CONJUNTO DE HERRAMIENTAS QUE AUXILIAN EL DESARROLLO DE SISTEMAS GESTORES DE INFORMACIÓN EN DOMINIOS TEXTUALES

Michel Artiles Egüe, Leticia Arco García, Damny Magdaleno Guevara

{mae, leticiaa, dmg}@uclv.edu.cu

Departamento de Ciencia de la Computación, Universidad Central "Marta Abreu" de Las Villas, Cuba

RESUMEN

En el Centro de Estudios de Informática (CEI) de la Universidad Central "Marta Abreu" de las Villas (UCLV) se han desarrollado sistemas para la gestión de la información y el conocimiento como parte de su estrategia de informatización. Algunos de estos sistemas son CorpusMiner, SATEX y GARLucene, los cuales brindan amplias ventajas para la gestión de la información y del conocimiento. Sin embargo, el diseño de estos sistemas no permite la utilización de algunos de sus módulos en otras aplicaciones, o la incorporación de otras formas de representación textual, u otros métodos de indexado y recuperación de la información. Por otra parte, estas aplicaciones son de escritorio, limitándose significativamente los procesos de indexado y recuperación de la información.

Por tales motivos, en este trabajo se presenta un conjunto de herramientas creadas con el objetivo de auxiliar el desarrollo de sistemas gestores de información en dominios textuales. Estas herramientas trabajan de forma independiente, son extensibles y facilitan el intercambio de información. La información se intercambia mediante documentos textos con un formato estructurado, descritos usando el Lenguaje de Marcado Extensible (XML) y validados usando esquemas de XML. Las herramientas desarrolladas son: listar recursos, extraer contenido, indexar contenido extraído, recuperar información, transformar información y estructurar información. El desarrollo de las mismas se basó en las facilidades brindadas por sistemas anteriores y en sus conexiones con repositorios de información científico-técnica.

Palabras clave: gestión documental, gestión del conocimiento, automatización de la información.

ABSTRACT

At the Center for Computer Studies (CIS), Central University "Marta Abreu" de las Villas (UCLV) have developed systems for managing information and knowledge as part of its strategy of informatization. Some of these systems are CorpusMiner, SATEX and GARLucene, which provide broad benefits for the management of information and knowledge. However, the design of these systems does not allow the use of some of its modules in other applications, or the incorporation of other forms of textual representation, or other methods of indexing and retrieval of information. Moreover, these applications are desktop processes significantly limited indexing and retrieval of information.

For these reasons, this paper presents a set of tools created with the objective of assisting the development of information management systems in textual domains. These tools work independently, are extensible and facilitate the exchange of information. The information is exchanged through text documents to a structured format described using the Extensible Markup Language (XML) and validated using XML schemas. The tools developed are: to list resources, extract contents, index, contents removed, retrieve information, process information and structure information. The development of these relations was based on the facilities offered by previous systems and their connections to repositories of scientific and technical information.

Keywords: document management, knowledge management, automation of information.

1 INTRODUCCIÓN

Los sistemas de gestión de documentos[1] han tomado un gran auge en la actualidad [1]. Docyoument, Text Miner[2] , Worldox[3], Autonomy[4] , Knexa[5] y otros[6] que fueron identificados por el Instituto Kaieteur[7]constituyen algunos de los más referenciados. La gran mayoría son sistemas más dirigidos al comercio del conocimiento en Internet y no la gestión del conocimiento en las organizaciones [2].

Los sistemas dirigidos a las organizaciones tienen un alto precio en el mercado internacional, debido esencialmente a los beneficios que les reportan. Por eso resulta casi imposible adquirir estos tipos de sistemas a las instituciones cubanas. Las universidades no están exentas de esta limitante y son de las organizaciones en el país que más requieren manipular grandes cantidades de documentos científicos por la actividad académica y científica que desarrollan. En ellas, la gestión del conocimiento juega un rol importante porque sus procesos son estables, generan y preservan valiosa información proveniente de diversos procesos, tienen acceso a importantes fuentes de información externa, poseen capital humano bien capacitado y buen desarrollo de las tecnologías de la información [2].

En el Centro de Estudios de Informática (CEI) de la Universidad Central "Marta Abreu" de las Villas (UCLV) se han desarrollado sistemas para la gestión de la información y el conocimiento como parte de su estrategia de informatización. Algunos de estos sistemas son CorpusMiner, SATEX y GARLucene, los cuales brindan amplias ventajas para la gestión de la información y del conocimiento. Sin embargo, el diseño de estos sistemas no permite la utilización de algunos de sus módulos en otras aplicaciones, o la incorporación de otras formas de representación textual, u otros métodos de indexado y recuperación de la información. Por otra parte, estas aplicaciones son de escritorio, limitándose significativamente los procesos de indexado y recuperación de la información.

Por tales motivos, en este trabajo se presenta un conjunto de herramientas creadas con el objetivo de auxiliar el desarrollo de sistemas gestores de información en dominios textuales. Estas herramientas trabajan de forma independiente, son extensibles y facilitan el intercambio de información. Así, se pretende que el conjunto de herramientas desarrollada facilite la gestión de la información y contribuya significativamente a la gestión de la información textual en el CEI-UCLV, garantizando una adecuada conexión con sus repositorios de información científico-técnica.

2 ESQUEMA PARA DESARROLLAR SISTEMAS GESTORES DE INFORMACIÓN EN DOMINIOS TEXTUALES

La gestión de información en dominios textuales integra varias áreas del saber, entre las cuales están: el descubrimiento de conocimiento en bases de datos, la minería de datos y la minería de textos. 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 [7, 8].

Automatizar la gestión de información y el conocimiento conlleva al desarrollo de sistemas que ayuden a los usuarios al enfrentarse a grandes colecciones textuales, mediante la organización y extracción de conocimiento de las mismas [1]. La entrada a estos sistemas debe provenir del resultado de un proceso de recuperación de información [1], de un servidor o colección personal de información textual; y las salidas deben ser los grupos homogéneos de documentos afines, los términos relevantes y los documentos más

---

[1] Sistemas de almacenamiento, sistemas de soporte de búsquedas, modelos de categorización y análisis de contenidos, ontologías y servicios de control de acceso y coordinadores de trabajo colaborativo.

[2] http://www.sas.com/technologies/analytics/datamining/textminer

[3] http://www.worldox.com/

[4] http://www.autonomy.com

[5] http://www.knexa.com

[6] Knowinc.com, Keen.com, Yet2.com, iExchange.com, Saba.com, cordis.lu, IQ4Hire.com, petrocore.com y eBrainx.com

[7] Instituto para la gestión del conocimiento (Kaieteur Institute for Knowledge Management http://www.kikm.org)

representativos de cada grupo, así como los que se relacionan con ellos y la calidad con que fueron obtenidos los grupos; garantizando el control para la evaluación de los resultados del agrupamiento [1].
Los sistemas para la gestión de la información y el conocimiento (CorpusMiner, SATEX y GARLucene) desarrollados por especialistas del CEI-UCLV siguen el esquema propuesto en [1] para la confección de sistemas gestores de información en dominios textuales. En la Ilustración 1 se muestra la idea general de este esquema, el cuál está compuesto por cuatro módulos: recuperación de la información o especificación del corpus textual a procesar, representación del corpus textual obtenido o fijado por el usuario, agrupamiento de los documentos y valoración (validación y etiquetamiento) de los grupos textuales obtenidos.

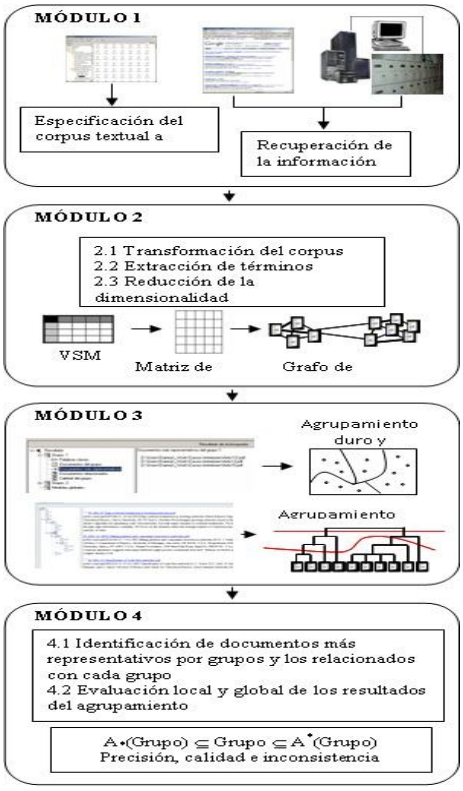

Ilustración 1. Esquema para desarrollar sistemas gestores de información en dominios textuales.

2.1 RECUPERACIÓN DE LA INFORMACIÓN

Recuperación de información (Information Retrieval; IR) puede ser definida como una aplicación de las tecnologías de la computación para la adquisición, organización, almacenamiento, recuperación, y distribución de información. IR está estrechamente relacionada con elementos prácticos y teóricos sobre la mejora de la tecnología de los motores de búsqueda, incluyendo la construcción y mantenimiento de grandes repositorios de información. En años recientes, los investigadores han incrementado y expandido su preocupación acerca de la bibliografía y búsqueda a texto completo en repositorios en la Web, tanto en datos de hipertextos pertenecientes a bases de datos como en multimedia.

2.2 REPRESENTACIÓN TEXTUAL

Al trabajar con textos (datos no estructurados) resulta indispensable una correcta representación textual para su procesamiento posterior. Una de las variantes más utilizadas para realizar la representación de grandes volúmenes textuales es la representación espacio-vectorial (Vector Space Model; VSM) [3]. La representación en VSM se caracteriza por ser efectiva para representar documentos, ajustarse a otras formas de indexado y ser ampliamente reconocida en la comunidad de minería de textos. Según VSM cada documento se identifica como un vector de rasgos en un espacio en el cual cada dimensión corresponde a términos indexados distintos (palabras). Un vector documento dado, en cada componente tiene un valor numérico para indicar su importancia.

2.2.1 TRANSFORMACIÓN DEL CORPUS

El objetivo de la transformación del corpus es convertir los datos correspondientes a ficheros de entrada en una secuencia de ítems lingüísticos (tokens[8] de palabras). En el paso subsiguiente a la extracción de términos, estos tokens serán usados para generar rasgos significativos (índices de términos) [2].
Aquí se reconocen los componentes textuales desde los diferentes formatos y la secuencia resultante de tokens debe ser transformada. Posibles transformaciones pueden ser: convertir las letras todas a mayúsculas o todas a minúsculas, eliminar las marcas de puntuación al final de los tokens, omitir los tokens que contienen caracteres alfanuméricos, identificar los nombres de personas, localidades y organizaciones, y sustituir las contracciones y abreviaturas por sus expresiones completas [4].

[8] Los tokens son cadenas de caracteres delimitadas por espacios en blanco (por ejemplo espacios, cambios de líneas, tabs).

Lograr una representación textual lo suficientemente buena implica la inclusión de varias sub-tareas como: transformación del corpus, extracción de términos, reducción de la dimensionalidad, y la normalización y pesado de la matriz [4].

2.2.2 EXTRACCIÓN DE TÉRMINOS

La extracción de términos parte de una secuencia de tokens, obtenida a partir de la transformación del corpus, y produce una secuencia de términos indexados basados en esos tokens. Una posibilidad bastante utilizada consiste en crear el vocabulario a partir de los términos indexados resultantes de la extracción y analizar léxicamente los documentos identificando las palabras simples como rasgos [5]. Así, se explota básicamente el plano estadístico de los textos y no se considera la secuencia de aparición de las palabras en un documento (modelo bolsa de palabras; bag-of-words model) [6], aunque alguna información sintáctica puede enriquecer posteriormente los resultados. El análisis léxico es ventajoso porque la definición de los términos es independiente del lenguaje y computacionalmente muy eficiente. Además la representación resultante es fácil de analizar por los humanos, por lo que se logra la interpretabilidad requerida en el post-agrupamiento. Una desventaja es que cada inflexión de una palabra es un posible rasgo y el número de éstos puede ser innecesariamente grande, requiriéndose la reducción de dimensionalidad [2].

2.2.3 REDUCCIÓN DE LA DIMENSIONALIDAD

Controlar la dimensionalidad del espacio del vector documento cuando se utilizan las palabras como los términos a indexar es esencial. Porque la complejidad de muchos algoritmos de agrupamiento depende crucialmente del número de rasgos y reducirlo hace tratables estos algoritmos [1]. Además existen palabras que son irrelevantes y producen peores resultados, así que eliminándolas se logra aumentar la eficiencia del agrupamiento a realizar. Existen varias técnicas para reducir la dimensionalidad del vector: selección de rasgos y reparametrización [10], o combinación de ambas [13]. Algunas de estas son: la eliminación de palabras de parada o gramaticales (stop word elimination), métodos de filtrado para decidir cuándo incluir un término en el vocabulario o no, la homogeneidad ortográfica (spelling), la reducción de las palabras a su forma raíz (stemming) [14] y otras como el análisis de latencia semántico (Latent Semantic Analysis) [15], el uso de tesauros y de ontologías [16, 17] que generalmente están asociadas a un dominio específico o requieren de una alta complejidad computacional.

2.2.4 NORMALIZACIÓN Y PESADO DE LA MATRIZ

Dadas las estadísticas de frecuencias de los términos en todos los documentos, se genera un vector pesado para cualquier documento basado en el vector de frecuencias de términos. Cada peso expresa la importancia de un término en un documento con respecto a su frecuencia en todos los documentos. Existen diferentes variantes que permiten pesar los términos indexados entre ellas están las formas de pesado global basadas en alguna variación de la fórmula TF-IDF [18, 19] y la normalización dividiendo la frecuencia de aparición de los términos por la longitud de los documentos para abstraerse de su variedad de tamaños [1].

2.3 AGRUPAMIENTO

El agrupamiento se considera una de las técnicas que permiten el análisis exploratorio de los datos. El análisis de grupos[9] permite descubrir la estructura interna de éstos e identificar distribuciones interesantes y patrones subyacentes en ellos, considerando muy poca o ninguna información a priori [7].

Los algoritmos de agrupamiento son usados para encontrar una estructura de grupos que se ajuste al conjunto de datos, logrando homogeneidad dentro de los grupos y heterogeneidad entre ellos [8]; de forma tal que exista un alto grado de asociación entre los objetos de un mismo grupo y un bajo grado entre los miembros de grupos diferentes [8].

---

[9] En este trabajo se utilizarán indistintamente los términos: grupos, conglomerados, comunidades, subconjuntos y clases

2.4 EVALUACIÓN

"El agrupamiento es un proceso subjetivo; el mismo conjunto de datos usualmente necesita ser agrupado de formas diferentes dependiendo de las aplicaciones" [9]. Esta subjetividad hace el agrupamiento difícil, y aún más su validación.

El procedimiento de evaluar los resultados de algoritmos de agrupamiento se conoce por validación del agrupamiento [10, 11]. Se dice medida de validación de grupos a una función que hace corresponder un número real a un agrupamiento, indicando en qué grado el agrupamiento es correcto o no [12]. Estas medidas en su mayoría son heurísticas por naturaleza para conservar una complejidad computacional plausible [2].

3 CONJUNTO DE APLICACIONES PARA EL PROCESAMIENTO TEXTUAL

Con el objetivo de independizar las futuras aplicaciones de los diferentes módulos inherentes al proceso de gestión documental según la ilustración 1, se desarrollaron un conjunto de herramientas compatibles con ese esquema y que abarcan los módulos de recuperación de la información y representación del corpus. Las herramientas desarrolladas trabajan de forma independiente y pueden ser utilizadas por cualquier otra aplicación como bibliotecas o ejecutables. Todas estas herramientas se desarrollaron utilizando el lenguaje de programación JAVA y el IDE NetBeans. Además, las bibliotecas y marcos de trabajo (framework) que son utilizados cumplen con la categoría de software libre; cumpliendo con la política trazada por la dirección de nuestro país que aboga por el uso de software que pertenezca a dicha categoría.

Como los resultados obtenidos por las aplicaciones son datos con un formato estructurado se decidió utilizar el Lenguaje de marcado extensible (XML) para representar esa información y el uso de esquemas de XML para validar la estructura del documento resultante [13].

Algunas herramientas como las que permiten la indexación, recuperación y transformación utilizan la biblioteca 3.0.2 de Apache Lucene [14] y son capaces de optimizar el uso del idioma original para obtener las raíces de las palabras. Hasta el momento ellas distinguen los siguientes idiomas: Inglés, Español, Francés, y Alemán; y otros más que pueden ser incorporados sin requerir mucho esfuerzo.

3.1 MÓDULO DE RECUPERACIÓN DE LA INFORMACIÓN

El Módulo de recuperación de la información, según se observa en la ilustración 1, fue dividido en dos submódulos para una mejor concepción de las herramientas. Un módulo que se encarga de la inserción de información en la base de datos y otro módulo que se encarga de la recuperación de la información dada una consulta.

El módulo de inserción está conformado por tres herramientas:

- Listar los recursos
- Extraer el contenido de los recursos
- Indexar el contenido de los recursos

Las herramientas que permiten listar y extraer los recursos pueden hacerlo desde un directorio local o un servidor que implemente el protocolo de transferencia de archivo (FTP) como el sitio ftp (ftp://ict.cei.uclv.edu.cu) que constituye el repositorio de información del CEI-UCLV.

3.1.1 LISTAR RECURSOS

Lista el contenido del directorio especificado si puede ser accedido por la aplicación, para que su contenido sea extraído posteriormente.

El pase de parámetros a la aplicación es el siguiente: "XML PROTOCOLO PWD", donde:

- XML: fichero XML resultante de la ejecución de la aplicación.
- PROTOCOLO: puede ser un DIRECTORIO o sitio FTP
  - DIRECTORIO: -d
  - FTP: -ftp servidor [-u user –p passwd]

- PWD: Directorio a listar, ya sea de un ftp o un directorio local.

El fichero XML resultante cumple con la estructura del esquema que se presenta en la Ilustración 2.

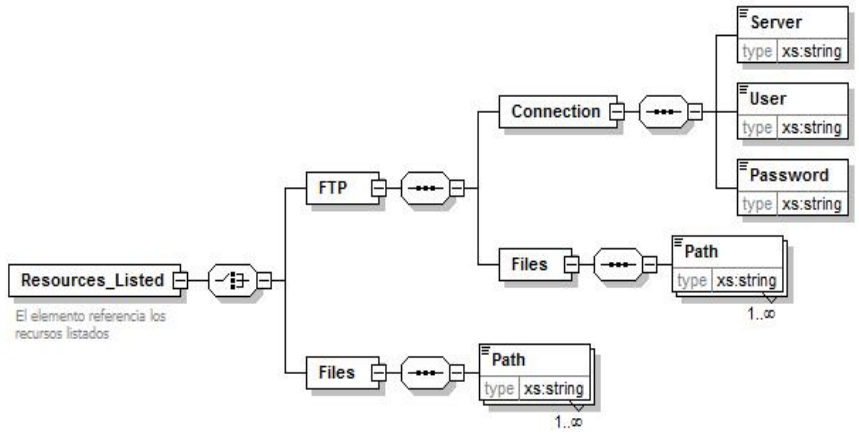

Ilustración 2. Esquema XML del listado de recursos.

El nodo raíz del documento XML resultante puede contener una de las dos opciones posibles según la localización de los archivos (FTP o locales). Si la localización de los archivos es un servidor FTP se guardan en el documento XML información de la conexión como el servidor, nombre de usuario y contraseña. En ambos casos la dirección física de los archivos se guarda dentro de los nodos denominados Path que están englobados en el nodo Files.

3.1.2 EXTRAER CONTENIDO

Extrae el contenido de los archivos resultantes de la aplicación encargada de listar los archivos. Para extraer el contenido se apoya en el marco de trabajo "Tika" que permite extraer contenido de múltiples tipos de ficheros [25].

Por defecto la aplicación extrae los siguientes metadatos:

- Nombre: nombre del documento (DOCUMENT-NAME)
- Contenido: todo el contenido del documento (FULL-TEXT)
- Lenguaje: lenguaje en que está escrito la mayor parte del documento (LANGUAGE)

La aplicación recibe como parámetros:

- Archivo XML el cual es validado según el esquema XML del listado de recursos que se muestra en la Ilustración 2.
- Archivo XML que contiene los campos o metadatos a extraer adicionalmente y que es validado usando un esquema que cumple con la especificación de metadatos de "Tika" [25], según se muestra en la Ilustración 3.
- Archivo XML resultante que contiene la información extraída de los documentos, separada por los campos de los metadatos especificados por el atributo "NODE-NAME" según se muestra en la Ilustración 4.

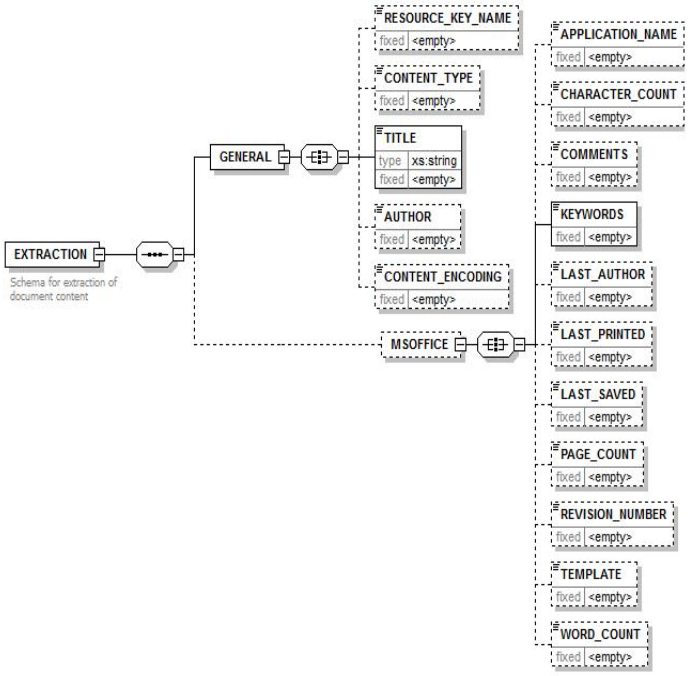

Ilustración 3. Metadatos de Tika.

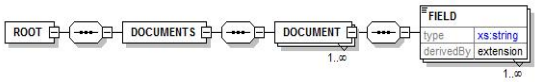

Ilustración 4. Esquema de información extraída de los documentos.

3.1.3 INDEXAR CONTENIDO

Indexa el contenido de los documentos almacenado por campos en un archivo XML que cumple con el esquema que se muestra en la Ilustración 3 teniendo en cuenta el campo que describe el lenguaje del contenido del documento original (LANGUAGE) si es que este existe en el documento XML. Separar los índices según el lenguaje permite que los índices sean más pequeños y la búsqueda por índice sea mucho más rápida y específica.

En caso de ocurrir algún error durante la indexación la aplicación imprime en la salida estándar los mensajes originados por las excepciones o errores reportados. Además, trata de recuperarse del error al seguir indexando el próximo contenido de documento.

El pase de parámetros es el siguiente: XML DIRECTORIO, donde:

- XML: es un archivo XML que cumpla con el esquema presentado en la ilustración 3.
- Directorio**:** directorio raíz donde se crearán los índices.

3.1.4 RECUPERAR INFORMACIÓN

Recupera información a partir de un índice de archivo originado por la herramienta que permite indexar el contenido o uno creado con cualquier aplicación que utilice la biblioteca Apache Lucene para crear el repositorio de índices. El pase de parámetros de la herramienta es el siguiente: SEARCH LANG QUERY SAVE, donde:

- SEARCH: dirección del documento XML donde están las especificaciones de la búsqueda con la estructura del esquema según la Ilustración 5.
- LANG: lenguaje en el que se realizará la búsqueda siempre en minúscula y se admiten los mismos lenguajes que en herramienta de indexar el contenido.
- QUERY: texto que conforma la consulta a realizar, se admiten los mismos tipos de consultas que Lucene 3.0.2 [14].
- SAVE: dirección del documento XML donde se almacenará la información recuperada con la estructura del esquema según la Ilustración 6.

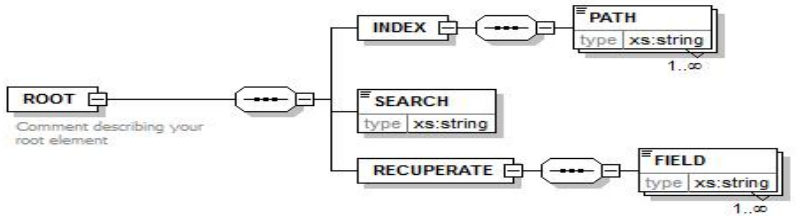

Ilustración 5. Esquema de las especificaciones de búsqueda.

Se optó por requerir un documento XML que especificara parámetros de configuración para realizar las búsquedas en vez de requerirlas como parámetros, debido a que estos en particular no cambian frecuentemente con las búsquedas. Los parámetros

que conforman el documento XML son: los directorios raíces de los índices, el campo por el cual se requiere que se efectúe la búsqueda y los campos que se requieren recuperar de los documentos que concuerden con el criterio de búsqueda. En la Ilustración 5 se muestra el esquema de las especificaciones de búsqueda.

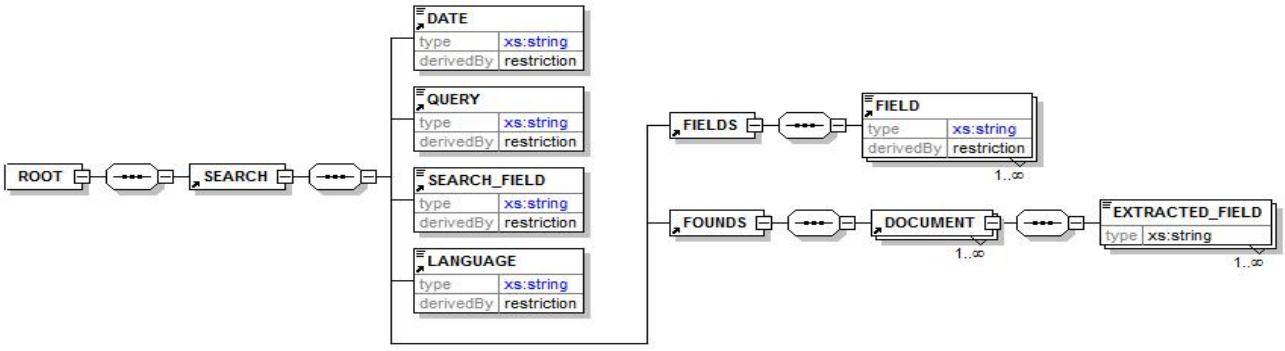

Ilustración 6. Información recuperada del repositorio.

Además de salvar los campos y la información contenida en los campos solicitados por el documento XML (SEARCH) la aplicación por defecto salva en el documento XML resultante algunos datos como la fecha en que fue realizada la consulta, la consulta en sí, el campo que fue consultado en la base de datos y el lenguaje en el cual se realizó la consulta. El esquema con el documento XML que contiene la información recuperada del repositorio se muestra en la Ilustración 6.

3.2 MÓDULO DE REPRESENTACIÓN DEL CORPUS TEXTUAL

El módulo de representación del corpus textual está dividido en los siguientes submódulos:

- Transformar Información
- Estructurar Información

3.2.1 TRANSFORMAR INFORMACIÓN

Transforma un documento XML de información que contenga una estructura de documento que concuerde con el esquema que se muestra en la Ilustración 6. La aplicación recibe como parámetros la dirección del documento XML con la información y una lista de nombres de campos a transformar. Además, la aplicación agrega un nuevo elemento denominado TRANSFORMATION al elemento ROOT del documento XML con la información resultante, detalles se muestran en la Ilustración 7.

El elemento TRANSFORMATION contiene dos elementos esenciales: FIELDS y TRANSFORMED.

- FIELDS: contiene una lista de elementos denominados FIELD que contienen los campos que fueron transformados.
- TRANSFORMED: contiene una lista de documentos formados por una lista de campos denominados FIELD que contienen una lista de campos TOKEN conformados por las raíces de las palabras.

Los tokens son obtenidos luego de haber pasado por un conjunto de filtros a partir de una modificación de la clase StandardAnalyzer de la biblioteca Apache Lucene para realizar las transformaciones que se aplican en este módulo, agregándole la eliminación de los tokens alfanuméricos y la obtención de raíces gramaticales, método que se encuentra en Lucene pero que fue enriquecido utilizando los stemmer del subproyecto de Lucene Snowball[10]. Algunos de los filtros utilizados fueron:

- Eliminar las marcas de puntuación al final de los tokens.
- Eliminar los apóstrofes del tipo ’s y la eliminación de los acrónimos al quitarles los puntos.
- Convertir las letras todas a minúsculas.

[10] http://snowball.tartarus.org/

- Eliminar las palabras conformadas por otros caracteres que no sean alfanúmericos.
- Quitar los tokens que están dentro de la lista de palabras de paradas del idioma correspondiente al texto. Las listas de paradas fueron conformadas utilizando las listas de paradas ofrecidas por el proyecto Snowball de Apache Lucene.

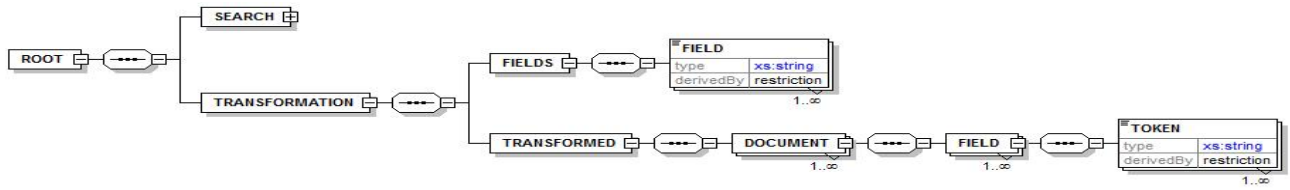

Ilustración 7. Transformación del texto de los campos.

3.2.2 ESTRUCTURAR INFORMACIÓN

Crea un VSM por cada campo que fue transformado a partir de sus tokens. La aplicación recibe como parámetro la dirección del documento XML que contiene las especificaciones de los campos transformados que cumpla con el esquema presentado en la Ilustración 6. Al campo ROOT se le agrega un elemento llamado "Vector Space Models" que cumple con la especificación del esquema que se muestra en la Ilustración 8.

El elemento denominado VECTOR-SPACE-MODELS contiene una lista de elementos VSM que está conformado por un campo KEY que representa el nombre del campo por el cual se creó la VSM, un elemento TRANSFORMATIONS que contiene una lista de elementos que contienen las transformaciones que ha sufrido la VSM, un elemento TERMS que contiene una lista de elementos con los términos que están representados en la VSM, un elemento VALUES conformado por elementos ROW que representan cada documento y cada elemento ROW contiene una lista de elementos COL representando a los términos con un valor de tipo double.

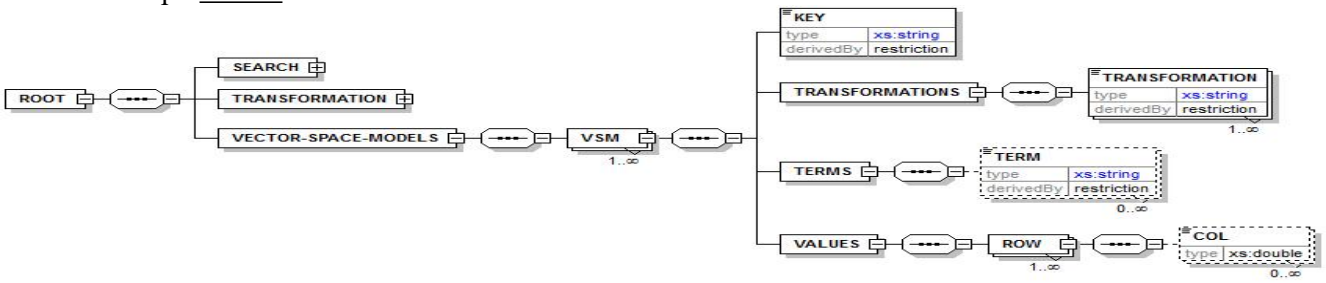

Ilustración 8. Modelos del espacio de vector.

Hasta el momento la aplicación no permite ninguna otra transformación que las que hace por defecto:

- Crear la representación especio vectorial basada en la frecuencia de aparición de los tokens en el documento.
- Aplicar TF-IDF clásica a la representación espacio vectorial.
- Reducir la dimensionalidad basándose en la calidad de términos, asociándole a cada término el valor de su calidad teniendo en cuenta la aparición de éste en la colección de documentos. La cantidad máxima de términos es 600, según se propone en [15].

CONCLUSIONES

En el presente trabajo se expuso un conjunto de herramientas desarrolladas en el laboratorio de Inteligencia Artificial del Centro de Estudios de Informática de la Universidad Central "Marta Abreu" de Las Villas que permiten auxiliar el desarrollo de sistemas gestores de información en dominios textuales. Herramientas que trabajan de forma independiente y realizan el intercambio de información en documentos de textos con un formato estructurado, descritos usando el Lenguaje de Marcado Extensible (XML) y validados usando

esquemas de XML. Además, el desarrollo de estas herramientas fue basado en experiencias acumuladas por las facilidades que brindan sistemas desarrollados anteriormente y en sus conexiones con repositorios de información científico-técnica. El conjunto de herramientas desarrolladas contribuyen al desarrollo de aplicaciones que permitan cumplir uno de los objetivos de la estrategia de informatización del CEI en la UCLV y su posible generalización a otras instituciones del país.

BIBLIOGRAFÍA

[1] R. M. Müller, *et al.*, "Electronic marketplaces of knowledge: Characteristics and sharing of knowledge assets," in *Proceedings of the International Conference on Advances in Infrastructure for e-Business (SSGRR 2002)*, L'Aquila, Italy, 2002.

[2] L. A. García, "Agrupamiento basado en la intermediación diferencial y su valoración utilizando la teoría de los conjuntos aproximados," CEI-AI, Universidad Central "Marta Abreu" de las Villas, Santa Clara, 2008.

[3] G. Salton, *et al.*, "A vector space model for automatic text retrieval," *Communications of the ACM,* vol. 18, pp. 613-620, 1975.

[4] C. Lanquillon, "Enhancing Text Classification to Improve Information Filtering," PhD. thesis, Research Group Neural Networks and Fuzzy Systems, University of Magdeburg "Otto von Guericke", Magdeburg, 2001.

[5] G. Salton and C. Buckley, "Term weighting approaches in automatic text retrieval," *Information Processing and Management,* vol. 24, pp. 513-523, 1988.

[6] D. D. Lewis and M. Ringuette, "A comparison of two learning algorithms for text classification," in *Proceedings of the Third Annual Symposium on Document Analysis and Information Retrieval*, University of Nevada, Las Vegas, 1994, pp. 81-93.

[7] E. Levine and E. Domany, "Resampling method for unsupervised estimation of cluster validity," *Neural Computation,* vol. 13, pp. 2573-2593, 2001.

[8] M. R. Anderberg, *Clustering Analysis for Applications*: New York: Academic, 1973.

[9] A. K. Jain, *et al.*, "Data clustering: a review," *ACM Computing Surveys,* vol. 31, pp. 264-323, 1999.

[10] S. Theodoridis and K. Koutroubas, *Pattern Recognition*: Academic Press, 1999.

[11] M. Halkidi, *et al.*, "Clustering validity checking methods: Part II," *ACM SIGMOD Record,* vol. 31, pp. 19-27, 2002.

[12] F. Höppner, *et al.*, *Fuzzy cluster analysis: methods for classification, data analysis and image recognition.* West Sussex, England: John Wiley & Sons Ltd., 1999.

[13] J. Sturm, *Developing XML Solutions*, 2000.

[14] E. Hatcher, *et al.*, *Lucene in Action*, 2010.

[15] M. W. Berry, *Survey of Text mining: Clustering, Classification, and Retrieval*. New York, USA: Springer Verlag, 2004.