recibirá por pantalla alguno de los siguientes mensajes de error.

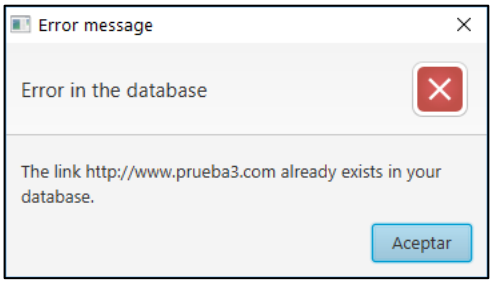

Fig. B.13.- Error importar datos: noticia existente.

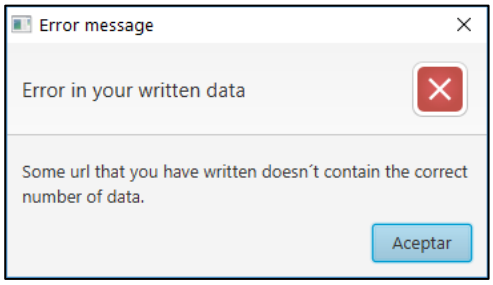

Fig. B.14.- Error importar datos: nº datos incorrecto.

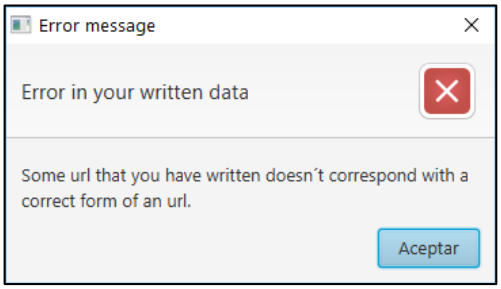

Fig. B.15.- Error importar datos: URL mal formada.

El error obtenido en la Fig. B.13 corresponde al hecho de que la noticia que se quiere importar ya existe en la base de datos.

El error que indica la Fig. B.14 hace referencia a que el número de datos que se ha escrito para guardar es incorrecto.

Por último, el error de la Fig. B.15 indica que la URL que se trata de almacenar no corresponde con una URL.

Para comprobar que sus noticias han sido importadas de manera correcta, diríjase a la pantalla de "Show views" (véase Fig. B.9) y compruebe que están entre sus noticias. En la Fig. B.16, puede apreciar como las tres noticias definidas en la Fig. B.12, han sido importadas con éxito a las noticias vistas.

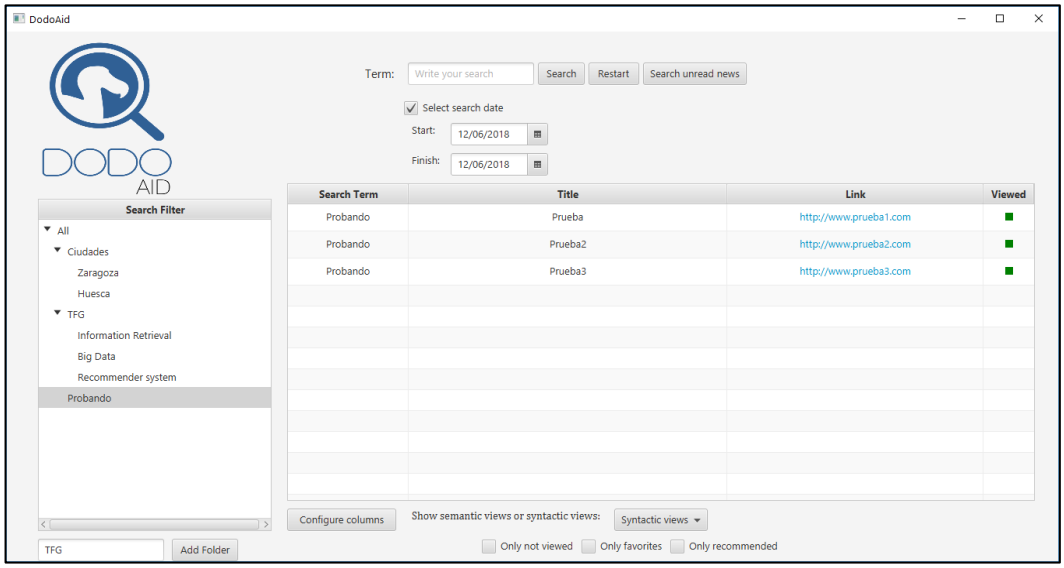

Fig. B.16.- Aparición de las noticias importadas en las noticias vistas.

Para aquellos usuarios que deseen importar un gran número de noticias, existe la posibilidad de importar tu propio conjunto de noticias o *dataset*. Para acceder a él, desde la pantalla principal (véase Fig. B.2) debe pulsar la opción "Import dataset/Train classifier". A continuación, obtendrá la pantalla que se muestra en la Fig. B.17.

Las opciones que dispone el usuario son varias:

- Open dataset: permite al usuario abrir un *dataset* con una estructura bien formada. Se considera una estructura bien formada, aquella que está constituida por una carpeta, subcarpetas y ficheros dentro de las propias subcarpetas. En la Fig. B.18, se aprecia un ejemplo de *dataset* importado siguiendo la estructura que se ha comentado.
- Train: entrena el *dataset* importado por el usuario. Para poder usar esta opción, es necesario que se haya importado previamente un *dataset*.

- Train from views: se entrena a partir de las noticias que ha visto el usuario. No es necesario que el usuario importe un *dataset*. Para que se entrene correctamente, las noticias almacenadas deberán seguir una estructura similar a la que se aprecia en la Fig. B.9.
- Classify: realiza la clasificación a partir del *dataset* importado. Previamente es necesario importar un *dataset* que actúe como conjunto de entrenamiento para el clasificador. Cuando se termina de entrenar un clasificador, se forma lo que se denominan las vistas semánticas, que hacen referencia a las categorías en las que se van a redistribuir los ítems que estén sin categorizar tras realizar la clasificación. Por tanto, el usuario va a poder distinguir entre las vistas semánticas, véase la Fig. B.19, y las vistas sintácticas, véase la Fig. B.20. Para comprender mejor ambos conceptos se van a explicar con un ejemplo. Imaginemos que el usuario ha buscado los términos "Data Mining" y "Big Data" y los guarda en una carpeta denominada "Programming". Al entrenar el clasificador con esta estructura, se crea la vista semántica "Programming" que incluye todos los objetos digitales correspondientes a las búsquedas sobre "Data Mining" y "Big Data". Estos objetos digitales son los que denominamos vistas sintácticas. Por tanto, ahora cuando el usuario quiera clasificar un nuevo ítem relacionado con los dos términos descritos, es probable que el clasificador decida categorizarlo como "Programming".
- Classify from views: realiza la clasificación a partir de las noticias vistas por el usuario. Su funcionamiento es similar al que se acaba de explicar para "Classify". La única diferencia es que no se necesita cargar un *dataset*, ya que el entrenamiento del clasificador se realiza a partir de las búsquedas que ha realizado el usuario.
- Transfer to views: permite transferir las noticias del *dataset* importado a las noticias vistas del usuario. En la Fig. B.20, se aprecia un ejemplo de cómo los datos importados a partir de un *dataset* han sido transferidos a las noticias vistas del usuario.

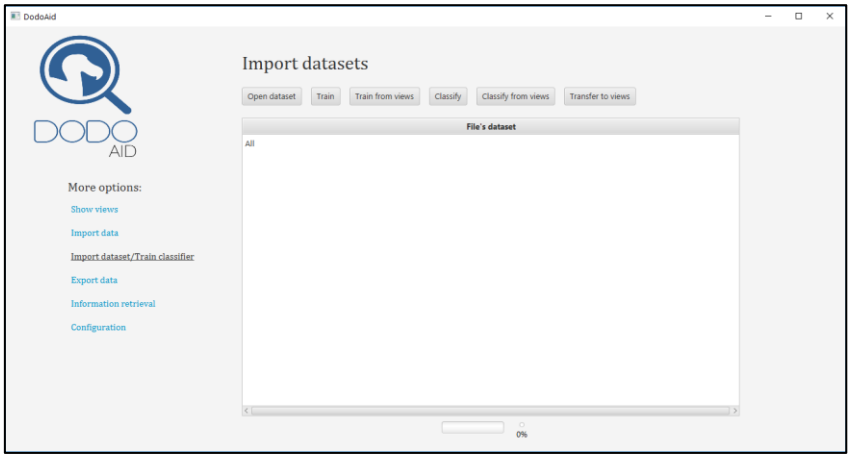

Fig. B.17.- Pantalla inicial para importar un dataset.

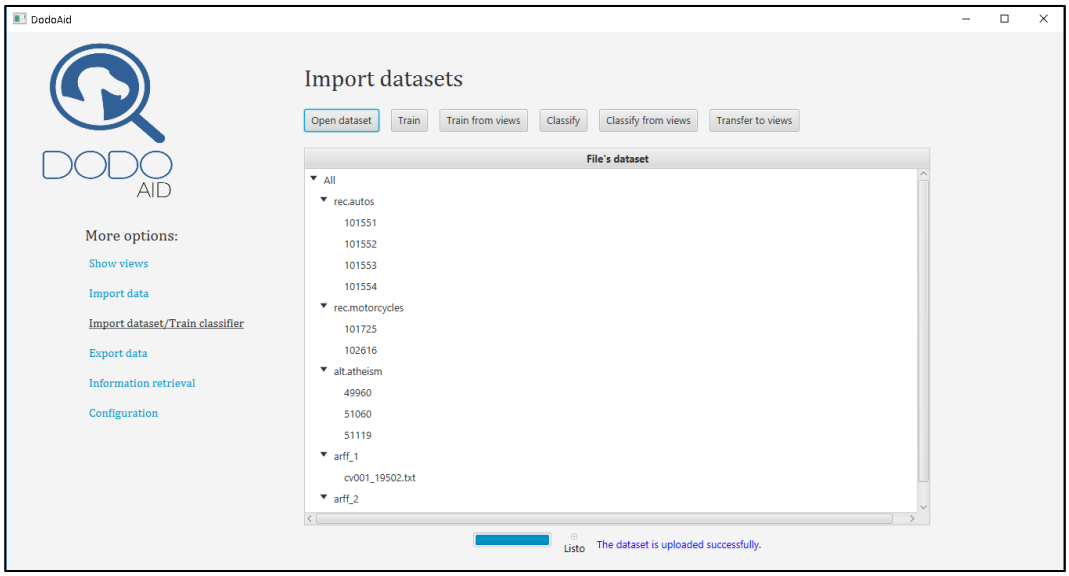

Fig. B.18.- Dataset importado.

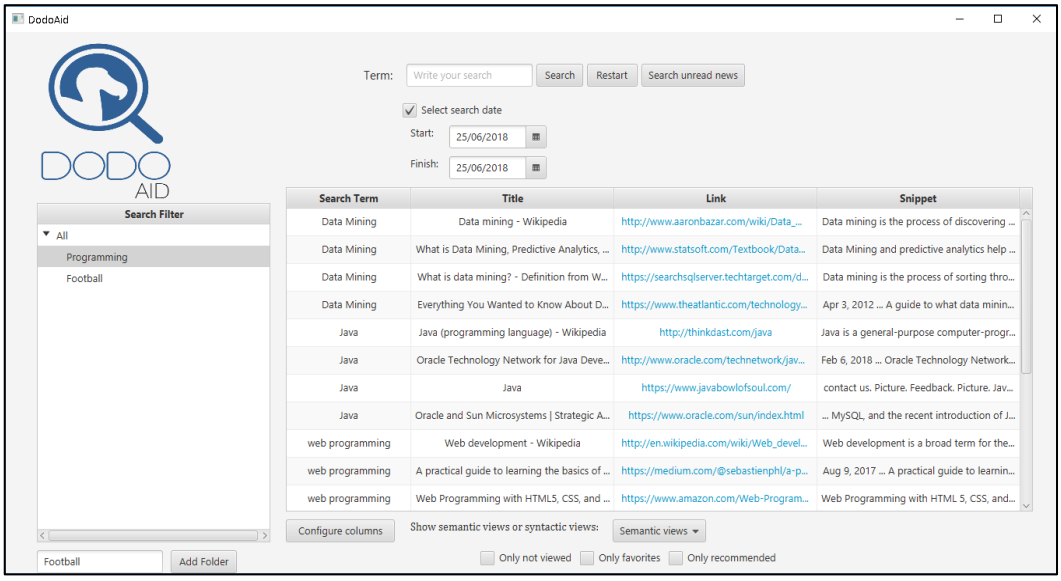

Fig. B.19.- Vistas semánticas tras realizar una clasificación.

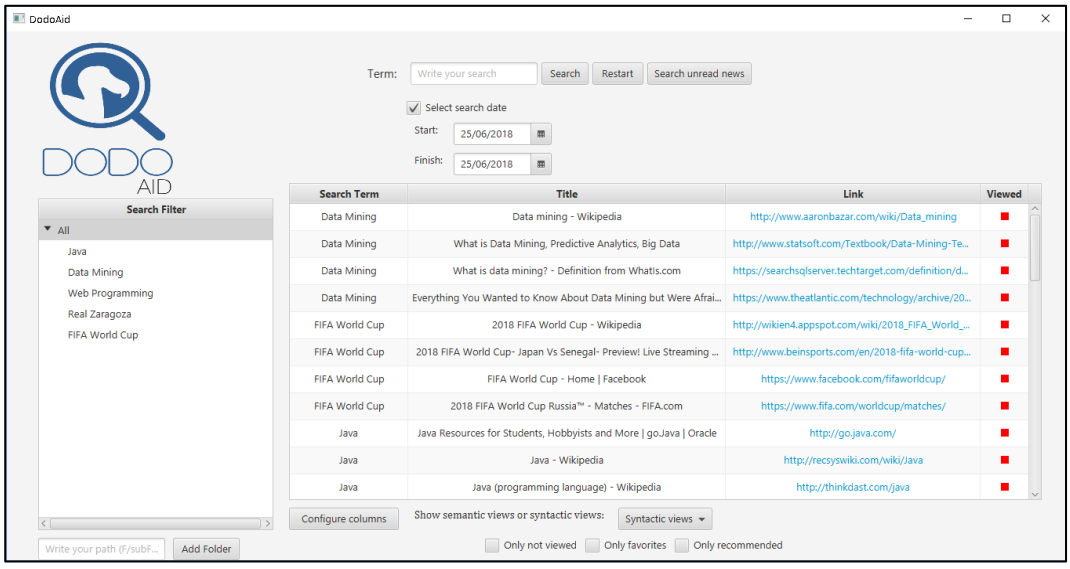

Fig. B.20.- Vistas sintácticas tras realizar una clasificación.

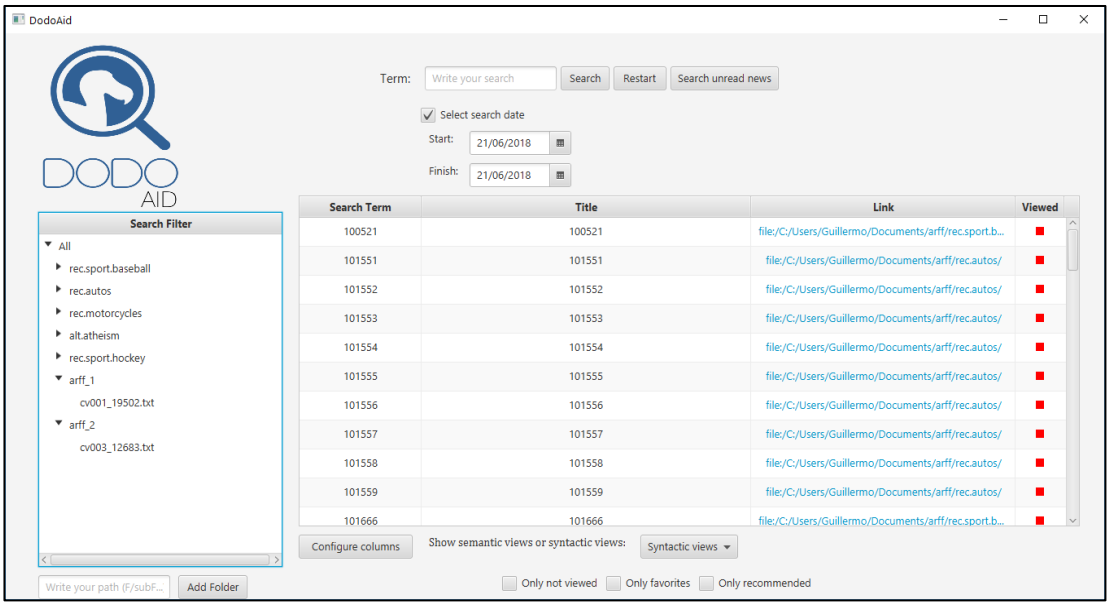

Fig. B.21.- Transferencia de las noticias importadas a través de un dataset a las noticias vistas del usuario.

Si el usuario es capaz de poder importar sus noticias, es posible que quiera poder descargarlas y guardárselas. Para ello, se le da la opción de exportarlas. Desde la pantalla principal (véase Fig. B.2), el usuario debe seleccionar la opción “Export data”. Una vez lo haya hecho, se le mostrará una ventana como la que se aprecia en la Fig. B.22, a partir de la cuál decidirá dónde almacenar sus noticias.

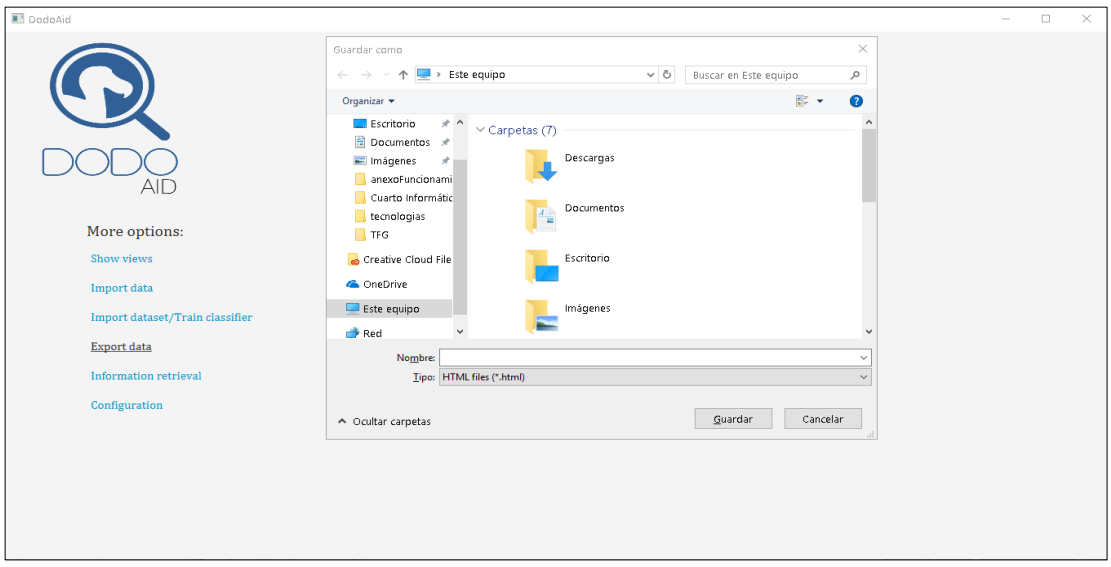

Fig. B.22.- Exportar datos.

Si por el contrario lo que desea es un análisis detallado sobre las noticias, existe la posibilidad de que se realice un procesamiento de la información para que el usuario pueda tener una mayor información sobre sus búsquedas. Desde la pantalla principal (véase la Fig. B.2), el usuario debe pulsar la opción “Information Retrieval” obteniendo como resultado la pantalla que se aprecia en la Fig. B.23.

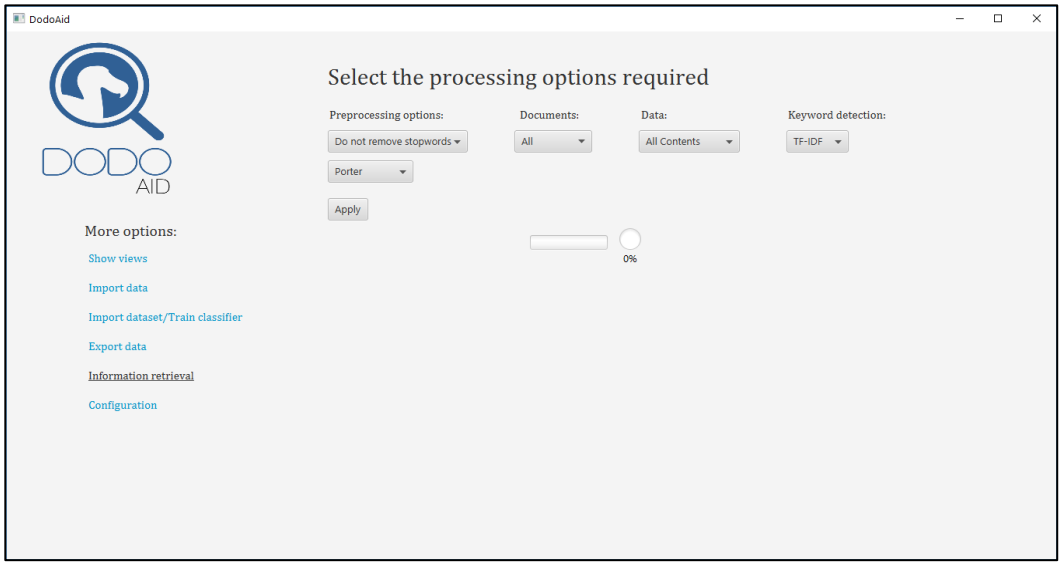

Fig. B.23.- Procesamiento de la información.

Es posible que si usted es un usuario primerizo con el procesamiento de información, no conozca el significado de algunos términos a elegir. Por ello, para aquellos usuarios que no están familiarizados con dicho tema, se va a explicar de manera resumida los términos que pueden ser de su ignorancia.

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 → Corresponde a la eliminación de los sufijos englobando todas esas palabras sobre la 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.

4.- TF (Term Frecuency) → Corresponde a la repetición de un determinado término a lo largo de todo el documento.

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

En las figuras Fig. B.24 y Fig. B.25 se pueden apreciar dos ejemplos tras el procesamiento de la información. En el primero de ellos, gracias a la opción de la lupa, se permite al usuario poder ver la noticia con la misma estructura en la que está en la web.

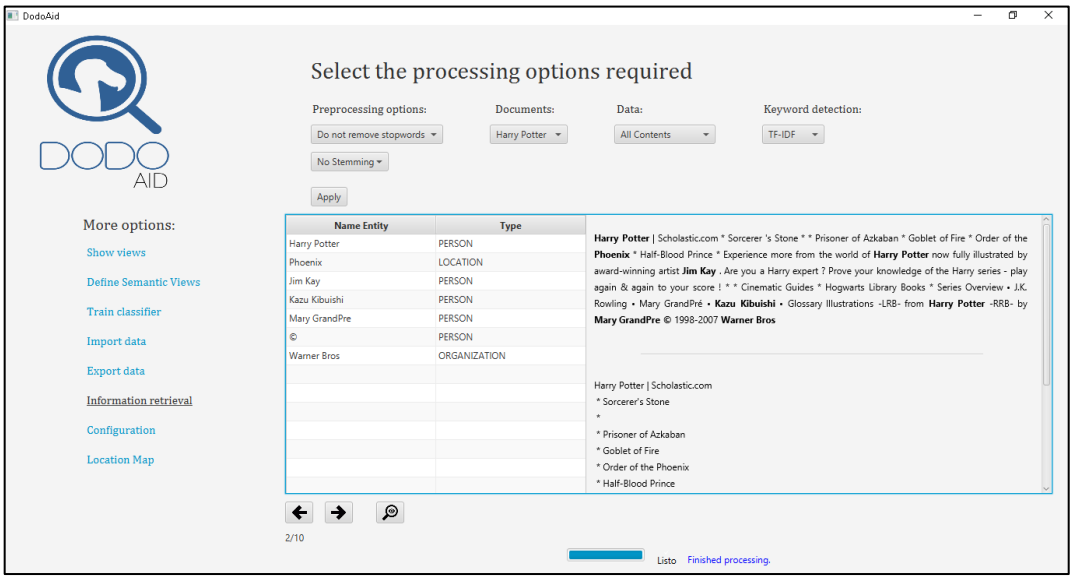

Fig. B.24.- Procesamiento de la información.

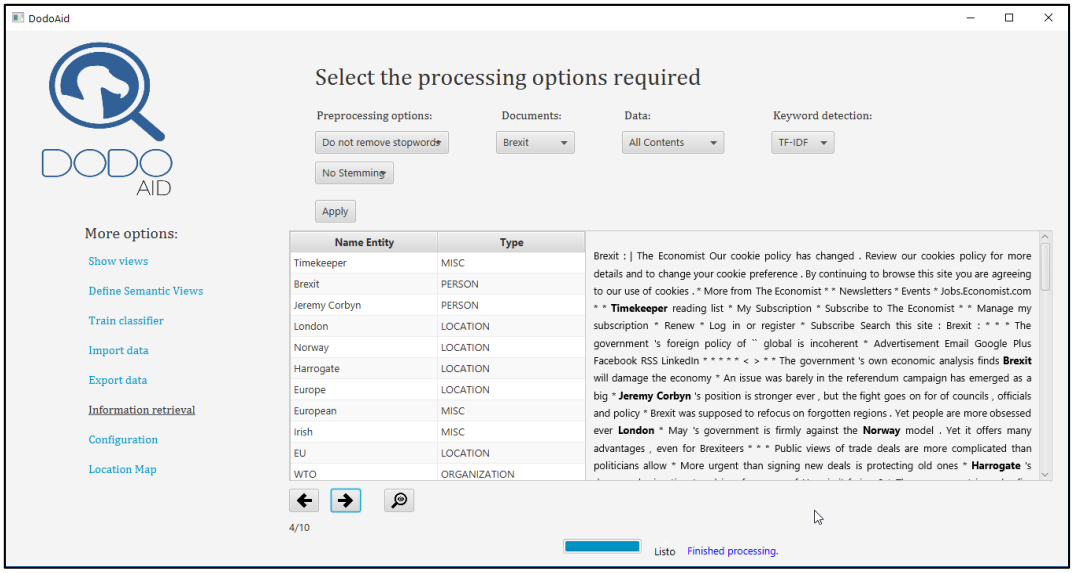

Fig. B.25.- Procesamiento de la información.

Tras haber realizado el procesamiento de la información, el sistema detecta nombres de entidades, palabras claves, personas, localizaciones, etc. En relación a las localizaciones que aparecen en el texto, se le permite al usuario poder situarlas en un mapa. Para ello, se debe pulsar sobre la opción “Location Map”, que únicamente aparece cuando se ha realizado un procesamiento de la información (véase la diferencia del menú izquierdo entre la Fig. B.23 y Fig. B.24). Como resultado tras decidir que quiere conocer la ubicación de las localizaciones, el usuario obtendrá un mapa similar al de la Fig. B.26, en el que puede ver dónde está ubicada una localización y, además, obtener información relevante acerca de ella. Para pasar de localización, el usuario puede pulsar sobre los puntos en el mapa o pulsar en el botón “Go to next Location”. Por último, en relación al mapa, se permite al usuario acceder a las noticias que contienen dicha localización. Por ejemplo, en la Fig. B.26, Paris aparece en 1/10 objetos digitales. Al pulsar

sobre “Go to digital objects”, el usuario obtiene la noticia en la que se ha nombrado dicha localización.

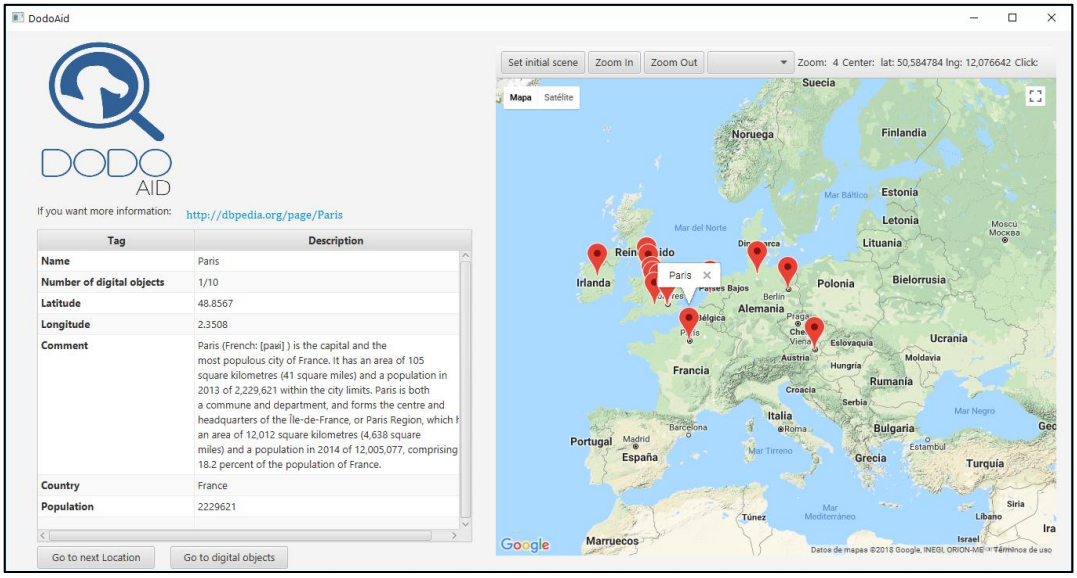

Fig. B.26.- Ubicaciones obtenidas en el procesamiento de información situadas en el mapa.

Para finalizar, es posible que el usuario quiera tener una configuración fija a la hora de manejarse desde la aplicación. Es por ello que se le ofrece la posibilidad de definir su propia configuración para que cada vez que acceda a la aplicación, se le muestre la información a su gusto.

Cuando el usuario quiera cambiar la configuración inicial del sistema, debe acceder a la pestaña “Configuration” desde la pantalla principal, véase Fig. B.2. Desde esta pantalla, figura adjunta Fig. B.27, el usuario va a poder modificar su nombre, definir las columnas a mostrar para las noticias y definir las opciones a seguir para realizar la tarea de recuperación de información (véase la Fig. B.24).

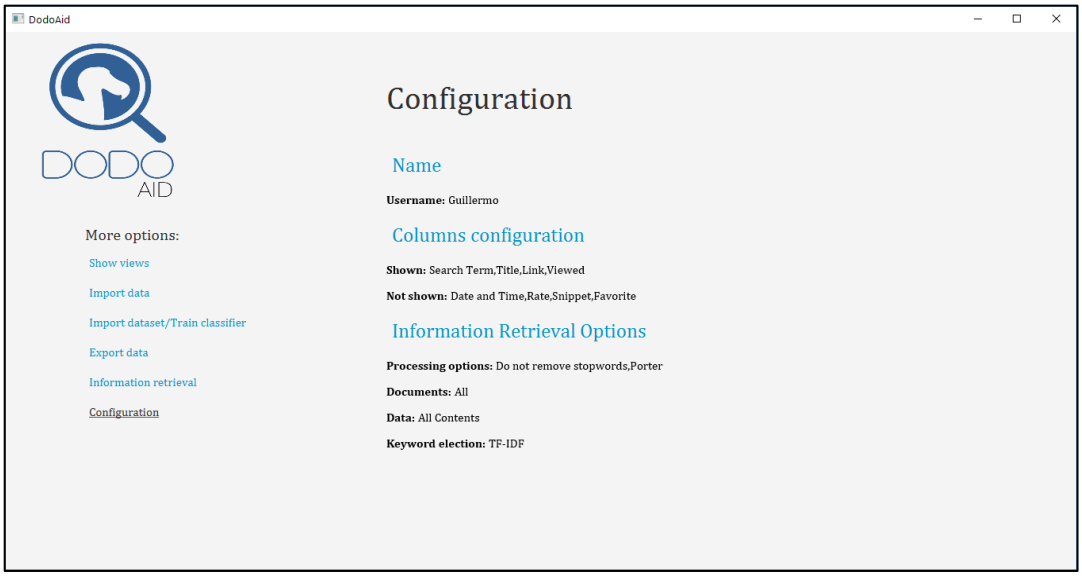

Fig. B.27.- Configuración del usuario

C. Evaluación del sistema de recomendación

A la hora de tratar de saber si el sistema que se ha generado realiza su trabajo de manera correcta o no, es necesario realizar diversas pruebas para comprobar si las recomendaciones que recibe el usuario son buenas, o si por el contrario el ratio de error es demasiado grande.

C.1 MAE y RMSE

En primer lugar, se va a hablar de dos métricas muy utilizadas para medir la precisión que tienen los sistemas de recomendación. Éstas son MAE (Mean Absolute Error) y RMSE (Root Mean Squared Error). A continuación, se va a entrar más en detalle a comprenderlas y se va a realizar un ejemplo relacionado con el sistema de recomendación de DodoAid para entender cómo se ha realizado su cálculo.

MAE mide el error medio en un conjunto de predicciones. Calcula el valor absoluto de la diferencia entre la valoración inicial del usuario y la predicción proporcionada por el sistema de recomendación.

$$\text{MAE} = \frac{1}{n}\sum_{j=1}^{n} |y_j - x_j|$$

RMSE, al igual que MAE, también mide el error medio en un conjunto de predicciones, pero a diferencia de la otra métrica ésta es calculada a partir del cuadrado de la resta entre la valoración inicial del usuario y la predicción proporcionada por el sistema. De este modo, cuándo se produce una recomendación que no debía haberse realizado, RMSE amplifica y penaliza con mayor fuerza aquellos errores de mayor magnitud.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{j=1}^{n} (y_j - x_j)^2}$$

De este modo, se llega a que el resultado de RMSE es siempre mayor o igual que el de MAE. Si todos los errores tienen la misma magnitud, entonces RMSE = MAE.

A continuación, se va a explicar un ejemplo en base al sistema de recomendación de noticias. Imaginemos que al usuario de DodoAid que se va a estudiar le gusta mucho la música. Una de sus búsquedas es “música clásica” a cuyos ítems los ha valorado con un rating entre 7 y 10. El sistema ya puede saber que aquellas noticias cuya similitud coseno sea similar a las que se han encontrado con el término de búsqueda “música clásica” es probable que le gusten al usuario. Éste, consciente de que el sistema conoce un poco más sus gustos, realiza nuevas búsquedas en la aplicación: “rock”, “música de moda”, “Mozart” o “Viena”. El sistema de recomendación va a calcular el rating para cada una de las nuevas noticias en función de lo que conoce sobre el usuario, en este caso el contenido textual de los ítems que ha obtenido tras la

búsqueda del término “música clásica”. Para poder calcular tanto el MAE como el RMSE, el usuario deberá valorar las noticias que se le han recomendado, ya que el cálculo de ambas métricas necesita tanto el rating predicho como el rating del usuario.

Siguiendo con el ejemplo, se va a adjuntar una tabla en la que se almacena un posible caso de cálculo:

Noticia	Rating predicho	Rating usuario	Diferencia
N1	7.5	8.0	0.5
N2	7.2	7.5	0.3
N3	6.8	4.0	2.8

Tabla C.1.- Posible ejemplo de interacción de un usuario al valorar un objeto digital.

El rating predicho corresponde con el valor calculado por el sistema de recomendación. En cambio, el rating del usuario hace referencia a la valoración que proporciona éste sobre el ítem que se le ha recomendado. Una vez que tenemos ambos ratings, ya se puede pasar a calcular tanto el MAE como el RMSE.

$$\text{MAE} = \frac{|8.0-7.5|+\ |\ 7.5-7.2|+\ |4.0-6.8|}{3} = 1.2$$

$$\text{RMSE} = \frac{(|8.0-7.5|)^2+\ (|7.5-7.2|)^2+\ (|4.0-6.8|)^2}{3} = 2.7266$$

Como se observa, al haber únicamente tres ítems sobre los que realizar la prueba, el valor de las métricas no va a ser muy representativo, pero se puede apreciar cómo el RMSE penaliza en mayor medida los fallos. Para el caso del objeto digital N3 se puede considerar que se ha producido un fallo, ya que el sistema de recomendación había predicho una valoración de 6.8, mientras que el usuario al verlo ha decidido que no le gusta y le ha dado una valoración baja.

A medida de que el sistema vaya entrenándose y sepa mejor los gustos del usuario, ambos errores irán disminuyendo, ya que cada vez la probabilidad de poder ofrecerle al usuario un ítem de su interés será mayor.

A la hora de calcular la valoración (un máximo de 10) dada por el usuario, se sigue un algoritmo en el que se tienen en cuenta tres características:

- Rating (α): 0..10
- Favourite (β): 1 ó 0 si no es favorito.
- Views (γ): 0..N

El objetivo es poder obtener una valoración del usuario sobre un determinado ítem en función de la interacción que realiza sobre él. De este modo, cada característica va a tener un peso diferente.

Inicialmente, se decidió darle un mayor peso a añadir un ítem a favoritos, ya que se consideraba que añadirlo a esa lista significaba que es un ítem que te había gustado bastante. Por tanto, en un comienzo se aplicaba la fórmula siguiente para la valoración sobre un ítem:

$$score(\alpha,\beta,\gamma) = \alpha \cdot 0.25 + \beta \cdot 5.0 + \gamma \cdot 0.25$$

Posteriormente se decidió estudiar a un usuario navegando por la aplicación. A partir de dicha observación, se llega a la conclusión de que para un usuario es más intuitivo valorar un ítem con una nota antes que añadirlo a favoritos. Por ello, el algoritmo inicial se vio modificado dándole más peso al rating que al hecho de añadirlo a favoritos.

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

A continuación, se van a adjuntar gráficas en relación a ambos casos para poder comprobar de manera numérica la diferencia de error entre ambos.

Caso 1: rate = 2.5; favorito = 5.0; número de visitas = 2.5

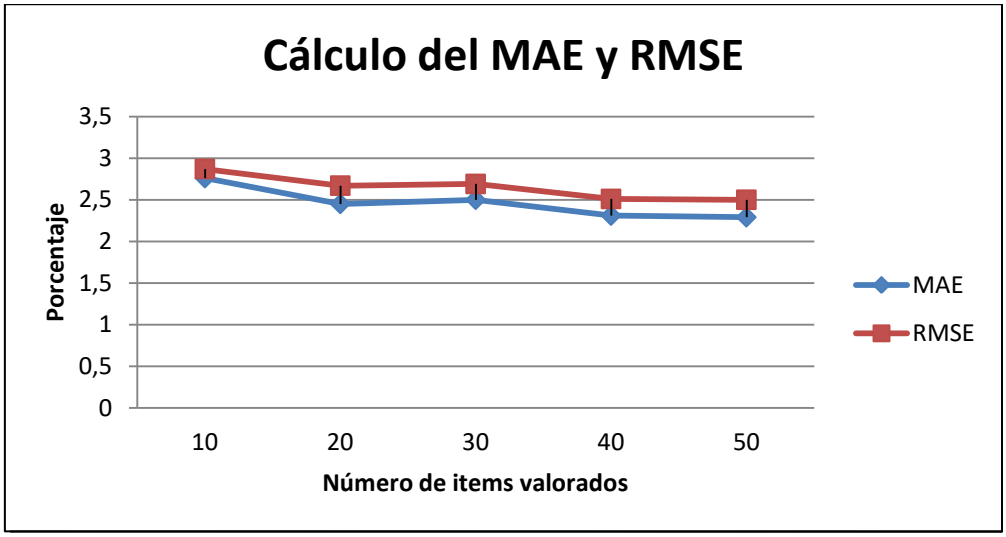

Fig. C.1.- MAE y RMSE para rate=2.5 y favorito=5.0

Para obtener dichos datos se ha analizado el comportamiento de un usuario al que se le ha dicho que añada ítems a favoritos y los vaya valorando para poder obtener recomendaciones. Una vez obtenidas, se le dice que las valore con el fin de poder saber el error que se ha dado en ellas. Al obtener los resultados y ver que el MAE y el RMSE eran tan altos, se decide analizar al usuario para ver su comportamiento. A partir de ello, se llega a la conclusión que al usuario le resulta más intuitivo valorar las noticias que añadirlas a favoritas. Por ello, si tiene alguna noticia como favorita y se le recomiendan algunas similares a ella, se observa que posteriormente el usuario, a la hora de valorarlas, no suele añadirlas a favoritas, por lo que la diferencia entre los ratings va a ser elevada.

A partir de este aspecto, se decide cambiar la puntuación en el algoritmo, dándole mayor peso al rating que al hecho de añadirla a favoritas.

Caso 2: rate = 5.0; favorito = 2.5; número de visitas = 2.5

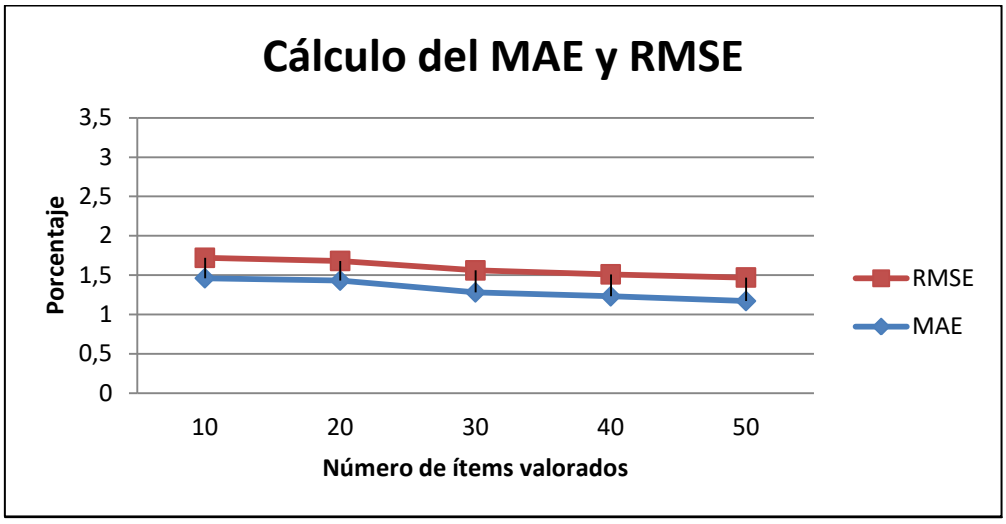

Fig. C.2.- MAE y RMSE para rate=5.0 y favorito=2.5

En relación a que el usuario va valorando cada vez un mayor número de ítems, el error se va reduciendo, ya que el sistema de recomendación cada vez dispone de más información sobre el usuario y va a poder realizar mejores recomendaciones.

Al fin y al cabo, la forma con la que se ejecuta ambos casos es la misma, pero analizando los resultados se observa que para el usuario es más intuitiva la valoración del 0 al 10, por lo que se decide cambiar el algoritmo complementario al cálculo de similitud coseno, definiendo que el rating tiene más peso que el hecho de añadir un ítem a favoritos.

C.2 Precisión y recall

Una vez explicados el MAE y el RMSE, se va a pasar a analizar dos técnicas que también nos permiten conocer cómo de bien hace su trabajo el sistema de recomendación generado.

Por un lado, se va a hablar de la precisión, la cual hace referencia al porcentaje de la fracción de recomendaciones top que son recomendaciones relevantes. Es decir, de los ítems que se nos recomiendan, cuántos aparecen entre los ítems relevantes para el usuario. Se considera ítem relevante aquel cuyo valor de rate sea mayor que el threshold (umbral definido para la recomendación).

$$\text{precision} = \frac{|\{relevant\ documents\} \cap \{retrieved\ documents\}|}{|\{retrieved\ documents\}|}$$

Por otro lado, la otra técnica que se va a analizar es el *recall*, la cual hace referencia al porcentaje de la fracción de ítems relevantes que aparecen en el top de recomendaciones. Es decir, del total de los ítems definidos como relevantes por el usuario, cuántos de ellos aparecen en el top de las recomendaciones.

$$\text{recall} = \frac{|\{relevant\ documents\} \cap \{retrieved\ documents\}|}{|\{relevant\ documents\}|}$$

Para el cálculo de ambos valores se necesita que el usuario haya valorado algún ítem previamente, ya que si no se conoce nada al respecto, es imposible obtener ambas métricas. Un usuario que accede por primera vez a la aplicación no va a tener ningún dato guardado en ella, por lo que el sistema es imposible que pueda recomendarle ítems de interés. Al no disponer de información previa, tampoco podemos saber si las recomendaciones que se le hacen al usuario son buenas o malas. Por ello, se ha decidido que para poder calcular ambas métricas se tiene que definir la "ground truth". Este término, en relación al machine learning, hace referencia a la precisión de clasificación de un conjunto de datos de entrenamiento, es decir, definidos unas valoraciones sobre un conjunto de ítems, comprobar si las predicciones que se obtienen se acercan a dichas valoraciones o no.

Una vez definidos ambos términos, se va a realizar el cálculo sobre la aplicación de DodoAid. El usuario inicial ha realizado la búsqueda de 10 términos, obteniendo como resultado 100 ítems diferentes. Tras analizar las búsquedas, ha decidido que todos los ítems devueltos para cada término buscado tengan la misma valoración. Por tanto, el usuario conectado a DodoAid tiene lo siguiente:

- Término buscado: "Music" → 10 ítems → Rating: 7.5
- Término buscado: "Football" → 10 ítems → Rating: 2.5
- Término buscado: "Amaral" → 10 ítems → Rating: 7.0
- Término buscado: "FIFA World Cup" → 10 ítems → Rating: 3.0
- Término buscado: "Salou" → 10 ítems → Rating: 5.0
- Término buscado: "Volleyball" → 10 ítems → Rating: 6.0
- Término buscado: "VAR" →10 ítems → Rating: 2.0
- Término buscado: "Arenal Sound" → 10 ítems → Rating 7.0
- Término buscado: "Maluma" →10 ítems → Rating 1.0
- Término buscado: "Reggaeton" → 10 ítems → Rating 0.0

Analizando la "ground truth" definida por el usuario y, estableciendo el threshold con un valor de 5.0, se observa que va a haber 50 ítems relevantes y otros 50 no relevantes, debido a que su rating no pasa el umbral definido.

Una vez se tiene la "ground truth", ya se pueden obtener resultados para ambas métricas. Cuando el sistema de recomendación trate de recomendar al usuario por primera vez, no va a tener definido ningún rating para dichos ítems, ya que esa valoración establecida es propia del usuario. Por ello, se va a entrenar el sistema de recomendación estableciendo el rating de un ítem de cada término buscado, teniendo así 10 términos utilizados para entrenar el sistema. Con este entrenamiento, haciendo que se recomienden únicamente las noticias que no han sido valoradas, el número de documentos relevantes e irrelevantes se queda en 45 ítems cada uno.

Tras el primer entrenamiento del sistema, la recomendación ha contado con 10 ítems que han tenido un rating predicho mayor del umbral establecido. Dentro de esos 10 ítems, 9 estaban entre los relevantes, mientras que 1 de ellos no, ya que corresponde al término "Maluma". Por tanto, tras este primer entrenamiento tenemos:

$$\text{Precision} = \frac{9}{10} = 0.9 \qquad \text{Recall} = \frac{9}{45} = 0.2$$

A continuación, se va a seguir con el ejemplo entrenando de nuevo el sistema de recomendación. Se valora un nuevo ítem de cada término, teniendo ahora 20 valoraciones hechas por el usuario y un total de 40 ítems relevantes y no relevantes. Tras este segundo entrenamiento, la recomendación ha contado con un total de 8 ítems que han sobrepasado el umbral de rating 5.0. Dentro de estos 8 ítems, 6 de ellos eran relevantes, mientras que 2 de ellos correspondían a la búsqueda "FIFA World Cup" y "Reggaeton". Por tanto, ahora tenemos:

$$\text{Precision} = \frac{6}{8} = 0.75 \qquad \text{Recall} = \frac{6}{40} = 0.15$$

Para finalizar con este ejemplo, se va a entrenar de nuevo el sistema con otra valoración más para un ítem de cada búsqueda. De este modo, el sistema estará entrenado con 30 ratings y quedarán 35 ítems relevantes y no relevantes todavía por recomendar. De nuevo 8 ítems han superado el umbral establecido, pero esta vez todos ellos pertenecen a los ítems relevantes establecidos. Por tanto, ahora la precisión y el *recall* son:

$$\text{Precision} = \frac{8}{8} = 1 \qquad \text{Recall} = \frac{8}{35} = 0.23$$

Con este ejemplo de cálculo se puede ver cómo el sistema de recomendación va aprendiendo conforme se van añadiendo nuevas valoraciones y cada vez la precisión con la que se recomiendan ítems relevantes para el usuario es mayor.

C.3 Pruebas con otros datasets

Además de realizar pruebas con las noticias que se van obteniendo tras las búsquedas del usuario, se han utilizado *datasets* conocidos, como por ejemplo el de MovieLens con tamaño 100K, para comprobar el funcionamiento de nuestro sistema sobre otros conjuntos de datos. Inicialmente, el fichero de MovieLens no dispone de información textual sobre las diferentes noticias. A partir del uso de KNIME se generó un *dataset* en el que los ítems tuviesen información textual. De este modo, el *dataset* estaba formado con el identificador de la película, el nombre de la película y una explicación sobre la misma. A partir de estos datos, ya se dispone de información textual en las características de los ítems para poder aplicar el sistema de recomendación basado en contenido. Para que éste pueda realizar su trabajo es necesario que el usuario haya valorado diferentes ítems, por lo que por otro lado se disponen de las valoraciones de los usuarios sobre dichos ítems.

Una vez ya se tiene preparado el *dataset* para trabajar, el objetivo es poder ver con qué precisión realiza las recomendaciones. Obtener las estadísticas para este caso ha resultado

bastante tedioso e inexacto, debido a que cada usuario almacenado en el *dataset* tiene un 2% de los ítems evaluados. Para poder calcular correctamente las estadísticas sobre un determinado usuario, se deberían entrenar todos los ítems valorados por éste. Es decir, si un usuario ha valorado 200 ítems, el conjunto de datos de entrenamiento correspondería a esos 200 ítems valorados.

A la hora de calcular las estadísticas con Apache Mahout, éste utiliza un conjunto de datos de entrenamiento y de validación que genera aleatoriamente, por lo que los resultados obtenidos tanto para la recomendación como para el *recall* son muy bajos (menos del 10%). Este aspecto implica que de todas las recomendaciones que se le están realizando al usuario, menos del 10% deberían haberse realizado. Para poder calcular métricas como la precisión y el *recall* sería necesario crear un conjunto de entrenamiento que incluyese todos los ítems valorados de un usuario específico. A partir de ello, definida la "ground truth" del usuario, es cuando se podrían calcular dichas métricas.

En resumen, si queremos calcular las estadísticas para un *dataset* como el de MovieLens, se debe entrenar el sistema de recomendación con los ítems valorados por dicho usuario. En este caso, no se ha sabido cómo separar los conjuntos de entrenamiento y de test para pasárselos a Apache Mahout, ya que la función que tiene implementada para calcular las estadísticas genera dichos conjuntos de manera aleatoria, provocando que los valores de recomendación y *recall* sean tan bajos.

D. Nombre y Logo

A continuación se va a realizar un análisis sobre el nombre escogido y el logo generado. Para la elaboración del logotipo, se ha contado con la ayuda de Daniel Ibáñez Parra, licenciado en Ingeniería de Diseño Industrial en la Universidad de Zaragoza.

El nombre escogido para nuestro sistema viene del hecho de ser un sistema que trabaja con documentos de texto, en este caso, noticias. Al ser un proyecto que sigue en desarrollo, se ha considerado que para el futuro pudiese tener soporte para otro tipo de objetos digitales, como por ejemplo vídeos o imágenes. De este pensamiento es cómo surge el nombre de DodoAid (Aid for Documents and Other Digital Objects). Se considera fundamental que la marca tenga un elemento representativo, por lo que en este caso se ha querido asociar el sistema creado con el Dodo, una especie de ave extinta que habitaba en las Islas Mauricio, en el Océano Índico.

Una vez establecido el nombre del sistema y saber que el Dodo va a ser la imagen representativa de éste, se va a explicar el diseño del logo.

El logo de DodoAid se compone de dos partes bien diferenciadas. Por un lado, el isotipo (la parte gráfica del logo) y, por otra parte, el logotipo (la parte textual).

El isotipo de DodoAid se compone a partir de la suma de dos elementos principales:

- Lupa: que aporta y define la parte funcional del sistema acerca de la búsqueda de noticias. Denota inteligencia, curiosidad, búsqueda, detalle, mirar más allá, observar, etc. Es un elemento circular y sin aristas, que aporta dinamismo y quita seriedad al logo. El mango de la lupa es un elemento que se coloca de forma asimétrico por inercia del uso dado a esta unidad gráfica en los medios visuales y como ayuda a identificar mejor el gráfico con el objeto real. Se ha colocado a la derecha por distintos motivos. En primer lugar, porque la mayoría de personas son diestras y por tanto, la visualización de una lupa sostenida por el individuo es desde esta perspectiva. Por otro lado, para generar un ritmo y una dirección más natural en la interpretación del logo, llevando la vista desde la esquina superior izquierda hasta la esquina inferior derecha, dónde está la última unidad de información “Aid”.
- Dodo: El Dodo, tomado como elemento diferenciador de la marca debido a la coincidencia con su nombre, es un animal extinguido ampliamente conocido. Su contorno es fácilmente identificable y la falta de detalles no genera confusión. El hecho de haberlo colocado en negativo sobre el fondo ayuda a dar más presencia al isotipo.

El isotipo en conjunto puede interpretarse como la búsqueda de algo pasado y sobre lo que se desea tener un conocimiento en mayor profundidad. También suscita la búsqueda de detalles sobre algo en concreto, llegando únicamente a ello a través del sistema.

Por otro lado, el logo también está conformado por el logotipo de DodoAid. Éste, a su vez, está conformado por dos unidades de texto. En primer lugar, la palabra "DODO" generada manualmente a partir de formas geométricas que posteriormente han sido redondeadas. Al ser una palabra compuesta por una sílaba repetida, el propio texto crea un ritmo de forma natural. El redondeo y la simplicidad de las formas ayudan a quitar seriedad y genera una unidad de texto amigable y fácilmente legible. En segundo lugar, la palabra "AID", en color negro, es la que aporta la seriedad y la parte técnica del sistema, estando conformada en su mayoría por líneas rectas de distinta dirección.

En cuanto al color corporativo, se ha elegido el PANTONE 653C, un color azul oscuro con buena visibilidad sobre fondo blanco y carácter constitucional y serio a la vez que diferenciador. El negro también se propone como color de posible utilización en sus distintos porcentajes de uso de tinta, ayudando a complementar la información principal, que irá siempre en azul.

Por último, la tipografía corporativa escogida es "Lane – Narrow Regular". Esta tipografía sin serifa es adecuada para títulos y frases de pequeña longitud. Es una tipografía muy regular que mantiene una estética parecida en todos sus caracteres, lo que le confiere mayor unidad a las frases compuestas por ella. Sin embargo, en tamaños muy pequeños o grandes cantidades de texto puede provocar confusión. Funciona muy bien con frases en mayúsculas.

En la siguiente página que se adjunta, se muestra el manual de identidad corporativa, el cual sirve para mostrar y restringir el uso del logotipo y conocer las pautas seguidas para la elaboración del logotipo.

DODO Aid - Imagen de marca

LOGO PRINCIPAL

LOGOS ALTERNATIVOS

PALETA DE COLORES

100%
80%
60%
40%
20%

PANTONE 653 C
CMYK | 85 59 19 4
RGB | 52 96 148

100%
80%
60%
40%
20%

BLACK
CMYK | 0 0 0 100
RGB | 29 29 27

TIPOGRAFÍA

Aa

Lane - Narrow
Regular

A B C D E F G H I J K L
M N O P Q R S T U V
W X Y Z a b c d e f g h i
j k l m n o p q r s t u v w
x y z 0 1 2 3 4 5 6 7 8 9

DODO Aid encuentra noticias adaptadas a tus gustos.

Towards the Development of a Tool to Keep Track of Interesting Information in a Sea of Digital Documents

Short Paper (Work in Progress)

Sergio Ilarri
University of Zaragoza, I3A
Zaragoza, Spain
silarri@unizar.es

Guillermo Azón
University of Zaragoza
Zaragoza, Spain
614627@unizar.es

ABSTRACT

Managing the current data deluge is a great challenge for users. Emails are constantly arriving, notifications of tweets and RSS feeds keep popping out, newspapers and blogs of different types publish potentially-relevant news every day, etc. If a user wants to keep track of certain topics in an efficient way, a careful filtering is needed in order to keep the number of items to review manageable, as otherwise the user may finally give up or just perform some random or casual reading. Automated tools can help the user to perform this initial selection, and thus to minimize the feeling of being overwhelmed that the user may experience.

In this short paper, we present our ongoing work for the development of DodoAid, a recommender of digital objects that attempts to alleviate the current user's overload when he/she wants to follow information about certain topics. Beyond the application of information retrieval and text mining techniques, it can also apply techniques from the field of recommender systems to suggest items that not only fit topics of interest for the user but are also expected to be valuable according to the individual user's preferences, which can be learnt automatically in an implicit way.

CCS CONCEPTS

• **Information systems** → **Information extraction**; **Recommender systems**; • **Computing methodologies** → **Natural language processing**;

KEYWORDS

Information Retrieval, text mining, recommender systems

ACM Reference Format:
Sergio Ilarri and Guillermo Azón. 2018. Towards the Development of a Tool to Keep Track of Interesting Information in a Sea of Digital Documents: Short Paper (Work in Progress). In *CERI 18: 5th Spanish Conference in Information Retrieval, June 26–27, 2018, Zaragoza, Spain.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3230599.3230610

CERI 18, June 26–27, 2018, Zaragoza, Spain

ACM ISBN 978-1-4503-6543-7/18/06…$15.00
https://doi.org/10.1145/3230599.3230610

1 INTRODUCTION

In the current Big Data area, users are frequently overwhelmed with the huge amount of information available in the Internet that may potentially be relevant to them. Thus, it is very challenging to keep track of all the digital objects generated in the Web that may fit the interests of a user. Therefore, applications that can help the user to find the relevant documents are helpful. However, even if the user were able to automatically filter the existing web pages, comments in blogs, tweets, etc., by taking into account whether he/she is interested in that topic or not, the number of documents to revise would still be unmanageable. Thus, beyond the specific topics of the documents, the implicit preferences of the user should also be considered.

In this short paper, we present our ongoing work towards the development of *DodoAid* (*Aid for Documents and Other Digital Objects*), a recommender of digital objects that attempts to alleviate the current user's overload when he/she wants to follow information about certain topics. On the one hand, DodoAid can apply information retrieval and text mining techniques, to automatically detect the topics of documents and perform a pre-filtering based on the types of topics that belong to the user's profile. On the other hand, it can also rank candidate items and suggest the ones estimated to be more appropriate by using techniques from the field of recommender systems. In this way, it can automatically learn the user preferences through the feedback received from the user. In the rest of the paper, we present our proposal. In Section 2, we describe some related work. In Section 3, we present the main ideas behind DodoAid and the intended functionalities. In Section 4, we present the current status of the development, focused on text documents, and some future steps planned.

2 RELATED WORK

A first component of DodoAid is the application of *Information Retrieval* (*IR*) [2, 13] and text mining [10, 20] techniques. Specifically, it has functionalities to pre-process digital documents and extract relevant data, such as keywords and named entities (including geographic locations). Besides, it can transform those documents into a vector space model, thus structuring the content for later analysis and data mining. Finally, classification algorithms can be applied to categorize the documents according to a predefined list of possible categories or topics.

Another key building block of DodoAid is the application of techniques from the field of *Recommender Systems* (*RS*) [4, 17, 18], which offer recommendations to the user based on estimations of the expected relevance of items for that user. Personalized recommender

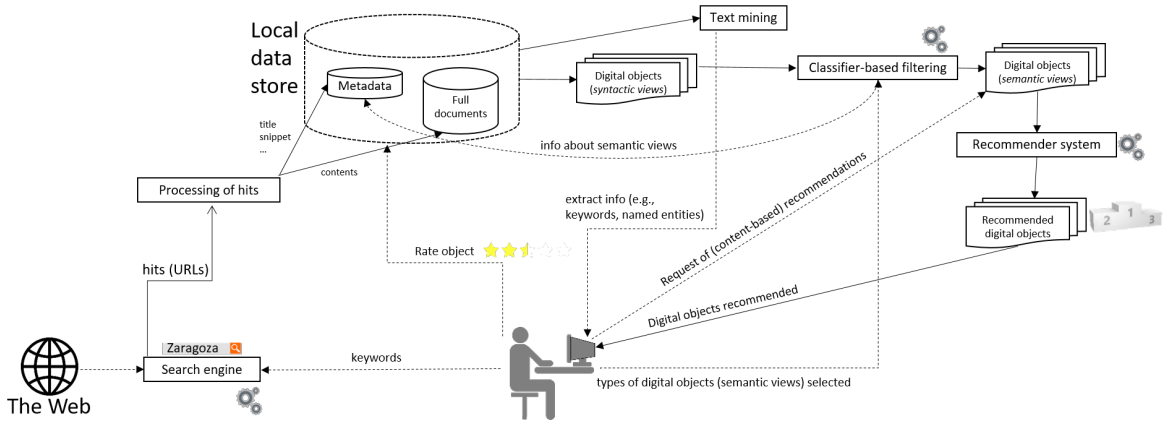

Figure 1: Main workflow of DodoAid

systems try to learn automatically the preferences of each user, usually by exploiting feedback such as numeric ratings provided by the user for the items consumed. From a predictive analytics perspective, we can see a recommender system as a system that estimates the rating that a user would provide for a given item, in order to recommend the item if the rating predicted is high enough; thus, it operates in a two-dimensional space $User \times Item \rightarrow Rating$.

RS have been proposed for the recommendation of items of very diverse types, including news [1, 6–9, 11, 12, 14, 16, 19, 21] and web pages [3, 5]. For example, [1] presents a recommendation approach that takes into account the location of the user and considers the serendipity (casual discovery) as an important objective. The approach presented in [6] focuses on the recommendation of news in a website and it is based on tracking the sequence of articles read when a user browses the website. The *PEN recsys* system [7] is also constrained to documents provided by a specific website (a newspaper website). Similarly, [8] evaluates several approaches for the recommendation of news in the *swissinfo.ch* website. In [11, 16], the focus is on *Usenet* news and the authors advocate performing a cost-benefit analysis to decide about the potential reading of an article, based on its *predictive utility*. Rather than exploiting rating information, the approach presented in [9] is based on the use of association rules obtained from interactions of the users with the system: it exploits the correlations between values of a number of attributes (*category*, *keyword*, *income*, *age*, *geo_user*, *geo_user_zip*, *weather*, *device*, *isp*, *browser*, *recommended_id*, and *position*) and reading certain items. As a final example, the mobile web news recommendation system called *MONERS* [12] focuses on mobile users (like [19, 21]) and highlights the importance of considering the *recency* of the news articles.

Regarding these existing approaches, in DodoAid we combine RS with IR techniques: the user plays an active role by searching digital objects of different types (not necessarily news), through the submission of keyword-based queries, and afterwards the objects retrieved go through the recommendation module, which ranks the hits obtained to highlight the most relevant ones to the user according to the preferences that the system learns along time.

3 FUNCTIONALITIES OF THE TOOL

Figure 1 shows the main workflow of the DodoAid tool and some examples of possible user interactions. Some basic aspects follow:

(1) The user can submit a keyword-based query against a web search engine and retrieve a number of hits that match the query (syntactic matching based on the search engine's indexing capability). Specifically, in our current prototype, we use Google, although any other search engine with an appropriate search API could be integrated. A snapshot showing some search results is shown in Figure 2.

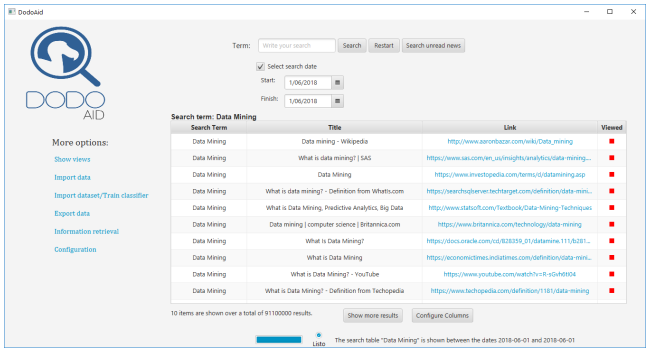

Figure 2: Keyword-based searching of documents

(2) The hits retrieved are processed, at least to obtain basic metadata such as the title and snippet; optionally, the user can also request to download whole documents. Besides, all the hits obtained with a specific keyword-based query define automatically a *syntactic view* that the user can handle in the application (see Figure 3). Syntactic views are organized in a hierarchical fashion similar to folders in a file system: the user can perform certain operations over the views, like changing their names, moving them in the hierarchy of views considered, deleting them, retrieving their contents (i.e., the digital objects that belong to each view), etc.

(3) A text mining process is applied, which includes transforming the available text about the digital objects (title, snippet,

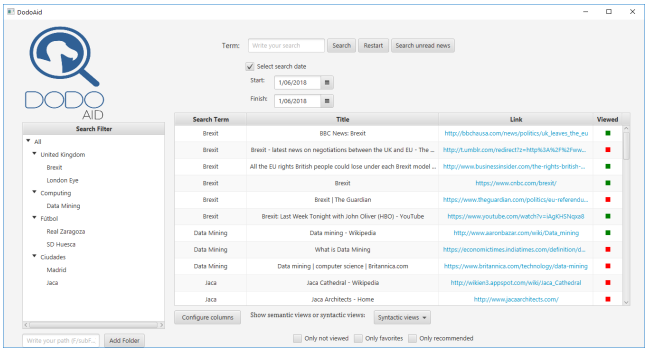

Figure 3: Syntactic views defined by previous searches

and/or body) according to the vector space model, after removing stopwords and applying stemming, as well as identifying *keywords* and *named entities* in the text. A number of parameters can be configured to perform these transformations. For example, for the identification of keywords, the user can select metrics such as the *TF* (*Term Frequency*) or *TF-IDF* (*Term Frequency – Inverse Document Frequency*), as shown in Figure 4. The user can visualize the information retrieved and geo-position the digital objects on a map, based on the geographic locations identified. Besides, the system can show extra information about the named entities by trying to link them to concepts in *DBpedia* (http://wiki.dbpedia.org/).

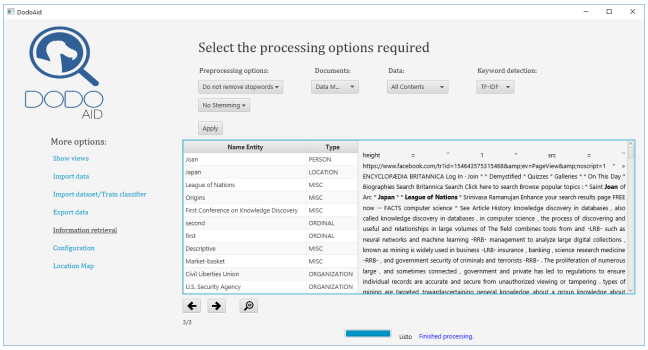

Figure 4: Processing the documents

(4) The digital objects can go through a classifier, that applies a classification algorithm to determine the most likely category they belong to. As a result of this classification-based pre-filtering, the digital objects are classified in a set of views that may be different from the original ones, and thus we call them *semantic views* as opposed to the original *syntactic views*. Obviously, in order to apply the classification algorithm, a previous training phase must take place: for that purpose, a set of documents must be previously labelled by an expert user according to the right category. DodoAid allows importing existing datasets of documents stored in a hierarchy of directories (for this purpose, the *TextDirectoryLoader* class provided by Weka is used), which is a useful functionality to facilitate the training of the classifier (besides, it might also be used in the future to enable testing different recommendation and classification approaches on top of well-known datasets). Two observations are relevant:
   - Adding a new category (semantic view) to consider would require retraining the model and adding labelled examples of digital objects in that category. Therefore, in a practical scenario, the user will probably want to define a set of semantic views that do not correspond to all the possible syntactic views. For example, a user interested in programming could define (through keyword-based searches) syntactic views such as "Java", "web programming", "IDEs", etc., but a single semantic view called *Programming* that includes all those syntactic views.
   - Applying the classification over the whole set of digital objects available to the user could be a costly process. Therefore, it is expected that the user will select only a subset of digital objects (such as objects belonging to one or more syntactic views selected), for example in order to filter out digital objects that are within a certain syntactic view but that do not really belong to the required semantic view. The result of the classification can be stored as part of the metadata in the local data store, but it must be taken into account that a change in the model (achieved through re-training) or in the semantic views defined could render a previous classification invalid.

(5) Then, a recommender system exploits information about the feedback of the user regarding digital objects that he/she has consumed in the past (explicit ratings, visualizations of the URLs in a browser, and/or adding the document to the list of favorites), in order to suggest the digital objects that are more relevant to the user for the semantic view he/she is interested in. For this purpose, a content-based recommendation system [15] is used.

Other minor functionalities include the capability to directly import documents from a list of URLs, which can be useful for example if a user wants to import his/her library of documents from other applications such as Pocket (https://getpocket.com/) or Instapaper (https://www.instapaper.com/).

Regarding the current implementation of the DodoAid prototype, it should be noted that for the moment we have focused on text documents. Besides, we can mention the use of the following technologies:

- The programming language used is *Java* (http://www.oracle.com/technetwork/java/javase/downloads). We have used the latest version available (JDK 8u171). For the development of the GUI, we have used *JavaFX* instead of *Swing*.
- As a search engine, our current prototype accesses Google by using the JSON API provided by Google (https://developers.google.com/custom-search/). A difficulty found when using the API is that, even though it allows filtering hits by date, no date metadata seem to be attached to the hits retrieved (or, at least, they are not accessible through the API).
- For the representation of maps, and the documents as markers on the maps, we have used *GMapsFX* (https://rterp.github.io/GMapsFX/), which provides a pure-Java wrapper to the

Google Map's JavaScript API (https://developers.google.com/maps). Another alternative could be to use *OpenStreetMap* (https://www.openstreetmap.org/).
- To obtain information from DBpedia, we submit SPARQL queries to a SPARQL endpoint for DBpedia (http://es.dbpedia.org/sparql). As a support, we have used *Jena* (https://jena.apache.org/) and its query engine ARQ.
- For text processing, we are using *Stanford CoreNLP* (https://stanfordnlp.github.io/CoreNLP/). We have performed tests with the available models for two languages: English (the corresponding jar file for this model takes up 1,04 GB) and Spanish (359,8 MB). Besides Stanford CoreNLP, we are also applying *Exude* (https://exude.herokuapp.com, https://github.com/uttesh/exude) to filter stopwords and for stemming purposes.
- To program the recommender, we are using *Apache Mahout* (https://mahout.apache.org/), which is a Java library for machine learning that incorporates functionalities for the development of recommender systems.
- For classification, we currently consider the classification algorithms available in *Weka* (https://www.cs.waikato.ac.nz/ml/weka/), although other alternatives (including Mahout) could also be used.
- For the storage of documents and the associated metadata, we use *SQLite* (https://www.sqlite.org/), as it is lightweight and enough for our current purposes.

Furthermore, we envision the use of other technologies in the future, for comparative purposes and also to extend the current functionalities. For example, the current prototype looks for named entities in DBpedia, but specific repositories such as *GeoNames* (http://www.geonames.org/), which is interlinked with DBpedia, could be queried for named entities representing geographic locations.

4 CURRENT STATUS AND FUTURE STEPS

In this paper, we have described the current status of the development of DodoAid, a tool to help users to keep track of digital objects such as documents available on the web. It combines techniques from the fields of information retrieval, text mining/processing, and recommender systems. The tool described is a work in progress and many improvements and extensions are possible.

Important extensions should focus on improving the usability and visual aspect of the tool, to empower the user to work with his/her digital objects and perform a variety of tasks with them in an easy way, extending the current parametrization options available. Besides, evaluating the tool, and more specifically its capabilities to alleviate the information overload experienced by users, would be an important step to perform once the development has been completed in all its extent. Another relevant and non-trivial extension would be to add support for digital objects other than text documents (e.g., images or videos). Finally, the tool could be extended with a collaborative filtering component, assuming a scenario where users are willing to share their ratings with others; however, applying collaborative filtering recommendation techniques may be a challenge, given the expected sparseness of the resulting user-item matrix (i.e., for each document the percentage of users who have rated it will probably be very low, as there is an immense number of documents on the web).

ACKNOWLEDGMENTS

Work supported by the project TIN2016-78011-C4-3-R (AEI/FEDER, UE) and DGA-FSE (COSMOS research group). We thank Daniel Ibáñez Parra for his help with the design of the DodoAid logo.

REFERENCES

[1] Yonata Andrelo Asikin and Wolfgang Wörndl. 2014. Stories Around You: Location-based Serendipitous Recommendation of News Articles. In *Second International Workshop on News Recommendation and Analytics (NRA)*, Vol. 1181. CEUR Workshop Proceedings, 1–8.
[2] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. 1999. *Modern Information Retrieval*. Vol. 463. ACM Press New York.
[3] Marko Balabanović. 1997. An Adaptive Web Page Recommendation Service. In *First International Conference on Autonomous Agents (AGENTS)*. ACM, 378–385.
[4] Jesús Bobadilla, Fernando Ortega, Antonio Hernando, and Abraham Gutiérrez. 2013. Recommender Systems Survey. *Knowledge-Based Systems* 46 (2013), 109–132.
[5] Sara Cadegnani, Francesco Guerra, Sergio Ilarri, María del Carmen Rodríguez-Hernández, Raquel Trillo-Lado, Yannis Velegrakis, and Raquel Amaro. 2017. Exploiting Linguistic Analysis on URLs for Recommending Web Pages: A Comparative Study. In *Transactions on Computational Collective Intelligence XXVI*. Lecture Notes in Computer Science (LNCS), Vol. 10190. Springer, 26–45.
[6] 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.
[7] Florent Garcin and Boi Faltings. 2013. PEN recsys: A Personalized News Recommender Systems Framework. In *International News Recommender Systems Workshop (NRS)*. ACM, 3–9.
[8] 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.
[9] Christián Golian and Jaroslav Kuchar. 2017. News Recommender System based on Association Rules at CLEF NewsREEL 2017. In *Eighth International Conference of the CLEF Initiative*, Vol. 1866. CEUR Workshop Proceedings.
[10] Andreas Hotho, Andreas Nürnberger, and Gerhard Paaß. 2005. A Brief Survey of Text Mining. *LDV Forum - GLDV Journal for Computational Linguistics and Language Technology* 20, 1 (May 2005), 19–62.
[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] 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.
[13] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Vol. 1. Cambridge University Press Cambridge.
[14] Tim Miranda, Mark Claypool, Anuja Gokhale, Tim Mir, Pavel Murnikov, Dmitry Netes, and Matthew Sartin. 1999. *Combining Content-Based and Collaborative Filters in an Online Newspaper*. Computer Science Technical Report WPI-CS-TR-99-16. Worcester Polytechnic Institute. 1–11 pages.
[15] Michael J. Pazzani and Daniel Billsus. 2007. Content-based recommendation systems. In *The Adaptive Web: Methods and Strategies of Web Personalization*, Peter Brusilovsky, Alfred Kobsa, and Wolfgang Nejdl (Eds.). Lecture Notes in Computer Science (LNCS), Vol. 4321. Springer, Chapter 10, 325–341. https://doi.org/10.1007/978-3-540-72079-9{_}9
[16] 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.
[17] Paul Resnick and Hal R. Varian. 1997. Recommender Systems. *Commun. ACM* 40, 3 (1997), 56–58.
[18] Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor. 2011. *Recommender Systems Handbook*. Springer.
[19] 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.
[20] Ah-Hwee Tan. 1999. Text mining: The state of the art and the challenges. In *PAKDD Workshop on Knowledge discovery from Advanced Databases (KDAD)*. 71–76.
[21] Kam Fung Yeung and Yanyan Yang. 2010. A Proactive Personalized Mobile News Recommendation System. In *Developments in E-systems Engineering (DESE)*. IEEE, 207–212.