Benemérita Universidad Autónoma de Puebla
Facultad de Ciencias de la Computación

Un método para el agrupamiento con traslape para documentos basado en k-means traslapado

por

Beatriz Beltrán Martínez

Tesis para obtener el grado de Doctor en
Ingeniería del Lenguaje y del Conocimiento

2020

Director:
Dra. Darnes Vilariño Ayala

Co-Director:
Dr. José Francisco Martínez Trinidad

Dedicatoria

A Adrián quien me ha acompañado durante este tiempo, me ha apoyado, me ha animado a continuar y sobretodo a no claudicar. Por el tiempo que no pudimos estar juntos y entender. Hijo, porque eres lo más importante para mí, hoy y siempre.

¡Te quiero!

A mi mamá, aunque no tengo palabras para expresar todo lo que siento, pero si por lo que te extraño y por lo que me haces falta...

Gracias ¡má!

A mis hermanos Vero, Raúl y Paty, por todo el apoyo brindado, por formar parte de esta historia, por estar juntos en las buenas, en las no tan buenas, pero sobretodo en las malas.

Los quiero

A mi papá, siempre ha sido mi ejemplo a seguir; a pesar del tiempo te sigo extrañando como nunca, y me hubiera gustado mucho que estuvieras.

¡Gracias!

Agradecimientos

Mi más sincero agradecimiento a mis asesores Dra. Darnes Vilariño Ayala y Dr. José Francisco Martínez Trinidad por todo el apoyo que me brindaron, los consejos y el enriquecimiento al desarrollo de este trabajo, el cual no hubiera podido terminar sin todos sus consejos y observaciones. Este trabajo no ha sido fácil, pero sin sus valiosos comentarios hubiera sido imposible.

A mis comité tutorial Dra. María Josefa Somodevilla García, Dr. David Eduardo Pinto Avendaño y Dr. Arturo Olvera López, por todo el apoyo brindado a este trabajo para enriquecerlo.

A todos mis amigos y para no olvidarme de ninguno, por los momentos que pasamos de estrés, de estar al borde del colapso, todo el trabajo que hemos tenido, las discusiones, los apoyos y las platicas.

Gracias a la Benemérita Universidad Autónoma de Puebla por las facilidades brindadas para llevar a cabo mis estudios.

Gracias al CONACyT por el financiamiento otorgado para el desarrollo de ésta investigación bajo el número de beca 481750.

Resumen

Existe una gran cantidad de algoritmos propuestos en la literatura y que generan grupos sin traslape, a pesar de que dichos algoritmos resuelven muchos problemas que han sido planteados durante su desarrollo, en la actualidad existen algunos problemas que requieren una solución donde el agrupamiento no sea disjunto, es decir, obteniendo grupos que consideren un traslape.

En la presente investigación se desarrolla un método de agrupamiento con traslape para documentos basado en el algoritmo k-means traslapado con peso (WOKM). Además, se propone un nuevo método de inicialización de los centroides basado en el Punto de Transición (PT); se analiza el comportamiento de esta nueva propuesta tomando una ventana alrededor del PT y se propone además una nueva forma de calcular el PT, considerando no cada documento por separado, sino a partir de un corpus conformado por todos los documentos.

Los experimentos realizados con diferentes corpus muestran que método propuesto k-means traslapado con peso, usando el punto de transición (WOKMTP) obtiene mejores resultados en términos de la métrica *FBCubed* que los algoritmos de agrupamiento con traslape del estado del arte.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1. Introducción

Las formas de comunicación han sido elementos indispensables a lo largo de la historia de los seres humanos, las cuales han ido evolucionando; así se tiene desde idiomas primitivos hasta el habla. Además, se puede considerar otros medios de comunicación, tales como pinturas, gráficas, escritura, entre otras y de forma no verbal (expresiones y lenguaje corporal). En particular la paralingüística sirve para enfatizar la comunicación verbal (Cangelosi (2001)).

En el caso particular del desarrollo de la comunicación escrita y considerando el avance del Internet, se dio inicio a la conexión de varias computadoras y al envío de mensajes entre ellas. En la actualidad, se manejan grandes volúmenes de información lo que es conocido como big data; además, se han desarrollado servicios que permiten manipular estos grandes volúmenes de información en la nube. Los servicios que se ofrecen están clasificados en tres capas (software como servicio (SaaS), plataforma como servicio (Paas) e infraestructura como servicio (Iaas)), que son necesarias para el manejo de la información en la nube. Como se planteó con anterioridad el agrupamiento de la información ayuda a la manipulación de los datos y a la disminución de la cantidad de información a procesar.

El agrupamiento construye grupos que están compuestos por elementos u objetos que son similares (Jain y Dubes (1988)). Por otra parte, el agrupamiento con traslape hace referencia a la construcción de grupos, donde los elementos u objetos pueden ser asignados a más de uno de ellos. Por ejemplo, en los sistemas de recomendación se pueden observar personas que tienen gustos culinarios en común, pero pueden también tener diferentes niveles académicos, por lo cual estarían en los dos grupos simultáneamente.

En la presente investigación los elementos u objetos se consideran documentos escritos en texto plano; a pesar de que en la actualidad no existen grandes problemas de almacenamiento, continua presentándose el problema del manejo de la memoria.

La mayoría de los algoritmos de agrupamiento desarrollados, construyen grupos sin traslape, esto es debido a que muchos problemas requieren de grupos disjuntos; aunque también existen algoritmos de agrupamiento que construyen grupos con traslape.

Existen diferentes formas de clasificar a los algoritmos de agrupamiento, en particular, por el tipo de agrupamiento obtenido. Jain y Dubes (1988) proponen, que se puede tener la clasificación de los algoritmos de agrupamiento como:

- **Disjuntos**, esto es, que en los grupos construidos un objeto pertenece a un grupo y no al resto.
- **Difusos**, cuando en los agrupamientos construidos un objeto pertenece a todos los grupos, pero con un cierto grado de pertenencia.
- **Traslapados**, es donde un objeto puede pertenecer a más de uno de los grupos

construidos.

A continuación se define el problema que se aborda en la presente investigación.

1.1. Definición del problema

Este trabajo de investigación, tiene como objetivo el desarrollo de un método de agrupamiento con traslape para documentos basado en el algoritmo k-means traslapado con peso.

El método permite que se pueda especificar el número de grupos a formar y se hace mención de que el número de atributos a considerar no es fijo, para los elementos a agrupar, esto conduce al planteamiento del objetivo general y los objetivos particulares.

1.2. Objetivo general y objetivos específicos de la investigación

A continuación, se presentan los objetivos, primero se enuncia el objetivo general y posteriormente los objetivos particulares que se han planteado:

Objetivo general: Desarrollar un método de agrupamiento con traslape para documentos basado en el algoritmo k-means traslapado con peso, cuyos resultados superen significativamente a los algoritmos de agrupamiento con traslape reportados en la literatura.

Objetivos particulares:

1. Evaluar las técnicas y algoritmos de agrupamiento desarrollados para determinar su rendimiento en términos de precisión de los resultados y complejidad

de los algoritmos.

2. Evaluar diferentes métodos de inicialización de centros.

3. Desarrollar un método para inicializar los centros para el algoritmo k-means traslapado con peso.

4. Diseñar e implementar el método de agrupamiento con traslape basado en el algoritmo k-means traslapado con peso.

1.3. Preguntas de investigación

Una vez identificadas las limitantes de los algoritmos de agrupamiento actual, se establecen las siguientes preguntas de investigación:

1. ¿Será posible desarrollar un método de agrupamiento con traslape basado en el algoritmo k-means traslapado con peso, que obtenga resultados estadísticamente superiores a los presentados en el estado del arte?

2. ¿Existirá una relación entre las medidas de similitud y el tipo de algoritmo de agrupamiento con traslape, desarrollado para medir la calidad del método?

3. ¿Existe relación entre la calidad de los resultados obtenidos por el método desarrollado y la cantidad de datos de entrada?

1.4. Hipótesis

Con base en las preguntas que se proponen, se establecen las siguientes hipótesis:

Hipótesis 1: El rendimiento del método con traslape propuesto es estadísticamente superior a los presentados por otros modelos de agrupamiento con traslape existentes en la literatura, mediante una representación adecuada.

Hipótesis 2: El método de agrupamiento con traslape presenta un rendimiento por encima de los algoritmos clásicos, cuando el conjunto de datos es mayor a los reportados por la literatura.

1.5. Justificación

En la actualidad, el desarrollo de métodos de agrupamiento de documentos que permitan traslape entre los grupos creados, proporciona una herramienta útil en problemas que actualmente son de interés en diferentes ámbitos, por ejemplo, los sistemas de recomendación.

En los sistemas de recomendación una persona tiene diferentes preferencias y éstas pueden ser utilizadas en las áreas comerciales, telecomunicaciones, comercio electrónico, etc. Por otra parte, el perfilado de usuarios, podría servir para identificar a una persona y personalizar por ejemplo sus compras, o su posible identificación ante un delito. Los ejemplos anteriormente mencionados dejan claro que en general los seres humanos pueden pertenecer a más de un grupo según su preferencia, lo que establece la necesidad de desarrollar métodos eficientes para manejar información traslapada.

1.6. Límites de la tesis

No se desarrollarán nuevas métricas para determinar la calidad de los grupos obtenidos. En los casos en los que la cantidad de datos sobrepase la capacidad de los algoritmos reportados en la literatura, no se podrá comparar su rendimiento con respecto a la aproximación propuesta en esta tesis.

1.7. Estructura de la tesis

La estructura de este trabajo de investigación está dada de la siguiente manera: En el capítulo I (Introducción) se discute el problema a resolver, el objetivo general y los particulares, así como las preguntas de investigación y las hipótesis planteadas.

En el capítulo II, se ofrece el marco teórico necesario para comprender el trabajo desarrollado en esta tesis. El capítulo III hace un recorrido entre las diferentes publicaciones que representan el estado del arte de los algoritmos de agrupamiento con traslape para documentos.

El capítulo IV presenta el método propuesto basado en el algoritmo de k-means traslapado con peso.

En el capítulo VI se evalúa el método propuesto en corpora estándar y públicos y se compara con los algoritmos del estado del arte. Finalmente, se ofrecen las conclusiones, aportaciones y el trabajo a futuro.

Capítulo 2. Marco Teórico

En esta sección se presentan los conceptos teóricos necesarios para la comprensión del problema que se aborda en esta tesis. Iniciamos con el concepto de agrupamiento que es la base de esta investigación.

2.1. Agrupamiento

La clasificación de objetos es realizada por el ser humano desde que es pequeño, colocar objetos por colores o formas, son actividades que se desarrollan desde los primeros años de vida. Por lo anteriormente planteado la formalización, el desarrollo de algoritmos y de métodos para agrupar o clasificar es de suma importancia; por lo que el problema de agrupar objetos está siendo ampliamente estudiado en la literatura.

La importancia del agrupamiento es tal que el agrupamiento puede ser aplicado al desarrollo de resúmenes automáticos y en la mercadotecnia.

Según Aggarwal y Reddy (Aggarwal y Reddy (2013)) el agrupamiento es definido, como un conjunto de objetos particionados en diferentes conjuntos, tal que los objetos sean tan similares como sea posible.

Por otra parte, Jain y Dubes (Jain y Dubes (1988)), consideran que en un agrupamiento los objetos semejantes pertenecen al mismo grupo, mientras que los objetos que no son semejantes pertenecen a diferentes grupos.

Finalmente, para Gan (Gan et al. (2007)) en el agrupamiento de objetos los grupos son creados con aquellos objetos que son muy similares, utilizando para ello una medida de semejanza.

Como nuestra investigación se basa en el desarrollo de un método de agrupamiento basado en el algoritmo k-means traslapado con peso, se hace necesario explicar este algoritmo de agrupamiento.

2.2. Algoritmo k-means traslapado con peso

En la actualidad, existen problemas que pueden ser tratados utilizando algoritmos de agrupamiento, en los cuales se consideren objetos que se encuentren en más de un grupo; lo que ha originado la necesidad de diseñar algoritmos de agrupamiento con traslape.

A continuación se presenta el algoritmo k-means traslapado con peso (WOKM por sus siglas en inglés), en el cual se apoya el método propuesto en esta tesis.

El algoritmo k-means traslapado con peso fue propuesto por Cleuziou (2010), y se basa en el algoritmo k-means, pero a diferencia de éste, construye grupos que tienen traslape, utilizando un peso en cada uno de ellos.

Sea $\chi = \{x_1, x_2, ..., x_n\}$ un conjunto de objetos, donde cada objeto es descrito por p características. Además, sea $\mathfrak{C} = \{C_1, C_2, ..., C_k\}$ el conjunto de grupos tras-

lapados que se construirán y $\mathfrak{M} = \{m_1, m_2, ..., m_k\}$ los centros de cada uno de los grupos, respectivamente.

El algoritmo WOKM desarrolla la misma idea del algoritmo k-means, pero buscando el mínimo de la siguiente función objetivo:

$$\varphi(\{C_i\}_{i=1}^{k}) = \sum_{x_i \in \chi} \sum_{v=1}^{p} \gamma_{i,v}^{\delta} |x_{i,v} - \phi_v(x_i)|^2 \tag{2.1}$$

donde $\mathfrak{C}$ es un cubrimiento de χ, considerando que cada objeto x_i debe pertenecer al menos a un grupo. La ecuación 2.1, $\phi_v(x_i)$ denota la "imagen" de x_i, en términos de la característica v que está definida por la combinación de los prototipos ($m_c \in \mathfrak{M}$) al que x_i pertenece, y es obtenida como:

$$\phi_v(x_i) = \frac{\sum_{m_j \in A_i} \lambda_{j,v}^{\delta} m_{j,v}}{\sum_{m_j \in A_i} \lambda_{j,v}^{\delta}},$$

donde $A_i = \{m_j | x_i \in C_j\}$ lo que significa que A_i es el conjunto de centros, a los que el objeto x_i pertenece. En otras palabras, $\phi_v(x_i)$ es un promedio con peso de los centros de los grupos a los que x_i pertenece.

En 2.1 $\gamma_{i,v}$ es calculada como sigue:

$$\gamma_{i,v} = \frac{\sum_{m_j \in A_i} \lambda_{j,v}}{|A_i|}$$

donde $\lambda_{j,v}$ es el peso asociado a la característica v en el grupo j, tal que $\forall j$, $\sum_{v=1}^{p} \lambda_{j,v} = 1$. Además, el parámetro $\delta > 1$ que se encuentra en 2.1 sirve para regular la influencia del peso $\lambda_{j,v}$.

Dado k centros tomados de χ o generados aleatoriamente, y dado el peso inicial asociado a la característica v se toma como $\lambda_{j,v} = 1/p$ para $v = 1, ..., p$ y $j = 1, ..., k$. La optimización de la función objetivo 2.1 es alcanzada por una ejecución iterativa de los siguientes 3 pasos:

1. **Multi-asignamiento:** En este paso cada objeto x_i es asignado a uno o más grupos, tal que la función objetivo es minimizada; y como resultado se obtiene el conjunto A_i mediante la siguiente heurística:

   Dado $\mathfrak{C} = \{C_1, C_2, ..., C_k\}$ que es un conjunto de grupos traslapados, con centros $\mathfrak{M} = \{m_1, m_2, ..., m_k\}$, y donde el objeto x_i es asignado a uno o más grupos de $\mathfrak{C}$; para lograr esto inicialmente $A_i = \emptyset$, por lo tanto los centros de cada grupo se evalúan de manera iterativa, desde el centro más cercano al más lejano de x_i.

   Para construir el multiasignamiento se asigna x_i al grupo del centro más cercano m_r y se evalúa la ecuación 2.1, posteriormente se asigna x_i al centro m_r y a m_s (siendo m_s el segundo centro más cercano), si esta asignación produce un valor más pequeño de la ecuación 2.1 que la primera asignación (es decir, se mejora), entonces ahora se prueba la asignación de x_i a m_r, m_s y m_t (siendo m_t el tercer centro más cercano), si esta asignación produce un valor más pequeño en 2.1 que la asignación anterior entonces se continua con este procedimiento probando la asignación de x_i, cuando una asignación no produzca un mejor resultado en 2.1 que la asignación anterior se detiene el proceso y en A_i estarán los centros a los que x_i fue asignado, es decir A_i tendrá el multiasignamiento

de x_i.

2. **Actualización de los centros de cada grupo:** Para el cálculo del nuevo centro m_j^* se toma en cuenta, si x_i pertenece a varios agrupamientos, si es así, éste aporta menos para el cálculo del nuevo centroide (el promedio o la media), y se expresa como se muestra en la siguiente ecuación:

$$m_j^* = \frac{1}{\sum_{x_i \in C_j} \alpha_i} \sum_{x_i \in C_j} \alpha_i (|A_i| x_i - \sum_{m_c \in A_i \setminus \{m_j\}} m_c)$$

donde $\alpha_i = 1/|A_i|^2$.

3. **Actualización de los pesos:** Una vez que los grupos y centros han sido calculados, los pesos locales se actualizan por medio de la siguiente ecuación:

$$\lambda_{j,v} = \frac{(\sum_{x_i \in c_j |\ |c_i|=1} (x_{i,v} - m_{j,v})^2)^{1/(1-\delta)}}{\sum_{u=1}^{p} (\sum_{x_i \in c_j |\ |c_i|=1} (x_{i,u} - m_{j,u})^{1/(1-\delta)}}$$

Los pasos anteriormente descritos se repiten hasta que los grupos no tienen cambios, o bien debido a que se alcanza un número máximo de iteraciones. Finalmente, el algoritmo WOKM obtiene k grupos traslapados.

2.3. Medidas de semejanza entre documentos

Las medidas de semejanza permiten cuantificar el grado de asociación entre los objetos de un grupo (Tan et al. (2005)).

Para comparar documentos, en la literatura se han propuesto varias medidas de semejanza o distancia, las medidas presentadas en la tabla 2.1 son las más utilizadas para determinar la similitud entre documentos.

Tabla 2.1: Medidas de semejanza entre documentos.

Medida de semejanza	Año	Fórmulas
Euclidiana	300 AC	$\mathcal{D}(x_i, x_j) = \|\|x_i - x_j\|\| = \sqrt{\sum_{l=1}^{n}(x_{il} - x_{jl})^2}$ $\mathcal{D}_W(x_i, x_j) = \|\|x_i - x_j\|\|_W = \sqrt{(x_i - x_j)^T W (x_i - x_j)}$
Jaccard	1901	$\mathcal{J}(x_i, x_j) = \frac{\|x_i \cap x_j\|}{\|x_i \cup x_j\|}$
Dice	1945	$\mathcal{S} = \frac{2\|x_i \cap x_j\|}{\|x_i\| + \|x_j\|}$
Jaccard extendido (Tanimoto)	1960	$\mathcal{EJ}(x_i, x_j) = \frac{x_i \cdot x_j}{\|\|x_i\|\|^2 + \|\|x_j\|\|^2 - x_i \cdot x_j}$
Coseno	1979	$\mathcal{D}_{cos}(x_i, x_j) = 1 - \cos(x_i, x_j) = 1 - \frac{x_i^T x_j}{\|\|x_i\|\| \|\|x_j\|\|} = 1 - \frac{\sum_{l=1}^{n} x_{il} x_{jl}}{\|\|x_i\|\| \|\|x_j\|\|}$

La distancia Euclidiana se usa si las características que se toman en cuenta permiten describir a los objetos de $\mathscr{X}$ cuantitativamente, donde cada objeto $x_i \in \mathscr{X}$ es un vector n-dimensional $x_i = (x_{i1}, x_{i2}, ..., x_{in})^T$ (Wierzchon y Kłopotek (2018)).

La distancia coseno es utilizada cuando el problema a resolver es de alta dimensionalidad, la cual está definida como 1 menos el coseno del ángulo creado entre los vectores x_i y x_j.

El coeficiente de Jaccard por su parte mide el grado de similitud entre conjuntos que son representados mediante una matriz de semejanza. Sea x_i la $i-$ésima fila y x_j la $j-$ésima columna dentro de dicha matriz, entonces el grado de similitud entre x_i y x_j queda en la intersección de la fila y la columna, no importando el tipo de dato que se trabaja (Tan et al. (2005)).

El Coeficiente de Jaccard extendido también conocido como coeficiente de

Tanimoto, se ha utilizado para encontrar la similitud entre documentos, su comportamiento es muy similar a la medida de similitud de Jaccard (Tanimoto (1958)).

El coeficiente de Dice, es un estadístico que se utiliza para comparar pares de documentos (Dice (1945)). El resultado de esta medida no varía mucho al valor que se obtiene utilizando el coeficiente de Jaccard (Jackson et al. (1989)).

2.4. Inicialización de los centros en el algoritmo k-means

En esta sección se discuten los métodos k-means armónico y *canopy* (Zhang et al. (1999); McCallum et al. (2000)), al utilizar el algoritmo k-means armónico, se toman los centroides de la última iteración, para usarlos como semillas iniciales para el algoritmo k-means, esto define un método de búsqueda o de selección de centros iniciales. De manera semejante se realiza en el algoritmo *canopy*. Se eligen éstos dos métodos porque han sido reportados en la literatura, como los que obtienen los mejores resultados.

En la presente investigación se propone un método para buscar centros iniciales para el algoritmo de agrupamiento WOKM, que será comparado contra los métodos que se explican a continuación.

2.4.1. Inicialización de centros usando el algoritmo k-media armónica

El algoritmo de k-media armónica (KHM, por sus siglas en inglés) (Zhang et al. (1999, 2000)), está basado en el algoritmo k-means, el cual utiliza la media armónica en lugar del promedio como centroide de los grupos, ya que la media armónica de un

conjunto de valores se define como el inverso de la media aritmética de los recíprocos; y se considera que el resultado no está influido por valores extremos (Zhang et al. (2000)).

Sea $\mathfrak{X} = \{x_1, x_2..., x_n\}$ un conjunto de n números, la media armónica H se define:

$$H = \frac{n}{\sum_{i=1}^{n} \frac{1}{x_i}}$$

Sean $\mathfrak{M} = \{m_1, m_2, ..., m_k\}$ los centros de cada uno de los grupos; la función objetivo a optimizar se presenta en la siguiente ecuación:

$$\varphi(\{m_l\}_{l=1}^{k}) = \sum_{i=1}^{n} \frac{k}{\sum_{j=1}^{k} \frac{1}{\|x_i - m_j\|^t}},$$

donde t es un parámetro ($t \geq 2$), la expresión $\sum_{j=1}^{k} \frac{1}{\|x_i - m_j\|^t}$ es la media armónica.

Cada centro se actualiza utilizando la ecuación:

$$m_j = \frac{\sum_{i=1}^{n} z(m_j|x_i) w(x_i) x_i}{\sum_{i=1}^{n} z(m_j|x_i) w(x_i)}, \tag{2.2}$$

donde $z(m_j|x_i)$ es la función miembro de x_i al grupo m_j, el cual es calculado como:

$$z(m_j|x_i) = \frac{\|x_i - m_j\|^{-p-2}}{\sum_{j=1}^{k} \|x_i - m_j\|^{-p-2}},$$

y $w(x_i)$ es el peso asociado a cada x_i, que se determina como:

$$w(x_i) = \frac{\sum_{j=1}^{k} \|x_i - m_j\|^{-p-2}}{(\sum_{j=1}^{k} \|x_i - m_j\|^{-p})^2}$$

El algoritmo KHM inicia con k centros elegidos aleatoriamente, cada instancia se asocia al grupo con centro más cercano, posteriormente el algoritmo actualiza los centros usando la ecuación 2.2, y el proceso se repite hasta que los grupos no cambian o bien cuando el número de iteraciones es alcanzado.

En Khanmohammadi et al. (2017) se mostró que utilizar los centros de la última iteración de k-means armónico como los centros iniciales para el algoritmo k-means traslapado, produce mejores resultados que seleccionar los centros iniciales de manera aleatoria. Por lo tanto, utilizaremos este método en nuestros experimentos, para ser comparado contra nuestra propuesta.

2.4.2. Inicialización de centros usando el algoritmo *canopy*

Uno de los problemas que se presentan en los algoritmos de agrupamiento es la alta dimensionalidad y el manejo de grandes conjuntos de datos, por lo cual en McCallum et al. (2000) se presenta una técnica de agrupamiento que pretende abordar el problema antes mencionado.

La idea principal consiste en utilizar una medida de distancia no costosa para dividir los datos en subconjuntos traslapados, a los que se les llama *canopies*. Posteriormente, se ejecuta un algoritmo de agrupamiento, pero sin calcular la distancia entre puntos que no pertenecen al mismo *canopy*. Bajo esta idea, se considera que

se reduce el costo computacional.

Los *canopies* tienen la ventaja de aceptar una gran variedad de dominios para ser utilizados en diferentes algoritmos de agrupamiento, por ejemplo: k-means, máxima verosimilitud (EM), y el algoritmo aglomerativo ávido. Los *canopies* son creados considerando que aquellos puntos que no están en el mismo grupo, son tan lejanos como sea posible.

Para generar los *canopies* el proceso se divide en dos etapas:

1. En la primera etapa se tiene una métrica sencilla cuyo objetivo es la obtención de los grupos *canopies* traslapados; a partir de un conjunto de puntos. Se debe considerar que todos los elementos pertenecen al menos a un grupo.

2. La segunda etapa utiliza alguno de los algoritmos de agrupamiento, considerando no calcular la distancia entre elementos, que no se encuentran en el mismo *canopy*.

En McCallum et al. (2000) se mostró que utilizar los centros obtenidos por canopy, como los centros iniciales para el algoritmo k-means, reduce el tiempo computacional y produce mejores resultados, que seleccionar los centros iniciales de manera aleatoria. Por lo tanto, utilizaremos este método en nuestros experimentos, para ser comparados con nuestra propuesta desarrollada.

2.5. Punto de transición

El punto de transición (PT) (Urbizagástegui (1999)) hace referencia a palabras en el vocabulario de un texto, el cual divide al vocabulario V en palabras de alta

y baja frecuencia. Urbizagástegui se refiere a este concepto utilizando la ley de Zipf (Zipf (1949)), lo que permite representar a un documento mediante la vecindad de las palabras que se describen alrededor del punto de transición. En el trabajo de Urbizagástegui se demuestra que estas palabras son las que deben indexar a cada documento.

En Manning et al. (2008) se utiliza la ley de Zipf como la distribución de las palabras en una colección. Se basa en contar las palabras de las más frecuentes a las menos frecuentes, teniendo de esta forma a t_1 como la palabra más frecuente en la colección, t_2 es la siguiente palabra más frecuente y así sucesivamente.

Zipf considera que es más fácil representar un texto considerando solamente utilizar aquellas palabras más conocidas, sin preocuparse por aquellas que no se conocen. La Ley de Zipf demuestra que la representación de un documento se puede calcular como la frecuencia cf_i de la $i-$ésima palabra, y esto es proporcional a $1/t$, donde t es el número total de palabras a considerar, para la representación del documento completo:

$$cf_i \alpha \frac{1}{i}$$

Como cf_i se calcula para la $i-$ésima palabra, el número de palabras a considerar se va decrementando; de manera equivalente se puede reescribir la ley de Zipf como:

$$cf_i = ci^k$$

o como:

$$\log cf_i = \log c + k \log i$$

considerando a $k = -1$ y $c \approx \frac{1}{13}$ como una constante.

El proceso para la identificación de palabras alrededor del punto de transición es bastante simple y eficiente. Dado un vocabulario V:

$$V = \{(t_1, f(t_1)), ..., (t_n, f(t_n))\}$$

donde t_i es una palabra en el vocabulario y $f(t_i)$ es su correspondiente frecuencia, se calcula l_1 como el número de palabras que tienen frecuencia 1.

$$l_1 = \#\{t_i | (t_i, f(t_i)) \in V, f(t_i) = 1\}$$

$t_P T$ es aquella palabra del vocabulario que tiene la frecuencia más cercana al PT, y permite detectar al conjunto de palabras de alta frecuencia dentro documento y al conjunto de palabras de baja frecuencia. Por lo tanto el PT se define como:

$$PT = \frac{\sqrt{1 + 8l_1} - 1}{2}$$

La representación de un documento con las palabras cercanas al PT permite manejar al documento completo, con un número menor de palabras; es decir, aquellas palabras más representativas de dicho documento. Cuando se habla de las palabras

alrededor del punto de transición se busca que el documento sea representado por aquellas palabras más importantes.

Capítulo 3. Estado del arte

En este capítulo se describen los algoritmos de agrupamiento con traslape para documentos reportados en la literatura.

Uno de los primeros trabajos en el ámbito de los algoritmos de agrupamiento con traslape es presentado en Arabie et al. (1981), el cual relaja la restricción de que los objetos estén agrupados en una categoría exhaustiva y mutuamente exclusiva, permitiendo el establecimiento de agrupamientos traslapados. Es notorio que muchos conjuntos de datos al ser agrupados, no tienen porqué estar en grupos exclusivos, es de ahí que se tiene la necesidad de una solución con traslape, pero crear todos los posibles conjuntos traslapados es muy costoso computacionalmente, por lo que se deben desarrollar heurísticas para seleccionar los grupos potenciales. ADCLUS asume que se tienen n objetos para agrupar y una matriz de proximidad, la cual previamente se ha normalizado en el intervalo $[0, 1]$, además se asigna un peso no negativo que representa la propiedad del rasgo más sobresaliente; el modelo ADCLUS resuelve el problema evaluando la expresión que se muestra a continuación:

$$\hat{S}_{i,j} = \sum_{k=1}^{m} w_k P_{ik} P_{jk} \tag{3.1}$$

donde $\hat{S}_{i,j}$ es la semejanza entre los objetos i y j, m es el número de grupos traslapados, cada uno con un peso numérico no negativo asociado, w_k (donde $k = 1...m$) y

$$P_{ik} = \begin{cases} 1 \text{ si el objeto } i \text{ tiene la propiedad } k \\ 0 \text{ en otro caso} \end{cases}$$

ADCLUS calcula la semejanza $\hat{S}_{i,j}$ de cualquier par de objetos como la suma de los pesos de los grupos donde i y j pertenecen.

Por otro lado, el algoritmo MAPCLUS construye una representación matricial del modelo ADCLUS, con la diferencia de que en este algoritmo el usuario puede definir un pequeño número de grupos. El algoritmo MAPCLUS fue probado con un corpus creado con comentarios hechos en desayunos. Los autores proponen aplicar dicho algoritmo con un corpus general que sea independiente del dominio.

En el algoritmo Star (Aslam et al. (1998)) se presenta un algoritmo de agrupamiento con traslape incremental. El algoritmo Star parte de un grafo no dirigido, donde cada vértice del grafo representa a un documento y las aristas entre vértices representan la semejanza entre documentos (usando la función de similitud coseno). Si se consideran solamente aristas entre vértices del grafo con una semejanza mayor a un cierto umbral β, entonces a este grafo se le denomina grafo de β-semejanza. El algoritmo Star construye un cubrimiento del grafo antes mencionado mediante subgrafos tipo estrella, donde, un subgrafo tipo estrella es un grafo en el que existe un vértice llamado centro y hay una arista entre este vértice y el resto de

los vértices, llamados satélites. Cada grafo tipo estrella seleccionado conformará el agrupamiento y en la lista de grupos X se guardarán los vértices centro, al final, esta lista incluirá también los vértices aislados, a los que se les conoce como grafos tipo estrella degenerados. Star trabaja en dos pasos para construir un cubrimiento del grafo de β-semejanza. En el primer paso se calcula el grado de los vértices del grafo β-semejanza y los ordena en una lista L de posibles vértices centro en forma descendente, en el segundo paso y de manera iterativa, se toma el vértice de mayor grado en L como el centro de un grafo tipo estrella, con sus vértices adyacentes como satélites, el cual forma un grupo. Todos los vértices del nuevo grafo tipo estrella se eliminan de la lista L y se agrega el centro del grafo tipo estrella a la lista X. Este proceso se repite hasta obtener un conjunto de subgrafos tipo estrella que cubren totalmente al grafo original. De esta forma los grupos obtenidos en la lista X tienen traslape, debido a que los vértices pueden ser adyacentes a más de un vértice centro. Para las pruebas realizadas los autores utilizaron el corpus TREC-6[1].

Zamir y Etzioni (1998), encontraron que los grupos creados a partir de fragmentos cortos devueltos por los motores de búsqueda Web (resúmenes), son tan buenos como los grupos creados usando el texto completo de los documentos Web. Para crear los grupos, los autores utilizaron el algoritmo de agrupamiento de árboles de sufijos (STC), el cual es incremental y obtiene los grupos mediante la identificación de las frases comunes entre los documentos, se entiende por frase una o más palabras consecutivas dentro de un documento. Se define un grupo base como el conjunto de aquellos documentos que comparten una frase; el algoritmo considera

[1]https://trec.nist.gov/

tres pasos: en el primero se realiza una limpieza de los documentos (eliminación de sufijos, prefijos, reducción de plural a singular, eliminación de la mayoría de los signos de puntuación, números y etiquetas HTML); en el segundo paso, se identifican los grupos base utilizando un índice invertido de frases y usando la estructura de árbol de sufijos; en el tercer paso se realiza la combinación de los grupos base con un alto traslape, considerando una medida de similaridad binaria; los autores demuestran que su algoritmo es más rápido que los métodos estándar de agrupamiento. Las pruebas se realizaron con un conjunto de documentos construidos con los resultados de realizar consultas en un buscador de *MetaCrawler*.

En Gil-García et al. (2003), se propone una extensión al algoritmo de agrupamiento Star. Se introduce un nuevo concepto de grafo tipo estrella y también proponen dos cambios al algoritmo Star, el primero es la definición del grado de complemento, el cual considera los nodos adyacentes con una $\beta-$similaridad, que no han sido considerados en otros grupos; definiendo una manera diferente de construir el grafo tipo estrella, y el segundo cambio, es que este algoritmo resuelve el problema de la dependencia del orden de los datos y los grupos ilógicos del algoritmo Star. El procedimiento realizado dentro de esta extensión, sigue los mismos pasos que el algoritmo Star, utilizando la nueva forma de construir el grafo tipo estrella, y considerando el grado de complemento, que consiste en obtener el grado de un objeto tomando los vecinos adyacentes no incluidos en ningún agrupamiento; de esta forma se generan grupos traslapados. Este algoritmo obtiene un número pequeño de grupos, las pruebas son realizadas con diferentes colecciones de datos del TREC. El algoritmo fue comparado con el algoritmo Star obteniendo mejor tiempo

de ejecución, además de obtener menos grupos.

En Aslam et al. (2004) se desarrollan dos nuevas propuestas del algoritmo expuesto en Aslam et al. (1998), el cual considera el algoritmo de agrupamiento Star para sistemas de información estáticos (off-line) y dinámicos (on-line). Para ambas versiones, tanto estática como dinámica, los documentos se representan siguiendo el modelo de espacio vectorial, que considera la ocurrencia de las palabras en el documento, y para comparar documentos utilizan la similitud coseno. El algoritmo estático (off-line) se puede utilizar como un paso de procesamiento previo en un sistema de información estático, o como un paso de procesamiento posterior en los documentos específicos recuperados por una consulta. Como preprocesado, ayuda a los usuarios a decidir cómo explorar una base de datos de documentos de texto. Estos datos al estar agrupados son útiles para delimitar la base de datos sobre la que se pueden formular consultas. Como posprocesado, se clasifican los datos recuperados en grupos que capturan categorías de temas; los autores demuestran que el algoritmo dinámico es más complejo que el algoritmo estático, y prueban además que éste no produce una cobertura única. Por otra parte, el algoritmo dinámico soporta la inserción y eliminación de documentos. Se demuestra la eficiencia del algoritmo dinámico, utilizando 2,000 documentos de la colección de TREC-FBIS.

En Hammouda y Kamel (2004) los autores proponen dos elementos clave para el agrupamiento de documentos. El primer elemento clave es un modelo de índice de documentos, se genera un grafo de índice de documentos (DIG por sus siglas en inglés), el cual permite la construcción incremental eficiente de un índice de los documentos basándose en frases. La representación de los documentos se realiza de

la siguiente manera: se toma cada frase en el documento considerando la cantidad de términos y se le otorga un peso a cada una, en lugar de basarse solo en índices de un solo término. El DIG captura diferentes niveles de significancia de las oraciones originales, y entonces se utiliza un grafo de sufijos. El segundo elemento clave, es un algoritmo de agrupamiento de documentos incremental, basado en maximizar la rigidez de los grupos, observando cuidadosamente la distribución de similitud de documentos por pares dentro de los grupos. Para el análisis de la estructura se proponen 3 niveles de significancia dentro del documento (alta - encabezado, título, etc., media - palabra en cursiva o negrita, hipervínculos, imágenes y tablas, baja - el cuerpo del documento). Las pruebas se realizaron con tres conjuntos de datos, dos de documentos web y el tercero de noticias de USENET (acrónimo de *Users Network*) que es un sistema global de discusión en Internet.

Gil-García et al. (2005) introducen un algoritmo de agrupamiento jerárquico llamado *algoritmo compacto jerárquico*, que trata tanto datos estáticos como dinámicos. El algoritmo trabaja de manera iterativa; para determinar los nuevos grupos se utiliza una medida de similitud entre grupos. Se considera el grafo de β-similaridad sin tener en cuenta la orientación de las aristas; obteniendo un grafo no dirigido (llamado grafo no dirigido $max{-}S$) y una rutina de cobertura, que encuentra las componentes conectadas a dicho grafo, se forman conjuntos compactos. Este algoritmo puede generar grupos de diferentes formas y los grupos creados en cada nivel de la jerarquía son únicos. La diferencia entre la versión estática y la dinámica reside en que la rutina de cobertura se lleva a cabo de manera dinámica para dicha versión, esto es que en cada nivel de la jerarquía se pueden actualizar los grafos de

cobertura. Diferentes algoritmos jerárquicos aglomerativos pueden ser obtenidos mediante el algoritmo descrito, especificando una medida de semejanza diferente. Los experimentos demuestran que el algoritmo requiere menos tiempo computacional en comparación con los métodos UPGMA (Jain y Dubes (1988)) y BKM (Steinbach et al. (2000)). Para analizar su comportamiento se utilizó la colección TREC-5 con artículos publicados por la Agence France-Presse (AFP), que es la agencia de noticias más antigua en el mundo y una de las más grandes, junto con Reuters, Associated Press y EFE (que es una agencia de noticias internacional fundada en Burgos, España).

Jing et al. (2007) presentan una extensión del algoritmo k-means, para objetos de alta dimensionalidad, por ejemplo, para agrupamiento de textos. El algoritmo se llama entropía pesada de k-means (EWKM, por sus siglas en inglés) en donde se considera que el peso de una dimensión en un grupo se representa por la probabilidad de contribución de esa dimensión en la formación del agrupamiento. La entropía de las dimensiones pesadas representa la certeza de las dimensiones en la identificación de un agrupamiento. Por lo tanto, se modifica la función objetivo en el algoritmo k-means, para agregar la entropía pesada; de tal forma que permita minimizar la dispersión del agrupamiento y al mismo tiempo maximizar el peso de la entropía negativa, para estimular más dimensiones y contribuir a la identificación de los grupos. Para esto se agrega un paso adicional en el algoritmo k-means, el cual, calcula el peso de todas las dimensiones en cada agrupamiento. Los experimentos tanto en datos sintéticos como en datos reales muestran que este algoritmo genera mejores grupos que otros algoritmos de agrupamiento. Para las pruebas rea-

lizadas inicialmente se consideraron grupos de datos sintéticos, y posteriormente se utilizaron datos tomados de la universidad de California, (UCI) Machine Learning Repository.[2]; sin embargo los corpus utilizados no son traslapados.

El algoritmo Generalizado Star (GStar), propuesto por Pérez-Suárez y Medina-Pagola (2007), es una generalización del algoritmo Star. Se introduce una manera diferente de construir el grafo tipo estrella. El algoritmo GStar trabaja de manera semejante al algoritmo Star, en donde en el primer paso se calcula el grado de cada vértice, pero considerando para cada vértice v únicamente a los vértices adyacentes que tienen un grado menor que el grado de v, y se ordenan descendentemente en la lista L, posteriormente en el segundo paso se procede como en el algoritmo Star, obteniendo en la lista X los agrupamientos con traslape. Se realizan evaluaciones en colecciones de documentos estándar, tales como TREC-5 y Reuters-21578, obteniendo pequeños grupos. GStar es simple y eficiente. Se resuelve la dependencia que tiene el algoritmo Star con respecto al orden de entrada de los datos,y con la generación de grupos ilógicos y redundantes. Para los experimentos los documentos se representaron con el modelo de espacio vectorial, y los términos índice, como los lemas de las palabras; y se realizó la eliminación de palabras cerradas y los términos son pesados usando la frecuencia de los mismos.

Basándose en el algoritmo GStar se propuso el algoritmo ACONS (Alonso et al. (2007); Pérez-Suárez et al. (2008)), este es un nuevo algoritmo de agrupamiento llamado estrella condensado; ACONS es la evolución del algoritmo Star y resuelve el inconveniente observado en los algoritmos Star y Star extendido; es decir, la

[2]https://archive.ics.uci.edu/ml/datasets.php

dependencia del orden de los datos, la producción de vértices no cubiertos y la generación de grupos ilógicos y redundantes; en este algoritmo a diferencia de GStar se reduce la cantidad de grupos generados. Para lograr esto, se ejecutan los pasos que sigue el algoritmo GStar, pero construyendo el grafo tipo estrella como en ACONS; y, posteriormente se eliminan aquellos subgrafos (grupos), tales que todos sus vértices se encuentren contenidos en otros subgrafos; estos grupos son considerados grupos redundantes. La evaluación en colecciones estándares de documentos muestran que el algoritmo obtiene pequeños grupos, siendo un algoritmo simple de implementar y eficiente. Se realizan evaluaciones en colecciones de documentos estándar, tales como TREC-5 y Reuters-21578, los resultados demuestran que supera a los algoritmos Star y GStar.

Un algoritmo con una idea semejante a los anteriores, fue propuesto en Pérez-Suárez et al. (2009) llamado algoritmo de agrupamiento incremental por decisión fuerte (ICSD, por sus siglas en inglés), el cual obtiene grupos traslapados y densos, usando un grafo heurístico de cobertura, reduciendo el costo computacional al mantener de forma incremental la estructura del agrupamiento. Este algoritmo obtiene grupos traslapados mediante la cobertura del grafo de $\beta-$semejanza con grafos tipo estrella, con la diferencia de que para la obtención de los mismos solo se incluyen como satélites los vértices adyacentes al nodo centro, que tengan menor grado que el centro; con esto, el número de nodos satélites asociados a un nodo centro es menor y de este modo se reduce el traslape entre grupos. Los experimentos muestran que el algoritmo ICSD supera a otros algoritmos basados en grafos, con un mejor tiempo.

Se usaron colecciones de datos TREC-5, Reuters-21578 y TDT2[3].

En Abella-Pérez y Pagola (2010) se presenta el método IClustSeg, para la segmentación de textos por tópico y utilizando un algoritmo de agrupamiento incremental con traslape. Al ser un algoritmo incremental, permite que se vayan agregando nuevos documentos y de acuerdo a los cambios se actualizan los resultados usando la información previa. Esta aproximación propone un método de segmentación lineal por tópicos, se utiliza el modelo de espacio vectorial, y se considera cada párrafo como la unidad mínima. En primer lugar se preprocesa cada párrafo eliminando las palabras cerradas. El algoritmo tiene tres niveles: la búsqueda de cohesión por tópico, la detección de límites de segmentos de tópicos y detección de límites de segmentos de documentos. En el primer nivel, se aplica el algoritmo ICSD y los grupos traslapados obtenidos sirven de entrada al siguiente nivel. En el segundo nivel, con los grupos obtenidos se procesa cada uno de ellos utilizando una ventana, para verificar si dos párrafos adyacentes en un grupo están cerca. En el último nivel, se concatenan segmentos que al menos tengan un párrafo en común, para obtener la segmentación lineal. Para este algoritmo se hace uso de la similitud coseno para los valores de similitud. Las pruebas se realizaron con artículos tomados de las memorias del ICPR'2006 (International Conference Pattern Recognition 2006).

Otra extensión del algoritmo GStar fue reportada en Pérez-Suárez et al. (2013a), al cual denominaron OCDC (Agrupamiento con traslape basado en densidad y compacidad), a diferencia de GStar en el primer paso para ordenar la lista de posibles

[3]https://catalog.ldc.upenn.edu/LDC2001T57

centros L, OCDC considera el promedio de la densidad y compacidad entre vértices. La densidad de un vértice v es el número de vértices adyacentes a v, con un grado menor que v y la compacidad de un vértice v es el número de vértices adyacentes, cuya semejanza entre grupos de un vértice es menor a la semejanza entre grupos de v. La semejanza entre grupos de un vértice es el promedio de la semejanza entre este vértice y sus vértices adyacentes. En el segundo paso se construye la lista X de manera similar a Star; posteriormente para reducir el número de grupos se eliminan aquellos grafos que comparten todos sus satélites con otro grafo. Con esta nueva propuesta se logran resolver algunas de las deficiencias de los algoritmos anteriores, manteniendo una complejidad computacional aceptable. De acuerdo a los autores, OCDC obtiene menos grupos traslapados y con menor traslape que GStar. Para las pruebas se utilizan 5 corpus: AFP, CISI, CACM, Reuters-21578 y TDT2.

En Pérez-Suárez et al. (2013c) se presenta el algoritmo de agrupamiento con traslape OClustR (Agrupamiento con traslape basado en relevancia), el cual se basa en OCDC, pero a diferencia de OCDC obtiene la lista de posibles centros L de los grafos tipo estrella, calculando el promedio de la densidad relativa y compacidad relativa. La densidad relativa se calcula como el promedio de la densidad con respecto al número total de vértices adyacentes, y la compacidad relativa se calcula como el promedio de la compacidad entre el número total de vértices adyacentes. Posteriormente se procede como en el algoritmo OCDC. Además, OClustR construye grupos con traslape más cercanos a los traslapes reales en las colecciones generadas por otros algoritmos de agrupamiento con traslape. Las pruebas se realizaron comparando con los algoritmos Star, GStar, ACONS, DCS (Suárez et al. (2012))

entre otros, en términos de calidad de los grupos, número de grupos y tiempo de ejecución, llegando a la conclusión de que OClustR es más preciso. Los experimentos se realizaron con los corpus AFP, Reuters-21578 y TDT2.

Pérez-Suárez et al. (2013b) presentan un algoritmo que permite el traslape entre sus grupos, llamado DClustR, como una alternativa para el análisis en redes sociales, recuperación de información y bioinformática. El algoritmo DClustR es un algoritmo de agrupamiento con traslape dinámico, basado en OClustR. DClustR construye los agrupamientos de manera similar a OClustR, pero al ser un algoritmo dinámico, permite agregar o eliminar documentos sin tener que construir los agrupamientos desde el inicio, esto se logra mediante la detección de las componentes conexas, del grafo de β-similaridad, en las cuales los documentos fueron agregados o eliminados y solo se actualizan los grafos tipo estrella involucrados en estas componentes conexas. Este algoritmo se comparó contra los algoritmos GSTAR y ACONS, y se evalúo considerando la calidad de los grupos, número de grupos, traslape en los grupos y tiempo de ejecución. En general el algoritmo DClustR muestra mejores resultados que el resto de los algoritmos. La colección de documentos utilizados para la realización de los experimentos fueron AFP, Reuters-21578, TDT2, CISI y CACM.

González-Soler et al. (2015) presentan el algoritmo CUDA-DCLus, versión paralela basado en GPU del algoritmo DClustR, donde se mejora la eficiencia, se muestra un buen rendimiento en términos de eficiencia y consumo de memoria. Además, este algoritmo está diseñado exclusivamente para el procesamiento de documentos; a diferencia de DClustR; se puede fácilmente extender a otras aplicaciones. La forma

en que trabaja consiste en dividir el conjunto de datos en pequeños subconjuntos, de manera tal que cada subconjunto pueda ser almacenado en la matriz de memoria compartida; posteriormente se aplica la técnica Reduction para obtener las sumas acumulativas y se aplica el algoritmo DClustR. Se realizaron las pruebas con ambos algoritmos, con las 6 colecciones que se crearon, hasta llegar a 5000 documentos y demostrando que esta propuesta escala su velocidad a medida que el tamaño de la colección crece y que se utiliza menos memoria que su contra parte no paralela.

En Dey et al. (2016), se presenta un algoritmo de agrupamiento, el cual descubre inicialmente la frecuencia de ocurrencia de conceptos y posteriormente realiza el agrupamiento. Las palabras que co-ocurren dentro de un documento son consideradas como un concepto. El algoritmo inicia extrayendo los conceptos de los documentos y representando cada documento como un grafo de conceptos, en dicho grafo los vértices son los conceptos y las aristas unen pares de conceptos, si tienen un valor de significancia mayor que cierto umbral, donde, el valor de significancia se calcula como el número total de documentos, donde co-ocurren las palabras, dividido entre el número de documentos donde aparecen solas. Cada grafo de conceptos forma inicialmente un grupo, llamado grupo de conceptos. Posteriormente de manera iterativa se unen todos los pares de grupos de conceptos que comparten un cierto porcentaje de conceptos. Al llevar a cabo este procedimiento un grupo puede unirse por separado con diferentes grupos, creando el traslape. El procedimiento se detiene cuando ya no se pueden unir grupos de conceptos. El algoritmo se probó con un repositorio real que incluye títulos de noticias, quejas, soluciones técnicas y datos para la manufactura de automóviles.

France et al. (2016) examinan la escalabilidad de los modelos ADCLUS (ADitive CLUStering), que es un modelo aditivo que determina la similitud de dos objetos, utilizando un modelo de programación no lineal, con variables binarias. El modelo INDCLUS (INdividual Differences CLUStering) es una generalización del algoritmo propuesto para resolver el modelo ADCLUS, basado en el algoritmo que implementa a ambos, denominado MAPCLUS (MAthematical Programming CLUStering); que son técnicas que pueden ser usadas para extraer agrupamientos con traslape. El algoritmo SINDCLUS es una extensión del algoritmo ADCLUS que maneja una función objetivo cuadrática para obtener el óptimo de la misma se aplica un algoritmo de mínimos cuadrados. El algoritmo SINDCLUS no garantiza una solución simétrica de agrupamiento; es por ello que se propone una modificación llamada SYMPRES la cual garantiza la simetría de la solución; mediante un algoritmo llamado KL-Break-Out, que es un algoritmo progresivo que crea una lista de los agrupamientos que han sido cambiados recientemente. El uso de la memoria o lista de los recientes movimientos puede ser resuelto utilizando la heurística de búsqueda tabú, propuesta por Glover (1989). Se propone además una extensión del algoritmo SINDCLUS y del algoritmo SYMPRES utilizando la heurística de recocido simulado (Laarhoven y Aarts (1987)). Los autores probaron los algoritmos descritos anteriormente en una colección de documentos, haciendo uso de un conjunto de datos más grande, extraídos de la literatura sobre minería de datos y estadística; iniciando la experimentación en un conjunto de datos de tamaño pequeño-mediano (de 10 a 50 documentos); posteriormente se experimentó con conjunto de datos desde 125 hasta 500 documentos.

En Gilpin y Davidson (2017), se plantea el problema de agrupamiento jerárquico como un problema de programación lineal entera (ILP), con una función objetivo explícita, para ser optimizada globalmente. Se parte de que inicialmente cada documento conforma un grupo. Los grupos se van uniendo en cada iteración, mediante la unión de aquellos dos grupos que sean más similares, buscando optimizar la función objetivo. Encontrar el óptimo donde las variables tomen valores enteros, se convierte en un problema difícil de resolver computacionalmente, ya que el espacio de búsqueda es no convexo. Todos los algoritmos exactos que implementan dicho modelo son de orden exponencial. Para crear grupos traslapados se permite que un grupo se pueda unir de forma independiente a más de un grupo, utilizando el valor obtenido para la función objetivo. Los algoritmos jerárquicos generan un dendograma que debe cumplir las propiedades: simétrica, reflexiva y transitiva. Al relajar la propiedad transitiva, en particular, se pueden descubrir grupos con traslape. Los resultados, muestran que con pocos datos se logra encontrar los óptimos globales. Las pruebas se realizaron utilizando los conjuntos de datos fMRI (functional magnetic resonance imaging), Movie lens (corpus creado con texto e imagen para manejar diferentes interfaces), 20 newsGroups (es una colección de aproximadamente 20,000 grupos de noticias) y un conjunto artificial de datos de los cuales se conocen los grupos.

En Guo et al. (2017), se propone un algoritmo basado en grafos, se refiere a grafos de red como un super-grafo, donde cada nodo es un grafo en sí mismo. Un nodo de un super-grafo se denomina super-nodo. El agrupamiento por grafos en red busca encontrar nodos con contenido e información de estructura similares. El algoritmo parte de una representación de sus elementos como un super-grafo, el

cual contiene un número determinado de super-nodos interconectados; se calcula la semejanza entre los super-nodos, primero con aquellos conectados, es decir, donde existe una arista entre cada par de super-nodos. Las semejanzas entre los super-nodos no vinculados (no son adyacentes), se calcula mediante un recorrido aleatorio ponderado entre super-nodos. La similaridad de los super-nodos vinculados se divide en dos partes (similitud de atributos de nodos y la similitud de estructura de nodos), es necesario combinar ambas medidas de similitud. Se eligen algunos super-nodos con estructura compartida para tener una conexión más estrecha. Se realiza un ajuste iterativo y así se obtiene el agrupamiento final. Para medir la similitud del contenido del nodo, se usó la distancia de similitud coseno, considerando los atributos superpuestos entre dos super-nodos. Para calcular la semejanza entre super-nodos vinculados y además para extender la similaridad entre super-nodos no vinculados se propone un kernel de paso aleatorio atribuido (ARWK por sus siglas en inglés). Para los experimentos se utilizaron los corpus DBLP (bibliografía de ciencias de la computación), PubMed (30,000,000 de citas a artículos biomédicos) y BlogCatalog (base de datos constituida por datos de comercio electrónico).

A continuación en la tabla 3.1 se muestra una comparativa de los algoritmos descritos anteriormente, resaltando el modelo, las ventajas y las desventajas de cada uno de ellos.

Como se puede ver en la primer columna de la tabla 3.1, en las últimas 5 décadas se han desarrollado algoritmos de agrupamiento con traslape para documentos, los cuales pueden ser clasificados en algoritmos jerárquicos, basados en grafos y particionales. En la columna de *ventajas* se puede observar que la mayoría

de éstos algoritmos han atacado los problemas presentados en tres direcciones, primero mejorar el comportamiento computacional de los mismos, segundo tratar de no depender del orden de los datos de entrada y por último mejorar la calidad de los grupos obtenidos. En la columna *desventajas* puede observarse que a la mayoría de los algoritmos no se les puede especificar el número de grupos a formar; algunos de ellos no garantizan la convergencia al óptimo global y manejan super-grafos que son complejos a la hora de su manipulación. De ahí la necesidad de proponer un nuevo método para agrupar documentos con traslape, que obtenga mejor calidad en los agrupamientos y al que se le pueda especificar el número de grupos a crear.

Tabla 3.1: Síntesis de los algoritmos del estado del arte

Algoritmo	**Metodología**	**Ventajas**	**Desventajas**
Arabie et al. (1981)	Para el modelo ADCLUS se ha desarrollado el algoritmo MAPCLUS, el cual asume a n objetos que van a ser agrupados como datos de entrada, implementando para resolver el modelo un algoritmo de optimización con función objetivo no lineal, sin restricciones y el dominio de las variables en $\{0,1\}^n$.	Manejan una matriz de peso donde el usuario puede definir un pequeño número de grupos.	Solamente han sido probados en un dominio particular; con comentarios hechos en desayunos.
Aslam et al. (1998)	Desarrolla un método a partir de grafos para obtener grupos traspalados; el algoritmo propuesto maneja la cantidad de documentos a agrupar de forma estática y de forma dinámica.	El algoritmo propuesto para resolver el modelo tiene un comportamiento en tiempo polinomial a partir de los datos de entrada.	No se tiene un control adecuado de los grupos a crear.
Zamir y Etzioni (1998)	Se crean grupos que consideran resúmenes por extracción del documento completo, devueltos por los motores de búsqueda en la Web.	Es más rápido que los algoritmos estándar de agrupamiento, ya que de cada documento se almacena un fragmento del mismo. Los grupos obtenidos son tan buenos como los grupos creados usando el documento completo.	Este algoritmo debe ser probado con resúmenes obtenidos utilizando la técnica de abstracción; la cual genera resúmenes extrayendo la idea esencial de cada documento, igual que lo harían los seres humanos.
Gil-García et al. (2003)	Es una extensión del algoritmo de agrupamiento Star que introduce un nuevo concepto de grafo tipo estrella y por consiguiente se obtienen diferentes grupos de tipo estrella.	Este algoritmo resuelve el problema de la dependencia del orden de los datos y de los grupos ilógicos que se obtienen con el algoritmo de agrupamiento Star original.	No permite procesar grandes conjuntos de datos. No existe un control en el número de grupos creados.
Aslam et al. (2004)	Es el mismo proceso del algoritmo de agrupamiento Star para sistemas de información estáticos (off-line) y dinámicos (on-line).	El algoritmo dinámico soporta la inserción y eliminación de documentos.	No existe un control en el número de grupos creados.
Hammouda y Kamel (2004)	Se genera un grafo de índice de documentos, el cual permite la construcción incremental de un índice de los documentos basándose en frases, enfatizando la eficiencia y generando diferentes niveles de significancia.	Es un algoritmo de agrupamiento de documentos incremental, basado en maximizar la rigidez de los grupos.	Se requiere una mejora en la exactitud del método. Solo ha sido aplicado a documentos web.
Gil-García et al. (2005)	Algoritmo iterativo que une las aristas desconectadas, por medio de un promedio de grupos, utilizando una medida de similitud entre grupos.	El algoritmo requiere menor tiempo computacional que los métodos dinámicos para conjuntos de datos.	No existe un control en el número de grupos creados.
Jing et al. (2007)	Una extensión del algoritmo k-means, para objetos de alta dimensionalidad.	Simultáneamente minimiza la dispersión del grupo y maximiza el peso de la entropía negativa.	La sensibilidad en la inicialización de los grupos por ser una extensión del algoritmo k-means.
Pérez-Suárez y Medina-Pagola (2007)	Es una generalización del algoritmo Star	Se resuelve la dependencia que tiene el algoritmo Star con respecto al orden de entrada de los datos.	No existe un control en el número de grupos creados.
Alonso et al. (2007); Pérez-Suárez et al. (2008)	Es una extensión del algoritmo GStar que reduce el número de grupos generados.	Se resuelve la dependencia que tiene el algoritmo Star con respecto al orden de entrada de los datos.	No existe un control en el número de grupos creados.

Pérez-Suárez et al. (2009)	Es un algoritmo de agrupamiento incremental, el cual obtiene grupos traslapados y densos.	Reduce el costo computacional.	No existe un control en el número de grupos creados.
Abella-Pérez y Pagola (2010)	Se segmentan los textos por tópico y se utiliza un algoritmo de agrupamiento incremental con traslape.	Se obtienen segmentos más cohesivos, incrementándose significativamente su rendimiento.	No se probó con grandes conjuntos de datos.
Pérez-Suárez et al. (2013a)	Es una extensión del algoritmo de agrupamiento GStar.	Se resuelven algunas deficiencias de los algoritmos anteriores.	No existe un control en el número de grupos creados.
Pérez-Suárez et al. (2013c)	Algoritmo de agrupamiento con traslape basado en relevancia.	Construye grupos con traslape más cercanos a los traslapes reales en las colecciones generadas por otros algoritmos de agrupamiento con traslape.	No existe un control en el número de grupos creados.
Pérez-Suárez et al. (2013b)	Algoritmo de agrupamiento con traslape dinámico.	Construye grupos con traslape más cercanos a los traslapes reales en las colecciones generadas por otros algoritmos de agrupamiento con traslape. Permite que se agreguen más documentos.	No existe un control en el número de grupos creados.
González-Soler et al. (2015)	Versión paralela del algoritmo DClustR.	Se mejora la eficiencia, se muestra un buen rendimiento en términos de eficiencia y consumo de memoria.	No existe un control en el número de grupos creados.
Dey et al. (2016)	Es un algoritmo de agrupamiento que descubre inicialmente la frecuencia de ocurrencia de conceptos semánticos y posteriormente realiza el agrupamiento.	Construyen los conceptos basado en la co-ocurrencias de palabras para determinar la similitud semántica entre estas co-ocurrencias.	No existe un control en el número de grupos creados.
France et al. (2016)	Se examina la escalabilidad de los modelos ADCLUS e INCLUS y el comportamiento del algoritmo MAPCLUS.	Son técnicas que pueden ser usadas para extraer agrupamientos con traslape a partir de datos semejantes.	Los algoritmos que implementan a dichos modelos no garantizan la convergencia al óptimo global.
Gilpin y Davidson (2017)	Plantea el problema de agrupamiento jerárquico como un problema de programación lineal entera (ILP), con una función objetivo explícita, para ser optimizada globalmente.	Se demuestra que con pocos datos se logra encontrar los óptimos globales.	No existe un control en el número de grupos creados.
Guo et al. (2017)	Se parte de una representación de sus elementos como un super-grafo, el cual contiene un número determinado de super-nodos interconectados.	Formula un nuevo problema de agrupamiento de grafos en red, donde cada grafo esta sujeto a relaciones interconectadas y el objetivo de la agrupación es encontrar un grupo de grafos que comparten contenido y estructuras similares.	No existe un control en el número de grupos creados. Maneja super-grafos que son complejos a la hora de su manipulación.

Capítulo 4. Método Propuesto

A partir lo discutido en el capítulo 3, en este capítulo se introduce un método para agrupar documentos con traslape, en el cual se pueda especificar el número de grupos a formar.

Es por ello que se propone un método de agrupamiento con traslape basado en el algoritmo k-means traslapado con peso (WOKM), que hasta donde se sabe no ha sido utilizado para agrupar documentos. Es seleccionado el algoritmo WOKM porque hasta el momento es el algoritmo de agrupamiento con traslape al que se le puede especificar el número de grupos, que ha obtenido los mejores resultados (Aroche-Villarruel et al. (2014)).

El algoritmo WOKM presenta el problema de la inicialización de los centros de toda la familia de los algoritmos k-means; lo que hace necesario desarrollar un método de inicialización de los centroides; por lo que también se propone un método que utiliza el punto de transición (PT), para seleccionar los k centros iniciales para WOKM.

Para la representación del corpus, se utiliza el modelo de espacio vectorial; que consiste en representar cada palabra del vocabulario, por su frecuencia de aparición; esto quiere decir que el vocabulario está compuesto por todas las palabras del corpus completo, esto representa las columnas en el modelo de espacio vectorial y las filas representan a cada documento del corpus.

Se parte de la representación del corpus y se obtiene el punto de

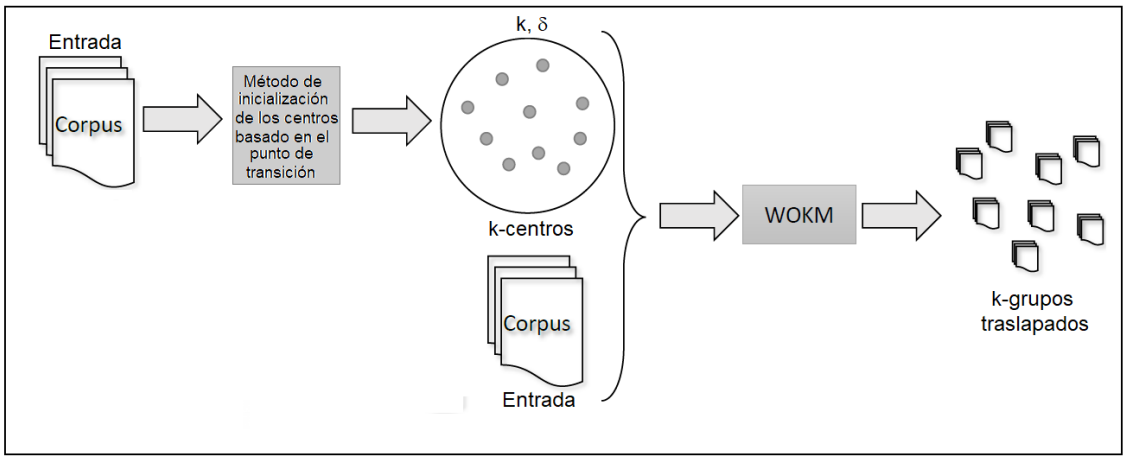

Figura 4.1: k-means traslapado con peso usando el punto de transición

transición del mismo, considerando como hipótesis del que al trabajar con las palabras más relevantes del vocabulario de todo el corpus, se reduce el número de palabras relevantes del vocabulario a utilizar. En un paso posterior se ordenan los documentos a partir de las palabras relevantes alrededor del PT, desde los documentos que poseen más palabras, hasta los que poseen menor número de palabras relevantes. Se toman los k primeros documentos como centros. Como parámetros de entrada al algoritmo WOKM se consideran los centros, el corpus y el parámetro δ; obteniendo finalmente los k grupos traslapados. La figura 4.1 muestra el proceso que lleva a cabo el procedimiento descrito.

4.1. Método de inicialización de los centroides basado en el punto de transición

Se han discutido diversos métodos para encontrar los centroides iniciales de cada grupo; porque de esto depende el comportamiento de los algoritmos de la familia k-means, se propone representar a cada documento dentro del corpus con las palabras más representativas de todo el corpus; es decir las palabras alrededor del punto de transición

(PT), por lo que surge la siguiente pregunta ¿si se disminuye el número de palabras para representar cada documento, se disminuye el tiempo de ejecución para calcular los centroides, y se obtiene mayor precisión? A continuación se propone un nuevo método basado en el PT.

Se parte de que en Urbizagástegui (1999) se representa a un documento a partir de las palabras que se encuentran alrededor del PT. Profundizando en lo que se planteó anteriormente en esta investigación se modifica la forma en que se obtiene el PT por cada documento. Como se necesita obtener los k centroides del corpus se propone por primera vez calcular el punto de transición no por cada documento, sino considerando el corpus completo. Posteriormente son seleccionados aquellos documentos que tengan mayor número de palabras alrededor del PT de todo el corpus. Queda claro que cada documento de todo el corpus va a ser representado por aquellas palabras alrededor del PT del corpus completo. Esto provoca que siempre se van a considerar el mismo número de palabras para representar a cada documento; es decir la matriz asociada a la representación vectorial del corpus va a considerar a cada documento con una menor cantidad de palabras, que si se consideran las palabras alrededor del PT de cada documento por separado.

Considerando la ecuación 4.1 l_1 se calcula como la cantidad de palabras de frecuencia 1 en todo el corpus. Además, el PT divide a las palabras en dos conjuntos, el conjunto de las palabras de alta frecuencia y el conjunto de las palabras de baja frecuencia, se propone una ventana alrededor del valor del PT y se consideran t palabras dentro de dicha ventana. Se tomarán aquellos k documentos que tengan dichas palabras con mayor frecuencia. Esto se ha considerado bajo la hipótesis de que estas palabras van a permitir generar buenos centros, porque como se plantea en Urbizagástegui (1999) considerando

las palabras representativas de cada documento se obtiene mayor precisión en los resultados, que considerando todas las palabras de cada documento.

$$PT = \frac{\sqrt{1 + 8l_1} - 1}{2} \tag{4.1}$$

Una vez obtenido el PT para todo el corpus, se ordenan las palabras del corpus completo, desde las más frecuentes hasta las menos frecuentes. En el trabajo de Urbizagástegui se demuestra que las t palabras alrededor del PT son las que deben indexar a cada documento del corpus. En esta investigación se seleccionan las t palabras que deben indexar a cada documento del corpus completo.

Se comienza seleccionando aquellos documentos que contienen a las t palabras; posteriormente se buscan los documentos que contienen a las $t - 1$ palabras y así sucesivamente hasta llegar a los documentos que tengan una sola palabra. Esta forma de selección de los documentos permite escoger los k centroides; donde cada centroide representa a un documento.

En caso de que el número de documentos que tienen las palabras elegidas sea mayor a k, se consideran aquellos k documentos con mayor número de palabras. Por la forma en que se van seleccionando los documentos no es posible obtener menor cantidad de centros que el valor de k, ya que se parte de que el número de documentos siempre es mayor o igual que k, por lo tanto, el número k coincide con el número de documentos del corpus.

La entrada del método para obtener los centroides usando el PT es el corpus ($\mathfrak{D}$), el número de centroides (k) y la ventana (t). La salida es el conjunto de centroides (C). Se calcula el número de palabras de frecuencia 1 en el corpus. Se determina el PT del corpus completo; se seleccionan aquellas palabras que se encuentran alrede-

dor del PT, las cuales se consideran en el conjunto tPT. Para cada documento que pertenece al corpus se verifica si contiene las palabras del conjunto tPT, de ser así dicho documento va a formar parte del conjunto de $Candidatos$. A continuación se ordenan los documentos del conjunto $Candidatos$ descendentemente por el número de palabras pertenecientes a tPT. Finalmente, se seleccionan los k primeros documentos del conjunto $Candidatos$. El siguiente pseudocódigo 1 muestra la implementación de cómo se crean los centroides a partir del PT. El procedimiento que se sigue es el que se mencionó anteriormente.

Entrada: Corpus $\mathfrak{D}$, número de centroides k, ventana t
Salida : Conjunto de centroides C
Inicio
  $Candidatos \leftarrow \emptyset$
  $l_1 \leftarrow$ # de palabras de frecuencia 1 en $\mathfrak{D}$
  $PT \leftarrow (\sqrt{1+8l_1}-1)/2$
  $tPT \leftarrow \{p \mid \forall p \in \mathfrak{D} \wedge freq(p) \in [PT - t/2, PT + t/2]\}$
  Para $d \in \mathfrak{D}$ **hacer**
    Si $tPT \subseteq d$ **entonces**
      $Candidatos \leftarrow Candidatos \cup d$
    FinSi
  FinPara
  Ordenar $Candidatos$ descendentemente por el número de palabras pertenecientes a tPT
  $C \leftarrow k$ primeros documentos de $Candidatos$
  Regresa C
Fin

Algoritmo 1: Pseudocódigo para obtener los centroides usando el PT

4.2. k-means traslapado con peso para documentos usando el punto de transición como método de inicialización

En esta sección se discute el método propuesto k-means traslapado con peso, usando el punto de transición (WOKMTP por sus siglas en inglés).

En la literatura, el algoritmo k-means ha sido extendido para generar grupos traslapados, específicamente los algoritmos k-means traslapado (OKM) Guillaume (2008), k-means traslapado con peso (WOKM) y k-medoides traslapado (OKMED) Cleuziou (2010). En Aroche-Villarruel et al. (2014) se realizó una comparación experimental entre los algoritmos OKM, OKMED y WOKM. A partir de dicho estudio, los autores concluyen que el algoritmo WOKM tiene los mejores resultados, en términos de la calidad de los grupos, sobre los algoritmos OKM y OKMED. Debido a este resultado, en la presente investigación se considera al algoritmo WOKM para construir los grupos.

Por otro lado, es conocido que el algoritmo k-means tiene problemas con respecto a la inicialización de los centros, ya que los grupos generados pueden ser diferentes, dependiendo de los centros iniciales; este problema es heredado al algoritmo WOKM; es por ello que se propone utilizar el PT para generar los k centroides y de esta forma se propone el método WOKMTP.

A continuación se presenta la modificación realizada al algoritmo WOKM para considerar documentos.

Sea $\chi = \{d_1, d_2, ..., d_n\}$ un conjunto de documentos, donde cada documento es descrito por p palabras. Además, sea $\mathfrak{C} = \{C_1, C_2, ..., C_k\}$ el conjunto de grupos traslapados que se construirán y $\mathfrak{M} = \{m_1^{PT},$

$m_2^{PT}, ..., m_k^{PT}\}$ los centros de cada uno de los grupos, respectivamente.

La función objetivo a minimizar es:

$$\varphi(\{C_i\}_{i=1}^{k}) = \sum_{d_i \in \chi} \sum_{v=1}^{p} \gamma_{i,v}^{\delta} |d_{i,v} - \phi_v(d_i)|^2 \tag{4.2}$$

donde $\mathfrak{C}$ es un cubrimiento de χ, considerando que cada documento d_i debe pertenecer al menos a un grupo. La ecuación 4.2, $\phi_v(d_i)$ denota la "imagen" de d_i, en términos de la palabra v que está definida por la combinación de los prototipos ($m_c^{PT} \in \mathfrak{M}$) al que el documento d_i pertenece, y es obtenida como:

$$\phi_v(d_i) = \frac{\sum_{m_j^{PT} \in A_i} \lambda_{j,v}^{\delta} m_{j,v}}{\sum_{m_j^{PT} \in A_i} \lambda_{j,v}^{\delta}},$$

donde $A_i = \{m_j | d_i \in C_j\}$ lo que significa que A_i es el conjunto de centros, a los que el documento d_i pertenece. En otras palabras, $\phi_v(d_i)$ es un promedio con peso de los centros de los grupos a los que d_i pertenece.

La (4.2) $\gamma_{i,v}$ es calculada por medio de la siguiente ecuación:

$$\gamma_{i,v} = \frac{\sum_{m_j \in A_i} \lambda_{j,v}}{|A_i|}$$

donde $\lambda_{j,v}$ es el peso asociado a la característica v en el grupo j, tal que $\forall j, \sum_{v=1}^{p} \lambda_{j,v} = 1$. Además, el parámetro $\delta > 1$ que se encuentra en 4.2 sirve para regular la influencia del peso local.

Dado k centros tomados de χ o generados aleatoriamente, y dado

el peso inicial asociado a la palabra v se toma como $\lambda_{j,v} = 1/p$ para $v = 1, ..., p$ y $j = 1, ..., k$. La optimización de la función objetivo 4.2 es alcanzada por una ejecución iterativa de los pasos de multi-asignamiento, cálculo de centros y cálculo de pesos presentados en la sección 2.2 del marco teórico.

El método propuesto considera como entrada el corpus ($\mathfrak{D}$), el número de grupos (k). Como salida los grupos traslapados ($\mathfrak{C}$). Se determina el PT considerando todo el corpus. Posteriormente se aplica el algoritmo WOKM considerando la inicialización de los centros obtenidos con el método 1, y de manera iterativa se aplican los pasos de multi-asignamiento, actualización de los centros y actualización de los pesos hasta que converge al óptimo; esto se tiene en el pseudocódigo de esta propuesta (ver pseudocódigo 2).

Entrada: Corpus $\mathfrak{D}$, k número de grupos a encontrar
Salida : Grupos traslapados $\mathfrak{C}$

Inicio
    `/* Punto de transición */`
    Obtención de centros con PT (Método 1)
    `/* Algorimto WOKM */`
    Inicializar los k centros obtenidos de PT
    repetir
        Multi-asignamiento
        Actualización de los centros de los grupos
        Actualización de pesos
    hasta que *converge*;
    Regresa $\mathfrak{C}$
Fin

Algoritmo 2: Pseudocódigo del método WOKMTP

Capítulo 5. Resultados experimentales

En este capítulo se presenta la evaluación del método propuesto (WOKMTP); el cual ha sido organizado de la siguiente manera; primeramente se discute la preparación de los diferentes experimentos; se presentan los resultados experimentales obtenidos por el método propuesto WOKMTP y se compara contra uno de los algoritmos del estado del arte más reciente y exitoso (OClustR), además con el algoritmo WOKM y con el algoritmo OKM. Posteriormente se evalúa la efectividad del método de inicialización de centros iniciales propuesto, usando el PT y se compara contra el mismo método WOKM, pero utilizando otros dos métodos de inicialización de centros que se han reportado en la literatura (armónico y *canopy*). Y finalmente se evalúa el método propuesto usando diferentes medidas de similitud de documentos.

5.1. Preparación de experimentos

Los corpus que se consideraron para los experimentos, se tomaron del repositorio *Mulan repository: A Java Library for Multi-Label Learning*[1] Tsoumakas et al. (2011) dicho corpora es una biblioteca de Java, de código abierto de conjuntos de datos multi-etiquetados. En particular se consideraron aquellos que contenían textos y no se tomaron en cuenta el resto de los conjuntos de datos. En los corpus considerados en este repositorio las palabras cerradas fueron eliminadas, no se eliminan números, ni se considera la lematización de cada palabra.

Para los experimentos, se eligieron 12 corpus, los cuales son de diferentes dominios, entre los que se pueden mencionar: páginas web, leyes, medicina, información bibliográfica y correos electrónicos, entre otros; además, contienen desde 978 hasta 87,856 documentos. Por otro lado el número de palabras en cada corpus varía desde 500 hasta 37,187 y finalmente, el número de grupos varía entre 26 y 983. En la tabla 5.1 se presenta el número de documentos, palabras y grupos de los corpora seleccionados.

Los resultados de los agrupamientos fueron evaluados usando la medida *FBcubed* Amigó et al. (2009), la cual es una medida externa, especialmente diseñada para evaluar algoritmos de agrupamiento con

[1] http://mulan.sourceforge.net/datasets.html

traslape. Para los algoritmos OKM y WOKM se utilizó la misma inicialización aleatoria. Para el método WOKMTP la inicialización fue generada utilizando el punto de transición.

Tabla 5.1: Resumen de los corpora utilizados en los experimentos realizados.

Corpus	**No. Documentos**	**No. Palabras**	**No. Grupos**
Arts	7,485	23,146	26
Bibtex	7,395	1,836	159
Bookmarks	87,856	2,150	208
Business	5,505	21,924	30
Delicious	16,105	500	983
Education	6,030	27,534	33
Enron	1,702	1,001	53
EUR-Lex (DC)[2]	17,407	5,000	412
EUR-Lex (SM)[3]	19,348	5,000	201
Health	4,558	30,605	32
Medical	978	1,449	45
Science	6,428	37,187	40

Cabe hacer mención que las implementaciones de OKM, WOKM y OClustR fueron provistas por los autores de Aroche-Villarruel et al. (2014); Pérez-Suárez et al. (2013c). Además, la implementación de KHM se ejecutó desde la herramienta de procesamiento de discursos de Matlab: Speech Processing Toolbox[4]. La implementación de *canopy* se ejecutó desde el entorno para el análisis del conocimiento Weka[5]. Para los experimentos se utilizaron 3 servidores:

1. Procesador Intel Xeon E5540 2.67 GHz con 4 núcleos físicos y 12 Gbytes de memoria RAM corriendo el sistema operativo Linux

[2]Directory Codes
[3]Subject Matters
[4]https://la.mathworks.com/matlabcentral/fileexchange/39015-speech-processing-toolbox
[5]https://www.cs.waikato.ac.nz/ml/weka/

con la distribución de Ubuntu.

2. Procesador Intel Xeon CPU E5-2680 2.50GHz con 24 núcleos físicos y 128 GB de memoria RAM corriendo el sistema operativo Windows Server.

3. Procesador Intel(R) Xeon Phi(TM) CPU 7250 @ 1.40GHz con 68 núcleos físicos y 195 Gbytes de memoria RAM corriendo el sistema operativo Linux con la distribución de Ubuntu.

En los tres servidores se realizaron los diferentes experimentos, en el primero se desarrollaron los experimentos con la distancia Euclidiana, el segundo y tercer servidor fueron utilizados en paralelo por la cantidad de experimentos a desarrollar. Los corpus Bibtex, Bookmarks, Eurlex-dc y Eurlex-sm, consumen mucho tiempo computacional.

5.2. Resultados experimentales de WOKMTP

En esta sección, se discuten los resultados obtenidos al comparar los algoritmos de agrupamiento OKM, WOKM, OClustR y WOKMTP. Se realiza la comparación del método WOKMTP con estos algoritmos, ya que son los que ofrecen mejores resultados en términos de la calidad de los grupos creados, esto fue presentado en las secciones 3 y 4.2. Para la mayoría de los corpus la representación utilizada

de los documentos es el modelo de espacio vectorial. En la Tabla 5.2 se muestran los resultados de los experimentos realizados, en las columnas se tienen los algoritmos antes mencionados y las filas son los corpus mostrados en la Tabla 5.1.

La métrica utilizada para evaluar la calidad del agrupamiento construido es *FBcubed*, lo cual permite evaluar los diferentes algoritmos, ya que se usa la misma métrica. Para realizar el agrupamiento se uso la distancia Euclidiana. Por cada corpus (fila), se ha resaltado en negrita el resultado más alto correspondiente al algoritmo (columna) en términos de la métrica *FBCubed*, para un mayor valor de *FBCubed* se obtienen los mejores resultados.

Tabla 5.2: Resultados de la métrica FBcubed de la evaluación de los algoritmos con traslape para documentos OKM, WOKM, OClustR y WOKMTP sobre el corpora de la Tabla 5.1.

Corpus	**OKM**	**WOKM**	**OClustR**	**WOKMTP**
Arts	0.2541	0.2442	0.0307	**0.3702**
Bibtex	0.1639	0.1640	**0.1671**	0.1347
Bookmarks	0.2134	0.2131	0.1812	**0.3570**
Business	0.6599	0.6532	0.0087	**0.8226**
Delicious	0.0666	0.0686	0.0309	**0.6518**
Education	0.2022	0.2023	0.0288	**0.3891**
Enron	0.5658	0.5704	0.1018	**0.6098**
EUR-Lex (DC)	0.2099	0.2064	0.0351	**0.3449**
EUR-Lex (SM)	0.2613	0.2708	0.0231	**0.4387**
Health	0.4149	0.3389	0.0228	**0.5471**
Medical	0.3535	0.3588	**0.5265**	0.3589
Science	0.1899	0.1670	0.0257	**0.2398**

Los resultados mostrados en la Tabla 5.2 se obtuvieron colocando el valor de k como el número de grupos de cada corpus, dicho

número se encuentra en la última columna de la Tabla 5.1. Además, el valor para el parámetro δ en WOKM se determinó en $\delta = 2$, esto por sugerencia de Aroche-Villarruel et al. (2014). Adicionalmente, el algoritmo OClustR requiere como parámetro el valor de β, para la obtención del grafo de β-similaridad. Para determinar este valor, se realizaron pruebas, con diferentes valores entre 0.1 y 0.5 (como se sugiere en Pérez-Suárez et al. (2013c)), con incrementos de 0.1, y por lo tanto se eligió el valor $\beta = 0.4$, ya que fue el valor que obtuvo mejores resultados para *FBcubed*. El valor de la ventana utilizada para el método WOKMTP fue de $t = 10$, porque al realizarse diferentes experimentos con diversos valores de t, con este se obtuvo los mejores resultados.

Los resultados obtenidos en la Tabla 5.2 muestran que el método propuesto WOKMTP supera en la mayoría de los diferentes corpus utilizados a los algoritmos contra los que se compara. También, es importante mencionar que solo en dos corpus no se ha logrado mejorar los resultados obtenidos.

El PT se utiliza para seleccionar las palabras más representativas de cada documento. Cada documento es representado con las palabras alrededor del PT, esto ha permitido generar buenos centroides. Esto ofrece mejores resultados que al inicializar aleatoriamente los centros, para utilizarse en el algoritmo WOKM. En este experimento pode-

mos apreciar que cuando se utiliza el método propuesto basado en el PT para determinar los centros iniciales de WOKM (ver la columna WOKMTP), se tienen mejores resultados que usando únicamente una inicialización aleatoria de los centros(columna WOKM).

El método propuesto WOKMTP obtiene el mismo número de grupos, que el número de clases en las que se encuentran agrupados los documentos de los corpus mostrados en la Tabla 5.1, siendo esto una ventaja sobre otros algoritmos en los cuales se puede llegar a tener un gran número de grupos. Por ejemplo, el algoritmo OClustR no requiere que se especifique el número de grupos a formar, en la Tabla 5.3 se muestra el número de grupos generados para cada corpus. En dicha tabla es posible observar que el número de grupos creados por el algoritmo OClustR es muy grande con respecto al número real de clases en cada corpus (comparando con la última columna de la Tabla 5.1).

5.3. Comparación de WOKMTP contra WOKM y otros métodos de inicialización

En esta sección, se evalúa el método propuesto (WOKMTP) cambiando el método de inicialización de centroides. En los experimentos se usaron los métodos de inicialización de centroides KHM y *Canopy* los cuales se explicaron en las secciones 2.4.1 y 2.4.2, y se denotan

Tabla 5.3: Grupos obtenidos por el algoritmos OClustR.

Corpus	**Clases**
Arts	3884
Bibtex	2957
Bookmarks	24429
Business	3536
Delicious	1086
Education	3141
Enron	497
EUR-Lex (DC)	1306
EUR-Lex (SM)	1297
Health	2489
Medical	88
Science	2091

como HWOKM y CWOKM respectivamente.

Tabla 5.4: Comparación de los resultados de la métrica FBcubed de la evaluación de los algoritmos con traslape para documentos HWOKM, CWOKM y WOKMTP sobre el corpora de la Tabla 5.1.

Corpus	**HWOKM**	**CWOKM**	**WOKMTP**
Arts	0.3397	0.3589	**0.3702**
Bibtex	**0.2052**	0.1408	0.1347
Bookmarks	0.2331	0.2458	**0.3570**
Business	0.4677	0.5174	**0.8226**
Delicious	0.5294	0.6010	**0.6518**
Education	0.2428	0.3016	**0.3891**
Enron	0.6005	0.3577	**0.6098**
EUR-Lex (DC)	0.2894	0.2584	**0.3449**
EUR-Lex (SM)	0.3798	0.4191	**0.4387**
Health	0.5057	0.5358	**0.5471**
Medical	**0.3861**	0.3589	0.2600
Science	0.2058	0.2101	**0.2398**

Los resultados mostrados en la Tabla 5.4 se obtuvieron colocando el valor de k para el número de clases, dependiendo de cada corpus a obtener, dicho número se encuentra en la última columna de la Tabla 5.1.

De los resultados obtenidos en la Tabla 5.4, se puede notar que

en general el método WOKMTP propuesto obtiene mejores resultados que el resto de los métodos. La columna de WOKMTP muestra en negrita los resultados obtenidos para la mayoría de los corpus.

El método WOKMTP en los corpus Bibtex y Medical no mejoró al método HWOKM; analizando la representación del corpus proporcionado por Mulan Repository se observó que cada palabra del vocabulario es representada con $\{0, 1\}$; lo que quiere decir que la palabra no pertenece al documento o pertenece al documento, a diferencia de los otros corpus que utilizan para representar a los documentos el modelo de espacio vectorial. Esto debe ser probado con otros corpus que tengan la misma representación.

5.4. Resultados de los experimentos realizados con diferentes medidas de similitud

En la sección 5.1, se realizaron los experimentos utilizando la distancia Euclidiana para la obtención de los grupos. En esta sección se muestran con diferentes medidas de similitud entre documentos a saber: Coseno, coeficiente ó índice de Sørensen-Dice y coeficiente de Jaccard.

En la tabla 5.5 se muestran los resultados en términos de la medida *FBcubed* de los métodos HWOKM, CWOKM y WOKMTP utilizando la similitud Coseno para todos los corpus.

Tabla 5.5: Comparación de los resultados de la métrica FBcubed de la evaluación de los algoritmos con traslape para documentos HWOKM, CWOKM y WOKMTP utilizando la medida de similitud Coseno sobre el corpora de la Tabla 5.1.

Corpus	**HWOKM**	**CWOKM**	**WOKMTP**
Arts	0.277984	0.276974	**0.349411**
Bibtex	0.118588	0.117251	**0.158209**
Bookmarks	**0.090589**	0.073812	0.074302
Business	0.333279	0.644423	**0.717694**
Delicious	0.599726	0.600220	**0.670864**
Education	0.213168	0.336103	**0.350930**
Enron	0.107514	0.185090	**0.217156**
Eurlex-DC	0.058196	0.089904	**0.094149**
Eurlex-SM	0.205133	0.174937	**0.221394**
Health	0.436666	0.539573	**0.547438**
Medical	0.214834	**0.298040**	0.256327
Science	0.223173	0.251685	**0.255027**

Analizando los resultados obtenidos puede apreciarse que no existen grandes diferencias, entre la distancia Euclidiana y la similitud Coseno. En general el método WOKMTP obtiene los mejores resultados.

Se realizan otros experimentos utilizando el coeficiente ó índice de Sørensen-Dice. Éstos resultados se muestran en la tabla 5.6.

La última tabla 5.7 muestra los resultados obtenidos utilizando el coeficiente de Jaccard como medida para obtener los grupos.

El método WOKMTP obtiene mejores resultados que todos los otros métodos, independientemente de la medida de similitud utilizada.

Tabla 5.6: Comparación de los resultados de la métrica FBcubed de la evaluación de los algoritmos con traslape para documentos HWOKM, CWOKM y WOKMTP utilizando el coeficiente ó índice de Sørensen-Dice sobre el corpora de la Tabla 5.1.

Corpus	HWOKM	CWOKM	WOKMTP
Arts	0.299741	0.308491	**0.343506**
Bibtex	**0.297582**	0.152480	0.199252
Bookmarks	0.170446	0.254475	**0.274484**
Business	0.615932	0.768498	**0.841291**
Delicious	0.620428	0.634581	**0.692824**
Education	0.335114	0.314757	**0.352708**
Enron	0.672425	0.674762	**0.688024**
Eurlex-DC	0.148531	0.198752	**0.201585**
Eurlex-SM	0.194752	0.204400	**0.204525**
Health	0.521975	0.529014	**0.539439**
Medical	0.235712	**0.250839**	0.252841
Science	0.425875	0.314587	**0.552114**

Tabla 5.7: Comparación de los resultados de la métrica FBcubed de la evaluación de los algoritmos con traslape para documentos HWOKM, CWOKM y WOKMTP utilizando el coeficiente de Jaccard sobre los corpus de la Tabla 5.1.

Corpus	HWOKM	CWOKM	WOKMTP
Arts	0.241080	0.298147	**0.324085**
Bibtex	**0.160988**	0.084752	0.108271
Bookmarks	0.082110	0.088015	**0.119555**
Business	0.587991	0.701401	**0.857413**
Delicious	0.230538	0.574834	**0.622782**
Education	0.303803	0.323739	**0.363521**
Enron	0.270686	0.306911	**0.688024**
Eurlex-DC	0.058347	0.074191	**0.109028**
Eurlex-SM	0.197326	0.206203	**0.211572**
Health	0.544547	0.541591	**0.550979**
Medical	0.235712	0.309401	**0.311035**
Science	0.253997	0.253516	**0.276071**

Capítulo 6. Conclusiones

En el presente trabajo de investigación se cumplió el objetivo general al desarrollar un método de agrupamiento con traslape para documentos, basado en el algoritmo k-means traslapado, cuyos resultados superan a los algoritmos de agrupamiento con traslape del estado del arte.

Se desarrolló un método de agrupamiento traslapado para documentos basado en el algoritmo k-means traslapado con peso (WOKM), que incluye un método para la inicialización de centros iniciales basado en el concepto de punto de transición.

Para cumplir con el objetivo general se evaluaron los métodos y los algoritmos de agrupamiento con traslape para documentos del estado del arte, para determinar su rendimiento en términos de la calidad de los grupos creados.

Se compararon además diferentes métodos de inicialización de los centros iniciales para el algoritmo k-means traslapado con peso (inicialización aleatoria, k-means armónico y algoritmo *canopy*).

A lo largo del proceso de experimentación se ha podido observar que el método WOKMTP supera a los reportados en la literatura para el agrupamiento traslapado de documentos.

Otra conclusión importante, es que el método propuesto para buscar centros iniciales, permite a WOKM obtener mejores grupos traslapados de documentos que, los métodos comúnmente utilizados para k-means.

Se ha demostrado con los experimentos desarrollados que el método propuesto obtiene buenos resultados independientemente de la medida de similaridad utilizada.

Finalmente, los resultados experimentales se han realizado con corpus variados en número de documentos, términos y clases o grupos. Es importante destacar que independientemente de la diversidad antes mencionada, el método propuesto obtiene en general buenos resultados.

6.1. Aportaciones

Las aportaciones que se tienen a partir de la investigación desarrollada son:

- Un método de agrupamiento con traslape para documentos, al que se le especifica el número de grupos a construir.
- Un método para la búsqueda de centros iniciales para agrupa-

miento con traslape para documentos de la familia k-means.

6.2. Trabajo a futuro

Con base en los resultados obtenidos durante la investigación desarrollada, se propone el siguiente trabajo a futuro:

- Desarrollar un algoritmo de agrupamiento con traslape no basado en k-means, que use el método propuesto para obtener centros iniciales basado en el PT.
- Proponer una mejora al método WOKMTP incluyendo diferentes ventanas alrededor del PT.
- Proponer un método de búsqueda de centros iniciales para algoritmos de agrupamiento particionales, basado en el PT considerando datos que no sean documentos.

6.3. Publicaciones

Derivado de la investigación llevada a cabo durante los estudios de doctorado se tienen:

- Towards the Construction of a Clustering Algorithm with Overlap Directed by Query (2017). Beatriz Beltran, Darnes Vilariño Ayala, David Pinto, Rodolfo Martínez. **Research in Computing Science** Vol. 145

- Desarrollo de un algoritmo de agrupamiento con traslape dirigido por consulta (2018). Beatriz Beltrán Martínez, Darnes Vilariño Ayala. **Capítulo de libro:** Tópicos actuales en la Ingeniería del lenguaje y del conocimiento. Ed. Montiel & Soriano Editores S. A. de C. V.

- Un algoritmo de agrupamiento con traslape para documentos (2019). Beatriz Beltrán, Darnes Vilariño. **Capítulo de libro:** Avances en tecnologías del lenguaje y el conocimiento. Ed. Montiel & Soriano Editores S. A. de C. V.

- Comparison of Clustering Algorithms in Text Clustering Tasks (2020). Rafael Gallardo García, Beatriz Beltrán, Darnes Vilariño, Claudia Zepeda, Rodolfo Martínez. **Computación y Sistemas** 24(2)

- Survey of Overlapping Clustering Algorithms (2020). Beatriz Beltrán, Darnes Vilariño. **Computación y Sistemas** 24(2)

- K-means based Method for Overlapping Document Clustering (2020). Beltrán, Beatriz, Vilariño, Darnes, Martínez-Trinidad, José Fco., Carrasco-Ochoa, J.A., Pinto, David. **Journal of Intelligent & Fuzzy Systems**, Vol. 39, No. 2 (JCR)

Bibliografía

Abella-Pérez, R. and Pagola, J. (2010). An Incremental Text Segmentation by Clustering Cohesion. In *Handling Concept Drift in Adaptive Information Systems: Importance, Challenges and Solutions - HaCDAIS 2010*, volume 6419, pages 261–268.

Aggarwal, C. C. and Reddy, C. K. (2013). *Data Clustering: Algorithms and Applications*. Chapman & Hall/CRC, 1st edition.

Alonso, A. G., Suárez, A. P., and Medina-Pagola, J. E. (2007). ACONS: A new algorithm for clustering documents. In *Progress in Pattern Recognition, Image Analysis and Applications, 12th Iberoamericann Congress on Pattern Recognition, CIARP 2007, Valparaiso, Chile, November 13-16, 2007, Proceedings*, pages 664–673.

Amigó, E., Gonzalo, J., Artiles, J., and Verdejo, F. (2009). A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information Retrieval*, 12(4):461–486.

Arabie, P., J., C. D., Wayne, D., and Jerry, W. (1981). Overlapping Clustering: A New Method for Product Positioning. *Journal of Marketing Research*, 18(3):310–317.

Aroche-Villarruel, A. A., Carrasco-Ochoa, J. A., Martínez-Trinidad, J. F., Olvera-López, J. A., and Pérez-Suárez, A. (2014). Study of overlapping clustering algorithms based on kmeans through fbcubed metric. In Martínez-Trinidad, J. F., Carrasco-Ochoa, J. A., Olvera-Lopez, J. A., Salas-Rodríguez, J., and Suen, C. Y., editors, *Pattern Recognition*, pages 112–121, Cham. Springer International Publishing.

Aslam, J., Pelekhov, K., and Rus, D. (1998). Static and Dynamic Information Organization with Star Clusters. In *Proceedings of the Seventh International Conference on Information and Knowledge Management*, CIKM '98, pages 208–217, New York, NY, USA. ACM.

Aslam, J. A., Pelekhov, E., and Rus, D. (2004). The Star Clustering Algorithm for Static and Dynamic Information Organization. *Journal of Graph Algorithms and Applications*, 8(1):95–129.

Cangelosi, A. (2001). Evolution of communication and language using signals, symbols, and words. *IEEE Transactions on Evolutionary Computation*, 5(2):93–101.

Cleuziou, G. (2010). Two Variants of the OKM for Overlapping Clustering. In Guillet, F., Ritschard, G., Zighed, D. A., and Briand, H., editors, *EGC (best of volume)*, volume 292 of *Studies in Computational Intelligence*, pages 149–166. Springer.

Dey, L., Ranjan, K., Verma, I., and Naskar, A. (2016). A Semantic Overlapping Clustering Algorithm for Analyzing Short-Texts. In *Rough Sets*, pages 470–479, Cham. Springer International Publishing.

Dice, L. R. (1945). Measures of the Amount of Ecologic Association between Species. *Ecology*, 26(3):297–302.

France, S. L., Chen, W., and Deng, Y. (2016). ADCLUS and INDCLUS: Analysis, experimentation, and meta-heuristic algorithm extensions. *Advances in Data Analysis and Classification*, 11(2):371–393.

Gan, G., Ma, C., and Wu, J. (2007). *Data Clustering: Theory, Algorithms, and Applications (ASA-SIAM Series on Statistics and Applied Probability)*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.

Gil-García, R., Badía-Contelles, J. M., and Pons-Porrata, A. (2003). Extended Star Clustering Algorithm. In *Progress in Pattern Recognition, Speech and Image Analysis, 8th Iberoamerican Congress on Pattern Recognition, CIARP 2003, Havana, Cuba, November 26-29, 2003, Proceedings*, pages 480–487.

Gil-García, R., Badía-Contelles, J. M., and Pons-Porrata, A. (2005). Dynamic Hierarchical Compact Clustering Algorithm. In Sanfeliu, A. and Cortés, M. L., editors, *Progress in Pattern Recognition, Image Analysis and Applications*, pages 302–310, Berlin, Heidelberg. Springer Berlin Heidelberg.

Gilpin, S. and Davidson, I. (2017). A flexible ILP formulation for hierarchical clustering. *Artificial Intelligence*, 244(C):95–109.

Glover, F. W. (1989). Tabu Search - Part I. *INFORMS J. Comput.*, 1(3):190–206.

González-Soler, L. J., Pérez-Suárez, A., and Chang-Fernández, L. (2015). Algoritmo incremental de agrupamiento con traslape para el procesamiento de grandes colecciones de datos. *GECONTEC: Revista Internacional de Gestión del Conocimiento y la Tecnología*, 3(2):1–12.

Guillaume, C. (2008). An extended version of the k-means method for overlapping clustering. *2008 19th International Conference on Pattern Recognition*, pages 1–4.

Guo, T., Wu, J., Zhu, X., and Zhang, C. (2017). Combining Structured Node Content and Topology Information for Networked Graph Clustering. *ACM Trans. Knowl. Discov. Data*, 11(3):29:1–29:29.

Hammouda, K. M. and Kamel, M. S. (2004). Efficient phrase-based document indexing for Web document clustering. *IEEE Transactions on Knowledge and Data Engineering*, 16(10):1279–1296.

Jackson, D. A., Somers, K. M., and Harvey, H. H. (1989). Similarity Coefficients: Measures of Co-Occurrence and Association or Simply Measures of Occurrence? *The American Naturalist*, 133(3):436–453.

Jain, A. K. and Dubes, R. C. (1988). *Algorithms for Clustering Data*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Jing, L., Ng, M. K., and Huang, J. Z. (2007). An entropy weighting k-means algorithm for subspace clustering of high-dimensional sparse data. *IEEE Transactions on Knowledge and Data Engineering*, 19(8):1026–1041.

Khanmohammadi, S., Adibeig, N., and Shanehbandy, S. (2017). An Improved Overlapping K-means Clustering Method for Medical Applications. *Expert Syst. Appl.*, 67(C):12–18.

Laarhoven, P. J. M. and Aarts, E. H. L. (1987). *Simulated Annealing: Theory and Applications*. Kluwer Academic Publishers, USA.

Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press, USA.

McCallum, A., Nigam, K., and Ungar, L. H. (2000). Efficient clustering of high-dimensional data sets with application to reference matching. In *KDD '00: Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 169–178, New York, NY, USA. ACM Press.

Pérez-Suárez, A., Martínez-Trinidad, J. F., Carrasco-Ochoa, J. A., and Medina-Pagola, J. E. (2008). A new graph-based algorithm for clustering documents. In *Workshops Proceedings of the 8th IEEE International Conference on Data Mining (ICDM 2008), December 15-19, 2008, Pisa, Italy*, pages 710–719.

Pérez-Suárez, A., Martínez-Trinidad, J. F., Carrasco-Ochoa, J. A., and Medina-Pagola, J. E. (2009). A New Incremental Algorithm for Overlapped Clustering. In *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications, 14th Iberoamerican Conference on Pattern Recognition, CIARP 2009, Guadalajara, Jalisco, Mexico, November 15-18, 2009. Proceedings*, pages 497–504.

Pérez-Suárez, A., Martínez-Trinidad, J. F., Carrasco-Ochoa, J. A., and Medina-Pagola, J. E. (2013a). A New Overlapping Clustering Algorithm Based on Graph Theory. In *Advances in Artificial Intelligence*, pages 61–72, Berlin, Heidelberg. Springer Berlin Heidelberg.

Pérez-Suárez, A., Martínez-Trinidad, J. F., Carrasco-Ochoa, J. A., and Medina-Pagola, J. E. (2013b). An algorithm based on density and compactness for dynamic overlapping clustering. *Pattern Recognition*, 46(11):3040–3055.

Pérez-Suárez, A., Martínez-Trinidad, J. F., Carrasco-Ochoa, J. A., and Medina-Pagola, J. E. (2013c). OClustR: A new graph-based algorithm for overlapping clustering. *Neurocomputing*, 121:234–247.

Pérez-Suárez, A. and Medina-Pagola, J. E. (2007). A clustering algorithm based on generalized Stars. In Perner, P., editor, *Machine Learning and Data Mining in Pattern Recognition*, pages 248–262, Berlin, Heidelberg. Springer Berlin Heidelberg.

Steinbach, M., Karypis, G., and Kumar, V. (2000). A comparison of document clustering techniques. *Proceedings of the International KDD Workshop on Text Mining*.

Suárez, A. P., Trinidad, J., Carrasco-Ochoa, J. A., and Medina-Pagola, J. E. (2012). A dynamic clustering algorithm for building overlapping clusters. *Intell. Data Anal.*, 16:211–232.

Tan, P.-N., Steinbach, M., and Kumar, V. (2005). *Introduction to Data Mining*. Addison Wesley, us ed edition.

Tanimoto, T. T. (1958). *An elementary mathematical theory of classification and prediction by T.T. Tanimoto*. International Business Machines Corporation New York.

Tsoumakas, G., Spyromitros-Xioufis, E., Vilcek, J., and Vlahavas, I. (2011). Mulan: A Java Library for Multi-Label Learning. *Journal of Machine Learning Research*, 12:2411–2414.

Urbizagástegui, A. R. (1999). Las posibilidades de la ley de Zipf en la indización automática. *B3 : Revista Electrónica de Bibliotecología*.

Wierzchon, S. and Kłopotek, M. (2018). *Modern Algorithms of Cluster Analysis*, volume 34 of *Studies in Big Data*. Springer.

Zamir, O. and Etzioni, O. (1998). Web Document Clustering: A Feasibility Demonstration. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 46–54, New York, NY, USA. ACM.

Zhang, B., Hsu, M., and Dayal, U. (2000). K-Harmonic Means - A Spatial Clustering Algorithm with Boosting. In *Proceedings of the First International Workshop on Temporal, Spatial, and Spatio-Temporal Data Mining-Revised Papers*, TSDM '00, pages 31–45, London, UK, UK. Springer-Verlag.

Zhang, B., Hsu, M., Dayal, U., and Data, M. (1999). K-Harmonic Means - A Data Clustering Algorithm. *Hewlett Packard Research Laboratory Technical Report*.

Zipf, G. K. (1949). *Human Behaviour and the Principle of Least Effort*. Addison-Wesley.