Universidad Central “Marta Abreu” de Las Villas
Facultad de Matemática, Física y Computación
Licenciatura en Ciencia de la Computación

Trabajo de Diploma

Implementación de algoritmos para el agrupamiento documental utilizando OverallSimSUX

Autor
Ernesto Julio Cabrera González

Tutores
MSc. Damny Magdaleno Guevara
Lic. Ivett E. Fuentes Herrera

Consultante
Lic. María Matilde García Lorenzo

Santa Clara, 2015

Hago constar que el presente trabajo fue realizado en la Universidad Central “Marta Abreu” de Las Villas como parte de la culminación de los estudios de la especialidad de Ciencia de la Computación, autorizando a que el mismo sea utilizado por la institución, para los fines que estime conveniente, tanto de forma parcial como total y que además no podrá ser presentado en eventos ni publicado sin la autorización de la Universidad.

_____
Firma del autor

Los abajo firmantes, certificamos que el presente trabajo ha sido realizado según acuerdos de la dirección de nuestro centro y el mismo cumple con los requisitos que debe tener un trabajo de esta envergadura referido a la temática señalada.

_____
Firma del tutor

_____
Firma del jefe del Seminario de Inteligencia Artificial

“Muchas cosas se juzgan imposibles de hacer, antes de que estén hechas”

Plinio

AGRADECIMIENTOS

RESUMEN

En el Centro de Estudios de Informática (CEI) de la Universidad Central "Marta Abreu" de las Villas (UCLV) se ha propuesto una metodología para el agrupamiento de documentos XML combinando estructura y contenido, a través de la confección de una nueva función de similitud. Esta metodología está soportada en un procedimiento general implementado en un sistema para el agrupamiento de artículos científicos en formato XML. Aunque esta metodología muestra buen desempeño, avalado por experimentos con varios corpus textuales y pruebas estadísticas, al tener implícito una sola técnica de agrupamiento, *K-Star*; se desconoce el efecto que sufriría al remplazarla por otra con características disímiles.

En este trabajo se implementaron varios algoritmos de agrupamiento documental, siguiendo la metodología para el cálculo de la función de similitud *OverallSimSUX*, para documentos XML.

Se realizó la implementación de la herramienta OSSM Clustering, que incluye varias técnicas de agrupamiento de documentos, acopladas a la metodología mencionada y permite además, incorporarle otras técnicas de manera sencilla.

Se comprobó que la metodología se comportó de manera similar, al variar los algoritmos y funciones de similitud, lo que demuestra la estabilidad de la misma.

ABSTRACT

At the Center for Informatic Studies (CEI) of Universidad Central "Marta Abreu" of Las Villas (UCLV) has proposed a methodology for clustering XML documents by combining structure and content, through the making of a new function similarity. Is methodology is supported by general procedure implemented in a system for clustering of scientific articles in XML format (LucXML) recovered. Although this method shows good performance, supported by experiments with various text corpora and statistical tests, having a single technique implicit clustering, K-Star; the effect would suffer to replace it by another with different characteristics is unknown.

In this paper several documentary clustering algorithms were implemented following the methodology for calculating the similarity function *OverallSimSUX* for XML documents.

OSSM Clustering implementing the tool, which includes various techniques for grouping of documents, coupled to the above methodology was performed.

Tabla de Contenidos

INTRODUCCIÓN

XML (*Extensible Markup Language*) es un metalenguaje desarrollado por el W3C[1] proveniente de GML (*Generalized Markup Language*) que surgió por la necesidad que tenía la empresa de almacenar grandes cantidades de información. Un documento XML es una estructura jerárquica autodescriptiva de información, que consiste en un conjunto de átomos, elementos compuestos y atributos (Dalamagas et al., 2006).

A esto se añade que los documentos XML contienen su información en forma semiestructurada (Abiteboul, 1997) ya que incorporan estructura y datos en una misma entidad. Las etiquetas existentes en los documentos XML permiten la descripción semántica del contenido de los elementos. De este modo, la estructura de los documentos puede ser explotada para realizar recuperación de documentos relevantes (Guerrini et al., 2006).

Gestionar el conocimiento a partir de la información encontrada es fundamental en el trabajo científico (Passoni, 2005). Sin embargo, la gestión de información científica 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, lo cual constituye la motivación principal de este trabajo.

Existen varias formas de gestionar el conocimiento: la categorización, la clasificación y el agrupamiento (Dixon, 1997).

Particularmente, el agrupamiento nos permite organizar la información, delimitar la información relevante y descubrir nuevo conocimiento a partir de la información disponible en una colección especificada u obtenida como resultado de un proceso de recuperación de información. Para una eficiente organización y recuperación de los documentos XML relevantes, una posible solución es agruparlos basándose en su estructura y/o en su contenido (Tien T., 2007).

El desarrollo de sistemas que faciliten a los usuarios gestionar grandes colecciones de documentos, mediante la organización y extracción del conocimiento es una necesidad real.

[1] http://www.w3c.org

Estos sistemas a partir de una colección personal presentada como entrada, deben proponer como salida, grupos homogéneos de documentos afines y la calidad con que fueron obtenidos los grupos, proporcionando el control para la evaluación de los resultados del agrupamiento obtenido (Arco, 2009).

En el Centro de Estudios de Informática (CEI) de la Universidad Central "Marta Abreu" de las Villas (UCLV) se ha propuesto una metodología para el agrupamiento de documentos XML combinando estructura y contenido, a través de la confección de una nueva función de similitud. Es metodología está soportada en un procedimiento general implementado en un sistema para para el agrupamiento de artículos científicos en formato XML (*LucXML*), recuperados (Fuentes, 2013). Aunque esta metodología muestra buen desempeño, avalado por experimentos con varios corpus textuales y pruebas estadísticas, al tener implícito una sola técnica de agrupamiento, *K-Star* (Shin and Han, 2003); se desconoce el efecto que sufriría al remplazarla por otra con características disímiles.

Lo antes expuesto ratifica una problemática que la ciencia aún no aborda de manera completa y justifica el siguiente **planteamiento de investigación**:

La metodología propuesta presenta en sus etapas de agrupamiento una sola técnica, *K-Star*. Se desconoce el efecto que sufriría al remplazarla por otra con características disímiles. Sin embargo para lograr avalar totalmente la metodología, es necesario validarla con otras técnicas de agrupamiento de diversas clasificaciones.

Por lo que el **objetivo general** de esta investigación consiste en implementar varios algoritmos de agrupamiento documental, siguiendo la metodología para el cálculo de la función de similitud *OverallSimSUX*, para documentos XML.

Este objetivo se desglosa en los siguientes **objetivos específicos**:

1. Analizar los algoritmos de agrupamiento reportados como clásicos para determinar las variantes a implementar.
2. Implementar los algoritmos seleccionados.
3. Diseñar e implementar una herramienta de ayuda al agrupamiento de documentos XML.
4. Evaluar los resultados obtenidos a partir de corpus de XML.

Las **preguntas de investigación** planteadas son:

1. ¿Cómo acoplar las técnicas seleccionadas, de diversas características, a la metodología?
2. ¿En qué medida las técnicas seleccionadas afectarán el desempeño de la metodología?

Como respuestas a las preguntas de investigación y después de haber realizado el marco teórico se formuló la siguiente **hipótesis de investigación**:

H1: El cambio de técnica de agrupamiento en la metodología a comprobar, no afecta significativamente o mejora los resultados del agrupamiento de documentos XML.

La **tesis** está **estructurada** en tres capítulos. En el Capítulo 1 se realiza un estudio de las técnicas de agrupamiento existentes, especificando en las destinadas a los documentos XML. En el Capítulo 2 se presenta la implementación de varias técnicas de agrupamiento acopladas a la metodología a evaluar, así como una herramienta que las soporta. En el Capítulo 3 se presenta la evaluación del efecto de las técnicas en la metodología y la descripción a nivel de usuario de la herramienta OSSM Clustering. Este documento culmina con las conclusiones, recomendaciones, referencias bibliográficas y anexos.

1

ACERCA DE LAS TÉCNICAS DE AGRUPAMIENTO

1. ACERCA DE LAS TÉCNICAS DE AGRUPAMIENTO

El conocimiento se puede gestionar de diversas formas y hacerlo requiere de la integración de varias áreas del saber: el descubrimiento de conocimiento en bases de datos, la minería de datos y 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 (Dixon, 1997, Tan, 1999). Dentro de éstos, el agrupamiento permite organizar la información, determinar información relevante y crear nuevo conocimiento a partir de la información disponible. En este capítulo se referirán algunas técnicas existentes para realizar el agrupamiento.

1.1 Agrupamiento

Existen varias definiciones de agrupamiento, entre estas se encuentra la formalizada en (Jain et al., 1999): "El análisis de grupos organiza los datos mediante la extracción de la estructura subyacente en ellos como una partición de individuos o como una jerarquía de grupos. La representación puede entonces ser analizada para ver si los datos fueron agrupados acorde a ideas preconcebidas o es necesario sugerir nuevos experimentos". Así, el análisis de grupos es descrito como una herramienta para el descubrimiento porque tiene la potencialidad de revelar relaciones basadas en datos complejos no detectadas previamente.

Un algoritmo de agrupamiento intenta encontrar grupos naturales de datos basándose principalmente en la similitud y relaciones de los objetos, de forma tal que se obtenga una distribución interna del conjunto de datos mediante su particionamiento en grupos. Cuando el agrupamiento se basa en la similitud de los objetos, se desea que los objetos que pertenecen al mismo grupo sean tan similares como se pueda y los objetos que pertenecen a grupos diferentes sean tan diferentes como sea posible (Höppner et al., 1999, Kruse et al., 2007). En otras palabras, seguir el principio de maximizar la similitud dentro del grupo y minimizar la similitud entre los grupos (Anderberg, 1973).

El concepto de "similitud" tiene que ser especificado acorde a los datos. En la mayoría de los casos, los datos son vectores de valores reales, entonces se requieren algunas medidas (distancias, similitudes, o disimilitudes) para cuantificar el grado de asociación entre ellos.

Algunos algoritmos de agrupamiento tienen un requerimiento teórico para el uso de una medida específica, pero lo más común es que el investigador seleccione qué medida utilizará con determinado método. Las similitudes más utilizadas para comparar documentos son *Jaccard*, *Dice* y *Coseno* (Frakes and Baeza-Yates, 1992). Una valoración del impacto de la distancia *Euclidiana* y los coeficientes *Jaccard* y *Coseno* en dominios textuales se presentó en (Strehl et al., 2000) mostrando que para dominios textuales la distancia *Euclidiana* no reporta buenos resultados del agrupamiento. El cálculo de la similitud entre un par de documentos según *Jaccard*, *Dice* y *Coseno* se muestra en el Anexo 1.

Al mismo tiempo, es un reto descubrir grupos en datos que al relacionarse forman una estructura interesante para el análisis. Este tipo de datos ha tenido una mejor descripción cuando se representa como una colección de objetos interrelacionados y enlazados (Getoor and Diehl, 2005). El enlace entre objetos es un conocimiento que puede ser explotado en el agrupamiento, ya que rasgos de objetos enlazados están correlacionados, y es probable la existencia de enlaces entre objetos que tienen elementos comunes (Arco, 2009).

1.2 Algunas clasificaciones de las técnicas de agrupamiento

Para realizar análisis de grupos han sido propuestos una gran variedad de algoritmos de agrupamiento. Estos pueden ser clasificados de diversas formas: tipo de los datos de entrada del algoritmo, criterios para definir la similitud entre los objetos, conceptos en los cuales se basa el análisis y forma de representación de los datos, entre otros. En general, en esta clasificación se distinguen dos tipos: aquellos que forman particiones y los jerárquicos (Kruse et al., 2007).

Los métodos que forman particiones tienen como objetivo encontrar la mejor partición de los datos en *k* grupos ($k \in \mathbb{N}, k > 0$) basado en una medida de similitud dada y conservar el espacio de particiones posibles en *k* subconjuntos solamente. La mayoría de los algoritmos que siguen esta técnica son esencialmente basados en prototipos, comienzan con una partición inicial, usualmente aleatoria, y proceden con su refinamiento (Kruse et al., 2007). Uno de los algoritmos perteneciente a esta clasificación y que ha sido ampliamente utilizado es el *k-medias* (*k-means*) (Xiong et al., 2006). Otro algoritmo dentro de esta categoría es *PAM* (Partitioning Around Medoids) (Porter, 1980a); el objetivo de *PAM* es determinar un objeto representativo para cada grupo. Por otra parte *CLARA* (Clustering Large Applications), es una implementación

de *PAM* en un subconjunto de los datos, (Aslam, 2000). *K-Mode* y *K-prototypes* (Olmos and Martinez, 2005) están basados en *k-means*.

Por otra parte, los algoritmos jerárquicos hacen una descomposición jerárquica de los objetos. Dentro de ellos, los aglomerativos (bottom-up), comienzan considerando que cada objeto constituye un grupo, por tanto inicialmente existen tantos grupos como objetos tiene la colección, y sucesivamente los van uniendo, hasta que todos los objetos formen un único grupo, generalmente considerando una medida de distancia. Mientras que los divisivos (top-down) consideran inicialmente que existe un único grupo al cual pertenecen todos los objetos y sucesivamente van dividiendo los grupos en grupos más pequeños, hasta que cada grupo contenga un único objeto. La construcción de la jerarquía se puede detener por criterios automáticos o del usuario. Trabajos como (Cheng et al., 2006, Cheng et al., 2005) combinan la estrategia divisiva y la aglomerativa. Por su parte BIRCH (Zhang et al., 1996) utiliza una estructura jerárquica denominada *CF-Tree* para particionar los datos analizados de una forma incremental y dinámica. Otros trabajos dentro de esta clasificación son CURE (Guha et al., 1998) y ROCK (Guha et al., 1999).

Otra clasificación, no mutuamente excluyente a las ya presentadas, considera la forma de manipular la incertidumbre en términos del solapamiento de los grupos: agrupamiento duro y borroso (Höppner et al., 1999).

Las técnicas duras pueden ser deterministas o con solapamiento. Las deterministas crean una partición, donde los grupos son mutuamente excluyentes y exhaustivos del universo de objetos. Los algoritmos con solapamiento crean un cubrimiento, donde un objeto puede pertenecer a más de un grupo.

Las borrosas se subdividen en probabilísticas y posibilistas (Kruse et al., 2007). Dentro de este tipo de técnicas se encuentra la propuesta por (Aslam et al., 2004) que propone un cambio al clásico fuzzy *c-means*, que consiste en una nueva estrategia para seleccionar los centros iniciales de los grupos.

En la **Figura 1.1** se muestra una taxonomía según alguna de las clasificaciones de las técnicas de agrupamiento.

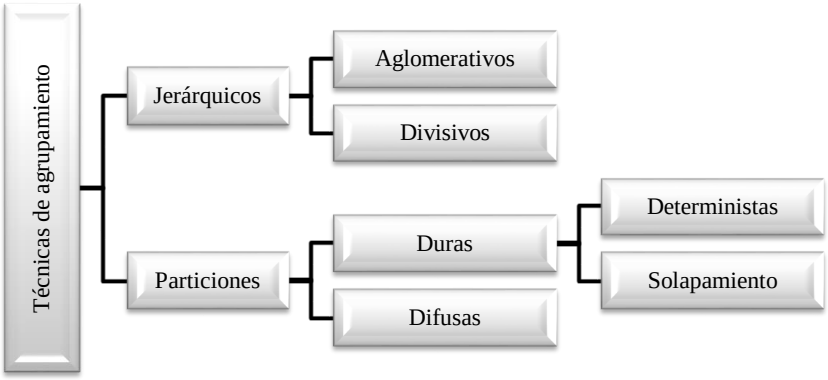


Figura 1.1 Taxonomía según alguna de las clasificaciones de las técnicas de agrupamiento.

Las Redes Neuronales Artificiales (RNA) han sido utilizadas extensamente en las últimas décadas tanto para agrupamiento como para clasificación. Entre las principales características se tiene la arquitectura de procesamiento paralela y distribuida, la capacidad de aprender relaciones no lineales complejas, el entrenamiento y la adaptación al dominio de datos. Las redes neuronales competitivas son las más usadas para el agrupamiento de datos. (Perez-Suarez and Medina-Pagola, 2007).

Comentado [u1]: Revisar

Por otra parte existen métodos evolutivos, que hacen uso de operadores evolutivos y una población de soluciones para obtener la partición de datos globalmente óptima. Las soluciones candidatas para el problema de agrupamiento son codificadas como cromosomas. Los operadores transforman uno o más cromosomas de entrada en uno o más de salida, siendo los más populares: selección, combinación y mutación. Adicionalmente se aplica una función de evaluación a los cromosomas para determinar la probabilidad de un cromosoma de pasar a la siguiente generación. (Perez-Suarez and Medina-Pagola, 2007).

Los métodos evolutivos se destacan por ser técnicas de búsqueda global, a diferencia del resto de métodos que realizan búsquedas locales.

1.3 Agrupamiento de documentos XML

Algunos autores plantean que los documentos son unidades indivisibles e independientes (Martín, 2007). Estas unidades pueden representar obras literarias, artículos científicos,

imágenes, etc. Reflexionando brevemente sobre el concepto de documento, se pueden encontrar múltiples tipos en los que resulta más natural tratarlos como un conjunto de partes; entre estos se encuentran los artículos científicos, que normalmente constan de título, resumen, palabras claves, una serie de secciones (que pueden dividirse en varias subsecciones y así sucesivamente), conclusiones, entre otras (Gil-García et al., 2003). Consecuentemente un conjunto dado de documentos $D = \{D_1,\ldots, D_m\}$, se corresponden con un conjunto de unidades estructurales $UE = \{UE_1,\ldots, UE_n\}$. De esta forma, desaparece el concepto de documento como unidad indivisible (Martín, 2007). Un tipo de documento que defiende claramente esta última idea son los documentos XML.

1.3.1 **Qué es XML**

Un documento XML es una estructura jerárquica autodescriptiva de información, que consiste en un conjunto de átomos, elementos compuestos y atributos (Dalamagas et al., 2006). A esto se añade que los documentos XML contienen su información en forma semiestructurada (Abiteboul, 1997) ya que incorporan estructura y datos en una misma entidad.

En la **Figura 1.2** se muestra un ejemplo de documento XML correspondiente a un artículo científico, el árbol que contiene la estructura de este documento se muestra en la **Figura 1.3**.

En este ejemplo es posible observar como el artículo está dividido en varias partes, que a su vez pueden estar divididas en subpartes. En un documento XML estas partes se llaman elementos, y se les señala mediante etiquetas.

Una etiqueta consiste en una marca hecha en el documento, que señala una porción de éste como un elemento, representando de esta forma un fragmento de información con un sentido claro y definido. Las etiquetas tienen la forma `<nombre> contenido </nombre>`, donde *<nombre>* es el nombre del elemento que se está señalando y *</nombre>* indica el fin de la misma.

Las etiquetas existentes en los documentos XML permiten la descripción semántica del contenido de los elementos. De este modo, la estructura de los documentos puede ser explotada para realizar recuperación de documentos relevantes (Guerrini et al., 2006).

El agrupamiento de los documentos XML basándose en su estructura y/o en su contenido contribuye a una eficiente organización y recuperación de los documentos relevantes (Tien T., 2007).

Estos algoritmos de agrupamiento se mantienen con un gran auge en los últimos tiempos, como consecuencia del crecimiento de datos electrónicos en este formato. El agrupamiento es fundamental para una eficiente organización y recuperación de los documentos XML relevantes. Sin embargo, la mayoría de los métodos existentes explotan solo la información incluida en el contenido o sólo la información contenida en la estructura (Tien T., 2007).

```
<?xml version="1.0" encoding="ISO-8859-1" ?>

<Artículo>
   <Título>
      "Agrupamiento de documentos estructurados"
   </Título>
   <Resumen>
      En este trabajo se propone realizar un agrupamiento…
   </Resumen>
   <Introducción>
      XML es el lenguaje mas utilizado en para…
   </Introducción>
   <Secciones>
      <Sección1>
         La estructura de los documentos XML juega un papel[1]…
      </Sección1>
      <Sección2>
         Un algoritmo de agrupamiento…
      </Sección2>
                                  …
      <Secciónn>
         La estructura de los documentos XML juega un papel…
      </Secciónn>
   </Secciones>
                                  …
   <Referencias>
      1. Autor, XML, su estructura…
   </Referencias>
</Artículo>
```

Figura 1.2 Ejemplo de un documento XML correspondiente a un artículo científico.

1.3.2 Técnicas para el agrupamiento de documentos XML

Cuando se trata de documentos XML, los algoritmos de agrupamiento se clasifican principalmente en tres grupos: los que se centran solo en el contenido de los documentos (Kurgan et al., 2002, Shen and Wang, 2003), realizando un análisis solamente léxico, o incluyendo elementos sintácticos o semánticos en el estudio, aquellos que realizan análisis léxico generalmente consideran los documentos como una bolsa de palabras; sin embargo, un buen proceso de agrupamiento no puede descartar el uso de la estructura (Tran et al., 2008b), por lo que están los algoritmos que utilizan solo la estructura, considerando que esta juega un papel importante en el agrupamiento para ciertas aplicaciones específicas y los que combinan

ambas 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).

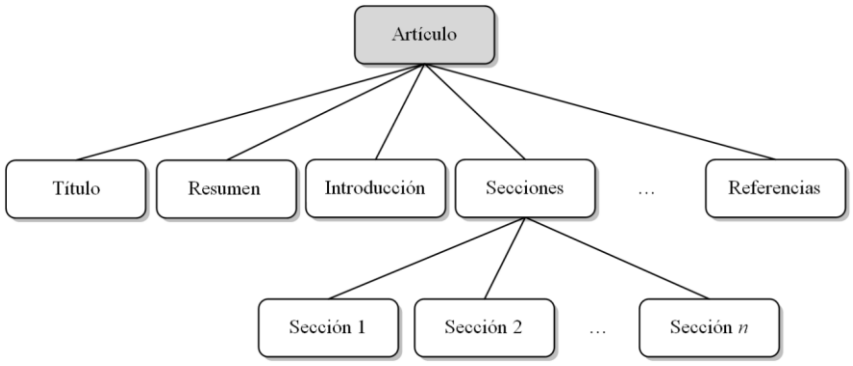


Figura 1.3 Ejemplo de un árbol correspondiente a un artículo científico.

1.3.2.1 Algoritmos que utilizan solo la estructura de los documentos

La estructura jerárquica de los documentos XML juega un papel importante en el agrupamiento (Nayak, 2006). Varios trabajos utilizan la estructura en forma de árbol para realizar el agrupamiento, por lo que dado un documento XML, el agrupamiento se realizaría obviando el contenido, utilizando solamente la estructura correspondiente. A continuación se mencionan algunos de estos.

Una de las variantes para comparar árboles es utilizar la distancia *Tree-Edit* (TE), que intenta transformar un árbol A_1 en un árbol A_2, realizando una secuencia de operaciones (inserción, eliminación y substitución de nodos. De manera que mientras menor sea la cantidad de operaciones necesarias en la transformación, mayor será la similitud entre los árboles correspondientes a los documentos comparados. Una gran cantidad de trabajos utilizan TE (Flesca et al., 2005, Dalamagas et al., 2006, Nierman and Jagadish, 2002, Chawathe et al., 1996, Chawathe, 1999, Zhang and Shasha, 1989, Selkov, 1977) o alguna de sus variantes.

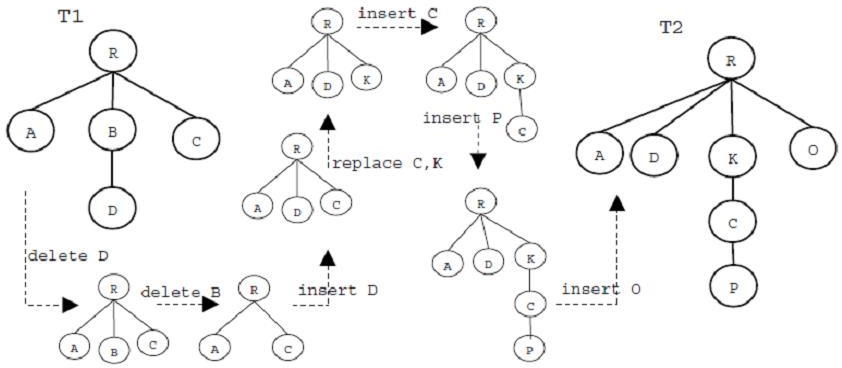


Figura 1.4 Uso de la distancia Tree-Edit.

Otra variante es a través del cálculo de los *Closed Frequent Subtrees* (CFST). Los autores del artículo referenciado en (Kutty et al., 2007) plantean que dado un conjunto de árboles *T*, existe para un árbol T_i un subárbol ST_i que mantiene la misma relación padre-hijo que T_i; se calcula la frecuencia $f(ST_i)$, que no es más que la cantidad de árboles pertenecientes a *T* de los que ST_i es subárbol, y este es frecuente (*FST*) si $f(ST_i)$ es mayor que un umbral determinado. Se puede decir que dos árboles pertenecen a un mismo grupo si tienen el mismo *FST*.

1.3.2.2 Algoritmos que combinan estructura y contenido

Sin embargo, para obtener mejores resultados en el agrupamiento, es esencial utilizar ambas dimensiones (Kutty et al., 2008). A continuación, se mencionan algunos trabajos existentes en la literatura.

Una primera variante muy sencilla 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, Giannopoulos and Veltkamp., 2002, Karmarkar, 1984, Yang and Chen, 2002).

Otro enfoque se muestra en (Tran et al., 2008a), ver **Figura 1.5**, donde realiza primeramente un agrupamiento teniendo en cuenta solo la estructura de los documentos, posteriormente proponen el uso del Latent Semantic Kernel (Cristianini et al., 2002) para determinar la similitud entre el contenido de los documentos y realiza un agrupamiento teniendo en cuenta el contenido.

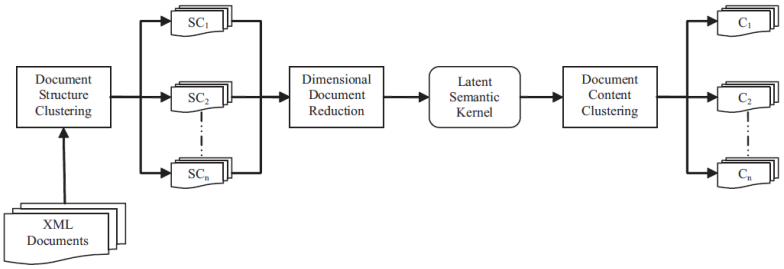


Figura 1.5 Esquema de algoritmo que utiliza estructura y contenido para agrupar documentos XML.[2]

1.3.3 Algoritmo de agrupamiento basado en la similitud OverallSimSUX

En (Gil-García et al., 2003) se propone una metodología general para la aplicación del agrupamiento de documentos XML, con el propósito de facilitar a los usuarios enfrentarse a grandes colecciones de documentos, a partir de su organización; contribuyendo a la extracción de conocimiento relevante. Este modelo se inicia a partir del resultado de un proceso de recuperación de información (Berry, 2004) 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.

Comentado [u2]: Arreglar referencia a (Fuentes, 2013)

Una visión gráfica del esquema del modelo general presentado en este trabajo se muestra en la **Figura 1.6**. En la **Figura 1.7** se muestra los cuatros módulos principales que contiene el modelo propuesto.

[2] Tomado de TRAN, T., KUTTY, S. & NAYAK, R. 2008a. Utilizing the Structure and Data Information for XML Document Clustering. *INEX*, 402-410.

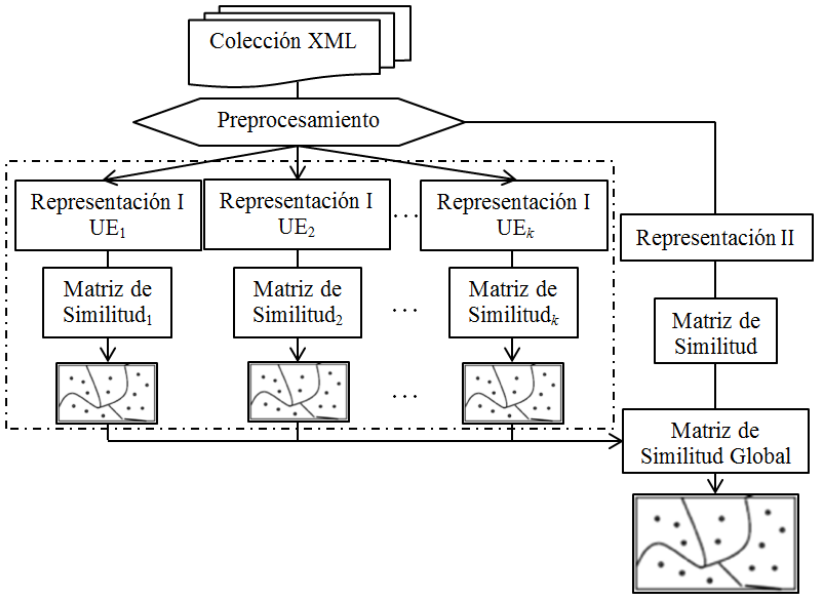


Figura 1.6 Esquema que muestra el modelo general para el agrupamiento por OverallSimSUX.

Módulo 1. Recuperación de la información o especificación del corpus textual a procesar, identificando en cada documento recuperado las Unidades Estructurales (UE).

Módulo 2. Representación del corpus textual obtenido.

Submódulo 2.1. Tratar cada UE como una colección diferente. Obtener por cada UE una representación basada en la VSM clásica, denominada en este trabajo *Representación I*.

Submódulo 2.2 A partir de las UE identificadas. Obtener una representación global que tendrá en cuenta el contenido en función de la estructura. Esta representación, es denominada *Representación II*.

Módulo 3. Agrupamiento de los documentos.

Submódulo 3.1. Realizar un agrupamiento por cada UE, a partir de la matriz de similitud resultante de la *Representación I*.

Submódulo 3.2. Obtener la matriz de similitud, a partir de la *Representación II*.

Submódulo 3.3. Realizar el agrupamiento general a partir del cálculo de la función de semejanza, propuesta en esta investigación, que utiliza como entrada el resultado de los submódulos 3.1 y 3.2.

Módulo 4. Valoración (validación y verificación) de los grupos obtenidos.

Figura 1.7 Módulos principales del modelo para el agrupamiento por *OverallSimSUX*.

El modelo presentado en (Gil-García et al., 2003) se basa en la similitud OverallSimSUX allí propuesta y que en la siguiente sección se detalla.

Comentado [u3]: Arreglar referencia a (Fuentes, 2013)

1.3.3.1 Similitud Coseno, función de semejanza OverallSimSUX

A partir de la función de semejanza β y del conjunto de documentos *MI*, se construye una matriz de similitud que refleja las relaciones de semejanza entre todos los objetos sujetos a estudio. Se considera la pertenencia a un grupo analizando el comportamiento global de las semejanzas entre los objetos. Esto se logra siguiendo el criterio β-semejantes que se describe en (Ruiz-Shulcloper, 1995). En (Gil-García et al., 2003) se puede observar la definición de (β-semejantes).

Comentado [u4]: Arreglar referencia a (Fuentes, 2013)

La relación estructural existente entre los documentos XML puede aportar mejores resultados al agrupamiento, cuando se utiliza el contenido en función de la relación entre sus unidades estructurales (Gil-García et al., 2003). En (Gil-García et al., 2003) se propone la medida de similitud *OverallSimSUX* que facilita capturar el grado de se semejanza entre estos documentos, tomando como génesis la relación existente entre sus unidades estructurales, cuando se manipulan como colecciones independientes y la similitud global. Esto se expresa formalmente a través de las definiciones de λ-pertenencia y *OverallSimSUX*, relacionadas a continuación.

Comentado [u5]: Arreglar

Comentado [u6]: Arreglar

Definición 1 (λ-pertenencia) Dados los objetos *i, j* se define la λ-pertenencia como la relación de pertenencia de ambos objetos a un mismo grupo, a partir de los resultados del agrupamiento. Esta pertenencia se formaliza en la ecuación (1).

$$\lambda(i,j) = \begin{cases} 1, & \{i,j\} \epsilon grupo_k \\ 0, & i \,\epsilon grupo_n \wedge j \,\epsilon grupo_m \end{cases} \quad m \neq n \qquad (1)$$

Definición 2 (*OverallSimSUX*) La similitud *OverallSimSUX* entre objetos *i, j* está dada por la expresión 2, en esta: $A = \{a_1, a_2, \dots, a_k\}$, donde a_k es el resultado del agrupamiento para la *Representación* I_k; s_g es la matriz de similitud coseno que se obtiene a partir de la *Representación II* y w_k es la ponderación de la UE_k.

$$f(A, s_g, i, j) = \frac{\sum_{k=1}^{m} (w_k * \lambda_{k(i,j)} + s_{g(i,j)})}{\sum_{k=1}^{m} w_k + 1} \qquad (2)$$

OverallSimSUX considera *m* como la cantidad de UE identificadas en los documentos. Esta función de similitud está normalizada por la sumatoria de los pesos de las *m* UE y el máximo

valor de la similitud global s_g (i.e. 1). Por tanto, su máximo (i.e. 1) se alcanza cuando los documentos *i*, *j* pertenecen al mismo grupo en todos los *k*-agrupamientos (i.e. $\lambda_k = 1$) y el valor de s_g es máximo (Gil-García et al., 2003).

1.3.3.2 Un algoritmo de agrupamiento basado en la similitud OverallSimSUX

En la **Figura 1.8** se muestra el algoritmo de agrupamiento propuesto en (Gil-García et al., 2003) basado en la estrategia del algoritmo de agrupamiento *K*-Star (Shin and Han, 2003, Pinto et al., 2009) y la matriz similitud *OverallSimSUX*.

Comentado [u7]: Arreglar

1. Construcción de la matriz de similitud OverallSimSUX.
2. Estimación del umbral de similitud.
3. Determinación de los núcleos iniciales del agrupamiento mediante el cálculo de la máxima similitud entre dos objetos, no asignados.
4. Asignación de los objetos que no pertenecen a los núcleos a partir de su umbral de pertenencia a los grupos ya formados.

Figura 1.8 Algoritmo de agrupamiento basado en *K*-Star.

1.4 Consideraciones finales del capítulo

Existen múltiples clasificaciones y técnicas para realizar el agrupamiento. Específicamente, las que abordan el agrupamiento de documentos XML, se dividen en: las que utilizan solo el contenido, las que utilizan solo la estructura y las que combinan ambas dimensiones. En [REFERENCIAR] se propone una metodología para el agrupamiento de documentos XML, que resultan más natural tratarlos como un conjunto de unidades estructurales y no como una unidad indivisible, entre ellos, los artículos científicos. Aunque esta muestra buen desempeño, avalado por experimentos con varios corpus textuales y pruebas estadísticas, al tener implícito la utilización de algoritmos para documentos que no poseen estructura; resulta necesario observar cuán susceptibles puede ser esta técnica al cambio del método de agrupamiento.

2

IMPLEMENTACION DE TÉCNICAS DE AGRUPAMIENTO UTILIZANDO OVERALLSIMSUX

2. IMPLEMENTACION DE TÉCNICAS DE AGRUPAMIENTO UTILIZANDO OVERALLSIMSUX

En este capítulo se especifican los detalles de diseño e implementación de una herramienta que contiene una serie de algoritmos "clásicos" de agrupamiento documental escogidos a partir de la revisión bibliográfica realizada. Estos algoritmos son insertados en la metodología para el agrupamiento de documentos XML que utiliza OverallSimSUX.

2.1 Breve descripción de los algoritmos escogidos

En esta sección se describirán de forma breve los algoritmos seleccionados para incluir en esta primera versión de la herramienta computacional. Fue de interés escoger algoritmos sencillos, pero con buenos resultados reportados en la literatura en colecciones textuales. Por otra parte se seleccionaron algoritmos de diferentes clasificaciones, específicamente: un algoritmo duro y determinista, uno duro y con solapamiento y un algoritmo difuso.

2.1.1 Algoritmo K Means

Este algoritmo representa los grupos a través de representantes o centroides que se calculan por la media (o promedio pesado) de los elementos pertenecientes al grupo. Inicialmente, el algoritmo selecciona aleatoriamente *k* elementos como centros iniciales de los grupos y procesa cada objeto restante asignándolo al grupo con el cual tenga la mínima distancia o mayor semejanza. A partir de este punto se recalculan los centroides y se vuelve a iterar por los objetos re-asignando, cada uno a su grupo más cercano.

Este proceso se repite hasta que se alcance algún criterio de convergencia. Usualmente el criterio puede ser: (a) no se afecta el agrupamiento; es decir, ningún objeto cambia de grupo o (b) la función objetivo deja de crecer o decrecer (en dependencia si se utilizó una función de minimización de distancia o de maximización de similitud. Los grupos que se obtienen producto de este algoritmo son grupos disjuntos. La complejidad computacional de este algoritmo es O(*nkT*), donde *T* representa la cantidad de iteraciones hasta la convergencia del algoritmo, *n* la cantidad de elementos y *k* la cantidad de grupos a formar. (Peng and Zhu., 2006)

El Pseudo-código del algoritmo puede observarse en la **Figura 2.1**

```
Input: D := {d_1, d_2, ..., d_n}- document collection,
       k - number of groups to be formed
Output: SC - Set of k clusters
1  SC := ∅;
   // Selecting initial k seeds
2  for i = 1 to k do
3  |  r := {d | d = Rand(D) ∧ d was not previously selected};
4  |  C_i := {r};
5  |  C_i.Center := r;
6  |  SC := SC ∪ C_i;
7  end
8  repeat
   |  // forming the set of clusters
9  |  forall d_j ∈ D do
10 |  |  i := arg_i min{F(d_j, C_i.Center), C_i ∈ SC};
11 |  |  C_i := C_i ∪ d_j;
12 |  end
13 |  cond := Evaluate(a,b) ;                    // Evaluating the stop-conditions
   |  // updating cluster's representative
14 |  forall C_i ∈ SC do
15 |  |  C_i.Center := Average(C_i);
16 |  |  C_i := ∅;
17 |  end
18 until cond ≠ true ;
```

Figura 2.1 Pseudo-código del algoritmo *K-Means*

En el pseudo-código presentado, se asumió que existe una función *Rand* que selecciona aleatoriamente un documento de un conjunto dado y una función *Evaluate* que verifica el cumplimiento de las condiciones de parada (a) y (b).

Este algoritmo tiene varias deficiencias importantes. En primer lugar sólo puede ser utilizado si los objetos están descritos en un espacio métrico, no agrupa correctamente los objetos aislados y puede caer en extremos locales de la función objetivo. Otra deficiencia importante es que, la calidad del agrupamiento resultante está determinada por la selección inicial de los centroides. Por último, es importante mencionar que en este algoritmo es necesario determinar a priori el número de grupos en los que se desea particionar la colección.

2.1.2 **Algoritmo Generalized Star**

El algoritmo *Generalized Star* fue propuesto por Pérez y Medina (Perez-Suarez and Medina-Pagola, 2007) y se apoya en el planteamiento de Aslam (Aslam et al., 2004) de que, el

Comentado [u8]: Trata de hacerlo con el mismo formato que el procedimiento del capítulo 1, porque esta imagen dice algoritmo 2 y no queda elegante en la redacción; si es mucho trabajo arreglala y quitale algoritmo 2

cubrimiento de G_β a través de sub-grafos en forma de estrella permite obtener grupos con una semejanza relativamente alta entre los documentos que lo componen. Utilizando este planteamiento y definiendo un nuevo tipo de sub-grafo en forma de estrella, este algoritmo a partir del cubrimiento de G_β con sub-grafos de este tipo, construye un conjunto de grupos que pueden ser solapados.

A diferencia de *Star*, donde sólo se utiliza el grado de los vértices para definir el sub-grafo, *Generalized Star* define un grupo de conjuntos para cada uno de los vértices del grafo, y apoyados en estos conjuntos, define lo que es un sub-grafo en forma de estrella generalizada y posteriormente, una heurística que define cómo será el agrupamiento que se obtenga.

Los conjuntos de Satélites Débiles (*WeakSats*) y Satélites Potenciales (*PotSats*) de un vértice *v*, se definen por las expresiones (1) y (2):

$$v.WeakSats = \{s \in v.Adj || v.Adj| \geq |s.Adj|\} \quad \textbf{(1)}$$

$$v.PotSats = \{s \in v.Adj || v.WeakSats| \geq |s.WeakSats|\} \quad \textbf{(2)}$$

El grado WeakSats y PotSats de un vértice *v*, se define como la cardinalidad de los conjuntos de satélites débiles y potenciales de *v* respectivamente.

Teniendo en cuenta las definiciones anteriores, un sub-grafo en forma de estrella generalizada (sub-grafo *EG*) se define como un sub-grafo de *m*+1 vértices, en el cual existe un vértice *c* denominado "centro" y *m* vértices llamados "satélites", cumpliéndose que: (i) existe una arista entre cada satélite y el centro, (ii) el centro satisface la expresión (3).

$$\forall s \in c.PotSats, |c.PotSats| \geq |s.PotSats| \quad \textbf{(3)}$$

El pseudo-código del algoritmo *GStar* puede observarse en la ¡Error! No se encuentra el origen de la referencia.**2**.

Input: $D := \{d_1, d_2, \ldots, d_n\}$ - document collection,
β - similarity threshold
Output: SC - Set of clusters

1 "Build $G_\beta = \langle V, E_\beta \rangle$ from D";
2 **forall** *vertex* $v \in V$ **do**
3 $v.WeakSats := \{s \mid s \in v.Adj \wedge |v.Adj| \geq |s.Adj|\}$;
4 **forall** *vertex* $v \in V$ **do**
5 $v.PotSats := \{s \mid s \in v.Adj \wedge |v.WeakSats| \geq |s.WeakSats|\}$;
6 $v.NecSats := v.PotSats$;
7 **end**
8 $L := V$;
9 $X := \emptyset$;
10 **while** $L \neq \emptyset$ **do**
11 $v := \arg\max\{|v_i.PotSats|, v_i \in L\}$;
12 `Update(d,X,L)`;
13 **end**
14 "Sort X in ascending order by PotSats degree";
15 $SC := \emptyset$;
16 **forall** *center* $c \in X$ **do**
17 **if** c *is redundant* **then** $X := X \setminus \{c\}$;
18 **else** $SC := SC \cup \{\{c\} \cup c.Adj\}$;

Comentado [u9]: Has lo mismo quítale algoritmo 14

El procedimiento *Update* (ver **Figura 2.13**) se aplica para marcar como centro al vértice seleccionado (*v*), eliminarlo de la lista de candidatos y para actualizar las propiedades de los vértices del conjunto Satélites Necesarios (*NecSats*) del vértice *v*.

El conjunto de (*NecSats*) de un vértice *v*, es el conjunto de los vértices adyacentes que dependen de *v* para pertenecer a algún grupo. Este conjunto es sólo necesario durante la selección de vértices centro. Inicialmente, el conjunto *v.NecSats* se inicializa con todos los vértices contenidos en el conjunto *v.PotSats*, pero puede decrecer en la medida que más vértices resulten cubiertos en el proceso de agrupamiento.

El funcionamiento de este algoritmo, se basa en una heurística que asume como criterio para estimar la densidad de un sub-grafo *EG*, el grado *PotSats* de su vértice centro. Como primer paso del algoritmo se construye el grafo G_β que representa a la colección. Posteriormente se construye una lista *L* con todos los vértices del grafo y utilizando ésta, se realiza un proceso iterativo que selecciona en *L* el vértice *v* de mayor grado *PotSats*, se adiciona este a *X*, se

actualizan sus vértices adyacentes consecuentemente y se elimina de *L*. El proceso termina cuando la lista *L* queda vacía.

Al terminar el proceso de cubrimiento, se ordenan los vértices en *X* de acuerdo a su grado *PotSat* y se verifica cada centro con el objetivo de eliminar aquellos centros redundantes que pudieron ser seleccionados en el proceso anterior. Para determinar si un centro es redundante se utiliza el concepto de conjunto de Centros Potenciales de un vértice.

Procedimiento `Update`

Input: v - Selected center,
C- Set of clusters centers,
L - Set of unprocessed vertices
Output: C, L

```
1  X := X ∪ {v};
2  L := L \ {v};
3  forall s ∈ v.NecSats do
4  |  s.NecSats := s.NecSats \ {v};
5  |  if s.NecSats = ∅ then L := L \ {s};
6  |  forall c ∈ s.Adj \ {v} do
7  |  |  c.NecSats := c.NecSats \ {s};
8  |  |  if (c.NecSats = ∅) ∧ (c.Adj ∩ X ≠ ∅) then L := L \ {c};
9  |  end
10 end
```

Figura 2.1 Pseudo-código del procedimiento *Update*.

El conjunto de Centros Potenciales (*PotCenters*) de un vértice *v*, es el conjunto de todos los vértices adyacentes a *v* que potencialmente pueden formar un sub-grafo *EG* en el cual esté incluido *v*. Este conjunto está definido por la siguiente expresión (4):

$$v.PotCenters = \{c \in v.Adj | c.PotSats \neq\} \quad \textbf{(4)}$$

Un centro $c \in X$ se considera redundante si cumple las condiciones siguientes:

1. $c.PotCenters \cap X \neq 0$; es decir, c tiene algún vértice centro adyacente.
2. $\forall s \in c.PotSats, (s \in X) \vee (s.PotCenters \cap (X\{c\}) \neq 0)$; es decir, cada satélite potencial de *c*, es centro o tiene algún centro *c'* adyacente en *X* tal que *c'*≠ *c*.

Este algoritmo tiene una complejidad computacional de $O(n^3)$, no necesita del número de grupos a obtener, no produce grupos ilógicos ni redundantes y no deja elementos sin cubrir, solucionando deficiencias del algoritmo *Star*.

Este algoritmo puede construir diferentes agrupamientos para una misma colección puesto que sólo selecciona en cada iteración un único elemento para incluir en *X*, por lo que es dependiente de los datos. Sin embargo, al permitir que los centros de los sub-grafos sean adyacentes, el algoritmo *GStar* evita descubrir grupos ilógicos, reduce el número de configuraciones en las que el algoritmo pueda dar soluciones diferentes, además de que garantiza obtener una cantidad de grupos menor que los algoritmos anteriores, reduce la influencia del orden de análisis en la calidad de los grupos, obteniendo agrupamientos con valores de calidad muy cercanos.

2.1.3 **Algoritmo Fuzzy K Means**

Este algoritmo mantiene el mismo principio planteado en el *K-Means* para el cálculo de los centroides, solo que en lugar de calcular a que grupo pertenece cada elemento, se actualiza en cada iteración la matriz *U*, que especifica que grado de pertenencia tiene cada elemento a cada grupo.

Dado un conjunto de documentos $X = \{x_1, x_2, \ldots, x_n\}$, el algoritmo FCM divide el conjunto en c $(1 < c < n)$ grupos difusos con vértices $v = \{v_1, v_2, \ldots, v_c\}$ minimizando la función objetivo que se define a continuación, en la expresión (5).

$$J_m = \sum_{j=1}^{c} \sum_{i=1}^{n} u_{ij}^{m} d_{ij}^{2} \qquad \textbf{(5)}$$

Donde d_{ij} es la distancia del documento i al centro del grupo j, y u_{ij} es el grado de membrecía del documento i al grupo j y $m \in [1, \infty)$ que es el grado de pertenencia de los grupos finales. Los centroides de los grupos son calculados de la siguiente manera:

$$v_j = \frac{\sum_{i=1}^{n} u_{ij}^{m} x_i}{\sum_{i=1}^{n} u_{ij}^{m}} \qquad \textbf{(6)}$$

Comentado [u10]: Teclear fórmula editada

La matriz de pertenencia debe satisfacer:

- $u_{ij} \epsilon [0,1]$ Δ_i=1,2,. . .,n ; Δ_j=1,2,. . .,c
- $\sum_{j=1}^{c} u_{ij}$=1 , Δ_i=1,2,. . .,n (7)
- 0 < $\sum_{i=1}^{n} u_{ij}$ < n , Δ_j=1,2,. . .,c & 1 < n < ∞

El Pseudocódigo del algoritmo FCM se describe a continuación:

Paso 0: Seleccionar *k* centroides iniciales del conjunto de documentos

Paso 1: Inicializar los valores de pertenencia U^{t-1}= $[u_{ij}^{t-1}]$ de cada documento x_i al centroide del grupo v_j para *i* < *i* < *n* y *1* < *j* < *k* (inicialmente $t = 1$) y seleccionar m ($m > 1$).

Paso 2: Calcular los centroides de los grupos v_t = $[v_1^t, v_2^t, v_3^t, \dots, v_k^t]$ usando (2)

Paso 3: Calcular la distancia entre x_i y v_j^t para *1* ≤ *i* ≤ *n*, *1* ≤ *j* ≤ *c*.

Paso 4: Actualiza los valores de pertenencia de x_i en *t* mediante, la expresión (8):

Paso 5: Si J_m^t <= J_m^{t-1} entonces parar, sino t ← t – 1 y volvemos al Paso 2

$$u_{ij}^{t} = \frac{1}{\sum_{c=1}^{k} \left(\frac{d_{ij}}{d_{ic}}\right)^{\frac{2}{(m-1)}}} \quad (8)$$

2.1.3.1 Distancia Chebyshev

Esta medida de distancia se basa en la diferencia máxima entre los elementos. También se conoce como la distancia del tablero de ajedrez, norma *Chebyshev* o norma *L*∞. La distancia *Chebyshev* entre un elemento *x* y un centroide *v* se define como se observa en la expresión (9), (Jafar and Sivakumar, 2013).

$$d(x, v) = Max_{i=1,2,\dots,n} |xi - vi| \quad (9)$$

2.2 Diseño e implementación de los algoritmos de agrupamiento y funciones de similitud

Con el objetivo de elaborar una herramienta completamente modular, donde la implementación de los algoritmos de agrupamiento y las funciones de similitud fueran totalmente independientes a los demás módulos del programa, se diseñó una jerarquía de clases que se explica a continuación. Esto posibilita añadirle nuevos algoritmos y funciones al programa sin tocar prácticamente nada de su código, solo añadiendo la clase correspondiente al nuevo algoritmo o función se similitud

2.2.1 **Clase abstracta Algorithm**

Se creó la clase abstracta *Algorithm* de la cual hereda cualquier clase que sea un algoritmo en la herramienta. A continuación se enumeran los atributos y métodos de esta clase y la descripción de cada uno de ellos.

- `protected double [][] ms` : Es la matriz de similitud que utiliza el algoritmo para agrupar.
- `protected ArrayList<Cluster> clusters` : Es la lista donde se dejan los grupos con sus elementos al finalizar de agrupar
- `public abstract String[] getParametersNames()` : Debe devolver los nombres de los parámetros que puede varias el usuario antes de ejecutar el agrupamiento. Es llamado por la interfaz visual para mostrar los nombres de dichos parámetros.
- `abstract String getParameterDescription(String parameter)`: Debe devolver una breve descripción del parámetro con nombre *parameter* que será mostrado en un *tooltip text* en la interfaz visual al pasar el mouse encima del atributo.
- `abstract String[] getPossiblesValues(String parameter)`: Debe devolver los posibles valores que puede tomar el atributo *parameter*, estos valores serán colocados dentro de un *combo box* para que el usuario elija cual valor tomará el algoritmo. Si el atributo tiene infinitos valores puede devolver *null* y la interfaz mostrará un campo de texto para que el usuario especifique el valor.

- `abstract String getParameterValue(String parameter)`: Este debe devolver el valor del atributo en ese momento.
- `abstract void setParametersValues(String[] values)`: Será llamado por la interfaz visual con todos los valores de los parámetros especificados por el usuario en mismo orden que se le dieron en el método *getParametersNames()*. Se debe asignar cada *String* en *values* a su correspondiente atributo.
- `abstract Algorithm getInstanceCopy()`: Es llamado por cada hilo de ejecución del programa. Aquí se debe devolver una instancia del algoritmo que extienda de la clase *Algorithm*, para que cada hilo tenga una copia diferente del algoritmo.
- `abstract long start()`: Es el método que realizará el agrupamiento. Luego de concluido esta operación en la lista ArrayList<Cluster> deberán estar todos los grupos obtenidos en el proceso. Este método devuelve el tiempo que demoró el algoritmo.
- `abstract String toString()`: Cada algoritmo debe devolver una cadena identificativa. Será mostrada en el *combo box* donde el usuario elige que algoritmo desea usar.

2.2.2 Clase abstracta Similarity

Se creó la clase abstracta *Similarity* de la cual hereda cualquier clase que sea una función de similitud en la herramienta. A continuación se enumeran los atributos y métodos de esta clase y la descripción de cada uno de ellos.

- `protected double [][] tf`: Es la matriz de frecuencia de los términos en los documentos.
- `abstract double similitud(int doci, int docj)`: Devuelve la similitud del documento *i* con el documento *j*.
- `abstract Similarity getInstanceCopy()`: Es llamado por cada hilo de ejecución del programa. Aquí se debe devolver una instancia de la función de similitud que extienda de la clase *Similarity*, para que cada hilo tenga una copia diferente de la función.

- `public abstract String toString()`: Cada función debe devolver una cadena identificativa. Será mostrada en el *combo box* donde el usuario elige que función desea usar.

2.3 Procedimiento general para el agrupamiento siguiendo OverallSimSUX

En (Gil-García et al., 2003) se propone un procedimiento general para realizar el agrupamiento siguiendo la metodología del cálculo de *OverallSimSUX*. En este trabajo se reutiliza este mismo procedimiento, pues tiene como fin aplicar otras técnicas de agrupamiento, pero utilizando esta misma metodología. Por tanto solo se describirá a grandes rasgos cada módulo del procedimiento general que se puede observar en el Anexo 2. Estos son:

1. Recuperación y creación de índices a partir del corpus de documentos XML.
2. Representación de la colección.
3. Agrupamiento General a partir de la matriz de similitud basada en el cálculo de la similitud *OverallSimSUX*.
4. Evaluación local y global de los resultados del agrupamiento.

2.3.1 Proceso 1: Recuperación y creación de índices a partir del corpus de documentos XML

Como ya se ha mencionado, la entrada al modelo lo constituye la colección de documentos XML que se desea procesar. A partir de esta especificación se comienza el proceso de recuperación utilizando *Tika*. La aplicación se adapta a cualquier estructura XML, posibilitando así el análisis de cualquier colección de documentos en ese formato. Luego se indexa de cada documento las unidades estructurales que el usuario identificó utilizando las facilidades de *Lucene*.

2.3.2 Proceso 2: Representación de la colección

En esta etapa se obtiene la *Representación I* asociada a cada UE y la *Representación II* que se obtiene de toda la colección. Específicamente para obtener la *Representación I* se construye la matriz VSM clásica, que contiene en sus filas el índice de términos construido 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.

Comentado [u11]: arreglar

Comentado [u12]: Revisar

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

Para cada representación resultante se calcula una matriz de similitud utilizando como medida la similitud coseno. Se genera un agrupamiento para cada UE a partir de la similitud asociada a la *Representación I*.

La matriz de similitud global se obtiene a partir del resultado de cada agrupamiento y la matriz de similitud asociada a la *Representación II*, utilizando como medida de similitud *OverallSimSUX*, ver ecuación 2. Finalmente el agrupamiento general es el resultado de aplicar algunos de los algoritmos descritos en las secciones 2.1.1–2.1.3, utilizando la matriz de similitud confeccionada con *OverallSimSUX*.

Como resultado se obtiene una partición de la colección inicial en grupos homogéneos de documentos.

2.3.4 Proceso 4: Evaluación local y global de los resultados del agrupamiento.

Para la evaluación de los resultados se han implementado medidas externas y las propuestas por *INEX*[3] para la evaluación de técnicas de clasificación supervisadas y no supervisadas de documentos XML. La medida externa implementada es *Overall F-measure* (Steinbach et al., 2000b), basada en Precisión (Pr) y cubrimiento[4] (Re) (Frakes and Baeza-Yates, 1992). Por último se incluye el cálculo de medidas basadas en *Purity*, *Micro-Purity* y *Macro-Purity* (Pinto et al., 2009).

2.4 Diseño del Sistema OSSM Clustering

En este trabajo se propone el Sistema para el análisis de algoritmos agrupamientos documental utilizando *OverallSimSUX* (OSSM Clustering), el cual implementa el procedimiento general descrito en la sección 2.3.

Entrada: Colección de documentos en formato XML **Salida:** Grupos homogéneos de documentos afines, resumen de cada documento, y calidad del agrupamiento.

[3] Initiative for the Evaluation of XML Retrieval.

[4] 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.

```
Inicio
   1. Realizar Preprocesamiento      /* normalización, convertir tokens a
                                        minúsculas, eliminar palabras de
                                        parada, Stemming */
        – Identificar Unidades Estructurales (UE)
   2. Para cada ue ∈ UE
        – Representación-I←Realizar_Representación-I(ue)
        – Matriz_Similitud←Similitud_Function(Representación-I)
        – Agrupamientos[ue]←Realizar_Agrupamiento_ue(Matriz_Similitud)
   3. Fin_Para
   4. Representación-II←Realizar_Representación-II
   5. Matriz_Similitud←Similitud_Function(Representación-II)
   6. Matriz_Similitud_Global←OverallSimSUX(Agrupamientos,Matriz_Similitu
      d)
   7. Realizar Agrupamiento_General(Matriz_Similitud_Global)
   8. Realizar Evaluación_Agrupamiento(Agrupamiento_General)
Fin
```

Figura 2.4 Procedimiento general para el agrupamiento usando OverallSimSUX.

2.5 Cómo agregar un nuevos Algoritmos y Funciones de Similitud a OSSM Clustering

Gracias al diseño de clases utilizado en la herramienta, es posible agregarle nuevas funciones de similitud y algoritmos al software en muy pocos pasos, y lo más importante, sin cambiar absolutamente nada de su código original. Esto constituye una ventaja pues el programador del nuevo algoritmo no tendrá que gastar su tiempo en estudiar el software para ver como introducir su código.

Solo deberá leer lo referido a las clases abstractas Algorithm y Similarity para añadir nuevos algoritmos y funciones de similitud respectivamente.

Pasos para agregar su nuevo algoritmo al programa:

1. Escribir una nueva clase dentro del paquete cluster que herede de la clase abstracta Algorithm. Esto implica que deberá implementar los métodos abstractos de la misma, necesarios para que la interfaz gráfica realice todas las operaciones posibles sobre el nuevo algoritmo. Leer la especificación de cada método de la clase Algorithm

2. Crear una instancia de la misma en el constructor de la clase principal del proyecto : MainWindows.java y añadir dicha instancia a la lista algorithms de tipo ArrayList<Algorithm>

Listo, ya su algoritmo está perfectamente integrado a la herramienta

Algo muy similar se hace con la incorporación de una nueva función de similitud:

1. Escribir una nueva clase dentro del paquete cluster que herede de la clase abstracta Similarity. Esto implica que deberá implementar los métodos abstractos de la misma, necesarios para que la interfaz gráfica realice todas las operaciones posibles sobre la nueva función de similitud. Leer la especificación de cada método de la clase Similarity
2. Crear una instancia de la misma en el constructor de la clase principal del proyecto : MainWindows.java y añadir dicha instancia a la lista similitudFunctions de tipo ArrayList<Similarity>

Listo, ya su función de similitud está perfectamente integrado a la herramienta

2.6 Conclusiones parciales

Se implementaron tres nuevas variantes de agrupamiento de documentos, acopladas a la metodología para el cálculo de la similitud *OverallSimSUX*.

Se implementó una herramienta computacional, capaz de mostrar de forma simultánea el resultado de las tres técnicas de agrupamiento implementadas.

3

EVALUACIÓN DE LAS TÉCNICAS DE AGRUPAMIENTO IMPLEMENTADAS Y DESCRIPCIÓN DE OSSM CLUSTERING A NIVEL DE USUARIO.

3. EVALUACIÓN DE LAS TÉCNICAS DE AGRUPAMIENTO IMPLEMENTADAS Y DESCRIPCIÓN DE OSSM CLUSTERING A NIVEL DE USUARIO.

La evaluación de los resultados de un agrupamiento es una tarea ardua; debido a que "El agrupamiento es un proceso subjetivo; el mismo conjunto de datos usualmente necesita ser agrupado de formas diferentes dependiendo de las aplicaciones" (Jain et al., 1999). En este capítulo se presentan los resultados de los experimentos diseñados para evaluar el modelo de agrupamiento propuesto en esta investigación. Además se realiza una descripción del sistema a nivel de usuario con el propósito de explicar cómo utilizar *OSSM Clsutering* para el agrupamiento de artículos científicos en formato XML, recuperados usando *Lucene*.

3.1 Evaluación de los resultados del modelo de agrupamiento de documentos XML

Para chequear la validez de los resultados obtenidos a partir del modelo de agrupamiento con los nuevos algoritmos, se han diseñado dos experimentos, aplicados a tres casos de estudio; con el propósito de realizar un análisis estadístico, que permita verificar si existen diferencias significativas entre los algoritmos incluidos a la metodología.

3.1.1 Definición de los casos de estudio para la aplicación del modelo de agrupamiento de documentos XML a través de OSSM Clustering

En el CEI-UCLV existe un gran número de artículos científicos y documentos relacionados con diversos temas de investigación, disponibles para la red del Ministerio de Educación Superior (MES). Este repositorio de información es visitado con frecuencia por los investigadores, profesores y estudiantes del centro con el propósito de seleccionar documentos relacionados con algún tema en específico o descubrir conocimiento entre los mismos, cuando comienzan la revisión del estado del arte en un área específica. Teniendo en cuenta estos antecedentes se decide conformar el primer caso de estudio a partir de archivos provenientes del sitio ICT[5], para comprobar las bondades de la nueva metodología a la recuperación de información y extracción de conocimiento que solicitan estos usuarios.

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

El segundo caso de estudio definido constituye una recopilación de documentos del repositorio IDE-Alliance, internacionalmente utilizados para evaluar el agrupamiento. Proporcionados por la Universidad de Granada, España.

Entre los corpus textuales publicados en Internet que se referencian en los artículos para evaluar algoritmos en el área de la minería de textos aplicados a los documentos XML, se destacan los experimentos que utilizan documentos de la colección de la Wikipedia, según se expone en (Denoyer and Gallinari, 2009) y (Campos et al., 2009), los que son publicados cada año por la **IN**iciativa para la **E**valuación de la recuperación de documentos **X**ML (INEX), entre otros. El tercer caso de estudio constituye una selección aleatoria de estos artículos, debido a que la colección contiene documentos clasificados en categorías y éstas a su vez se asocian a temas de diferentes áreas. 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.

En el Anexo 3 se muestra la descripción y la fuente de cada uno de los archivos de datos que conforman los casos de estudio antes mencionados. Todos los conjuntos de datos constan de un rasgo objetivo, por tanto existe la clasificación de referencia para cada uno de ellos, en específico para el primer caso de estudio este rasgo se obtuvo basado en el criterio de expertos. Las colecciones restantes fueron adquiridas con la clasificación de referencia.

3.1.2 Validación del agrupamiento

La validación del agrupamiento se conoce por el procedimiento de evaluar los resultados de algoritmos de agrupamiento (Theodoridis and Koutroubas, 1999, Halkidi et al., 2002). 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 (Höppner et al., 1999). Estas medidas en su mayoría son heurísticas por naturaleza para conservar una complejidad computacional plausible.

Atendiendo a la clasificación de las medidas para la evaluación del agrupamiento de (Höppner et al., 1999, Silberschatz and Tuzhilin, 1996, Kaufman and Rousseeuw, 1990), en esta investigación se seleccionaron medidas externa: *Overall F-measure* y el cálculo de medidas propuestas por INEX: *Purity*, *Micro-Purity* y *Macro-Purity*. En los Anexo 4 y Anexo 5 se puede

observar: un esquema de esta clasificación y cada una de las expresiones correspondientes a las medidas seleccionadas para la evaluación respectivamente.

Las medidas externas fueron seleccionadas para el estudio comparativo que se realiza, debido a que describen la calidad del resultado completo del agrupamiento usando un único valor real y se basan en una estructura previamente especificada que refleja la intuición que se tiene del agrupamiento de los datos (i.e. clasificación de referencia). *Overall F-measure* (Steinbach et al., 2000a) como medida externa, utiliza los criterios de Precisión (Pr) y cubrimiento[6] (Re) (Frakes and Baeza-Yates, 1992), que se calculan para un grupo j y una clase i dados, usando las expresiones $\mathrm{Pr}(i,j)=n_{ij}/n_j$ y $\mathrm{Re}(i,j)=n_{ij}/n_i$, respectivamente; donde n_{ij} es el número de objetos de la clase i en el grupo j, n_j es el número de objetos del grupo j y n_i es el número de objetos de la clase i. La medida-*F* (*F*-measure) se obtiene calculando la media armónica de precisión y cubrimiento. Se puede variar el umbral α ($0 \leq \alpha \leq 1$) para regular la influencia de precisión y cubrimiento en el cálculo de esta medida (Frakes and Baeza-Yates, 1992). Se utiliza α=0.5, para lograr una equidad en la importancia de estos criterios. Finalmente el valor global de la medida-*F* (Overall *F*-measure; OFM), se calcula usando el promedio ponderado de los valores máximos por clase de la medida-*F* sobre todos los grupos (Steinbach et al., 2000a).

La medida OFM fue seleccionada pues logra capturar de forma eficiente la correspondencia entre los resultados del agrupamiento con las clases de tomadas como referencia (Rosell et al., 2004).

Por último, se incluye el cálculo de medidas basadas en *Purity*: *Micro-Purity* y *Macro-Purity*. El criterio *Purity* es utilizado para determinar la calidad de los grupos del agrupamiento, se basa en la idea de maximizar su valor, para lo cual se desea que todos los elementos del grupo pertenezcan a una sola clase. *Purity* es una medida del mayor número de documentos con la misma etiqueta clase en el grupo, respecto al total de documentos. *Micro-Purity* y *Macro-Purity* se calculan para la solución completa del agrupamiento según se muestra en el epígrafe 3.1.3 (Gil-García et al., 2003). En general, en (Pinto et al., 2009) consideran que mayores valores de *Purity*, reportan mejores resultados del agrupamiento.

[6] 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.

3.1.3 **Diseño de los experimentos**

El primer experimento consistió en verificar cómo se comportan globalmente, sobre los tres casos de estudio descritos previamente, las nuevas variantes de agrupamiento incluidas en la metodología.

Para la aplicación de las variantes utilizadas para comparar los resultados del agrupamiento; fue necesario pre-procesar los corpus textuales, asociados a los tres casos de estudio que se muestran en el Anexo 3. Los experimentos realizados en esta investigación incluyeron en la transformación del corpus las operaciones siguientes: convertir todos los caracteres a minúscula, la sustitución de las contracciones por sus expansiones, de las abreviaturas por sus formas completas y la eliminación de números y símbolos y la segmentación por eliminación de afijos, basada en el método heurístico de *Porter* (Porter, 1980). Las formas de pesado se basan en la fórmula TF-IDF. La idea de una expresión TF-IDF es que el peso de los términos refleje la importancia relativa de un término en un documento con respecto a los otros términos en el documento. La reducción de la dimensionalidad del espacio de rasgos haciendo corresponder palabras morfológicamente similares con la palabra raíz asociada (Frakes and Baeza-Yates, 1992, Porter, 1980b) y la selección de aquellos 600 mejores términos, es decir, con calidad superior a determinado umbral considerando esencialmente las expresiones I y II de calidad de términos (Berry, 2004).

Para aplicar los algoritmos se siguió la metodología propuesta anteriormente para el agrupamiento de documentos XML. Lo anterior, incluyó: identificar en cada documento las UE; tratadas como colecciones diferentes. Para cada UE se obtuvo una representación (*Representación I*) basada en la Representación VSM clásica; se obtuvo además una representación global (*Representación II)* que tiene en cuenta el contenido en función de la estructura.

Como entrada los algoritmos tomaron los agrupamientos realizados por cada UE, a partir de la matriz de similitud coseno resultante de la *Representación I* y la matriz de similitud basada en el cálculo de la similitud coseno, a partir de la *Representación II*. Los documentos pertenecientes a cada una de las colecciones que se utilizan en los experimentos están etiquetados y se hace uso de esa clasificación para comparar los resultados del agrupamiento respecto a la clasificación de referencia. Por eso, como criterios para la validación de los resultados de los

algoritmos de agrupamiento que se comparan en el experimento 1 se utilizaron las medidas externas precisión, cubrimiento y OFM.

Para verificar si existen diferencias significativas entre los algoritmos, propuestos en esta investigación y el propuesto en la investigación anterior, se emplearon los resultados obtenidos por la medida de evaluación OFM, estos valores se pueden observar en la gráfica correspondiente a la Figura 3.1.

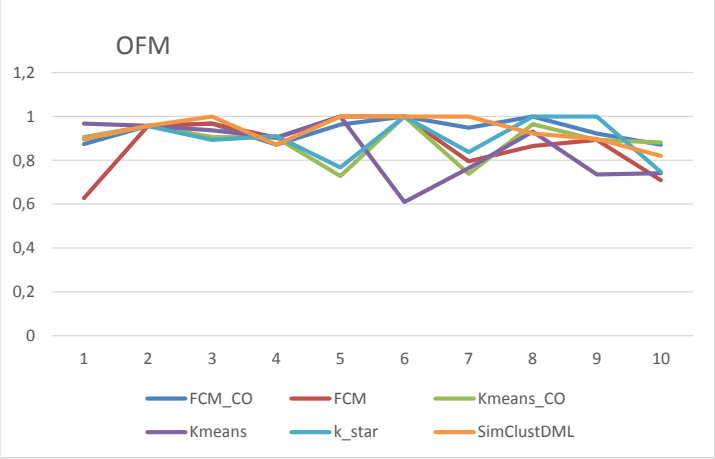


Figura 3.1 Valores de OFM de los algoritmos utilizados en el experimento 1.

Después de haber obtenido los resultados del desempeño de cada algoritmo estudiado, es vital entonces, realizar pruebas estadísticas, que permitan con un fundamento matemático y una descripción confiable; comparar y analizar el nivel de significación y comportamiento de tales algoritmos, con respecto a la medida OFM.

En este sentido, se realizaron para la comparación de los algoritmos aplicando para *k* muestras relacionadas la prueba de *Friedman*.

Para los casos en que se realizó una prueba de *Friedman* se aplicó el método de *Monte Carlo*, con intervalos de confianza del 95% y un número de muestras igual a 10000.

Los resultados de test no paramétrico de *Friedman* se muestran en la Figura 3.2 nos dice que no existen diferencias significativas para la prueba OFM

Estadísticos de contraste[a]

N			10
Chi-cuadrado			3,154
gl			5
Sig. asintót.			,676
Sig. Monte Carlo	Sig.		,685
	Intervalo de confianza de 95%	Límite inferior	,676
		Límite superior	,694

a. Prueba de Friedman

Figura 3.2 Resultados del test no paramétrico de *Friedman.*

En el segundo experimento se busca verificar cómo se comportan globalmente, sobre los tres casos de estudio descritos previamente, los algoritmos incluidos en la metodología, a través de un estudio comparativo, basado en las medidas *Micro-Purity* y *Macro-Purity* que utilizan en (Vries et al., 2011) para mostrar la calidad de los grupos obtenidos en cada agrupamiento.

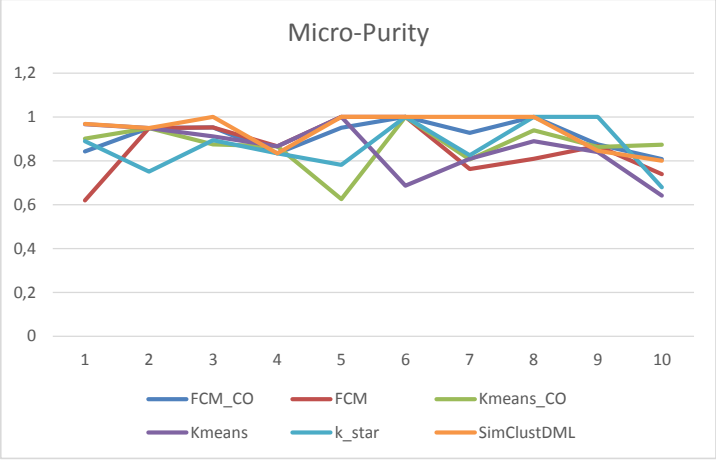


Figura 3.3 Valores de Micro-Purity de los algoritmos utilizados en el experimento 2.

A partir de los valores de Micro-Purity se realizó el test de Friedman bajo los mismos parámetros que para la prueba anterior, cuyos resultados se muestran en la Figura 3.4

Estadísticos de contraste[a]

N			10
Chi-cuadrado			5,205
gl			5
Sig. asintót.			,391
Sig. Monte Carlo	Sig.		,395
	Intervalo de confianza de 95%	Límite inferior	,385
		Límite superior	,404

a. Prueba de Friedman

Figura 3.4

Se puede apreciar que no existen diferencias significativas para la prueba Micro-Purity. Por tanto pasamos a la prueba Macro-Purity, cuyos resultados para el test de Friedman se pueden apreciar en la Figura 3.6

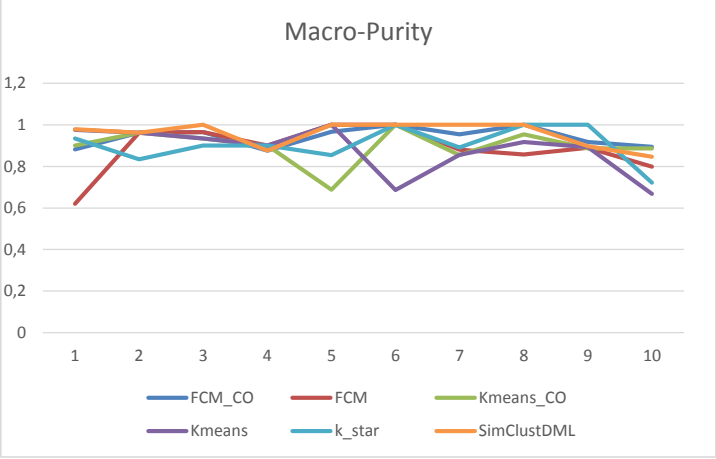


Figura 3.3 Valores de Macro-Purity de los algoritmos utilizados en el experimento 2.

Estadísticos de contraste[a]

N			10
Chi-cuadrado			7,180
gl			5
Sig. asintót.			,208
Sig. Monte Carlo	Sig.		,207
	Intervalo de confianza de 95%	Límite inferior	,199
		Límite superior	,215

a. Prueba de Friedman

Figura 3.6 Resultados del test no paramétrico de *Friedman.*

En esta prueba se obtienen los mismos resultados, no hay diferencias significativas en la prueba Macro-Purity.

3.2 Interfaz de usuarios de OSSM Clustering para la recuperación, indexación y agrupamiento de documentos XML

En este epígrafe se describe cómo utilizar en OSSM Clustering las opciones asociadas al procedimiento general para el agrupamiento de documentos XML.

3.2.1 ¿Cómo indexar colecciones de documentos XML?

OSSM Clustering permite indexar colecciones de documentos XML para su posterior recuperación, durante este proceso el sistema utiliza las facilidades de tika para extraer las unidades estructuradas seleccionadas por el usuario, que se le suministran a *Lucene* para crear los índices. El sistema crear un nuevo índice. La Figura 3.7 muestra la opción para crear un nuevo índice a partir de una colección personal de documentos XML.

Seleccionar en el menú “Index” la opción “doc XML”, esto va a indexar una serie de documentos XML, y va a guardar el índice en el directorio especificado en el diálogo que se muestra en la Figura 3.8.

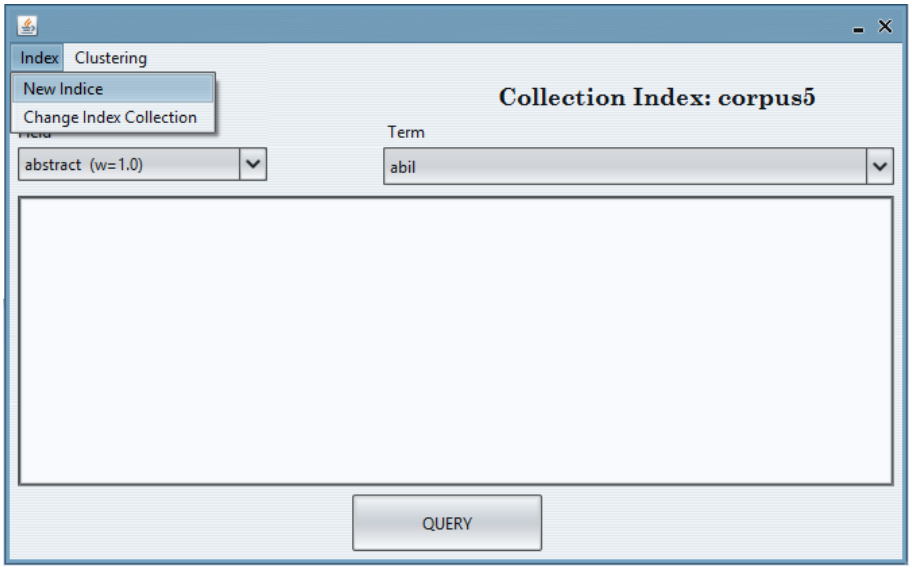


Figura 3.7 Opción para crear un nuevo index.

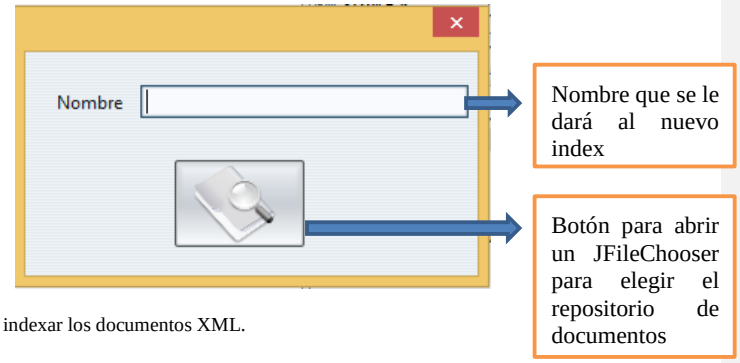


Figura 3.8 Ventana para indexar los documentos XML.

Una vez cargado el directorio OSSM Clustering procederá a leer cada archivo xml y extraer las unidades estructurales de la siguiente manera:

Sea LE una lista de estructuras xml

Para cada archivo f del directorio:

- Si f es xml válido:
    1. Extraer la estructura E de f.
    2. Si E está en la lista LE :
        - se toman las unidades estructurales de acuerdo a como se tomaron para esa estructura de LE y se pasa al siguiente archivo

    sino :

        - se muestra un JTree con la estructura E para que el usuario elija cuáles serán las unidades estructurales de ese documento.
        - Se guarda en E en LE y se continúa al siguiente archivo

A continuación se explica cómo funciona la ventana que muestra la estructura de un documento xml.

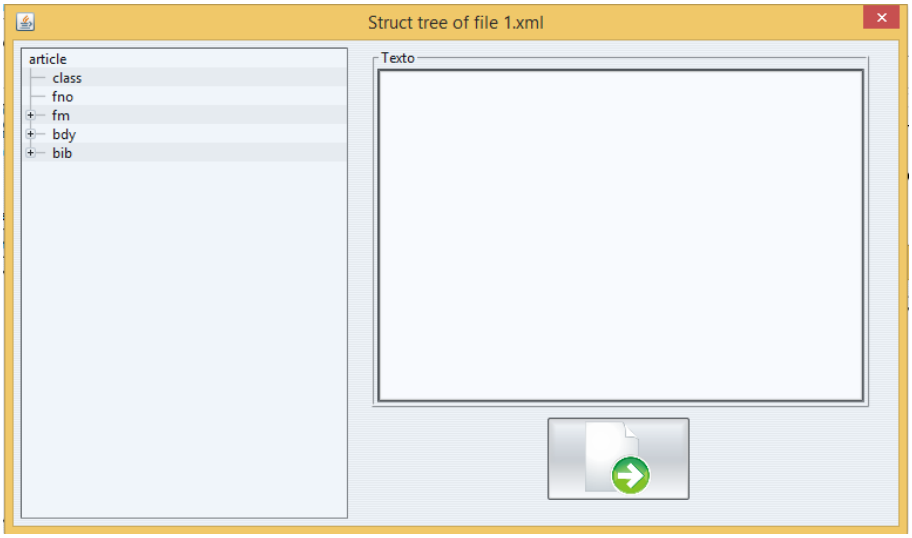


Figura 3.10

Presionando Enter sobre un elemento del documento se muestra un diálogo para cambiar el nombre de la etiqueta en la unidad estructural, y seleccionar si se quiere indexar o no. Todo lo

que se quiera tratar como una unidad estructural debe ser seleccionado para indexar tal y como se muestra en la Figura 3.10

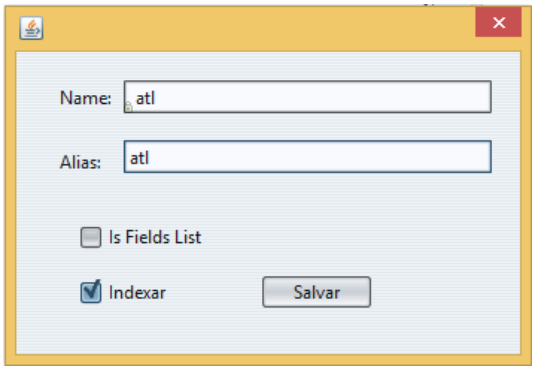


Figura 3.10

En alias se pone el nombre que quiere que tenga la unidad estructural. El check box "is Field List" se utiliza para documentos que tienen una lista de elementos, dentro de una etiqueta, que pertenecen a una misma unidad estructural. Las etiquetas a indexar se marcaran en el árbol con la terminación (i) y los que son listas (l). Presionando la tecla t sobre un elemento en el árbol se muestra en el JTextArea de la derecha el texto de esa etiqueta en el documento.

Una vez indexada una colección no hay necesidad de volver a hacerlo. Solo con cambiar de index como se indica a continuación, la herramienta cargará todos las unidades estructurales de esa colección que se indexaron.

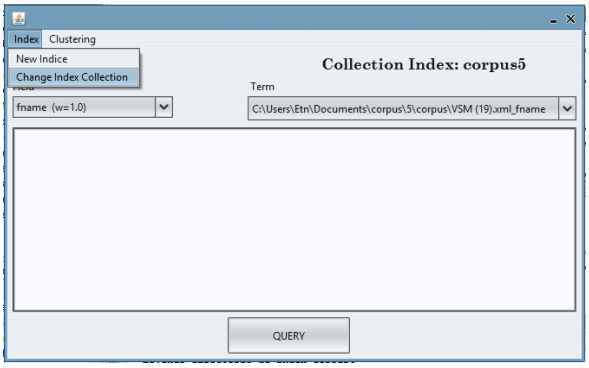


Figura 3.11

Esto nos mostrará una ventana con todos los índices disponibles, y algunos detalles de ellos

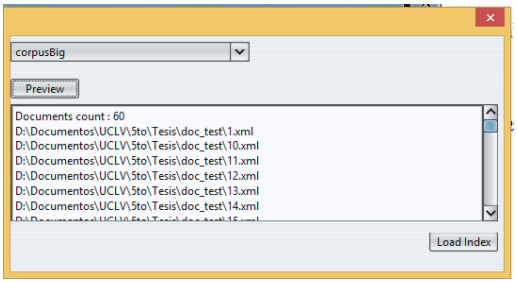


Figura 3.12

3.2.2 ¿Cómo configurar el agrupamiento de documentos XML?

En el menú Clustering-Wizard se accede a la ventana de configuración del agrupamiento, que contiene todos los parámetros que puede variar el usuario antes de comenzar el agrupamiento.

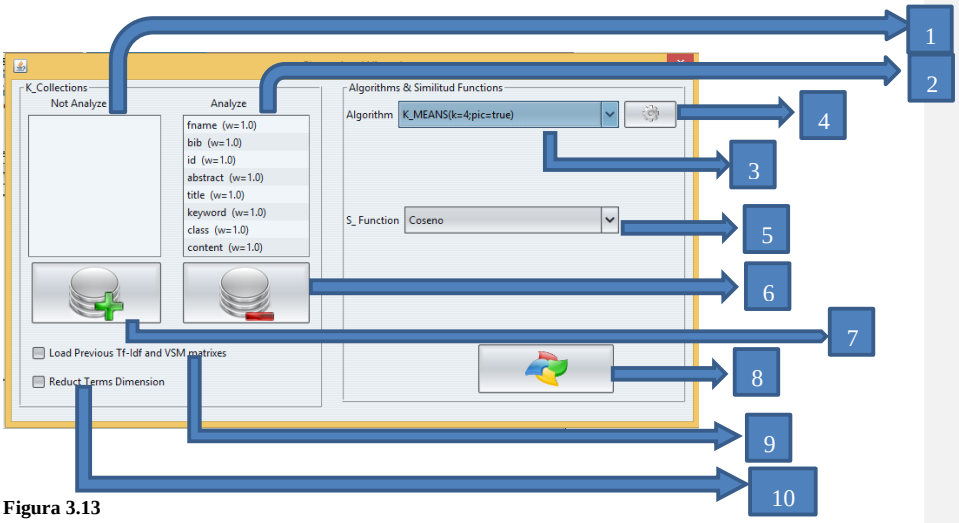


Figura 3.13

1. Es la lista de los campos que se indexaron que no son unidades estructurales. Aquí pueden estar el id del documento y la clase en el caso de documentos pertenecientes a los casos de prueba. Estos no serán analizados para el agrupamiento.
2. Es la lista de los campos que son unidades estructurales, y que serán analizados en el agrupamiento.
3. En este componente están todos los algoritmos cargados por el programa. El método toString de la clase Algorithm permite al programador de cada algoritmo ponerle un nombre personalizado. Se recomienda incluir en el nombre de los atributos los valores de los parámetros.
4. Este botón permite introducir los parámetros deseados al algoritmo seleccionado en el componente anterior.
5. En este componente están todas las funciones de similitud cargadas por el programa. El método toString de la clase Similarity permite al programador de cada función ponerle un nombre personalizado.
6. Este botón elimina el campo seleccionado de la lista de campos a analizar y lo adiciona en la lista de los que no se analizarán.
7. Este botón elimina el campo seleccionado de la lista de campos que no se analizarán y lo adiciona en la lista de los se analizarán.
8. Este botón inicia el proceso de agrupamiento. Este proceso inicia una ventana donde se muestra el proceso en cada una de sus partes

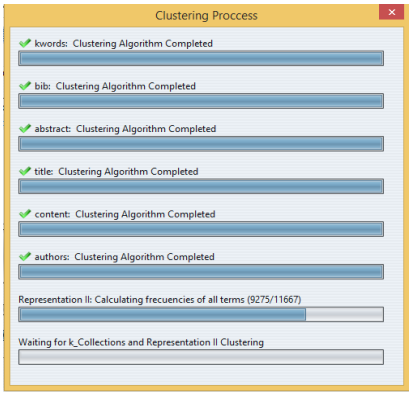


Figura 3.14

9. Si se selecciona, el proceso tomará las matrices tf-idf y de similitud calculadas en anteriores corridas, si existen. Esto deshabilita la opción para escoger la función se similitud, pues el proceso solo agrupará partiendo de las matrices leídas de los ficheros correspondientes.
10. Si está seleccionado, la matriz tf-idf se someterá a un proceso de reducción de dimensionalidad, quedándose con los 600 términos más valiosos con calidad superior a determinado umbral considerando esencialmente las expresiones I y II de calidad de términos (Berry, 2004).

En la siguiente figura se muestra la ventana de configuración del algoritmo K Means, que se muestra al dar click sobre el botón descrito en 8.

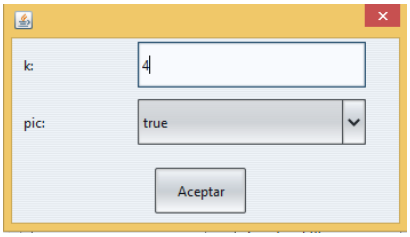


Figura 3.15

Esta ventana muestra a la izquierda los nombres de los atributos, que son obtenidos en la llamada al método String [] Algorithm. getParametersNames(). A la derecha se colocan los componentes necesarios para que el usuario coloque los valores de los parámetros. Se utilizan dos componentes: el JTextField para parámetros de infinitos o muchos valores posibles donde el usuario introduce el valor deseado, y JComboBox para parámetros de pocos valores posibles, donde el usuario elegirá el valor de un conjunto de posibles valores.

Esta implementación es posible gracias al método String[] Algorithm.getPossiblesValues(String parameter) . En el mismo el programador de su algoritmo retorna los valores posibles que puede tomar "parameter" y la interfaz visual los coloca en el JComboBox. Si el parámetro tiene infinitos valores, entonces el método debe devolver null, y se pondrá un JTextField.

3.2.3 ¿Cómo visualizar los resultados obtenidos luego del agrupamiento?

Hay tres formas diferentes de visualizar los resultados:

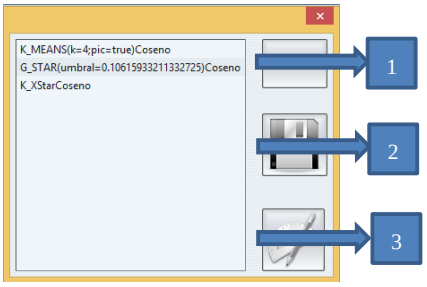


Figura 3.16

1. Un Jtree: Permite visualizar los grupos obtenidos en cada unidad estructural y en el agrupamiento final

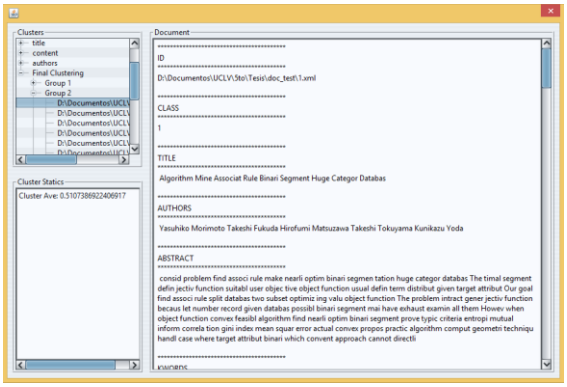


Figura 3.17

2. Fichero: Permite visualizar los grupos, matrices tf-idf y de similitud obtenidos en cada unidad estructural así como en el agrupamiento final en un fichero texto, para que pueda ser analizado por el usuario incluso luego de cerrar la aplicación.
3. Ventana de Validación: Para los documentos que pertenecen a los casos de prueba, muestra los valores de OFM, Micro Purity, Macro-Purity, entre otros, del agrupamiento final.

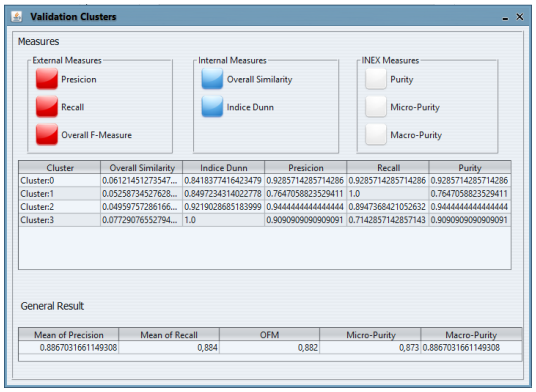


Figura 3.18

3.3 Conclusiones parciales

Utilizando tres casos de estudios se evaluaron los algoritmos seleccionados y no se encontraron diferencias significativas entre las técnicas analizadas con las tres medidas utilizadas (*OFM*, *micro-purity*, *macro-purity*) para evaluar el resultado de los agrupamientos.

La interfaz de usuario que incluye la implementación del procedimiento general es amigable y muestra de forma clara el resultado de los agrupamientos, permitiendo seleccionar cuales Unidades Estructurales tener en cuenta en el agrupamiento. La implementación modular utilizada en OSSM favorece la incorporación de nuevos algoritmos de agrupamiento a la metodología sin cambios perceptibles.

CONCLUSIONES

- Se implementaron dos variantes de algoritmos de agrupamiento por particiones duras y una variante difusa.
- Los algoritmos de agrupamiento documental implementados, permitieron verificar su efecto en la metodología. Obteniéndose como resultado todos los algoritmos se comportaron de forma estable; por lo que el uso de la metodología es independiente del algoritmo seleccionado.
- La implementación del sistema *OSSM Clustering* es totalmente extensible, permitiendo incorporar de forma natural otros algoritmos y funciones de similitud. El sistema es capaz de enfrentarse a cualquier tipo de estructura de XML, siempre que esté bien formado; permitiéndole al usuario escoger qué Unidades Estructurales desea tener en cuenta.

RECOMENDACIONES

Se recomienda:

- Agregar algoritmos del tipo jerárquico a la herramienta.
- Incorporar a la herramienta, los test estadísticos para la comparación de los algoritmos implementados.

REFERENCIAS BIBLIOGRÁFICAS

ABITEBOUL, S. 1997. Querying semi-structured data. *Proceedings of the ICDT Conference, Delphi, Greece*.

ANDERBERG, M. R. 1973. *Clustering Analysis for Applications*, New York: Academic.

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.

ASLAM, J., PELEKHOV, E. & RUS, D. 2004. The star clustering algorithm for static and dynamic information organization. *Journal of Graph Algorithms and Applications,* 8**,** 95.

ASLAM, J. P., K. RUS, D. . Scalable Information Organization. Proceedings of RIAO, 2000.

BATCHELOR, B. 1978. *Pattern Recognition: Ideas in Practice,* New York, Plenum Press.

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

CAMPOS, L. M. D., FERN´ANDEZ-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.

CRISTIANINI, N., SHAWE-TAYLOR, J. & LODHI, H. 2002. Latent semantic kernels. *JJIS'2002,* 18.

CHAWATHE, S. S. Comparing Hierarchical Data in External Memory. In Proceedings of International Conference on Very Large Databases, 1999. 90-101.

CHAWATHE, S. S., RAJARAMAN, A., GARCIA-MOLINA, H. & WIDOM, J. Change Detection in Hierarchically Structured Information. In Proceedings of the ACM International Conference on Management of Data, 1996. 493-504.

CHENG, D., KANNAN, R., VEMPALA, S. & WANG, G. 2006. A divide-and-merge methodology for clustering. *ACM Transaction on Database Systems (TODS),* 31**,** 1499-1525.

CHENG, D., VEMPALA, S., KANNAN, R. & WANG, G. A divide-and-merge methodology for clustering. Proceedings of the 24th ACM SIGMOD SIGACT SIGART Symposium on Principles of Database Systems (PODS 2005), 2005 Baltimore, Maryland. ACM Press, 196-205.

DALAMAGAS, T., CHENG, T., WINKEL, K.-J. & SELLIS, T. 2006. A Methodology for Clustering XML Documents by Structure. *Information Systems.*

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.

DIXON, M. 1997. An overview of document mining technology. http://www.geocities.com/ResearchTriangle/Thinktank/1997/mark/writings/dixm97_dm.ps.

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

DUCH, W. 2002. Similarity-based methods: a general framework for classification. *Control and Cybernetics,* 29**,** 937-968.

FLESCA, S., MANCO, G., MASCIARI, E., PONTIERI, L. & PUGLIESE, A. 2005. Fast detection of XML structural similarities. *IEEE Trans. Knowl. Data Engin.,* 7**,** 160-175.

FRAKES, W. B. & BAEZA-YATES, R. 1992. *Information Retrieval. Data Structure & Algorithms,* New York, Prentice Hall.

FUENTES, I. E. 2013. *Nuevo modelo de agrupamiento para documentos XML utilizando estructura y contenido.* Lic. Ciencia de la Computacion, Universidad Central "Marta Abreu" de Las Villas.

GETOOR, L. & DIEHL, C. P. 2005. Link mining: a survey. *SIGKDD Exploration Newsletter,* 7**,** 3-12.

GIANNOPOULOS, P. & VELTKAMP., R. C. A Pseudo-Metric for Weighted Point Sets. In Proceedings of the 7th European Conference on Computer Vision (ECCV), 2002. 715–730.

GIL-GARCÍA, J.M., B.-C. & PONS-PORRATA, A. Extended Star Clustering Algorithm. Proceedings o f CIARP, 2003.

GUERRINI, G., MESITI, M. & SANZ, I. 2006. An Overview of Similarity Measures for Clustering XML Documents. *Chapter in Athena Vakali and George Pallis.*

GUHA, S., RASTOGI, R. & SHIM, K. CURE: An Efficient Clustering Algorithm for Large Databases. Proceedings of the ACM SIGMOD Conference,, 1998.

GUHA, S., RASTOGI, R. & SHIM, K. ROCK: A Robust Clustering Algorithm for Categorical Attributes. Proceedings of the IEEE Conference on Data Engineering, 1999.

HALKIDI, M., BATISTAKIS, Y. & VAZIRGIANNIS, M. 2002. Clustering validity checking methods: Part II. *ACM SIGMOD Record,* 31**,** 19-27.

HAND, D. J. 1981. *Discrimination and classification,* John Wiley and Sons.

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

HÖPPNER, F., KLAWONN, F., KRUSE, R. & RUNKLER, T. 1999. *Fuzzy cluster analysis: methods for classification, data analysis and image recognition.,* West Sussex, England, John Wiley & Sons Ltd.

JAFAR, O. A. M. & SIVAKUMAR, R. 2013. A Comparative Study of Hard and Fuzzy Data Clustering Algorithms

with Cluster Validity Indices.

JAIN, A. K., MURTY, M. N. & FLYNN, P. J. 1999. Data clustering: a review. *ACM Computing Surveys,* 31**,** 264-323.

KARMARKAR, N. 1984. A new polynomial-time algorithm for linear programming. *Proceedings of the 16th Annual ACM Symposium on the Theory of Computing.*

KAUFMAN, L. & ROUSSEEUW, P. J. 1990. *Finding groups in data: an introduction to cluster analysis*, John Wiley and Sons.

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.

KURGAN, L., SWIERCZ, W. & CIOS, K. J. Semantic mapping of xml tags using inductive machine learning. 11th International Conference on Information and Knowledge Management., 2002 Virginia, USA.

KUTTY, S., TRAN, T., NAYAK, R. & LI, Y. 2008. Combining the structure and content of XML documents for clustering using frequent subtrees. *INEX***,** 391-401.

MARTÍN, C. 2007. *Aprendizaje Automático y Minería de Datos con Modelos Gráficos Probababilísticos.* DEA DEA, Universidad de Granada.

MICHALSKI, R. S., STEPP, R. E. & DIDAY, E. 1981. A recent advance in data analysis: clustering objects into classes characterized by conjuntive concepts. *Progress in Pattern Recognition,* 1**,** 33-56.

NAYAK, R. Investigating Semantic Measures in XML Clustering. Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence, 2006. IEEE.

NIERMAN, A. & JAGADISH, H. V. 2002. Evaluating structural similarity in XML documents. *5th Int. Conf. Computational Science (ICCS'05).*

NIU, Z.-Y., JI, D.-H. & TAN, C.-L. IN: EVANS, D. A. Document clustering based on cluster validation. . Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management (CIKM 2004), 2004 Washington, D.C., USA. ACM. 501-506.

OLMOS, Y. P. & MARTINEZ, J. M. M. 2005. *Estudio de metodos de agrupamiento en el contexto del resumen de corpus textuales.* Licenciado en Ciencia de la Computacion, Universidad Central "Marta Abreu" de Las Villas.

PASSONI, L. 2005. Gestión del conocimiento: una aplicación en departamentos académicos. *Gestión y Política Pública,* XIV**,** 57-74.

PENG, J. & ZHU., J. 2006. Refining spherical k-means for clustering documents.

PEREZ-SUAREZ, A. & MEDINA-PAGOLA, J. E. A clustering algorithm based on generalized stars. Proceedings of the 5th International Conference on Machine Learning and Data Mining in Pattern Recognition, MLDM 2007. LNAI 4571, 248.

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.

PORTER 1980a. An algorithm for suffix stripping. . *Program,* 14**,** 130-137.

PORTER, M. F. 1980b. An algorithm for suffix stripping. *Program,* 14**,** 130-137.

ROSELL, M., KANN, V. & LITTON, J.-E. Comparing comparisons: document clustering evaluation using two manual classifications. *In:* SANGAL, R. & BENDRE, S. M., eds. Proceedings of the International Conference on Natural Language Processing (ICON 2004), 2004 Hyderabad, India. Allied Publishers, 207-216.

RUIZ-SHULCLOPER, J. 1995. *Introducción al reconocimiento de patrones. Enfoque lógico combinatorio*, México, CINVESTAV IPN.

SELKOV, S. M. 1977. The Tree-to-Tree Editing Problem. *Information Processing Letters,* 6**,** 184-186.

SHEN, Y. & WANG, B. Clustering schemaless xml document. 11th international conference on Cooperative Information System., 2003.

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).

SILBERSCHATZ, A. & TUZHILIN, A. 1996. What makes patterns interesting in knowledge discovery systems. *IEEE Transactions on Knowledge and Data Engineering,* 8**,** 940-974.

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

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

STREHL, A., GHOSH, J. & MOONEY, R. Impact of similarity measures on Web-page clustering. Proceedings of the 17th National Conference on Artificial Intelligence (AAAI 2000): Workshop of Artificial Intelligence for Web Search, 2000 Austin, Texas. 58-64.

TAN, A. Text Mining: The state of the art and the challenges. Proceedings of the Conference Knowledge Discovery and Data Mining (PAKDD'99): Workshop Knowledge Discovery from Advanced Databases, 1999 Pacific Asia. 65-70.

THEODORIDIS, S. & KOUTROUBAS, K. 1999. *Pattern Recognition*, Academic Press.

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*.

TRAN, T., KUTTY, S. & NAYAK, R. 2008a. Utilizing the Structure and Data Information for XML Document Clustering. *INEX***,** 402-410.

TRAN, T., NAYAK, R. & BRUZA, P. Combining Structure and Content Similarities for XML Document Clustering. Seventh Australasian Data Mining Conference, 2008b Glenelg, Australia.

VRIES, C. M. D., NAYAK, R., KUTTY, S., GEVA, S. & TAGARELLI, A. 2011. Overview of the INEX 2010 XML mining track : clustering and classification of XML documents. *In Lecture Notes in Computer Science, Springer.* Amsterdam.

WILSON, D. R. & MARTÍNEZ, T. R. 1997. Improved heterogeneous distance functions. *Journal of Artificial Intelligence Research,* 6**,** 1-34.

XIONG, H., WU, J. & CHEN, J. K-means clustering versus validation measures: a data distribution perspective. Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD 2006), 2006 Philadelphia, PA, USA. ACM Press, 779-784.

YANG, W. & CHEN, X. O. 2002. A semi-structured document model for text mining. *Journal of Computer Science and Technology,* 17**,** 603-610.

ZHANG, K. & SHASHA, D. 1989. Simple Fast Algorithms for the Editing Distance Between Trees and Related Problems. . *SIAM Journal of Computing,* 18**,** 1245-1262.

ZHANG, T., RAMAKRISHNMAN, R. & LINVY, M. BIRCH: An Efficient Method for Very Large Databases. ACM SIGMOD, 1996 Montreal, Canada,.

ANEXOS

Anexo 1. Similitudes, distancias más usadas para comparar objetos y medidas de calidad

Sean los objetos O_i y O_j descritos por m rasgos, donde $O_i=(o_{i1}, \ldots, o_{im})$ y $O_j=(o_{j1}, \ldots, o_{jm})$

Distancia Euclidiana

$$D_{Euclideana}(O_i, O_j) = \sqrt{\sum_{k=1}^{m}(o_{ik} - o_{jk})^2} \qquad \text{(A1.1)}$$

Distancia Minkowski (Batchelor, 1978)

$$D_{Minkowski}(O_i, O_j) = \left(\sum_{k=1}^{m}|o_{ik} - o_{jk}|^{\gamma}\right)^{\frac{1}{\gamma}} \quad \text{donde } \gamma \geq 1 \qquad \text{(A1.2)}$$

La distancia Minkowsky es equivalente a la distancia Manhattan o city-block,y a la distancia Euclidiana cuando γ es 1 y 2, respectivamente (Batchelor, 1978). Para los valores de $\gamma \geq 2$, la distancia Minkowsky equivale a Supermum (Hand, 1981).

Distancia Euclidiana heterogénea (HeterogenousEuclidean – OverlapMetric; HEOM)

$$D_{HEOM}(O_i, O_j) = \sqrt{\sum_{k=1}^{m} d_{local}(o_{ik}, o_{jk})^2}, \text{ donde}$$

$$d_{local}(o_{ik}, o_{jk}) = \begin{cases} d_{Overlap}(o_{ik}, o_{jk}) & si\ k\ simbólico \\ d_{NormEuclidean}(o_{ik}, o_{jk}) & si\ k\ numérico \end{cases} \qquad \text{(A1.3)}$$

$$d_{Overlap}(o_{ik}, o_{jk}) = \begin{cases} 0, & si\ o_{ik} = o_{jk} \\ 1, & en\ otro\ caso \end{cases} \quad \text{y} \quad d_{NormEuclidean}(o_{ik}, o_{jk}) = \frac{|o_{ik} - o_{jk}|}{\max_k - \min_k}$$

Distancia Camberra (Michalski et al., 1981)

$$D_{Camberra}(O_i, O_j) = \sum_{k=1}^{m} \frac{|o_{ik} - o_{jk}|}{|o_{ik} + o_{jk}|} \qquad \text{(A1.4)}$$

Correlación de <u>Pearson</u> (Wilson and Martínez, 1997)

$$D_{Pearson}(O_i, O_j) = \frac{\sum_{k=1}^{m} \left(o_{ik} - \overline{atributo_k}\right)\left(o_{jk} - \overline{atributo_k}\right)}{\sqrt{\sum_{k=1}^{m} \left(o_{ik} - \overline{atributo_k}\right)^2 \sum_{k=1}^{m} \left(o_{jk} - \overline{atributo_k}\right)^2}} \quad \text{(A1.5)}$$

donde $\overline{atributo_k}$ es el valor promedio que toma el $atributo_k$ en el conjunto de datos.

Las expresiones de <u>Chebychev</u>, <u>Mahalanobis</u>, distancia de <u>Hamming</u> y la máxima distancia son otras variantes de cálculo de distancias entre objetos (Wilson and Martínez, 1997). En (Duch, 2002) se presentan formas de medir la similitud considerando una relación asimétrica entre los objetos y teniendo en cuenta la probabilidad condicional de un objeto respecto al otro.

A partir de estudios realizados, se ha demostrado que al agrupar documentos, los coeficientes *Dice*, *Jaccard* y *Coseno*, han reportado los mejores resultados (Frakes and Baeza-Yates, 1992). Una valoración del impacto de la distancia *Euclidiana* y los coeficientes *Dice*, *Jaccard* y *Coseno* en dominios textuales se presenta en (Strehl et al., 2000).

Coeficiente <u>Dice</u>

$$S_{Dice}(O_i, O_j) = \frac{2\sum_{k=1}^{m} \left(o_{ik} \cdot o_{jk}\right)}{\sum_{k=1}^{m} o_{ik}^2 + \sum_{k=1}^{m} o_{jk}^2} \quad \text{(A1.6)}$$

Coeficiente de <u>Jaccard</u>

$$S_{Jaccad}(O_i, O_j) = \frac{\sum_{k=1}^{m} \left(o_{ik} \cdot o_{jk}\right)}{\sum_{k=1}^{m} o_{ik}^2 + \sum_{k=1}^{m} o_{jk}^2 - \sum_{k=1}^{m} \left(o_{ik} \cdot o_{jk}\right)} \quad \text{(A1.7)}$$

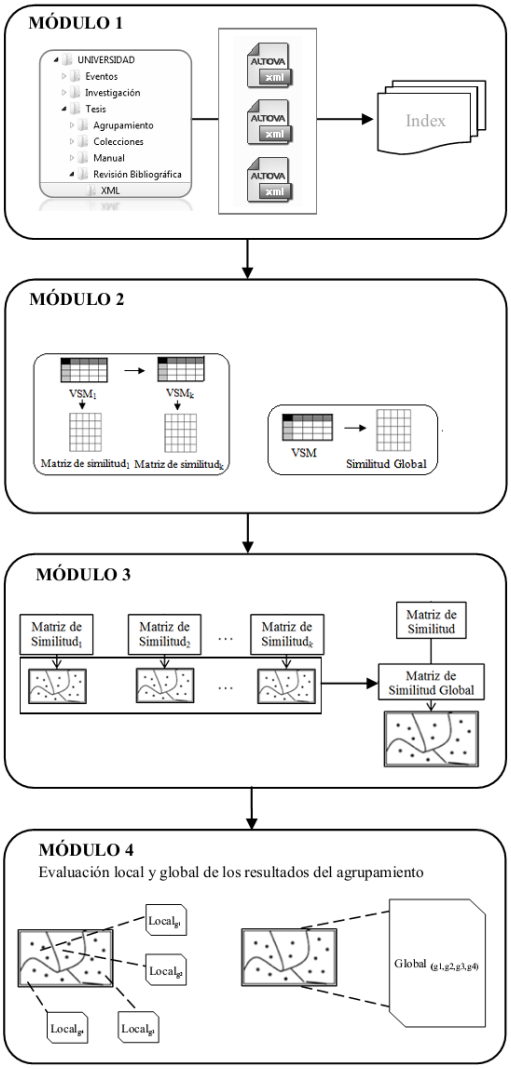
Coeficiente Coseno

$$S_{Coseno}(O_i, O_j) = \frac{\sum_{k=1}^{m}(o_{ik} \cdot o_{jk})}{\sqrt{\sum_{k=1}^{m} o_{ik}^2 \cdot \sum_{k=1}^{m} o_{jk}^2}} \qquad \text{(A1.8)}$$

Calidad de términos. Medir la calidad de los términos según las expresiones q_0 y q_1, la segunda constituye una variante de la primera donde n_1 es el número de documentos en los cuales *t* ocurre al menos una vez (Berry, 2004).

$$q_0(t) = \sum_{j=1}^{n}(tf_{d\,j}(t))^2 - \frac{1}{n}\left[\sum_{j=1}^{n} tf_{d\,j}(t)\right]^2 \quad q_1(t) = \sum_{j=1}^{n_1}(tf_{d\,j}(t))^2 - \frac{1}{n_1}\left[\sum_{j=1}^{n_1} tf_{d\,j}(t)\right]^2 \qquad \text{(A1.9)}$$

Anexo 2. Modelo general para el agrupamiento de documentos XML

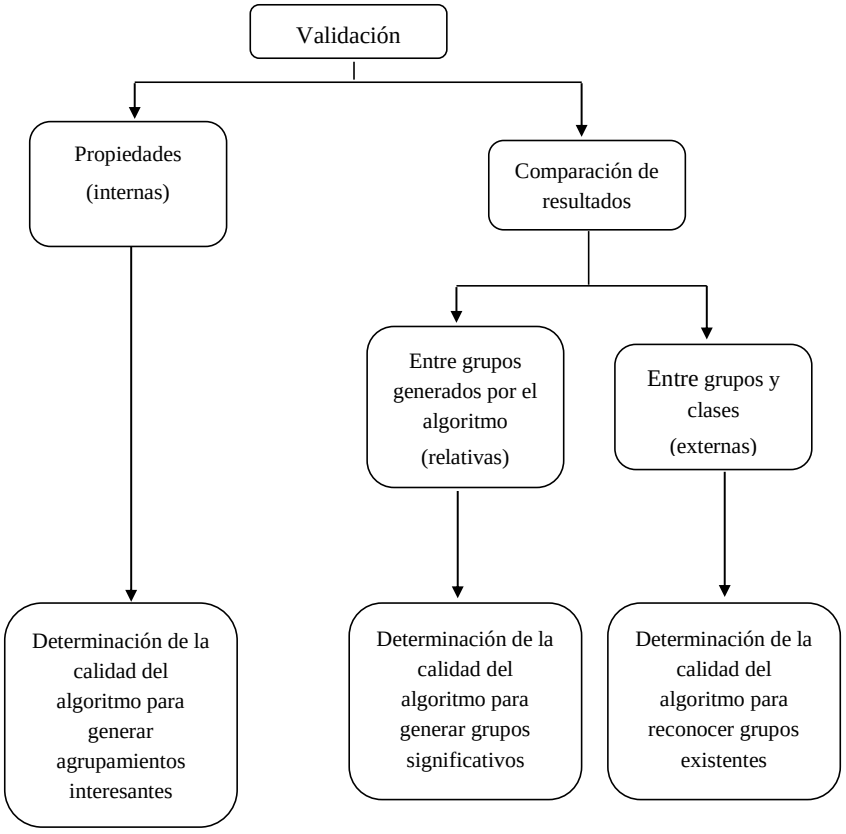


Anexo 3. Descripción de los archivos utilizados para evaluar la calidad del agrupamiento.

No. Corpus	Cantidad de documentos	Cantidad de clases	Valores ausentes
Conjuntos de documentos XML confeccionados a partir de documentos recuperados del sitio de ICT del Centro de Estudios de Informática de la Universidad Central "Marta Abreu" de Las Villas http://ict.cei.uclv.edu.cu			
5	34	2	Fuzzy Logic, SVM
6	30	2	Association Rules, SVM
7	30	2	Fuzzy Logic, Rough Set
8	49	3	Association Rules, SVM, Fuzzy Logic
9	49	3	Rough Set, SVM, Fuzzy Logic
10	64	4	Rough Set, Association Rules, SVM, Fuzzy Logic
11	45	3	Rough Set, Association Rules, Fuzzy Logic
12	34	2	Association Rules, SVM
13	35	2	Rough Set, SVM
14	49	3	Rough Set, Association Rules, SVM
15	30	2	Rough Set, Association Rules
Recopilación de documentos del repositorio IDE-Alliance, internacionalmente utilizados para evaluar agrupamiento. Proporcionados por la Universidad de Granada, España.			
1	21	2	Belief Propagation, CL
2	23	2	Belief Propagation, Copula
3	33	3	Copula, Belief Propagation, CL
4	22	2	CL, Copula
Recopilación de documentos del repositorio Wikipedia[7], internacionalmente utilizados para evaluar agrupamiento, a través de INEX.			
16	17	2	Politics, Party

7 http://www.inex.otago.ac.nz

Anexo 4. Clasificación simplificada de algunas técnicas para la validación de agrupamientos[8]



[8] Tomado de BRUN, M., SIMA, C., HUA, J., LOWEY, J., CARROLL, B., SUH, E. & DOUGHERTY, E. R. 2007. Model-based evaluation of clustering validation measures. Pattern Recognition, 40, 807-824.

Anexo 5. Algunas medidas externas para la validación del agrupamiento

Medidas externas

Medida-*F* Global (Overall *F*-Measure; OFM) (Steinbach et al., 2000b)

$$Overall\ F-Measure = \sum_{i=1}^{k} \frac{n_i}{n} \max\{F-Measure(i,j)\} \qquad \text{(A5.1)}$$

donde k es el número de clases, n_i es el tamaño de la clase i, n es el número total de objetos agrupados y *F-Measure*(i, j) se calcula según la expresión siguiente. Si α= 1, entonces OFM se nombra Purity (Rosell et al., 2004)

Medida-*F*(*F*-Measure) de la clase *i* respecto al grupo *j*

$$F-Measure(i,j) = \frac{1}{\alpha(1/\Pr(i,j)) + (1-\alpha)(1/\mathrm{Re}(i,j))} \qquad \text{(A5.2)}$$

Si α= 1 entonces *F-Measure*(i,j)coincide con precision, si α= 0 entonces *F-Measure*(i,j)coincide con cubrimiento. α= 0.5 significa igual peso para precisión y cubrimiento.

Micro-averaged precision y micro-averaged recall (NIU, 2004)

$$\text{MA - Pr} = \frac{\sum_{i=1}^{k} \alpha_i}{\sum_{i=1}^{k} (\alpha_i + \beta_i)} \quad \text{y} \quad \text{MA - Re} = \frac{\sum_{i=1}^{k} \alpha_i}{\sum_{i=1}^{k} (\alpha_i + \gamma_i)} \qquad \text{(A5.3)}$$

donde α_i es el número de objetos correctamente asignados a la clase i, β_i es el número de objetos incorrectamente asignados a la clase i y γ_i es el número de objetos incorrectamente no asignados a la clase i. MA-Pr = MA-Re si cada objeto pertenece a sólo un grupo y la clasificación de referencia también tiene una clasificación única para cada objeto.

Medidas propuestas por INEX

$$Purity(k) = \frac{NDMLC_k}{NDC_k} \qquad \text{(A5.4)}$$

$$Micro - Purity(k) = \frac{\sum_{k=0}^{n} Purity(k) * TotalFoundByClass(k)}{\sum_{k=0}^{n} TotalFoundByClass(k)} \quad \text{(A5.5)}$$

$$Macro - Purity(k) = \frac{\sum_{k=0}^{n} Purity(k)}{TotalofCategories} \quad \text{(A5.6)}$$

Donde: se asume el total de categorías como la cantidad de grupos encontrados.