Revista Cubana de Ciencias Informáticas
Vol. 10, No. 2, abril-junio, 2016
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 57- 72

Tipo de artículo: Artículo original
Temática: Inteligencia artificial
Recibido: 25/11/2014 | Aceptado: 24/11/2015

Recuperación de información para artículos científicos soportada en el agrupamiento de documentos XML

Information retrieval for scientific papers supported in the XML documents clustering

Damny Magdaleno[1*], Ivett E. Fuentes[1], Michel Cabezas[2], María M. García[1]

[1] Universidad Central "Marta Abreu" de Las Villas. Carretera a Camajuaní km 7½. Santa Clara, Villa Clara, Cuba. {dmg, mmgarcia}@uclv.edu.cu, ivett@uclv.cu

[2] XETIC. Calle 296-A e/ ave 207 y 203. Boyeros, La Habana, Cuba. michelc@uclv.cu

[*] Autor para correspondencia: dmg@uclv.edu.cu

Resumen

Cada día más datos electrónicos en formato semiestructurado, específicamente XML, se encuentran disponibles en el *World Wide Web*, intranets corporativas, y otros medios de comunicación. Por tal motivo la gestión de información se vuelve cada vez más compleja y desafiante, sobre todo porque las colecciones de documentos generalmente son heterogéneas, grandes, diversas y dinámicas. Superar estos desafíos es esencial para dar a los científicos mejores condiciones de administrar el tiempo necesario para procesar la información científica. En el laboratorio de Inteligencia Artificial de la Universidad Central "Marta Abreu" de las Villas se han obtenido varios sistemas que permiten manipular la información, como: SATEX, GARLucene y LucXML, este último da tratamiento de forma específica a los documentos XML, aunque no garantiza gestionar los documentos desde un repositorio en la red. En este trabajo se implementó una herramienta Web que usa las técnicas de recuperación inteligente, soportada en un algoritmo de agrupamiento de documentos XML que combina el contenido y la estructura existente en estos. Los principales resultados son: (1) el uso de la metodología para el agrupamiento de los documentos recuperados; (2) la utilización de herramientas especializadas en recuperación de información y manipulación de documentos; (3) al evaluar el sistema con datos representativos se obtuvieron resultados favorables lo que corrobora la validez de la implementación realizada.

Palabras clave: Recuperación de Información, Agrupamiento, XML

Revista Cubana de Ciencias Informáticas
Vol. 10, No. 2, abril-junio, 2016
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 57- 72

Abstract

Every day more electronic data in semistructured format, specifically XML, are available on the World Wide Web, intranets, and other media. By this, the information management becomes increasingly complex and challenging, especially since document collections are generally heterogeneous, large, diverse and dynamic. Overcoming these challenges is essential to give scientists better conditions to manage the time required to process scientific information. In the Artificial Intelligence Laboratory of Universidad Central "Marta Abreu" de Las Villas, they have obtained several systems that allow to manipulate information such as: SATEX, GARLucene and LucXML, the last one treats specifically to XML documents although it does not guarantee to manage the documents from a repository in the network. In this paper, a Web tool that uses smart recovery techniques, supported by a clustering algorithm of XML documents that combine existing content and structure these are implemented. The main results are: (1) the use of the methodology for the clustering of documents retrieved; (2) the use of specialized tools in information retrieval and document manipulation; (3) to evaluate the system with representing data, favorable results were achieved which confirms the validity of the implementation done.

Keywords: *Information Retrieval, Clustering, XML*

Introducción

La creación y diseminación de información en el *World Wide Web*, intranets corporativas, y otros medios de comunicación es soportada por un número creciente de herramientas, sin embargo, mientras la cantidad de información disponible está continuamente creciendo, la habilidad de procesarla y asimilarla no presenta el mismo ritmo de crecimiento. Este hecho hace que la gestión de información científica sea cada vez más compleja, al ser las colecciones textuales heterogéneas, grandes y dinámicas. Vencer estos desafíos es esencial para proporcionar a los científicos mejores condiciones de trabajo que aseguren una mayor productividad e inviertan un tiempo menor en procesar la información requerida, lo cual constituye la motivación principal de este trabajo. El conocimiento se puede gestionar de diversas formas y hacerlo requiere de la integración de varias áreas del saber: descubrimiento de conocimiento en bases de datos, minería de datos y de textos. Específicamente esta última integra la recuperación y extracción de información, el análisis de textos, el resumen, la categorización, la clasificación, el agrupamiento, la visualización, la tecnología de bases de datos, el aprendizaje automático y la minería de datos (Aggarwal and Zhai, 2012).

Particularmente, la Recuperación de Información (RI) abarca el conjunto de acciones, métodos y procedimientos para la representación, almacenamiento, organización y recuperación de la información; su objetivo fundamental es obtener

los documentos ordenados en función del grado de relevancia[1], para responder a las necesidades del usuario (Grossman and Frieder, 2012). Un Sistema de RI (SRI) es un programa que implementa un modelo de RI, posee tres componentes principales: la base de datos documental, el subsistema de consultas y el mecanismo de recuperación (Croft et al., 2010).

Por su parte, el agrupamiento permite organizar la información obtenida y descubrir nuevo conocimiento a partir del resultado de un proceso de recuperación de información (Afonso and Duque, 2014; Amoli and Sh, 2015; Yau et al., 2014; Guan et al., 2014; Shankar, 2012). El agrupamiento es una tarea del aprendizaje no supervisado que tiene como objetivo descomponer el conjunto de datos, de forma tal que los objetos que pertenecen al mismo grupo sean tan similares como sea posible y los objetos que pertenecen a grupos diferentes sean tan disimilares como sea posible. El análisis de grupos es una herramienta para descubrir una estructura previamente oculta en los datos, asumiendo que existe un agrupamiento natural o cierto en ellos. Sin embargo, la asignación de los objetos a las clases y la descripción de esas clases son desconocidas (Kruse et al., 2007).

La información que aparece en la web es variada, siendo actualmente la de formato semiestructurado la más utilizada (Algergawy et al., 2011). Ejemplos de estos formatos son AIML, WSDL y XML. Los documentos escritos en formato XML (*Extensible Markup Language*), el cual es un metalenguaje desarrollado por W3C[2] tienen una estructura jerárquica autodescriptiva de información, formada por átomos, elementos compuestos y atributos. Son extensibles, con estructura de fácil análisis y procesamiento, lo que le ha permitido convertirse en el formato estándar de intercambio de datos entre las aplicaciones Web (Piernik et al., 2015). Este hecho ha sido motivo para explotar la estructura de estos documentos en el proceso de recuperación de documentos relevantes (Watanabe et al., 2013). Por tanto, al enfrentarse a este tipo de colecciones los SRI se enfrentan a nuevos desafíos, entre estos: los usuarios en ocasiones requieren que el sistema devuelva como resultado de sus búsquedas partes de documentos y no documentos completos como es usual en los SRI clásicos; paralelo a este problema aparece el problema de cuál parte del documento indexar. Por otra parte cuando los algoritmos de agrupamiento se enfrentan a documentos XML, se clasifican principalmente en tres grupos: los que se centran solo en el contenido de los documentos (Algergawy et al., 2011), realizando un análisis solamente léxico, o incluyendo elementos sintácticos o semánticos en el estudio; existen otros trabajos que solo utilizan la estructura de los documentos para realizar el agrupamiento (Watanabe et al., 2013; Costa et al., 2013), considerando que esta juega un papel importante en el agrupamiento para ciertas aplicaciones específicas y los que combinan ambas

[1] Se refiere a la relevancia como una medida del grado de correspondencia del documento a la consulta realizada al sistema.
[2] http://www.w3c.org

componentes: estructura y contenido; lo cual, constituye un nuevo desafío, ya que la mayoría de los enfoques existentes no utilizan estas dos dimensiones dada su gran complejidad (Tien T., 2007).

Una primera variante muy sencilla de combinar contenido y estructura es mezclar en una representación Espacio Vectorial (*Vector Space Model;* VSM) (Salton et al., 1975) el contenido y las etiquetas del documento y aplicar un algoritmo de agrupamiento conocido. Otros trabajos realizan extensiones a la representación VSM, llamadas C-VSM y SLVM (Doucet and AhonenMyka, 2002). En (Tekli and Chbeir, 2011) fue propuesto un marco para trabajar con similitudes por estructura y por semántica. Este marco consiste de cuatro módulos principales para descubrir las estructuras comunes a través de los subárboles, identificando los subárboles con parecidos semánticos, aquí utilizan los costos basados en las operaciones de la distancia *tree-edit* (Chen and Zhang, 2012) para el cálculo de la distancia basada en este enfoque. En (Pinto et al., 2009) utilizaron técnicas no supervisadas con la intención de agrupar documentos de una colección de gran tamaño. Este enfoque utiliza un algoritmo de agrupamiento iterativo en un proceso de agrupamiento recursivo sobre subconjuntos de la colección completa. En (Magdaleno et al., 2015a) se propone una metodología para la aplicación del agrupamiento de documentos XML, combinando la estructura y el contenido, tomando el resultado de un proceso de recuperación de información (Buettcher et al., 2010; Chowdhury, 2010). Las salidas son grupos homogéneos de documentos afines, el resumen de cada documento, los documentos más representativos de cada grupo y la calidad del agrupamiento; garantizando el control para la evaluación de los resultados. *OverallSimSUX* logra capturar la similitud entre una pareja de documentos, teniendo en cuenta la relación existente entre las secciones de estos como colecciones independientes, a su vez trata los documentos como un todo.

En el Centro de Estudios de Informática (CEI) de la Universidad Central “Marta Abreu” de las Villas (UCLV) se han propuesto los sistemas para la gestión de la información y el conocimiento [SATEX (Arco et al., 2008b), GARLucene (Arco et al., 2008a)] que implementan el esquema propuesto por (Arco, 2009) para la confección de sistemas gestores de información en dominios textuales. Los mismos brindan amplias ventajas para la gestión de la información y del conocimiento, pero no incorporan un algoritmo de agrupamiento capaz de explorar la estructura de documentos XML. Por su parte, el sistema LucXML (Magdaleno et al., 2013) implementa la metodología propuesta en (Magdaleno et al., 2015a) y (Fuentes, 2013), por lo que permite el tratamiento de los documentos XML a partir de un algoritmo de agrupamiento que utiliza su estructura y contenido, sin embargo, el mismo no garantiza gestionar los documentos desde un repositorio en la red. Además, en el Centro de Estudios de Informática existe un gran número de artículos científicos de variados temas. Se mantiene el desafío de dar a los científicos mejores condiciones en su trabajo investigativo, de ahí que el objetivo general de este trabajo es implementar un esquema de recuperación inteligente de información soportado en el agrupamiento de documentos XML de artículos científicos mediante una herramienta Web.

Revista Cubana de Ciencias Informáticas
Vol. 10, No. 2, abril-junio, 2016
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 57- 72

Materiales y métodos o Metodología computacional

El proceso completo de Recuperación de Información consistirá en:

- Obtener mediante la indexación de una colección de documentos, el conjunto de términos asociados a cada documento.
- Obtener, la representación textual de la colección en forma de palabras claves o términos de indexación.
- Comparar cada uno de los documentos indexados con la consulta realizada, obteniendo en algunos casos el grado con el que el documento satisface a la consulta, aquellos que la satisfagan completamente.
- Presentar al usuario la salida del proceso de búsqueda que permite evaluar la salida y comprobar que es satisfactoria para su necesidad de información.

Para reducir el tiempo que los usuarios asimilan el resultado de la recuperación, se requiere que salida del sistema tenga algún nivel de organización, con este fin, en este trabajo se realiza un agrupamiento de la colección recuperada.
El procedimiento general que implementa esta herramienta cuenta de tres módulos principales. En la siguiente sección se exponen estos tres módulos, la puesta en práctica de algunos de los principios de la RI mencionados, así como las herramientas utilizadas para la elaboración del sistema implementado.

Implementación de RISADXML

En la Figura 1 se muestra un diagrama que contiene los tres módulos principales que se implementaron en el sistema para la **R**ecuperación de **I**nformación **S**oportado en el **A**grupamiento de **D**ocumentos **XML** (RISADXML); estos son: (1) Creación de índices y recuperación del corpus de documentos XML, (2) Representación de la colección y (3) Agrupamiento General a partir de la matriz de similitud basada en el cálculo de la función *OverallSimSUX*. Para la implementación se utilizó una arquitectura cliente/servidor; a través del cliente Web se logra el acceso a los paquetes implementados en la parte del servidor, destacándose el proceso de recuperación de la información y el agrupamiento de los documentos recuperados.

A continuación, se mencionan las clases fundamentales contenidas en la parte cliente, seguido de la explicación del funcionamiento de los módulos implementados en la parte del servidor.

Revista Cubana de Ciencias Informáticas
Vol. 10, No. 2, abril-junio, 2016
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 57- 72

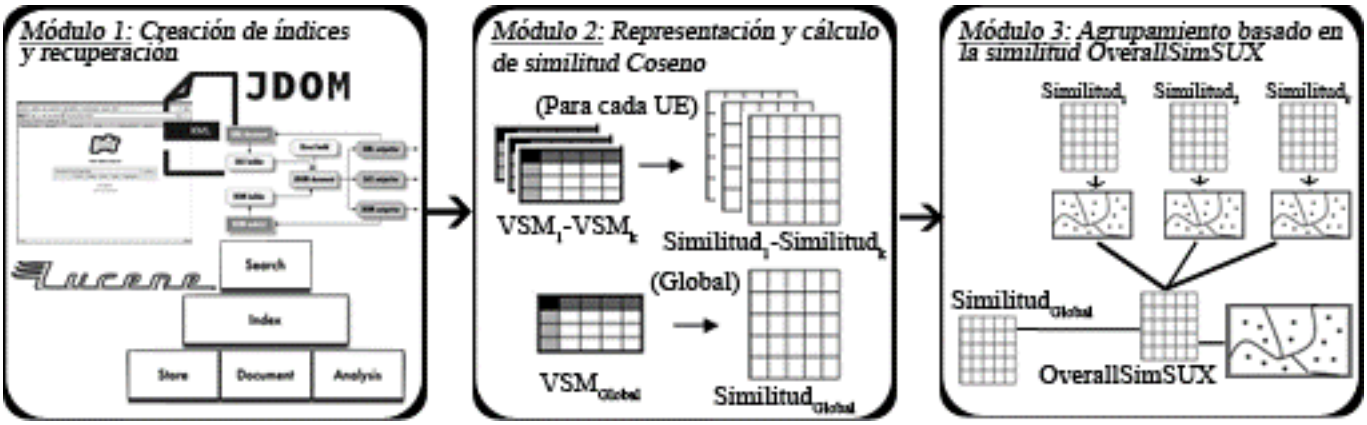

Figura 1. Módulos del sistema RISADXML

Cliente

- *MainLayout* y *WinConfiguration*: Clases visuales donde están todos los componentes que le son mostrados a los usuarios.
- *Controller*: Se utiliza para controlar la interconexión entre el cliente y el servidor, las llamadas a los métodos utilizados en el servidor y las respuestas de este al cliente.
- *RisadXML*: Encargada de iniciar la aplicación, es la primera clase que se ejecuta.
- *RisadXMLService*: Define los servicios de Llamada a Procedimientos Remotos (RPC) utilizados en la aplicación.
- *RisadXMLServiceAsync*: Esta interfaz es utilizada para la interconexión entre el cliente y el servidor en las RPC utilizadas en la ejecución de la aplicación.
- *ResultRecord*: Define cómo se van a mostrar los resultados de la búsqueda.

En la implementación se utilizó *GWT*[3], *framework* creado por Google que permite crear aplicaciones *AJAX*[4] en el lenguaje de programación *Java* que son compiladas posteriormente por GWT en código JavaScript ejecutable optimizado que funciona automáticamente en los principales navegadores.

Servidor

Módulo 1: Creación de índices y recuperación del corpus de documentos XML

[3] http://code.google.com/webtoolkit/

[4] http://ajax.asp.net/

En el proceso de RI, la indexación y la búsqueda son pasos claves. Para estas operaciones se utilizó *Lucene*[5], biblioteca implementada en *Java*, de código abierto. Permite fácilmente la integración con cualquier aplicación (Artiles, 2011) por lo que ha sido integrada a las funciones de búsquedas de muchas aplicaciones web y de escritorio; teniendo como factor clave su aparente simplicidad, pues realmente cuenta con complejos algoritmos que implementan técnicas de RI de última generación (Chriss A. and Zitting, 2012). Además, para utilizarla no es necesario un conocimiento profundo acerca de cómo se indexa y recupera información.

Indexación

Lucene crea de forma interna un índice compuesto de documentos; para cada uno de estos documentos, define un conjunto de campos con el texto. Una herramienta utilizada en este trabajo, que facilita la confección de los campos, es el API *Jdom*[6], especializada en la manipulación de documentos en formatos XML. Esta biblioteca permite identificar de forma natural los elementos existentes en un documento XML (Hatcher et al., 2009). Específicamente en este trabajo es muy útil para identificar las secciones de los documentos a agrupar, (denominadas en este trabajo Unidades Estructurales, UE) por ejemplo, en un artículo científico: resumen, introducción, materiales y métodos, entre otros y así poder extraer exactamente el texto contenido en una UE específica.

Otro de los motivos por los que se escogió *Lucene* es que, para la creación de los índices de términos, trabaja con la representación VSM, que es utilizada en el modelo implementado para realizar las representaciones de los documentos a agrupar. Para el preprocesamiento de la colección se utilizaron varias clases, entre estas: *StandardAnalyzer*, especializada en normalizar los *tokens* extraídos; *LowerCaseFilter*, convierte los *tokens* a minúsculas y *StopFilter* elimina palabras de parada (Singh and Siddiqui, 2012, Zaman et al., 2011, Amarasinghe et al., 2015). Adicionalmente, *Analyzer* obtiene las raíces de las palabras mediante heurísticas, y tratar la sinonimia y polisemia. La Figura 2 muestra las clases encargadas del proceso de indexación.

Recuperación

El proceso de búsqueda se realiza a partir del índice construido. Para ello se utilizaron las clases: *search* y *queryParser* de la biblioteca *Lucene*.

[5] http://lucene.apache.org/

[6] http://www.jdom.org/

Revista Cubana de Ciencias Informáticas
Vol. 10, No. 2, abril-junio, 2016
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 57- 72

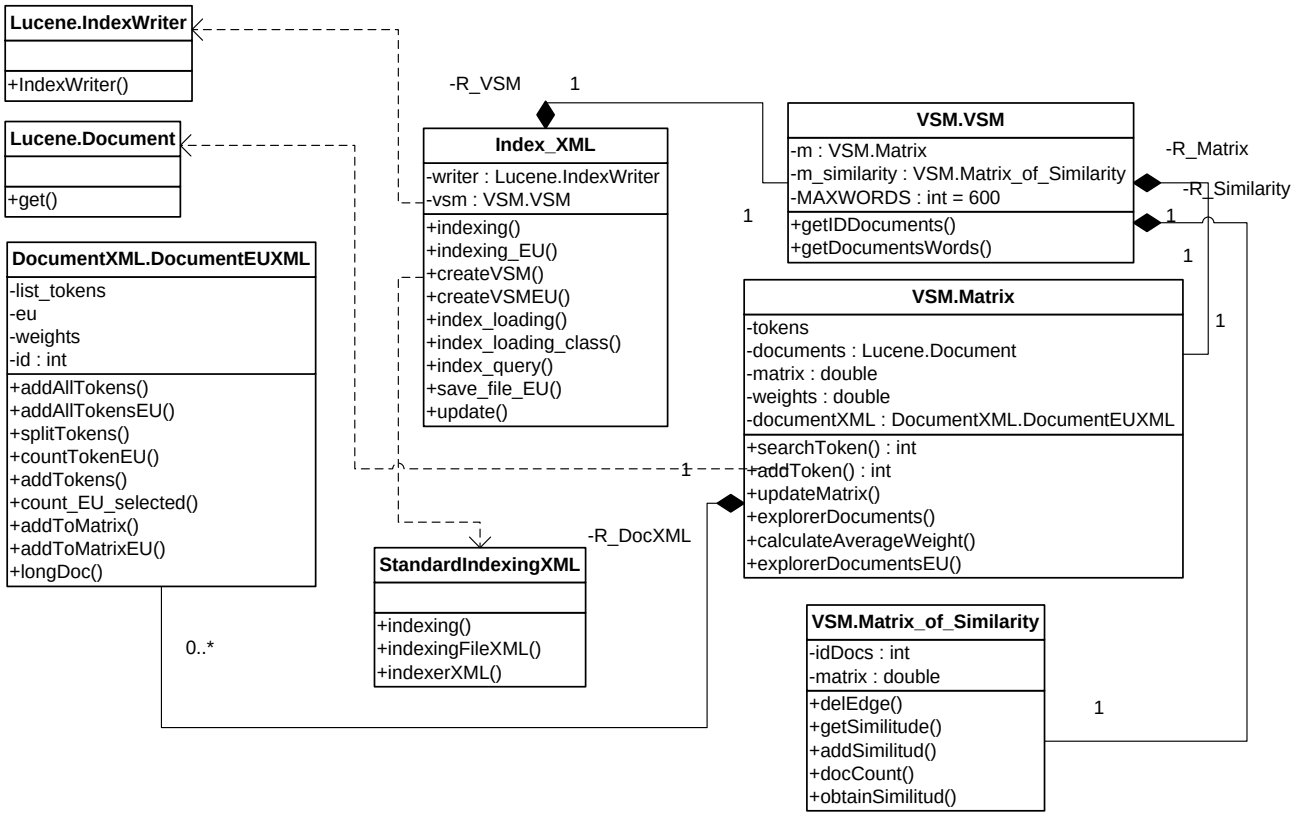

Figura 2. Diseño de clases relacionadas con el proceso de indexado

El procedimiento general empleado para la recuperación consistió en obtener la consulta indicada por el usuario y realizar la búsqueda sobre el índice a partir de las clases *IndexSearcher*, *QueryParser* y *Query*; de manera que los resultados obtenidos por la consulta son almacenados en un objeto de la clase *Hits*.

La clase *IndexSearcher* es usada para la búsqueda de documentos en un índice, provee una gran cantidad de métodos de búsqueda, entre los utilizados se encuentra *SpecificTerm*. La clase *QueryParser* de *Lucene* incluye métodos para la manipulación de expresiones regulares; instanciada suministrándole el nombre del campo sobre el que se realizará la búsqueda y un analizador, usado para procesar las condiciones de búsquedas impuestas. Esta clase contiene el método *parse* que necesita una consulta que contendrá la expresión a procesar

Módulo 2: Representación de la colección

El modelo escogido para agrupar los documentos XML realiza dos tipos de representaciones: *Representación I* asociada a cada UE y *Representación II* que se obtiene de toda la colección. Específicamente, para la *Representación I* se

Revista Cubana de Ciencias Informáticas
Vol. 10, No. 2, abril-junio, 2016
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 57- 72

construye la matriz VSM clásica, que contiene en sus filas el índice de cada término obtenido y los documentos de la colección en sus columnas, las celdas representan la frecuencia de aparición de cada término en la UE del documento que se procesa. La *Representación II* utiliza la misma estructura que la *Representación I*, pero en cada celda almacena la frecuencia pesada por la UE donde se encuentra el término. El cálculo de la frecuencia pesada así como la forma de calcular el peso de las UE se observan en las ecuaciones 1 y 2 (Magdaleno et al., 2011); donde, tf_{ij} es la frecuencia pesada del término *i* en el documento *j*, w_{kj} es el peso de la unidad estructural *k* en *j* y $frecuencia_{ik}$ es la frecuencia de aparición de *i* en *k*.

$$tf_{ij} = \sum_{k=1}^{n} (w_{kj} \times frecuencia_{ik}) \quad (1)$$

$$w_{kj} = \left(e^{\left(-L_{SU}/L_{Doc}\right)}\right)^{pot} \quad (2)$$

Módulo 3: Agrupamiento General a partir de la matriz de similitud basada en el cálculo de la función OverallSimSUX.

Para cada representación resultante se calcula la matriz de similitud utilizando como medida la similitud coseno, ecuación 3. Posteriormente se genera un agrupamiento para cada *Representación I* a partir de la similitud asociada.

$$S_{coseno(o_i,o_j)} = \frac{\sum_{k=1}^{m}(o_{ik} * o_{jk})}{\sqrt{\sum_{k=1}^{m} o_{ik}^2 * \sum_{k=1}^{m} o_{jk}^2}} \quad (3)$$

Para el agrupamiento final se calcula la matriz de similitud global utilizando la medida de similitud *OverallSimSUX*, ver ecuación 4, esta se obtiene a partir del resultado del agrupamiento realizado a cada *Representación I* y la matriz de similitud coseno asociada a la *Representación II*. Finalmente se realiza el agrupamiento general, utilizando la matriz de similitud confeccionada con *OverallSimSUX*.

$$S_{OSSUX}(i,j) = \frac{\sum_{k=1}^{n}(w_k \times \lambda_k(i,j)) + S_g(i,j)}{\sum_{k=1}^{n}(w_k) + 1} \quad (4)$$

Para realizar cada agrupamiento se utilizó el algoritmo de agrupamiento *K*-Star (Shin and Han, 2003). Como resultado se obtiene una partición de la colección inicial en grupos homogéneos de documentos.

Revista Cubana de Ciencias Informáticas
Vol. 10, No. 2, abril-junio, 2016
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 57- 72

Resultados y discusión

En este epígrafe se presenta el proceso de verificación del sistema. Una descripción de los requerimientos mínimos para su uso y finalmente una descripción a nivel de usuario con el propósito de explicar cómo utilizarlo.

Requerimientos de hardware

Para su funcionamiento, el sistema debe encontrarse instalado en un servidor de aplicaciones, como *Apache Tomcat*; debe contar con un hardware de respaldo, los requerimientos mínimos y software se especifican a continuación:

Parte del cliente

- Procesador Intel Pentium IV/1.5 GHz.
- 512 Mb de memoria RAM.
- Sistema operativo Windows XP o superior, Linux.
- Conexión mediante red al servidor de aplicaciones.
- Puede usarse como navegador web Firefox u Opera, se recomienda Firefox instalando el *plugin* de *Macromedia Flash Player* 10.

Parte servidor

- Procesador Intel Pentium IV/1.5 GHz.
- 1 Gb de memoria RAM.
- Sistema operativo Windows XP o superior, Linux.

La Figura 3 muestra la página principal del sistema después de realizar una recuperación; donde es posible también observar las funcionalidades que brinda:

1. Caja de texto para poder escribir la consulta.
2. Botón para realizar una consulta y brindar el resultado de la recuperación en 4.
3. Botón para configurar algunas opciones del sistema como: Seleccionar un repositorio local o remoto y escoger las Unidades Estructurales que debe tener en cuenta el recuperador.
4. Área con el resultado de la recuperación, para cada archivo recuperado se muestra: nombre, dirección, un fragmento del resumen y el grupo al que pertenece.

Revista Cubana de Ciencias Informáticas
Vol. 10, No. 2, abril-junio, 2016
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 57- 72

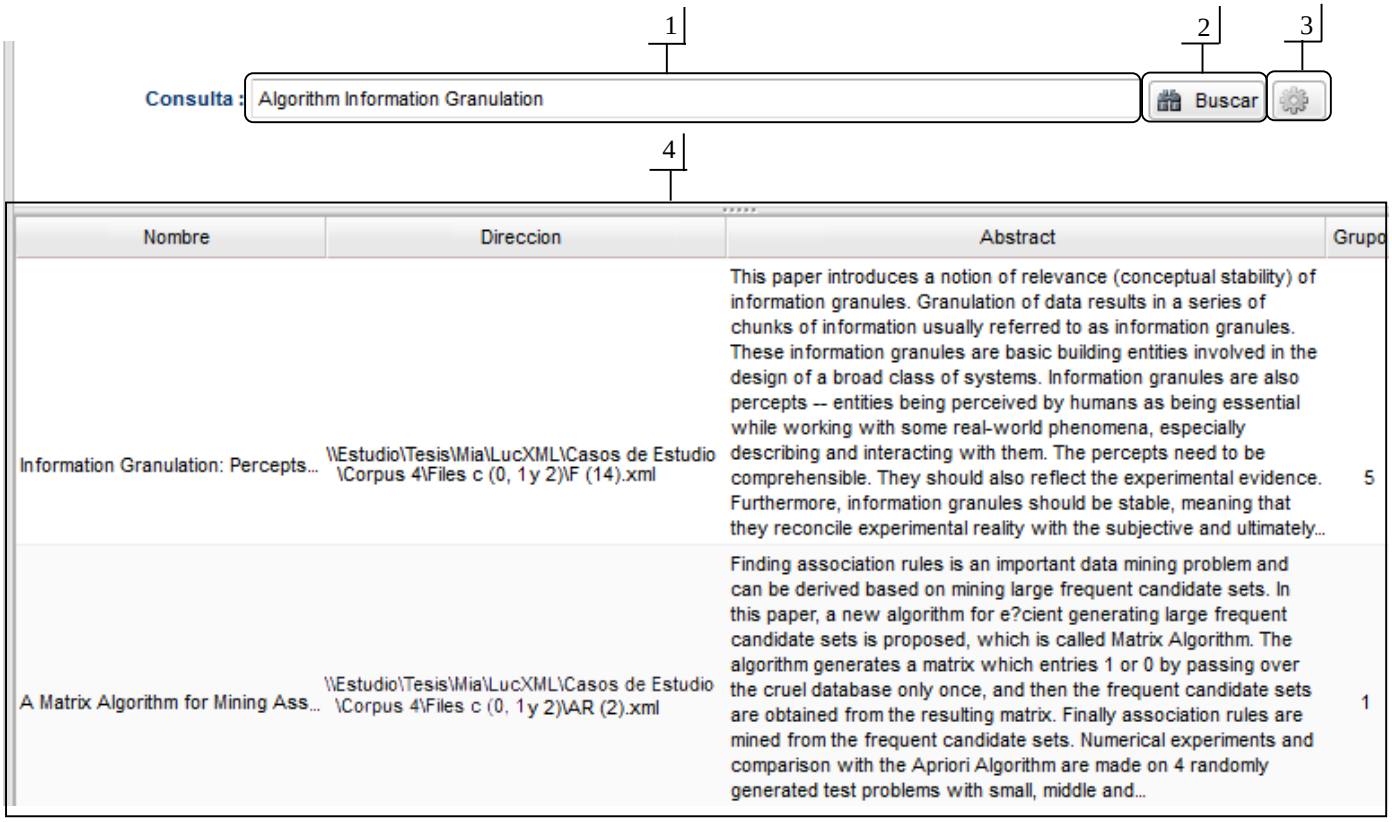

Figura 3. Ventana principal con el resultado final de una recuperación.

Entre los tipos de consulta que se pueden formular en RISADXML se encuentran:

- Palabras.
- Frases, ejemplo: “XML clustering”
- Apoyada por comodines de textos:
  - “?”, significa un carácter en frase o palabra incluyendo el carácter vacío. Ejemplo: “te?t” devuelve los artículos que contienen “text” o “test”.
  - “*”, significa varios caracteres en una frase o palabra. Ejemplo: “test*” devuelve los artículos que contienen “tests” o “tester”
- Uso de operadores booleanos:
  - “OR”, busca los documentos que tienen una frase o la otra.
  - “AND”, busca los documentos que tienen ambas frases.

Revista Cubana de Ciencias Informáticas
Vol. 10, No. 2, abril-junio, 2016
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 57- 72

- "+" busca los documentos que tienen la frase que sigue al símbolo y puedan contener la otra frase. Ejemplo: + "clustering" "XML"
- "NOT", Buscan los documentos que no contienen la frase que sigue al símbolo. Ejemplo: "structural clustering" NOT "content clustering". Este operador no puede ser usado cuando solo existe un término. Ejemplo: NOT " structural clustering"
- "-", Buscan los documentos que estrictamente no contienen la frase que sigue al símbolo.

Evaluación de la herramienta

Para chequear la validez de los resultados obtenidos por el sistema se han utilizado tres casos de estudio:

- El primer caso de estudio está conformado a partir de archivos provenientes del sitio ICT[7], para la recuperación de información y extracción de conocimiento que solicitan estos usuarios.
- El segundo caso de estudio constituye una recopilación de documentos del repositorio *IDE-Alliance*, internacionalmente utilizados para evaluar el agrupamiento. Proporcionados por la Universidad de Granada, España.
- El tercer caso de estudio constituye una selección aleatoria de documentos de la colección de la Wikipedia, publicados cada año por la **IN**iciativa para la **E**valuación de la recuperación de documentos **X**ML (INEX)[8]. Esta colección es referenciada en trabajos para evaluar algoritmos en el área de la minería de textos aplicados a los documentos XML (Denoyer and Gallinari, 2009, Campos et al., 2009). Esta colección tiene el problema que los textos contienen mucha información no útil y el formato en que se presentan es muy difícil de preprocesar.

Atendiendo a la clasificación de las medidas para la evaluación del agrupamiento (Rendón et al., 2011), en esta investigación se seleccionó la medida externa: *Overall F-measure*, OFM (Steinbach et al., 2000) para el estudio comparativo que se realiza entre el procesamiento realizado en (Magdaleno et al., 2015a, Magdaleno et al., 2015b) y los valores obtenidos por RISADXML con los 15 corpus conformados a partir de los tres casos de estudio descritos anteriormente. OFM utiliza los criterios de RI: Precisión (Pr) y cubrimiento[9] (Re).

[7] ftp://ict.cei.uclv.edu.cu

[8] *Initiative for the Evaluation of XML Retrieval*

[9] En este documento se utiliza cubrimiento como traducción de la medida *recall*. Adicionalmente, se utiliza el término cubrimiento para nombrar una forma de la división de los objetos después de un agrupamiento duro y con solapamiento.

Revista Cubana de Ciencias Informáticas
Vol. 10, No. 2, abril-junio, 2016
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 57- 72

Diseño del experimento

El experimento consistió en verificar cómo se comporta globalmente RISADXML con respecto a su predecesor LucXML, ambos implementan el modelo de agrupamiento mencionado anteriormente para documentos XML. En la Tabla 1 se puede observar que solo en cinco casos (cuatro a favor del sistema propuesto en este trabajo) los agrupamientos no se comportaron de forma similar, según la medida OFM.

Tabla 1. Valores de la medida *Overall F-Measure*, calculada a los agrupamientos obtenidos por los dos sistemas

Corpus	OFM LUCXML	OFM RISADXML	Corpus	OFM LUCXML	OFM RISADXML	Corpus	OFM LUCXML	OFM RISADXML
1	0.852	**0.902**	6	0.582	0.582	11	0.947	0.947
2	0.782	0.782	7	0.881	0.881	12	0.977	0.977
3	0.837	**0.851**	8	0.886	0.886	13	0.966	**0.970**
4	0.720	0.720	9	0.856	0.856	14	0.828	0.828
5	0.790	0.790	10	**0.9134**	0.874	15	0.908	**0.911**

Para demostrar lo anterior, se empleó la prueba no paramétrica de *Wilcoxon* (Wilcoxon, 1945) con los valores de la Tabla 1. En la Tabla 2 se puede observar que no existen diferencias significativas, pues en esta prueba estadística si la significación es mayor que 0.05, no se rechaza la hipótesis de que no existen diferencias significativas entre los pares de muestras comparadas.

Tabla 2. Resultados de aplicar Wilcoxon a los valores de la Tabla 1

		N	Mean Rank	Sum of Ranks
OFM_RISADXML - OFM_LucXML	Negative Ranks	1[a]	4.00	4.00
	Positive Ranks	6[b]	2.75	11.00
	Ties	10[c]		
	Total	15		

a. OFM_RISADXML < OFM_LucXML
b. OFM_RISADXML > OFM_LucXML
c. OFM_RISADXML = OFM_LucXML

OFM_RISADXML - OFM_LucXML[b]	
Z	-0.9438[a]
Asymp. Sig. (2-tailed)	**0.3452**

a. Based on negative ranks.
b. Wilcoxon Signed Ranks Test.

Conclusiones

El sistema implementado recupera (auxiliándose del API *Lucene*) los documentos en formato XML, correspondientes a artículos científicos provenientes de un servidor remoto o de un repositorio local; facilitando el trabajo de investigación de los científicos. La recuperación sigue el agrupamiento para tratar el contenido y la estructura de documentos utilizando la metodología basada en *OverallSimSUX*, la cual resulta valida comparada con su predecesor LucXML. Para trabajos futuros se pretende extender el sistema a otros tipos de documentos.

Referencias

AFONSO, A. R. & DUQUE, C. G. 2014. Automated text clustering of newspaper and scientific texts in brazilian portuguese: analysis and comparison of methods. *JISTEM-Journal of Information Systems and Technology Management,* 11**,** 415-436.

AGGARWAL, C. C. & ZHAI, C. X. 2012. *Mining Text Data,* Springer.

ALGERGAWY, A., MESITI, M., NAYAK, R. & SAAKE, G. 2011. XML data clustering: An overview. *ACM Comput. Surv.,* 43**,** 1-41.

AMARASINGHE, K., MANIC, M. & HRUSKA, R. Optimal stop word selection for text mining in critical infrastructure domain. Resilience Week (RWS), 2015, 2015. IEEE, 1-6.

AMOLI, P. V. & SH, O. S. 2015. Scientific Documents clustering based on Text Summarization. *International Journal of Electrical and Computer Engineering (IJECE),* 5.

ARCO, L. 2009. *Agrupamiento basado en la intermediación diferencial y su valoración utilizando la teoría de los conjuntos aproximados.* Doctorado en Ciencias Técnicas, Universidad Central "Marta Abreu" de Las Villas.

ARCO, L., ARTÍLES, M. & BELLO, R. 2008a. *Sistema para la Gestión de Artículos científicos Recuperados usando Lucene (GARLucene)*. Cuba patent application.

ARCO, L., MAGDALENO, D. & BELLO, R. E. 2008b. *Sistema para el agrupamiento y evaluación de colecciones textuales (SATEX)*. Cuba patent application.

ARTILES, M. 2011. *Herramientas de Minería de Textos e Inteligencia Artificial aplicadas a la gestión de la información científico-técnica.* Máster en Ciencia de la Computación, Universidad Central "Marta Abreu" de Las Villas.

BUETTCHER, S., CLARKE, C. L. A. & CORMACK, G. V. 2010. *Information Retrieval: Implementing and Evaluating Search Engines,* MIT Press.

CAMPOS, L. M. D., FERNÁNDEZ-LUNA, J. M. & J.F. HUETE, A. E. R. 2009. Probabilistic methods for link-based classification at INEX'08. *Proceedings of Initiative for the Evaluation of XML Retrieval,* 5631**,** 453–459.

CHEN, S. & ZHANG, K. 2012. An improved algorithm for tree edit distance with applications for RNA secondary structure comparison. *Combinatorial Optimization,* 27**,** 778-797.

CHOWDHURY, G. 2010. *Introduction to Modern Information Retrieval, Third Edition,* Facet Publishing.

CHRISS A., M. & ZITTING, J. L. 2012. *Tika in Action,* 20 Baldwin Road PO Box 261 Shelter Island, NY 11964, Manning Publications Co.

COSTA, G., MANCO, G., ORTALE, R. & RITACCO, E. 2013. Hierarchical clustering of XML documents focused on structural components. *Data & Knowledge Engineering,* 84**,** 26-46.

CROFT, W. B., METZLER, D. & STROHMAN, T. 2010. *Search Engines Information Retrieval in Practice* Pearson Education.

DENOYER, L. & GALLINARI, P. 2009. Overview of the inex 2008 xml mining track. In Advances in Focused Retrieval. *Proceedings of Initiative for the Evaluation of XML Retrieval,* 5631**,** 401–411.

DOUCET, A. & AHONENMYKA, H. 2002. Naive clustering of a large XML document collection. *INEX***,** 84-89.

FUENTES, I. E. 2013. *Nuevo modelo de agrupamiento para documentos XML utilizando estructura y contenido.* Licenciatura en Ciencia de la Computación Tesis de grado, Universidad Central "Marta Abreu" de Las Villas.

GROSSMAN, D. A. & FRIEDER, O. 2012. *Information retrieval: Algorithms and heuristics*, Springer Science & Business Media.

GUAN, R., YANG, C., MARCHESE, M., LIANG, Y. & SHI, X. 2014. Full Text Clustering and Relationship Network Analysis of Biomedical Publications.

HATCHER, E., GOSPODNETIC, O. & MCCANDLESS, M. 2009. *Lucene in Action*.

KRUSE, R., DÖRING, C. & LESOR, M.-J. 2007. Fundamentals of Fuzzy Clustering. *In:* OLIVEIRA, J. V. D. & PEDRYCZ, W. (eds.) *Advances in Fuzzy Clustering and its Applications.* Est Sussex, England: John Wiley and Sons.

MAGDALENO, D., FUENTES, I. E., ARCO, L., ARTILES, M., FERNANDEZ, J. M. & HUETE, J. 2011. New Textual Representation using Structure and Contents. *Research in Computing Science,* 54**,** 117-130.

MAGDALENO, D., FUENTES, I. E. & GARCÍA, M. M. 2013. *Sistema para el agrupamiento de artículos científicos en formato XML usando Lucene (LucXML)*. Cuba patent application.

MAGDALENO, D., FUENTES, I. E. & GARCÍA, M. M. 2015a. Clustering XML Documents using Structure and Content Based in a Proposal Similarity Function (OverallSimSUX). *Computación y Sistemas,* 19.

MAGDALENO, D., MIRANDA, Y., FUENTES, I. E. & GARCÍA, M. M. 2015b. Comparative Study of Clustering Algorithms using OverallSimSUX Similarity Function for XML Documents. *Inteligencia Artificial,* 18**,** 69-80.

PIERNIK, M., BRZEZINSKI, D., MORZY, T. & LESNIEWSKA, A. 2015. XML clustering: a review of structural approaches. *The Knowledge Engineering Review,* 30**,** 297-323.

PINTO, D., TOVAR, M. & VILARIÑO, D. BUAP: Performance of K-Star at the INEX'09 Clustering Task. *In:* GEVA, S., KAMPS, J. & TROTMAN, A., eds. INEX 2009 Workshop Pre-proceedings, 2009 Woodlands of Marburg, Ipswich, Queensland,Australia. 391-398.

RENDÓN, E., ABUNDEZ, I., ARIZMENDI, A. & QUIROZ, E. 2011. Internal versus external cluster validation indexes. *International Journal of computers and communications,* 5**,** 27-34.

SALTON, G., WONG, A. & YANG, C. S. 1975. A vector space model for automatic text retrieval. *Communications of the ACM,* 18**,** 613-620.

SHANKAR, R. 2012. *Evolutionary Document Clustering and Summarization of Scientific Articles using Frequent Itemsets.* International Institute of Information Technology Hyderabad.

SHIN, K. & HAN, S. Y. 2003. Fast clustering algorithm for information organization. *In:Proc. of the CICLing Conference.* Lecture Notes in Computer Science.Springer-Verlag (2003).

SINGH, S. & SIDDIQUI, T. J. Evaluating effect of context window size, stemming and stop word removal on Hindi word sense disambiguation. Information Retrieval & Knowledge Management (CAMP), 2012 International Conference on, 2012. IEEE, 1-5.

STEINBACH, M., KARYPIS, G. & KUMAR, V. A comparison of document clustering techniques. Proceedings of 6th ACM SIGKDD World Text Mining Conference, 2000 Boston. ACM Press, 1-20.

TEKLI, J. M. & CHBEIR, R. 2011. A Novel XML Document Structure Comparison Framework based-on Subtree Commonalities and Label Semantics. *Elsevier,* 11.

TIEN T., R. N. 2007. Evaluating the Performance of XML Document Clustering by Structure only. *5th International Workshop of the Initiative for the Evaluation of XML Retrieval*.

WATANABE, Y., KAMIGAITO, H. & YOKOTA, H. 2013. Similarity search for office XML documents based on style and structure data. *International Journal of Web Information Systems,* 9**,** 7.

WILCOXON, F. 1945. Individual comparisons by ranking methods. *Biometrics Bulletin,* 1**,** 80-83.

YAU, C.-K., PORTER, A., NEWMAN, N. & SUOMINEN, A. 2014. Clustering scientific documents with topic modeling. *Scientometrics,* 100**,** 767-786.

ZAMAN, A., MATSAKIS, P. & BROWN, C. Evaluation of stop word lists in text retrieval using Latent Semantic Indexing. Digital Information Management (ICDIM), 2011 Sixth International Conference on, 2011. IEEE, 133-136.