Revista Cubana de Ciencias Informáticas
Vol. 11, No. 2, Abril-Junio, 2017
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 46-60

Tipo de artículo: Artículo original
Temática: Inteligencia artificial
Recibido: 07/02/2017 | Aceptado: 25/04/2017

SemClustDML: algoritmo para agrupar artículos científicos basado en la información brindada por las referencias bibliográficas

SemClustDML: algorithm to clustering scientific papers based on information provided by bibliographic references

Lisvandy Amador [1*], María M. García [2], Daniel Gálvez Lío [2,3], Damny Magdaleno [2,3]

[1] Instituto de Biotecnología de las Plantas, Carretera a Camajuaní Km 5 ½ Santa Clara, Villa Clara, Cuba. C.P: 54830 lisvandy@ibp.co.cu
[2] Departamento de Computación, Universidad Central "Marta Abreu de Las Villas", Carretera a Camajuaní Km 5 ½ Santa Clara, Villa Clara, Cuba. C.P: 54830. {mmgarcia, dgalvez, dmg}@uclv.edu.cu
[3] Universidad Metropolitana del Ecuador (UMET), La Coruña N26-95 y San Ignacio, Quito, Ecuador. {d.galvez, dmagdaleno}@umet.edu.ec

[*] Autor para correspondencia: lisvandy@ibp.co.cu

Resumen

El agrupamiento de datos se ha convertido en una de las formas fundamentales de gestión del conocimiento. Particularmente gestionar el conocimiento a partir de la bibliografía científica disponible en internet resulta de gran importancia para los investigadores, es por ello que se han desarrollado técnicas especializadas en el agrupamiento de artículos científicos. Las publicaciones científicas siguen una estructura bien definida donde hay partes fundamentales que siempre están presente como: título, resumen, palabras claves y referencias bibliográficas. Específicamente, las referencias bibliográficas brindan información relevante en el momento de determinar si dos artículos dados tratan temas similares. Por lo cual, potenciar la información brindada por esta subunidad influye de manera significativa en el resultado del agrupamiento. Este trabajo tuvo como objetivo: desarrollar un algoritmo de agrupamiento que haga uso de las características especiales de la matriz de similitud obtenida con la función SimRefBib para mejorar los resultados del agrupamiento de artículos científicos basado en las referencias bibliográficas. Las pruebas realizadas demuestran que el algoritmo propuesto logra mejorar de manera significativa los resultados del agrupamiento de artículos científicos cuando este está basado únicamente en la información brindada por las referencias bibliográficas.

Palabras clave: agrupamiento de literatura científica, algoritmos de agrupamientos, gestión del conocimiento

Revista Cubana de Ciencias Informáticas
Vol. 11, No. 2, Abril-Junio, 2017
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 46-60

Abstract

Data clustering has become one of the key forms of knowledge management. Particularly knowledge management from the scientific literature available on the internet is very importance for researchers, that why, specialized techniques have been developed in scientific articles clustering. The scientific publications follow a well-defined structure where there are fundamental parts that are always present as: title, abstract, keywords and bibliographical references. Specifically, the bibliographical references provide relevant information when determining whether two articles address similar topics. Therefore, to enhance the information provided by this subunit has a significant influence on the clustering´s result. The objective of this work was to develop a clustering algorithm that makes use of the special characteristics of the similarity matrix obtained with the SimRefBib function to improve the results of scientific articles clustering based on bibliographic references. The tests show that the proposed algorithm improves significantly the results of the grouping of scientific articles when it is based only on the information provided by the bibliographic references.

Keywords: *Scientific Papers' Clustering, Clustering's algorithms, knowledge management*

Introducción

Los volúmenes de información disponibles a nivel mundial crecen a diario y las colecciones de datos se vuelven cada vez más heterogéneas, grandes, diversas y dinámicas (Magdaleno Guevara et al., 2016); por lo que es más complejo para los usuarios identificar la información relevante (Aljaber et al., 2010).

Uno de los principales métodos usados para la gestión del conocimiento es el agrupamiento (Qian and Zhang, 2003). El problema del agrupamiento consiste en encontrar grupos de objetos similares en un conjunto de datos, donde la semejanza entre un par de objetos se calcula usando una función de similitud (Aggarwal and Zhai, 2012).

Existe una gran cantidad de algoritmos de agrupamiento en la literatura (Bezdek et al., 1984; Guha et al., 1998; Pinto et al., 2010; Rajeshwari et al., 2015; Sert et al., 2015). Según (Magdaleno Guevara et al., 2015), estos se pueden clasificar siguiendo diversos criterios, como pueden ser: tipo de los datos de entrada, criterios para definir la similitud entre los objetos, conceptos en los cuales se basa el análisis y forma de representación de los datos. Si la participación del usuario influye en el agrupamiento, se tienen otras dos clasificaciones: algoritmos de agrupamiento automático y algoritmos de agrupamiento semiautomático. A continuación, se mencionan algunos de los algoritmos de agrupamiento existentes.

En (Pinto, et al., 2010) se presenta una variante del algoritmo de agrupamiento *K-Star*. Este, como la mayoría de estos algoritmos, requiere como parámetro una matriz de similitud que recoja el grado de semejanza entre cada par de objetos de la colección. La principal ventaja de este algoritmo radica en su capacidad durante el proceso iterativo, de descubrir automáticamente la cantidad de grupos que se deben formar.

En (Gil-García and Pons-Porrata, 2010) los autores desarrollan un algoritmo jerárquico dinámico para el agrupamiento de documentos. Este algoritmo representa la colección a agrupar mediante un grafo de $ß_0$-semejanza, donde cada vértice representa un grupo, por lo cual se parte de un grafo de *n* vértices, donde *n* es la cantidad de objetos a agrupar. Dos vértices estarán conectados únicamente si su semejanza supera un umbral definido. Luego se aplican sucesivas transformaciones al grafo a través de un algoritmo de cubrimiento hasta que se obtiene un grafo de $ß_0$-semejanza que sea completamente inconexo. Según (Domínguez et al., 2014), este algoritmo obtiene buenos resultados, pero en colecciones con un elevado número de objetos consume gran cantidad de memoria, lo que reduce la cantidad máxima de objetos a agrupar.

Específicamente, el agrupamiento de artículos científicos se torna una tarea de suma importancia; ya que es necesario dotar a los investigadores de herramientas capaces de agilizar el proceso de identificación de la información relevante y de esta manera puedan hacer un uso más eficiente del tiempo que disponen.

En (Magdaleno, 2015) se presenta una nueva metodología de agrupamiento de artículos científicos en formato semiestructurado. Esta metodología hace uso tanto de la estructura como del contenido del documento para lograr mejores resultados en el agrupamiento. La información brindada por las referencias bibliográficas es considerada muy relevante a la hora de determinar qué tan similares pueden ser dos artículos científicos, es por ello que el autor hace particular hincapié en esta unidad estructural y desarrolla una función de similitud que se adapte a las características de la misma. La matriz de similitud que se obtiene con la función propuesta (función *SimRefBib*) tiene ciertas características que la diferencian de matrices que pueden ser obtenidas con otras funciones, por ejemplo, *Dice* (Vargas Flores, 2016), *Jaccard* o *Coseno* (Lin et al., 2014). Es que los coeficientes de similitud para dos documentos supuestamente similares generalmente son bajos, pero para documentos que se suponen no son similares casi siempre el valor de similitud obtenido es cero o muy cercano a cero.

En general los algoritmos de agrupamiento asumen que documentos considerados similares presenten valores de similitud altos y los que no lo son presenten valores bajos; pero en muy pocos casos el valor de similitud es cero, dado que al analizar todo el documento existen elevadas probabilidades de que se encuentren términos comunes para determinados pares de documentos, a pesar de que ellos no traten un mismo tema. Al aplicar algunos de estos algoritmos

usando como entrada la matriz de similitud obtenida con la función *SimRefBib*, no se garantiza obtener siempre buenos resultados en el agrupamiento, debido en gran medida, por la forma en que internamente cada uno obtiene los grupos. Esto no significa que la función *SimRefBib* no sea capaz de discernir de manera correcta entre los elementos que deben pertenecer a cada grupo, porque el hecho que se obtenga valor de similitud cero para los documentos que deben pertenecer a grupos diferentes, garantiza que el diseño de un algoritmo que se adapte a estas características especiales favorecerá considerablemente el resultado del agrupamiento de artículos científicos. Es por ello que se propone como objetivo de este trabajo: Desarrollar un algoritmo de agrupamiento que haga uso de las características especiales de la matriz de similitud obtenida con la función *SimRefBib* para mejorar los resultados del agrupamiento de artículos científicos basado en las referencias bibliográficas.

Materiales y Métodos

Un agrupamiento eficaz debe tener en cuenta las preferencias y necesidades individuales para apoyar la personalización en el momento de la categorización (Wei et al., 2006). Para obtener los grupos (en este trabajo se usa indistintamente los términos grupo o clúster), el algoritmo propuesto se vale de algunos parámetros que facilitan al usuario lograr un agrupamiento que se ajuste a sus necesidades específicas. Estos parámetros son: umbral de similitud y longitud mínima de cada clúster.

El umbral de similitud, define el valor a tomarse en cuenta para considerar elementos que pertenecen a un mismo grupo. Se determina en un dominio entre 0 y 1 donde valores cercanos a 1 favorecen grupos más homogéneos al considerar niveles de similitud más altos. Así, por ejemplo, si se define 0.9 como umbral se obtendrá un número mayor de grupos, pero con elementos más similares entre sí, en cambio, si se define un umbral de 0.1 se obtendrán menos grupos, pero con un mayor índice de dispersión de los elementos dentro del grupo. De este modo el usuario puede variar el valor del umbral, dependiendo de qué tan compacto desea que sean los grupos. El segundo parámetro se aplicará dependiendo del agrupamiento generado por el umbral de similitud y se refiere a un segundo nivel de agrupamiento. La longitud mínima del clúster presupone reagrupar aquellos grupos conformados por un número de documentos menor que este parámetro. De no ser proporcionado por el usuario este segundo nivel de agrupamiento no tiene lugar.

Además, el algoritmo recibe como entrada la matriz de similitud que recoge el valor de semejanza que tiene cada par de objetos de la colección que se desea agrupar, específicamente, la obtenida con la función *SimRefBib* ya que el algoritmo está diseñado para explotar eficientemente las características especiales de esta matriz. Para matrices obtenidas con otras funciones de similitud no se garantizan buenos resultados en el agrupamiento.

Revista Cubana de Ciencias Informáticas
Vol. 11, No. 2, Abril-Junio, 2017
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 46-60

1.1 Algoritmo de agrupamiento SemClustDML

La idea general del algoritmo *SemClustDML*, es formar grupos preliminares sin hacer hincapié particular en la forma de seleccionar los centroides. El uso de la matriz obtenida con la función *SimRefBib* posibilita seleccionar como centroides el subgrupo máximo de documentos que no superan el valor $\gamma/2$, siendo γ el umbral de similitud que se define. Posteriormente se agrega el resto de los documentos a cada uno de estos centroides, con los cuales se supera el umbral de similitud. Dado que los documentos que deben pertenecer a grupos diferentes presentan similitud cero en la mayoría de los casos, no es necesario seleccionar como centroide el elemento más representativo del grupo, sino que cualquier documento del grupo puede en primera instancia ser considerado como centroide. Luego se aplican sucesivas transformaciones a estos grupos inicialmente formados y se obtienen los grupos finales. En la Figura 1 se formaliza el algoritmo.

Algoritmo 1. Algoritmo de agrupamiento *SemClustDML*
Entrada: Matriz de similitud *matriz*, Conjunto de n Objetos $(O = \{o_1, o_2, \dots, o_n\})$, umbral de similitud γ, longitud mínima de cada clúster l, cantidad de elementos aleatorios a seleccionar para comprobar si los clústers son agrupables v.
Salida: Lista de clústers formados (C)
Inicio:
1. Búsqueda de los centroides iniciales: $C = \{o_1, o_2, \dots, o_k\}$, $k \leq n$, donde cada o_i se considera un nuevo clúster c.
2. Asignación de cada objeto $o_i \notin C$ al c_j correspondiente.
3. Si $cup \leftarrow \bigcup_{g,h=1}^{k} cap(c_g, c_h) = \emptyset$, donde $cap = c_g \cap c_h$, entonces ir al paso 5.
4. Determinar para cada $o_i \in cap(c_g, c_h) \neq \emptyset$, el c_j correspondiente, donde $sim \leftarrow \frac{\sum_{r=1}^{m_j} matriz(o_i, c_{jr})}{m_j}$ es máxima, m_j cantidad de elementos en c_j. Ir paso 3.
5. $\forall o_i \notin C, c_j \leftarrow (c_j \cup o_i)$, donde $sim \leftarrow \frac{\sum_{r=1}^{m_j} matriz(o_i, c_{jr})}{m_j}$ es máxima.
Fin

Figura 1. Algoritmo de agrupamiento *SemClustDML*

1.1.1 Búsqueda de los centroides iniciales

Los centroides iniciales van a ser aquellos elementos a partir de los cuales se van a formar los grupos preliminares. Dado que dos elementos que tengan similitud menor que $\gamma/2$ difícilmente pertenecerán a un mismo clúster; el proceso de selección de los centroides iniciales se torna relativamente fácil y se convierte en la búsqueda de un grupo de elementos que tengan similitud menor que $\gamma/2$ tomados dos a dos. Para ello se añade a la lista de centroides el primer elemento de la colección, luego se compara cada uno de los siguientes elementos con los que ya forman parte de la lista de centroides, si este elemento no tiene similitud mayor que $\gamma/2$ con ninguno de los elementos que ya pertenecen a los

Editorial "Ediciones Futuro"
Universidad de las Ciencias Informáticas. La Habana, Cuba
rcci@uci.cu

Revista Cubana de Ciencias Informáticas
Vol. 11, No. 2, Abril-Junio, 2017
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 46-60

centroides, este elemento también pasa a formar parte de la lista. Si no se encuentra al menos un par de elementos cuya similitud sea menor que $\gamma/2$ el algoritmo devolverá un solo clúster formado por el conjunto de documentos de la colección. Es evidente que el orden en que se presenten los documentos al algoritmo, influye a la hora de determinar los documentos centroides. Sin embargo, los pasos posteriores del algoritmo garantizan que la efectividad del mismo no se vea afectada por la forma de seleccionar los centroides. La determinación de los centroides tiene una complejidad computacional de O$(nlogn)$.

1.1.2 Asignación de los elementos a los clústeres

La asignación de cada elemento que no fue seleccionado como centroide a cada uno de los clústeres es sencilla. Un elemento *i* pertenecerá a un clúster *C* si:

$$matriz(i,C_{centroide}) \geq \gamma \quad (1.1)$$

Siendo $C_{centroide}$ el centroide del clúster *C* y *matriz* la matriz de similitud obtenida con la función *SimRefBib*.

En este paso del algoritmo cada documento va a ser unido a todos aquellos centroides con los cuales supere el umbral de similitud. La asignación de los elementos a los grupos tiene una complejidad para el peor de los casos de O(n^2).

1.1.3 Grupos solapados

Definición 1 (Grupos solapados): Dos grupos C_i, C_j se dicen son solapados si $C_i \cap C_j \neq \emptyset$.

Pudiera ocurrir que existan elementos que superen el umbral de similitud con más de un centroide, es por ello que se hace necesario calcular para cada par de clúster solapados, la pertenencia a cada uno de estos clústeres de los elementos que se encuentran en la intersección. La α-pertenencia de un elemento *i* a un clúster C_j se define mediante la ecuación 1.2. En caso de que un elemento tenga el mismo valor de α-pertenencia para dos clústeres, el elemento será unido al primero de los clústeres.

$$\alpha(i,C_j) = \frac{1}{n_j} \sum_{k=1}^{n_j} \text{matriz}(i, C_{jk}) \quad (1.2)$$

En la ecuación anterior n_j indica la cantidad de elementos del clúster C_j.

Determinar si existen grupos que se interceptan tiene una complejidad O(n^2) para el peor de los casos y asignar los elementos que pertenecen a la intercepción al grupo al cual tengan mayor pertenencia total tiene una complejidad O(n^3) para el peor de los casos. Por lo cual la complejidad de este paso es O(n^3).

Editorial "Ediciones Futuro"
Universidad de las Ciencias Informáticas. La Habana, Cuba
rcci@uci.cu

Revista Cubana de Ciencias Informáticas
Vol. 11, No. 2, Abril-Junio, 2017
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 46-60

1.1.4 Elementos aislados

Al calcular los centroides iniciales y asignar cada uno de los elementos restantes a estos centroides se tendrán algunos elementos que no superen el umbral con ninguno de los centroides, por lo cual no serán unidos a ningún grupo, estos son los llamados elementos aislados.

Una vez formados los grupos se calcula la α-pertenencia de cada uno de estos elementos a cada uno de los clústeres, el elemento será añadido al clúster para el cual se obtenga el mayor valor de α-pertenencia. En caso de empate al calcular la α-pertenencia el documento será unido al primero de los clústeres en orden. Asignar cada elemento aislado al grupo al cual tiene mayor pertenencia total tiene complejidad para el peor de los casos de O(n^2).

1.1.5 Refinamiento del resultado del Agrupamiento

Una vez obtenidos los grupos, puede ser deseable para el usuario realizar un proceso de refinamiento de estos grupos, que le permita obtener un conjunto de grupos que se acerque más a sus necesidades de información, facilitando de esta forma la gestión del conocimiento. En la Figura 2 se muestran los tres pasos con que consta el proceso de refinamiento y en los siguientes subepígrafes se explican detalladamente estos pasos.

1. Para cada c_s formado seleccionar (si es posible) 2 nuevos centroides aplicar los pasos del 2 al 5 del algoritmo *SemClustDML*.
2. $\forall o_i \in c_s$ y longitud de c_s menor que l; $c_j \leftarrow (c_j \cup o_i)$, donde $sim \leftarrow \frac{\sum_{r=1}^{m} matriz(o_i, c_{jr})}{m}$ es máxima.
3. $\forall C_i, C_j \; con \; i,j = \overrightarrow{1..k}$, k cantidad de clúster, verificar si C_i, C_j son agrupables.

Figura 2. Pasos para refinar el resultado del agrupamiento

1.1.5.1 División de clúster

La necesidad de aplicar la división de los clústeres parte del problema que un objeto puede estar relacionado con objetos de dos o más grupos diferentes. Si en el proceso de selección de centroides este objeto resulta escogido, al aplicar la unión de elementos a los clústeres todos los elementos que estén relacionados con el objeto seleccionado formarán parte de un mismo clúster, lo cual no es un resultado deseado para el agrupamiento.

El proceso de división se aplica a cada clúster y consiste en buscar dos nuevos centroides en el grupo (de la misma manera que se seleccionan los centroides en el algoritmo original) y formar dos nuevos grupos con cada uno de estos centroides; si los clústeres formados no son agrupables, los dos nuevos grupos pasan a formar parte de los clústeres y el clúster original se elimina. La complejidad computacional de este paso se explica detalladamente en el subepígrafe 1.2.

Editorial "Ediciones Futuro"
Universidad de las Ciencias Informáticas. La Habana, Cuba
rcci@uci.cu

Revista Cubana de Ciencias Informáticas
Vol. 11, No. 2, Abril-Junio, 2017
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 46-60

1.1.5.2 Tamaño del clúster

Es posible que al usuario solo le interese obtener clústeres de tamaño mayor que *l* (*l* proporcionado por el usuario). Por tanto, en este paso del refinamiento de los grupos se seleccionarán aquellos que su tamaño sea menor que *l* y se calculará la α-pertenencia de cada uno de los elementos de estos grupos a los restantes clústeres. Cada elemento será unido al clúster con el cual tenga mayor α-pertenencia. Suponiendo que se obtienen *k* grupos con *n/k* elementos cada uno, la complejidad de reinsertar los elementos de aquellos clústeres de tamaño menor que *l*, en el clúster con respeto al cual tiene mayor α-pertenencia es $O\left(\frac{n^2}{k} log k\right)$.

1.1.5.3 Clústeres agrupables

Dos clústeres son agrupables si más de la mitad de los elementos del clúster de menor tamaño pueden formar parte del clúster de mayor tamaño. Un elemento puede ser cambiado de clúster si: supera el umbral de similitud con más de la mitad de los elementos del otro clúster o la α-pertenencia al clúster al que será cambiado el elemento supera el umbral definido. En la definición 2 se formaliza este planteamiento.

Definición 2 (Clúster agrupables): Dados los clústeres $C_i \; y \; C_j, C_i \leq C_j$ se dice que estos son agrupables si $CA(C_i, C_j) \geq 0{,}5$ donde:

$$CA\left(C_i, C_j\right) = \frac{1}{m_i} \sum_{k=0}^{m_i} EAC\left(C_{ik}, C_j\right) \tag{1.3}$$

En la ecuación anterior *m* indica la cantidad de elementos del clúster C_i y $EAC(C_{ik}, C_j)$ se define como:

$$EAC\left(i, C_j\right) = \begin{cases} 1 & si \; \alpha\left(i, C_j\right) \geq \gamma \;\; \text{o} \;\; \beta\left(i, C_j\right) \geq 0{,}5 \\ 0 & e.o.c \end{cases} \tag{1.4}$$

Donde γ es el umbral de similitud definido, y β es la β-pertenencia del elemento *i* al clúster C_j la cual se define como:

$$\beta\left(i, C_j\right) = \frac{1}{n_j} \sum_{k=1}^{n_j} ElemtSim\left(i, C_{jk}\right) \tag{1.5}$$

En la ecuación anterior *n* representa la cantidad de elementos del clúster C_j y $ElemtSim(i, j)$ se especifica en la ecuación 1.6:

$$ElemtSim(i, j) = \begin{cases} 1 & si \; SimRefBib(i, j) \geq \gamma \\ 0 & e.o.c \end{cases} \tag{1.6}$$

Editorial "Ediciones Futuro"
Universidad de las Ciencias Informáticas. La Habana, Cuba
rcci@uci.cu

Revista Cubana de Ciencias Informáticas
Vol. 11, No. 2, Abril-Junio, 2017
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 46-60

Cuando se verifica que dos clústeres son agrupables se puede obtener que $CA(C_i, C_j) < 0{,}5$, pero algunos elementos del clúster C_i pueden tener mayor α-pertenencia al clúster C_j que a C_i, estos elementos a pesar de que los clúster no sean unidos son cambiados al clúster con respecto al cual tienen mayor α-pertenencia.

Verificar para todos los pares de clúster si son agrupables o no tiene un costo computacional alto. Además, no es de interés hacer esta verificación para todos los pares, debido a que la misma forma de seleccionar los elementos que pertenecerán a cada grupo, y la forma que se refinan los grupos durante los pasos anteriores del algoritmo por sí solas evitan en gran medida que se obtenga varios pares de clúster que puedan resultar agrupables. Para evitar la verificación para todos los clústeres se recurren a la selección de *v* elementos aleatorios en cada clúster (*v* definido por el usuario), luego se toman dos a dos los subgrupos obtenidos y se verifica si ellos son agrupables. Solo se verificará si dos clústeres son agrupables si sus subgrupos correspondientes resultaron agrupables. Es importante aclarar que el parámetro *v* se usa únicamente como forma de disminuir la cantidad de elementos a analizar para saber si dos clústeres pueden ser agrupables. Es por ello que si el usuario no proporciona este parámetro, el algoritmo verifica para todos los pares de clústeres si estos son agrupables o no. Suponiendo que se obtienen *k* grupos con *n/k* elementos cada uno, la complejidad de verificar de manera exhaustiva para cada par de clústeres si estos son agrupables sería $O(\frac{n^2}{k} log k)$.

Resultados y discusión

Evaluar los resultados de un agrupamiento es un proceso complejo; 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). Para verificar la validez de los resultados obtenidos a través del algoritmo de agrupamiento propuesto, se diseñó un experimento con el propósito de realizar un análisis estadístico, que permita comprobar si existen diferencias significativas entre este algoritmo y el algoritmo usado en la investigación base (variante del algoritmo *K*-Star) (Pinto, et al., 2010). No se procedió a la comparación con otros algoritmos ya que en (Magdaleno, 2015) se realizó una evaluación de los resultados obtenidos con varios algoritmos de agrupamiento y se demostró que el algoritmo seleccionado es el que obtiene los mejores resultados. La evaluación incluye la verificación y validación.

El experimento desarrollado consistió en la aplicación de medidas externas para evaluar la calidad del agrupamiento. Las medidas externas fueron seleccionadas 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. Las medidas seleccionadas fueron la medida *Overall F-measure* (OFM) propuesta en

Editorial "Ediciones Futuro"
Universidad de las Ciencias Informáticas. La Habana, Cuba
rcci@uci.cu

Revista Cubana de Ciencias Informáticas
Vol. 11, No. 2, Abril-Junio, 2017
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 46-60

(Steinbach et al., 2000) y las medidas *Micro Purity* y *Macro Purity* propuesta su utilización por INEX(Costa and Ortale, 2013).

Como casos de estudios se utilizaron archivos provenientes del sitio ICT[1] y archivos pertenecientes al repositorio IDE-Alliance. Estos últimos, proporcionados por la Universidad de Granada en España, que son internacionalmente utilizados para evaluar resultados de agrupamiento. En la Tabla 1 se especifican las características de cada uno de los corpus utilizados. Note que los tres últimos corpus fueron creados tomando documentos de los dos repositorios anteriores.

Tabla 1. Descripción de los archivos utilizados para evaluar la calidad del agrupamiento.

No. Corpus	**Cantidad de documentos**	**Cantidad de clases**	**Temas que trata**
Conjuntos de documentos XML confeccionados a partir de documentos recuperados del sitio del ICT del Centro de Estudios de Informática de la Universidad Central "Marta Abreu" de Las Villas http://ict.cei.uclv.edu.cu			
1	32	2	Fuzzy Logic, SVM
2	25	2	Rough Set, Association Rules
3	32	2	Rough Set, SVM
4	28	2	Association Rules, Fuzzy Logic
5	32	2	Association Rules, SVM
Recopilación de documentos del repositorio IDE-Alliance, internacionalmente utilizados para evaluar agrupamiento. Proporcionados por la Universidad de Granada. España.			
6	28	3	Copula, Belief Propagation, CL
7	19	2	Copula, Belief Propagation
Documentos pertenecientes al sitio ICT y al repositorio IDE-Alliance			
8	41	4	Rough Set, Copula, Belief Propagation, CL
9	29	2	Copula, SVM
10	38	3	Copula, SVM, Belief Propagation

Como umbral de similitud para ambos algoritmos se usó la media de las similitudes. El cálculo del umbral de similitud tiene complejidad computacional O(*nlogn*), ya que consiste en recorrer la triangular superior (o inferior) de la matriz de similitud y dividir la suma de los valores entre la cantidad de documentos a agrupar.

En el caso del algoritmo *SemClustDML* se fijó la cantidad mínima de elementos de un clúster en seis y la cantidad de elementos aleatorios a seleccionar para comprobar si dos clústeres son agrupables en cuatro. Es válido aclarar que estos parámetros son proporcionados por el usuario y los grupos obtenidos pueden variar considerablemente en dependencia

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

Editorial "Ediciones Futuro"
Universidad de las Ciencias Informáticas. La Habana, Cuba
rcci@uci.cu

Revista Cubana de Ciencias Informáticas
Vol. 11, No. 2, Abril-Junio, 2017
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 46-60

de los valores que se asignen. Esto garantiza que el usuario pueda obtener grupos que se correspondan más a sus necesidades de información.

Para el caso de las medidas *Micro Purity* y *Macro Purity* no se obtuvieron diferencias significativas entre el algoritmo propuesto y la variante del algoritmo *K-Star* utilizada al aplicar la prueba no paramétrica de *Wilcoxon* (Wilcoxon, 1945).

La Figura 3 muestra el comportamiento de la medida OFM para los grupos obtenidos al aplicar a cada uno de los corpus utilizados el algoritmo *SemClustDML* y la variante del algoritmo *K-Star* respectivamente. En esta figura se puede observar que *SemClustDML* es el que obtiene mejores resultados. Para demostrar lo anterior, se empleó la prueba no paramétrica de *Wilcoxon* con los valores arrojados por la medida OFM. Como se puede observar en la Tabla 2, el test de *Wilcoxon* sugiere rechazar la hipótesis nula (p-value < 0,05) para todas las parejas comparadas, esto es que existen diferencias significativas entre los algoritmos comparados con los resultados de la medida OFM para los casos de estudio definidos.

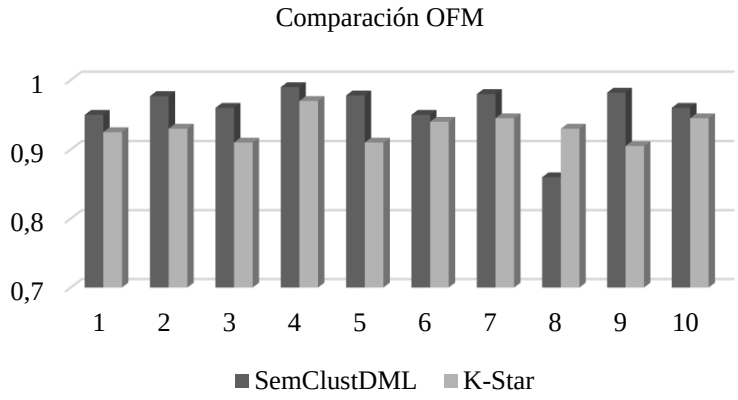

Figura 3. Comparación de los resultados obtenidos para la medida OFM al aplicar los algoritmos *SemClustDML* y variante del algoritmo *K-Star*.

Que existan diferencias significativas asociadas a los valores obtenidos para la medida OFM demuestra que el algoritmo propuesto es más preciso a la hora de obtener grupos de documentos afines que la variante del algoritmo K-*Star*, ya que esta medida combina los conceptos de precisión y cubrimiento. La precisión está referida, en el caso del agrupamiento, a que los documentos que sean ubicados en un grupo, pertenezcan en efecto a ese grupo según la clasificación de referencia. El cubrimiento busca que para cada grupo se logren asignar la mayor cantidad de documentos que según la clasificación de referencia debieran pertenecer al grupo. De esta manera al obtener valores cercanos a uno para la medida OFM se garantiza que los resultados del agrupamiento sean más eficaces.

Editorial "Ediciones Futuro"
Universidad de las Ciencias Informáticas. La Habana, Cuba
rcci@uci.cu

Revista Cubana de Ciencias Informáticas
Vol. 11, No. 2, Abril-Junio, 2017
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 46-60

Tabla 2. Resultado del test estadístico de Wilcoxon para las comparaciones en parejas

Experiment			**SemClustDML-INEXK_STAR**
Z			-2.293
Aymp. Sig (2-tailed)			0.022
Monte Carlo Sig (2-tailed)	Sig.		0.021
	95% Confidence Interval	Lower Bound	0.018
		Upper Bound	0.026
Monte Carlo Sig (2-tailed)	Sig.		0.010
	95% Confidence Interval	Lower Bound	0.008
		Upper Bound	0.012

1.2 Complejidad computacional

La complejidad computacional mostrada para cada uno de los cinco pasos iniciales del algoritmo lleva a la conclusión que el algoritmo propuesto asume en el peor de los casos la complejidad de la eliminación del solapamiento que es O(n^3). En el caso del refinamiento la mayor complejidad computacional la tiene determinar si se puede dividir alguno de los clústeres, y tiene una complejidad de O(n^3). Por lo cual el algoritmo *SemClustDML* en el peor de los casos presenta una complejidad computacional de O(n^3). Esta es mayor que la complejidad de la variante del algoritm *K-Star* con la que se compara la cual es O(kn^2). Sin embargo el diseño del algoritmo *SemClustDML* sigue el principio planteado por Ruiz-Shulcloper en (Ruiz-Shulcloper et al., 1995) donde dice que la definición del criterio de semejanza en el agrupamiento debe estar basada en el conocimiento que se tenga al respecto del problema en concreto que se está tratando, para poder definir así el tipo de comportamiento entre los objetos a partir de sus semejanzas que resulte, según el problema en particular, significativo. De este modo, el peor de los casos es muy poco frecuente en el algoritmo propuesto. Más bien el algoritmo se comporta de manera estable sobre el caso promedio, el cual tiene una complejidad computacional O(*nlog(kn)*), sin consideran el refinamiento y una complejidad de O(n^2) considerando el refinamiento. La complejidad es menor para el caso promedio que para el peor de los casos dado que el algoritmo está diseñado para adaptarse a las características especiales de la matriz *SimRefBib*, por lo que las partes del algoritmo donde la complejidad aumenta significativamente para el peor de los casos, para el caso promedio no se complejiza tanto ya que no se presentan muchos elementos que superen el umbral con más de un centroide.

La Tabla 3 muestra cómo se comportó el algoritmo *SemClustDML* aplicado a los corpus presentados como casos de estudio, con el objetivo de demostrar la complejidad computacional para el caso promedio.

Editorial "Ediciones Futuro"
Universidad de las Ciencias Informáticas. La Habana, Cuba
rcci@uci.cu

Revista Cubana de Ciencias Informáticas
Vol. 11, No. 2, Abril-Junio, 2017
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 46-60

Tabla 3. Comportamiento detallado del algoritmo SemClustDM

Corpus / Parámetro medido	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*
Cantidad de centroides seleccionados	4	2	5	2	4	5	3	7	4	5
Asignación de elementos (cantidad de iteraciones)	112	46	135	52	112	115	48	238	100	165
Elementos solapadores	14	6	19	0	12	8	6	24	12	17
Distribución	(9,2) (5,3)	(6,2)	(14,2) (5,3)	-	(7,2) (5,3)	(3,2) (5,3)	(3,2) (3,3)	(17,2) (7,3)	(8,2) (4,3)	(12,2) (5,3)
Cantidad de iteraciones	434	186	384	0	370	196	123	783	347	484
Cantidad de elementos aislados	0	0	0	0	0	0	0	0	0	0
División de clúster (cantidad de iteraciones)	1569	749	552	1856	467	328	201	769	450	1641

Al establecer una correspondencia entre la cantidad de iteraciones y la cantidad de elementos que presenta cada corpus, se obtiene que la complejidad de asignar los elementos solapadores se acota en O(*nlog(kn)*). Este es el paso más complejo del algoritmo, ya que encontrar los centroides en el peor de los casos asume complejidad de O(*nlogn*), que sigue siendo menor que la complejidad de asignar los elementos solapadores y la asignación de los elementos a los grupos se acota también en O(*nlogn*) para el caso promedio. No se encontraron elementos aislados por tanto este paso no se ejecuta.

Para el caso del refinamiento el paso más complejo es la división de clúster. Su complejidad se aproxima a $O(n^2)$ en el caso promedio. Los pasos dos y tres del refinamiento en el peor de los casos tienen complejidad $O\left(\frac{n^2}{k} logk\right)$, la cual es menor que $O(n^2)$, por lo cual no resulta necesario calcular la complejidad de estos pasos para el caso promedio ya que la mayor complejidad la seguirá aportando la división de clústeres.

Conclusiones

La función de similitud *SimRefBib*, especialmente diseñada para el agrupamiento de artículos científicos permite discernir de manera correcta entre los grupos que deben formarse para una colección de documentos dada, sin embargo, surge la necesidad de diseñar un algoritmo de agrupamiento que sea capaz de adaptarse a las características especiales de la matriz resultante del cálculo de esta función para lograr buenos resultados en el agrupamiento de este tipo de documentos.

Se implementó el algoritmo de agrupamiento para artículos científicos *SemClustDML* el cual hace uso de las características especiales de la matriz *SimRefBib* para mejorar el desempeño en el agrupamiento de este tipo de documentos. Este algoritmo cuenta con dos etapas: la etapa del agrupamiento propiamente dicha y una segunda etapa que consta de tres fases en las cuales se refina el resultado del agrupamiento.

La comparación del algoritmo *K-Star* con el algoritmo *SemClustDML* propuesto en esta investigación arrojó que existen diferencias significativas para la medida *OFM*, obteniéndose mejores resultados para el algoritmo *SemClustDML*.

Referencias Bibliográficas

AGGARWAL, C. C. AND C. ZHAI. A survey of text clustering algorithms. In C.C. AGGARWAL AND C. ZHAI eds. *Mining Text Data.* New york: Springer, 2012, p. 77-128.

ALJABER, B., N. STOKES, J. BAILEY AND J. PEI Document clustering of scientific texts using citation contexts. Information Retrieval, 2010, 13(2), 101-131.

BEZDEK, J. C., R. EHRLICH AND W. FULL FCM: The fuzzy c-means clustering algorithm. Computers & Geosciences, 1984, 10(2-3), 191-203.

COSTA, G. AND R. ORTALE. A latent semantic approach to xml clustering by content and structure based on non-negative matrix factorization. In *Machine Learning and Applications (ICMLA), 2013 12th International Conference on.* IEEE, 2013, vol. 1, p. 179-184.

DOMÍNGUEZ, Y. L., F. A. FUENTES, A. F. BRUZÓN AND R. O. BUENO Optimizaciones al Algoritmo de Agrupamiento Compacto Jerárquico Dinámico. Revista Cubana de Ciencias Informáticas, 2014, 8(Especial UCIENCIA 2014), 59-65.

GIL-GARCÍA, R. AND A. PONS-PORRATA Dynamic hierarchical algorithms for document clustering. Pattern Recognition Letters, 2010, 31(6), 469-477.

GUHA, S., R. RASTOGI AND K. SHIM. CURE: an efficient clustering algorithm for large databases. In *ACM Sigmod Record.* ACM, 1998, vol. 27, p. 73-84.

JAIN, A. K., M. N. MURTY AND P. J. FLYNN Data clustering: a review. ACM computing surveys (CSUR), 1999, 31(3), 264-323.

LIN, Y. S., J. Y. JIANG AND S. J. LEE A similarity measure for text classification and clustering. IEEE transactions on knowledge and data engineering, 2014, 26(7), 1575-1590.

MAGDALENO, D. Metodología para el agrupamiento de documentos semiestructurados. Universidad Central "Marta Abreu" de Las Villas, 2015.

MAGDALENO GUEVARA, D., I. E. FUENTES, M. CABEZAS AND M. M. GARCÍA LORENZO Recuperación de información para artículos científicos soportada en el agrupamiento de documentos XML. Revista Cubana de Ciencias Informáticas, 2016, 10(2), 57-72.

MAGDALENO GUEVARA, D., Y. MIRANDA, I. E. FUENTES AND M. M. GARCÍA Comparative Study of Clustering Algorithms using OverallSimSUX Similarity Function for XML Documents. Inteligencia artificial: Revista Iberoamericana de Inteligencia Artificial, 2015, 18(55), 69-80.

PINTO, D., M. TOVAR, D. VILARIÑO, B. BELTRÁN, et al. BUAP: Performance of K-Star at the INEX'09 Clustering Task. In S. GEVA, J. KAMPS AND A. TROTMAN eds. *Focused Retrieval and Evaluation: 8th International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2009, Brisbane, Australia, December 7-9, 2009, Revised and Selected Papers.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, p. 434-440.

QIAN, Y. AND K. ZHANG. A customizable hybrid approach to data clustering. In *Proceedings of the 2003 ACM symposium on Applied computing.* New York: ACM, 2003, p. 485-489.

RAJESHWARI, P., B. SHANTHINI AND M. PRINCE Hierarchical energy efficient clustering algorithm for WSN. Middle East Journal of Scientific Research, 2015, 23, 108-117.

RUIZ-SHULCLOPER, J., E. ALBA AND M. LAZO. Introducción al reconocimiento de patrones. Enfoque lógico combinatorio. In.: México, CINVESTAV IPN, 1995.

SERT, S. A., H. BAGCI AND A. YAZICI MOFCA: Multi-objective fuzzy clustering algorithm for wireless sensor networks. Applied Soft Computing, 2015, 30, 151-165.

STEINBACH, M., G. KARYPIS AND V. KUMAR. A comparison of document clustering techniques. In *KDD workshop on text mining.* Boston, 2000, vol. 400, p. 525-526.

VARGAS FLORES, S. I. Comparación de medidas de similitud para desambiguación del sentido de las palabras utilizando rankeo de grafos. Universidad Autónoma del Estado de México, 2016.

WEI, C.-P., R. H. CHIANG AND C.-C. WU Accommodating individual preferences in the categorization of documents: A personalized clustering approach. Journal of Management Information Systems, 2006, 23(2), 173-201.

WILCOXON, F. Individual comparisons by ranking methods. Biometrics Bulletin, 1945, 1(6), 80-83.