Universidad de Granada

Tratamiento Semántico de la Información Recuperada de Internet con Fines de Consulta y Exploración.

Úrsula Torres Parejo.

TRATAMIENTO SEMÁNTICO DE LA INFORMACIÓN RECUPERADA DE INTERNET CON FINES DE CONSULTA Y EXPLORACIÓN

Úrsula Torres Parejo

Departamento de Ciencias de la Computación e Inteligencia Artificial
ETS de Ingenierías Informática y de Telecomuncicación
Universidad de Granada

Trabajo de Investigación Tutelada.
Máster en Soft Computing y Sistemas Inteligentes.
Tutores: Miguel Delgado Calvo-Flores y Amparo Vila Miranda.

Índice general

Capítulo 1

INTRODUCCIÓN

Este trabajo se centra en la Recuperación de Información contenida en la Web y de la manipulación y representación de esta información, especialmente de los campos textuales, mediante formas intermedias atractivas para el usuario, con el fin de facilitar la consulta y la exploración.

El objetivo es crear una herramienta de visualización que permita al usuario navegar a través del contenido de una base de información y realizar consultas de una forma sencilla y eficiente.

Para ello, se han realizado experimentos reales sobre una base de datos en la que se ha almacenado información sobre noticias extraídas de Internet. Para la realización de estos experimentos, se ha seleccionado una muestra pequeña de tuplas, sobre la cual hemos probado el funcionamiento de los métodos que se expondrán a lo largo del desarrollo de este trabajo.

1.1. Planteamiento del Problema

Con el creciente uso de Internet y las nuevas Tecnologías, cada vez es mayor la cantidad de información que se acumula y que no llega a transformarse en información útil para el usuario.

Este problema se daba originalmente con datos estructurados, almacenados en bases de datos relacionales. Posteriormente, con el acceso a Internet, este problema se repite para datos no estructurados o semi-estructurados, almacenados en diferentes sistemas de información (Mar08).

No existe una forma de procesar y visualizar toda la información de manera eficiente y siempre queda información a la que el usuario es incapaz de acceder.

Con el fin de permitir el acceso a la información, los sistemas basados en *Tagging* permiten al usuario categorizar las fuentes de información mediante las denominadas etiquetas o *tags*, con el fin de poder recuperarlas posteriormente, lo cual es un método bastante cuestionable (ver subsección (2.1.2)),.

A su vez, estos sistemas han popularizado una herramienta de visualización de texto denominada "*Tag Cloud*", en la que habitualmente lo que se visualiza son las etiquetas más frecuentemente asignadas por los usuarios, pero que a veces también se construye a partir de términos extraídos del texto. La utilización de esta herramienta crece exponencialmente y son cada vez más los sitios web que la incorporan en sus páginas, a pesar de las numerosas y conocidas deficiencias tanto (ver subsecciones (2.1.3) y (2.1.8).

Resumiremos estas deficiencias en las siguientes:

- **Con respecto a la identificación del contenido.** La *Tag Cloud* muchas veces lleva a concepciones erróneas sobre el contenido de la información (ver subsección (2.1.5)
- **Con respecto a la semántica.** Los términos más frecuentes no son los más discriminantes y los términos más populares no tienen porqué ser los más relevantes, sin embargo, son estos los que aparecen representados en la *Tag Cloud* (ver subsección (2.1.3), que además tampoco permite inferir relaciones entre conceptos.
- **Con respecto a la teoría.** No está definida matemáticamente, aunque algunos autores han presentado un modelo formal para describirla (ver (Xex09)), pero este modelo no se ajusta a la *tag cloud* tradicional que encontramos en la web. Esta falta de definición acarrea numerosos problemas teóricos. A pesar de ello, podemos ver su utilización con fines analíticos.
- **Con respecto al método.** No existe un método estándar para su generación.

1.2. Objetivos

El objetivo principal es facilitar la búsqueda de información al usuario a través de un entorno amigable, que le resulte atractivo y fácil de usar y que facilite la exploración visual de la información permitiendo la identificación del contenido.

Por otro lado, se pretende incrementar la cantidad y calidad de la información recuperada a través de este entorno y solventar los principales problemas identificados en la *tag cloud* y en su estructura subyacente, estimando que la *tag cloud* es de las herramientas más utilizadas en la Web para estos propósitos, si no la que más.

Para ello, pretendemos aprovechar el llamativo diseño de la *Tag Cloud*, que tanto éxito ha tenido y sigue teniendo en la Web y solventar sus deficiencias, uniéndolo por un lado a la funcionalidad de la denominada Estructura-AP, desarrollada en el curso de otra investigación en nuestro grupo (Ver sección (2.2)) y por otro lado, creando lo que se denomina la "Estructura Multitérmino"(ver subsección (4.4.2)).

La estructura multitérmino es una forma de procesar la información que puede verse empleada y visualizada en forma de *tag cloud* multitérmino en algunos pocos sitios web, pero que sin embargo, carece de un modelo matemático y de un método estándar de extracción. Nosotros la hemos definido matemáticamente (ver sección (3)) y hemos realizado una modificación del Algoritmo "APriori" (ver subsección(3.3.2)) para su generación.

Veremos como mediante la estructura multitérmino y la estructura-AP resolveremos los principales problemas presentes en la generación y visualización de la *tag cloud* tradicional, o *tag cloud* que aparece con más frecuencia representada en la Web y también cuándo utilizar la estructura-AP para procesar el texto y cuándo la estructura multitérmino (ver sección (4), subsección (4.4.1)). En realidad, ambas son formas diferentes de procesar el texto, según el interés del usuario en cada momento, pero realizaremos su visualización de la misma manera, en forma de *tag cloud*.

Centraremos los objetivos con respecto a la mejora de la *tag cloud* en los siguientes aspectos:

- **Identificación del contenido.** Permitir la identificación del contenido mediante el uso de componentes multitérmino.
- **Semántica.** Permitir la discriminación entre términos mediante sus términos relacionados y facilitar el reconocimiento de las relaciones entre conceptos. Esto también se consigue permitiendo el uso de componentes multitérmino. Como sabemos, los términos más frecuentes son los peores discriminantes, véase el caso de "sistema"; si permitimos componentes multitérmino, se podría tener "sistema operativo" y "sistema nervioso", con lo

que se habrá inferido semántica y permitido relacionar conceptos como “sistema” y “nervioso” o “sistema” y “operativo”.

- **Teoría.** Presentar una base matemática sólida para definir los procedimientos mediante los cuáles se procesará y manipulará la información.
- **Método.** Presentar un método estándar para la extracción de la información que será procesada y representada en el entorno de visualización y navegación.

1.3. Desarrollo del trabajo

- En el capítulo 2, veremos los antecedentes tanto de la *tag cloud* como de la estructura-AP.

  Para la *tag cloud* se comenzará con una revisión bibliográfica, luego se hablará de los sistemas basados en *tagging* comentando sus aspectos positivos y negativos y luego se introducirá la *tag cloud* tal como la conocemos, comentando sus características, ventajas e incovenientes, nociones sobre el diseño, análisis de los métodos existentes para la extracción de etiquetas, etc. Posteriormente se hablará de de la *tag cloud* monotérmino “versus” la *tag cloud* multitérmino. Se terminará explicando algunas técnicas de *clustering* y herencia en la *tag cloud*, la *data cloud* o *tag cloud* propuesta por (Kou09a) para datos estructurados y se expondrán las conclusiones alcanzadas tras esta revisión.

  A continuación, se introducirá mediante un breve resumen la estructura-AP, para lo cual será necesario establecer el concepto de conjunto-AP y algunas de sus propiedades. Se comentarán algunas propiedades de la estructura-AP y se terminará hablando del acoplamiento de conjuntos de términos con estructuras-AP.

- En el capítulo 3, se establecerán los modelos matemáticos de la estructura-AP ponderada y la estructura multitérmino ponderada.

  Será necesario establecer una ponderación de la estructura-AP para poder visualizar ésta en forma de *tag cloud*. Se empezará definiendo los *itemsets* ponderados que compondrán los conjuntos-AP ponderados, para definir posteriormente la estructura-AP ponderada y algunas de sus propiedades y

se verá la visualización de ésta en forma de *tag cloud*.

Como de la estructura multitérmino no hay antecedentes, se empezará definiendo los conjuntos monotémino y multitérmino. Los conjuntos multitérmino porque serán los que compongan las estructura multitérmino y los conjuntos monotérmino porque son los que componen la estructura monotérmino, que es la estructura subyacente bajo la *tag cloud* que vemos visualizada en la web o *tag cloud* monotérmino y que también nos parece importante definirla, puesto que la usaremos para la comparación con la estructura-AP y estructura multitérmino y hasta el momento carece de definición.

Para entender los conjuntos monotérmino y multitérmino, se establecerá la definición de componente monotérmino y componente multitérmino. Tras definir los conceptos de estructura monotérmino y multitérmino, se comentarán algunas propiedades de los conjuntos y estructuras multitérmino.

Posteriormente y al igual que para la estructura-AP, con el fin de poder visualizar la estructura monotérmino y multitérmino en forma de *tag cloud*, se definirá un modelo para la estructura monotérmino y multitérmino ponderada. Para esta última se establecerá la definición de secuencia de elementos ponderada o *item-seq* ponderada, ya que estas *item-seqs* serán las que compongan los conjuntos multitérmino ponderados. Se verá una visualización tanto de la estructura monotérmino ponderada como de la estructura multitérmino ponderada en forma de *tag cloud* y finalmente se analizarán algunas propiedades de la estructura multitérmino ponderada y el acopliento de conjuntos multitérmino con estructuras multitérmino, considerando índices para la medida de este acoplamiento.

- En el capítulo 4 se verá un ejemplo práctico de cómo, a partir de la información contenida en una base de datos, se generarían la estructura-AP, la estructura monotérmino y la estructura multitérmino, con sus respectivas ponderaciones y cómo sería la visualización de estas tres estructuras en forma de *tag cloud*.

  Se comentarán paso a paso los métodos de extracción de la estructura-AP y la estructura multitérmino y se realizará una comparación entre ambas y de ambas con la estructura monotérmino. Veremos cuándo será mejor utilizar

una u otra estructura y cómo ambas mejoran la estructura monotérmino.

Finalmente se verá un ejemplo del cálculo de los índices de acoplamiento de un conjunto con ambas estructuras.

- En el capítulo 5 o último capítulo, se explicarán las conclusiones llevadas a cabo a partir de este trabajo y se realizará una serie de propuestas para trabajos futuros.

Capítulo 2

ANTECEDENTES

2.1. *Tag Cloud*

Aunque la mayoría de los autores y usuarios usa la terminología *tag cloud* para referirse a una visualización del texto en forma de nube de palabras con distintos tamaños de fuente indicando la popularidad de la palabra, muy pocos conocen el verdadero significado de las tag clouds. ¿Se extraen del texto? ¿Cuándo decimos popularidad nos referimos a la frecuencia o número de ocurrencias de la palabra en el texto?

Para contestar estas preguntas pasaremos a distinguir entre *word cloud* y *tag cloud*, aunque todos los autores usan la terminología *tag cloud* de manera indistinta para referirse a ambos conceptos.

Tag significa etiqueta o marca. Por lo tanto, la *tag cloud* es una nube de etiquetas o marcas. Las etiquetas pueden componerse de una o más palabras y son asignadas por el usuario para categorizar las fuentes de información encontradas en la Web. Los términos en una *tag cloud* no son extraídos del texto, si no que son marcas libres que han asignado anteriormente los usuarios a las fuentes de información. Las *tag clouds* permiten la navegación, por lo que las etiquetas trabajan como enlaces a las fuentes de información o páginas web marcadas. Cuando hablamos de frecuencia de las *tags*, hablamos del número de veces que esa etiqueta ha sido asignada, por lo que esta frecuencia también se conoce como popularidad. Así, lo que se representa en una *tag cloud* son las etiquetas más populares.

Sin embargo, muchos trabajos que hablan de la extracción de *tag clouds*, en realidad se refieren a la extracción de *word clouds* (Término introducido por Viégas y Wattenberg (Vie08)).

El objetivo de una *word cloud* es analizar el texto, permitiendo a los usuarios el examen de documentos secuenciales de una forma rápida. Lo que se muestra en este caso es la frecuencia de las palabras en un pasaje de texto en lugar de las etiquetas de un sitio web.

También en la *word cloud* los términos pueden componerse de una, dos o más palabras. Aunque, tradicionalmente tanto en la *word cloud* como en la *tag cloud* la tendencia es incluir términos de sólo una palabra.

Recientemente, han surgido gran número de herramientas para generar *word clouds*, bien de un texto proporcionado por el usuario o bien de cualquier sitio cuya dirección web facilita el usuario. Ejemplos de estas herramientas son ManyEyes (Vie07), Wordle (Fei09) , Word Clouds (Cla09), TagCrowd (Ste06) o Tag Cloud Generator (Nab09).

Existe una tercera nomenclatura en relación con las nubes de palabras: la *data cloud*.

Este término es introducido por Koutrika et al. (Kou09a), (Kou09b) para referirse a nubes construidas sobre bases de datos estructuradas a partir de los resultados obtenidos con la búsqueda por palabras clave, para guiar a los usuarios en el refinamiento de esas búsquedas.

Pero otros utilizan este término con acepción diferente. En la Web podemos encontrar referencias a la *data cloud* como nube de datos, es decir, como nube donde las etiquetas son dígitos y no palabras.

Por último, haremos referencia a la *text cloud*, aunque son más las nomenclaturas derivadas de este tipo de visualizaciones.

Hay quien utiliza el término *text cloud* como otra forma de llamar a la *word cloud*. Mayoritariamente la terminología *text cloud* se utiliza para referirse a una visualización de la frecuencia de las palabras que constituyen un texto presentada en forma de lista ponderada. Esta técnica es conocida por su uso para resaltar las palabras en los discursos políticos.El propósito de la *text cloud* es principalmente la comprensión del texto, mientras que el de la *tag cloud* es el acceso o la navegación a través de la información.

Podemos encontrar de forma resumida las diferencias entre *tag cloud, word cloud, data cloud* y *text cloud* en (30097)

Una vez definidas las diferentes terminologías, en este trabajo se utilizará indistintamente el término *tag cloud* por ser el comúnmente aceptado para cualquiera de estos tipos de estructuras de visualización, aunque basaremos nuestro interés en la *word cloud* y también se hablará de la *data cloud* tal como la definieron Koutrika et al. y de los aspectos clásicos de la *tag cloud* en el sentido estricto del término.

2.1.1. Revisión Bibliográfica

Todas las referencias que aparecen a continuación se encuentran recogidas en la bibliografía al final de este trabajo (5). La no especificación de páginas en algunas de estas referencias se debe a que, a pesar de pertenecer a una colección de comunicaciones o conferencias, han sido publicadas de forma aislada, bien en la página web de algún congreso o bien en la página web del mismo autor.

Revisión Bibliográfica de la *Tag Cloud*

Aunque el aspecto básico de la *tag cloud* (combinación de palabras con distintos tamaños) estriba desde hace más de 90 años, la *tag cloud* con el propósito que hoy le damos (representación visual de una colección de texto) tiene su primera aparición en el año 1976 en un experimento llevado a cabo por el psicólogo social Stanley Milgram. El experimento consistió en pedirle a la gente que nombrara puntos de interés en París con el fin de crear un mapa colectivo de la ciudad usando diferentes tamaños de fuente para mostrar la frecuencia en que se mencionó cada lugar (Mil76).

Casi 20 años después estos diagramas se creaban mediante ordenador, pero de modo ficticio en una novela de Douglas Coupland en 1995. En esta novela uno de los personajes hacía un programa para seleccionar al azar frases de su diario electrónico, frases que se mostraban en el libro.

En 1997, el programador Jim Flanagan, tomando las ideas de Milgram y Douglas, creó un script en Perl para añadir términos de búsqueda a su página web, variando el tamaño de los términos.

Sobre el 2001 las *tag clouds* empezaron a usarse en el mundo de las finanzas, la revista Fortune representó en un mapa el paisaje corporativo con masas circulares de texto que mostraban las 500 mayores corporaciones en el mundo. Cada nube representaba las compañías de cada país (For01).

En 2002 *Flickr* (Lud04) que es un sitio web bastante popular basado en compartir imágenes entre los usuarios, empezó a necesitar una forma de clasificar o etiquetar estas imágenes. Tomando la idea de Flanagan, creó una *tag cloud* que mostraba la popularidad de las etiquetas usando distintos tamaños de fuente (Vie08).

Por otro lado, existen numerosos sitios web que usaban el etiquetado incluso antes de que existiera, como *Yahoo* o *Open Directory (dmoz.org)*,que usaban técnicas semi-automáticas basadas en el control del vocabulario.

En 2005 Shaw (Sha05) representó la *tag cloud* como un grafo donde las etiquetas son representadas por nodos y las relaciones de similaridad entre nodos como ejes.

En 2006 Bielenberg y Zacher (Bie05) presentaron la *tag cloud* con forma circular. El tamaño de la fuente y su distancia al centro representaban la importancia de la etiqueta, pero la distancia entre etiquetas no representaba su similaridad.

En la actualidad, numerosos sitios web, entre ellos *Del.icio.us* (Sch03) presentan un sistema de etiquetado caracterizado porque permite que cualquier usuario pueda etiquetar cualquier recurso web de forma ciega, es decir, no puede ver las etiquetas asignadas por otros usuarios para la misma fuente de información.

Revisión Bibliográgica de la *Word Cloud*

El concepto de *word cloud* no es tan antiguo como el concepto de *tag cloud*. De hecho, seguramente nació a partir de ésta, aprovechando la forma de visualización, pero empleando esta forma en términos extraídos del texto en lugar de en etiquetas asignadas libremente, mediante un mismo criterio de selección: la frecuencia.

En esta línea tenemos algunos trabajos. Por ejemplo, en 2007, Kuo et al. (Kuo07) describieron una aplicación que se servía de *word clouds* para resumir los resultados de las consultas que se realizan sobre una base de datos biomédica.

En 2008, Viégas y Wattenberg (Vie08) criticaron la *tag cloud* cuando se usa con fines analíticos, proponiendo la *word cloud* como herramienta de análisis alternativa.

Estos mismos autores, en 2009 (Vie09), presentan una herramienta de visualización de texto en forma de nube, pero donde los términos se extraen directamente

del texto, es decir, presentan una herramienta para la generación de *word clouds* que actualemente es muy popular y usada por gran número de usuarios (Fei09).

También en 2009, Van Ham et al. (VH09) presentan una técnica para visulizar *word clouds* en forma de grafo.

En la actualidad, son muchas ya las herramientas que podemos encontrar en la Web para la generación de *word cloud*, algunas de las cuales ya han sido citadas anteriormente.

Revisión Bibliográfica sobre *cluestering* y herencia

En 2006 Begelman et al. (Beg06) afirmaron que la búsqueda es sólo el primer paso de la exploración y que el usuario continúa explorando, lo que es posible únicamente si las etiquetas están agrupadas en *clusters*.

También se estudió agrupar las etiquetas en clusters en una tesis en 2006 en la Universidad de Taiwan (Yu06).

Schmitz (Sch06) presentó una ontología de facetas derivada de Flickr con modelos basados en supuestos y métodos probabilísticos con la que también se recuperaban imágenes fácilmente.

Hsieh et al. (Hsi06) presentaron el concepto de herencia entre etiquetas, que permitía obtener mayor información para la recomendación y la búsqueda.

También otros autores introdujeron el mismo año la herencia entre etiquetas en sus trabajos, como es el caso de Heymann et al. (Hey06)

En 2007 Grahl et al. (Gra07) combinan los conceptos de *clustering* y herencia en las *folksonomías* (Se hablará de este término en la siguiente sección).

En la actualidad, sitios como *RawSugar* (D'S07) especifican relaciones de herencia entre etiquetas.

Flirckr utiliza *clusters* que proporcionan etiquetas populares agrupadas junto a sus etiquetas relacionadas.

HubLog (VI05) permite a los usuarios la exploración entre las etiquetas relacionadas de *Del.icio.us*.

Netr.it (VdB09) construye una red, extensible de forma manual, de las co-ocurrencias de las etiquetas personales de cada usuario de *Flirckr*.

Semantic Cloud (Sie09) genera una *tag cloud* semántica a través de técnicas *clustering* y ofrece la posibilidad de recuperación en forma de herencia.

Sin embargo, la aplicación de las técnicas de *clustering* dentro de las *folksonomías* continúa presentando retos importantes y autores como Papadopoulos et al. (Pap10) continúan investigando alternativas para la aplicación de estas técnicas, como sería la construcción de un esquema gráfico basado en *clustering* para la identificación de etiquetas relacionadas.

Revisión Bibliográfica sobre *Tag Clouds* Multitérmino

No existen muchos trabajos sobre *tag clouds* multitérmino.

En 2006, Panunzi et al. (Pan06) realizaron un estudio para evaluar la diferencia entre una técnica de extracción de palabras clave de un sólo término y otra que permitía extraer palabras clave de más de un término, demostrando que las palabras de más de un término se consideraban más descriptivas y permitían la identificación del contenido.

En 2007, Don et al. (Don07) señalaron algunas desventajas de las herramientas utilizadas en los sitios sociales como Delicious, indicando que estas se solventaban con uso de "multitérminos".

En 2008, Agili et al. (Agi08) utilizaron la técnica de Panunzi et al. (Pan06) para la extracción de palabras clave multitérmino.

Ese mismo año, Watters (Wat08) creó una herramienta que permitía el empleo de "multitérminos" en lugar de palabras simples.

En 2009, Vander Wal (VW05) en su *blog "Off de Top: Folksonomy entries"* realiza un análisis de las etiquetas multitérmino, resaltando muchas de sus ventajas frente a las etiquetas simples.

2.1.2. Sistemas Basados en Etiquetado o *Tagging* y *Folksonomía*

La definición de *folksonomía* según Vander-Wal (VW05) es: "Una *folksonomía* es el resultado del etiquetado libre y personal de información y objetos

(cualquier cosa con una URL) en Internet para la propia recuperación. El etiquetado es desempeñado en un ambiente social (compartido y abierto a otros). La acción de etiquetar se realiza por la persona que consume la información".

Los sistemas basados en etiquetado o *tagging* permiten a los usuarios categorizar las fuentes de información web mediante etiquetas o *tags* (palabras clave o *keyword* elegidas libremente), con el fin de encontrar posteriormente dichas fuentes de información. Parece ser un modo natural de los usuarios para clasificar objetos y una forma atractiva de descubrir nuevo material.

Mientras que los sistemas tradicionales de búsqueda, generan los resultados de la consulta basándose en métodos de *ranking*, por ejemplo asignando a los términos de sus bases de datos un peso de acuerdo a su frecuencia, los sistemas basados en etiquetado generan los resultados de la consulta recuperando todas los fuentes de información que previamente han sido etiquetadas con esa marca y ordenando los resultados en base a distintos factores, como pueden ser la actualidad con que se ha etiquetado, el número de veces que se le ha asignado esa etiqueta o el número de usuarios que han marcado ese recurso.

Tagging es también un proceso de indexación social, donde los usuarios pueden compartir las etiquetas y las fuentes de información construyendo un índice social de etiquetas o *tags* llamado *folksonomía*, así, las etiquetas pueden ser asignadas manualmente o mediante indexación automática. La indexación automática se produciría al compartir una fuente de información que ya ha sido etiquetada por otros usuarios y que, al formar parte de la *folksonomía*, se indexaría automáticamente a las etiquetas con que otros usuarios hubieran marcado previamente esa fuente de información.

Una *folksonomía* permite que cualquier usuario pueda acceder a cualquier fuente de información web previamente etiquetada, basándose en dos paradigmas principales: IF (*Information Filtering*) y IR (*Information Retrieval*)

En IF los usuarios juegan un papel pasivo, esperando que el sistema mande información a través de él de acuerdo con algún perfil previamente definido. Las herramientas de etiquetado social (*social bookmarking*) permiten el acceso IF desde que un usuario puede subscribirse a un conjunto de etiquetas específico vía *RSS/Atom Syndication* y estar alertado cuando un nuevo recurso sea indexado con este conjunto de etiquetas.

Por ejemplo, usamos IF cuando nos suscribimos a una lista con el fin de que el sistema nos mande información cuando algo nuevo aparezca. Supongamos que un

usuario está interesado en “viajes”, “gastronomía” y “música” y marca estos campos en un formulario como campos de su interés para que le envíen información a su correo. La información que reciba habrá sido obtenida mediante IF y “viajes”, “gastronomía” y ”música” habrán actuado como etiquetas.

Por otro lado, IR consiste en una búsqueda activa de la información, mediante consulta y exploración. Se busca por etiquetas, obteniendo una lista ordenada de fuentes de información en relación con esas etiquetas y posteriormente se escanea o explora dicha lista. El sistema puede proveer incluso una lista de etiquetas relacionadas, permitiendo la exploración de hipertexto.

Por ejemplo, usamos IR cada vez que buscamos a través de un buscador como Google. Esta búsqueda se considera activa. Los términos introducidos en la consulta actúan como etiquetas y el usuario es el que discrimina entre las fuentes de información recuperadas.

Sinclair y Cardew-Hall (Sin08) realizan un estudio sobre la utilidad de las *folksonomías*, llegando a la conclusión de que éstas son más útiles cuando la búsqueda es general que cuando es específica. En el caso de la búsqueda específica, los usuarios prefieren la búsqueda por palabras clave que el mismo usuario introduce.

Aunque una *folksonomía* se define comúnmente como un espacio plano de palabras clave sin ninguna relación semántica entre etiquetas previamente definida, diferentes estudios demuestran que las relaciones de asociación y herencia entre etiquetas se pueden inferir desde el análisis de la co-ocurrencia (HM06).

Aspectos Positivos y Negativos de los Sistemas Basados en Etiquetado o *Tagging*

- **Aspectos Positivos**

  Una gran ventaja del etiquetado aparece cuando la información que se necesita no está bien definida, ya que facilita la exploración a través de la *tag cloud*.

  Otras ventajas de la *folksonomía* según Hassan-Montero y Herrero-Solana (HM06) son las siguientes:

  - La *folksonomía* refleja directamente el vocabulario de los usuarios, lo que permite a los usuarios establecer sus necesidades reales y su

lenguaje. La mejor forma de obtener un índice centrado en el usuario es que el mismo usuario genere ese índice.

- Como las *folksonomías* emergen del acuerdo colectivo, las etiquetas son más precisas y meditadas y su significado más democrático que si hubieran sido asignadas por una sola persona.
- Cuando el proceso de indexación se obtiene mediante agregación se reduce la inconsistencia que existe cuando diferentes usuarios utilizan diferentes términos índice para describir el mismo documento.
- Las folksonomías permiten descubrir información por casualidad o azar.

- **Aspectos Negativos**

  La dificultad de los sistemas basados en etiquetado viene cuando distintos usuarios utilizan diferentes etiquetas para el mismo documento. Este tipo de problema se basa fundamentalmente en el lenguaje, en las relaciones léxicas entre las palabras, en la polisemia, sinonimia, así como de la opinión de la persona que añade la etiqueta a una determinada fuente de información. Teniendo en cuenta estas relaciones, la búsqueda a través de una *tag cloud* puede estar muy limitada, aunque puede mejorar si se le aplican técnicas de *clustering* (Beg06).

  Por otro lado, la forma de ordenar los resultados de la consulta de los sistemas basados en etiquetado puede no ser relevante, ya que muchos sistemas ordenan estos resultados basándose en actualidad con que los fuentes de información han sido etiquetados o el número de usuarios que los ha etiquetado (Sin08).

  Hsieh et al. (Hsi06) resumen en las siguientes las limitaciones de una *folksonomía*:

  1. Homonimia
  2. Sinonimia (incluyendo plurales y conjugados)
  3. Acrónimos
  4. Espacios, símbolos y palabras múltiples (ej. “nyc”, “New York”, “newyorkcity”,”newyork”, etc.)
  5. Ruido (etiquetas irrelevantes o mal escritas)
  6. Variación en el nivel básico (etiquetas como “Perl” o “JavaScript” pueden ser muy específicas para algunos usuarios, mientras “programación” puede ser demasiado general para otros).

7. Etiquetas públicas y privadas (“coche” es una etiqueta pública, “mi coche” sería una etiqueta privada)

Sin embargo, Wu et al. (Wu06) demostraron que es posible extraer etiquetas con utilidad colectiva de la suma de etiquetas asignadas libre e individualmente, resolviendo automáticamente los problemas de ambigüedad de etiquetas.

Por su lado, Bar-Ilan et al. (BI08) nos dicen que resulta más útil el etiquetado estructurado, es decir, proveer etiquetas según un contexto (como rellenar una lista de descripciones) que el etiquetado desestructurado (añadir etiquetas libremente), ya que proporciona mayor información descriptiva, aunque presenta algunos problemas y para la recuperación de información a veces es mejor el uso del etiquetado libre.

Pero a pesar del potencial de los sistemas basados en etiquetado para IR, no se ha investigado lo suficiente sobre la efectividad y utilidad de las folksonomías.

Evaluación de la Utilidad de la *Foksonomía*

Según Hassan-Montero y Herrero-Solana (HM06) la efectividad puede medirse mediante dos parámetros:

1. La especificidad término-etiqueta → Número de fuentes de información descritos por una etiqueta.
2. La exhaustividad indexación-etiqueta → Número de etiquetas asignadas a un recurso.

Una etiqueta “amplia” conlleva una alta recuperación de información y una baja precisión, por lo que es acertada en la exploración (búsqueda general) mientras que una etiquetada “escasa” conlleva baja llamada y alta precisión, lo que es más acertada en la consulta (búsqueda específica).

Suelen asignarse etiquetas amplias porque requieren menor esfuerzo cognitivo, por lo que suelen tener baja especificidad, lo que las hace mejores para explorar que para consultar.

La exhaustividad en etiquetado también suele ser bastante baja, ya que en el 90 % de los casos los usuarios asignan menos de 5 etiquetas a cada recurso.

Estos hechos son razonables si se considera que el bajo coste cognitivo del etiquetado es uno de los principales factores de su popularidad.

2.1.3. Características de la *Tag Cloud*

Funciones

La *tag cloud* se popularizó por parte de los sistemas basados en etiquetado o *tagging* como interfaz de recuperación visual de información, que además de una visualización del conocimiento permite la navegación (HM06).

En otras palabras, la *tag cloud* traslada el vocabulario emergente de una *folksonomía* en una herramienta social de navegación (Sin08). Cuando un usuario hace "click" con el ratón sobre una etiqueta, obtiene una lista ordenada de fuentes de información descritas por la etiqueta, así como otras etiquetas relacionadas, por lo que sirven como tablas de contenidos o índices que se crean automáticamente (Riv07).

En las *tag clouds* se representan las etiquetas usadas con más frecuencia. Se definen como colecciones de palabras usadas para representar los conceptos presentes en grandes bases de información, teniendo en cuenta la asociación entre etiquetas, la frecuencia de éstas y la actualidad, para lo que se representa cada etiqueta con diferente tamaño y color (el tamaño de las etiquetas determina su frecuencia, a mayor tamaño mayor frecuencia y el color su actualidad, por ejemplo puede utilizarse rojo brillante para la más reciente y gris oscuro para la más antigua) (Kuo07).

Entre las funciones que desempeñan las tag clouds cabe destacar (Riv07):

- La búsqueda o localización de un término específico o de un concepto.
- La exploración cuando no se tiene en mente ningún objetivo concreto.
- La captura de lo esencial cuando se mira a la *tag cloud* y se toma conciencia de los temas más relevantes.
- El reconocimiento de las entidades que probablemente están representadas.

Las *tag clouds* han ido incrementando su popularidad como visualizaciones en páginas web personales y comerciales, en *blogs* y en sitios que comparten información social como *Flickr* y *Del.icio.ous.*, aunque también se usa este tipo de visualización en otros muchos ámbitos, para analizar texto, para búsqueda o para representar categorías, pero la forma de extraer los términos de la *tag cloud* no será la misma dependiendo del ámbito en que se use.

Aspectos Positivos y Negativos

- **Aspectos Positivos**

  Según Hearst y Rosner (Hea08), la representación realizada a través de una *tag cloud* es compacta y facilita que el usuario se fije en los términos mayores o más importantes. Además permite que se representen simultáneamente hasta 3 dimensiones: las palabras, su importancia relativa y su orden alfabético.

  Las *tag clouds* creadas a partir de etiquetas asignadas por los usuarios, además son útiles para reflejar las nuevas tendencias, ya que los usuarios perciben más las etiquetas con mayor tamaño, con lo que se percatan cuando una nueva etiqueta con gran tamaño aparece dentro de la *tag cloud*. Esta nueva etiqueta refleja los nuevos intereses de los usuarios.

  Las *tag clouds* son útiles para sugerir términos de búsqueda y ahorrar al usuario esfuerzo cognitivo. Esta utilidad se incrementa cuando buscamos en páginas no escritas en nuestra primera lengua.

- **Aspectos Negativos**

  Muchos han criticado las *tag clouds* que derivan de las *folksonomías* en general, alegando entre otras cosas, que el hecho de que un término sea popular no significa que sea relevante, por lo que las *tag clouds* a veces dificultan la búsqueda de términos realmente útiles (Sin08).

  Es difícil en una *tag cloud* comparar entre sí las etiquetas que tienen un tamaño similar. Así mismo, se tiende a darle importancia a una etiqueta según su tamaño, lo que puede causar problemas, ya que la longitud de la etiqueta puede entrar en conflicto con el tamaño de la fuente, es decir, el usuario puede confundir las etiquetas de mayor longitud con las de mayor tamaño, por lo que la importancia vendría dada en función del número de caracteres

que posee.

Otro aspecto negativo de la *tag cloud* es que a través de ella no puede accederse a todos las fuentes de información de una base de datos de modo directo. Si se refina la búsqueda mediante etiquetas relacionadas, obtendremos las fuentes de información que estén etiquetadas con la etiqueta inicial y la relacionada, con lo que si un recurso no presenta una etiqueta en la nube inicial, es inaccesible desde la interfaz de búsqueda. El número de fuentes de información ocultas desde la *tag cloud* se incrementa proporcionalmente al número de fuentes de información totales (para un número fijo de etiquetas en la nube). En el experimento realizado por Sinclair y Cardew-Hall (Sin08) el porcentaje de fuentes de información ocultas permanecía casi constante en cada sesión con un valor aproximado al 55,5 %. Este porcentaje es mayor que si realizamos una búsqueda específica por palabras clave mediante el método tradicional de búsqueda.

A la hora de realizar una búsqueda mediante una *tag cloud*, es mayor el número de consultas que se realizan que introduciendo directamente las palabras clave para buscar, incluso cuando las palabras de la consulta aparecen en la *tag cloud*, lo que sugiere que los usuarios se entretienen explorando el entorno (Sin08).

Cuando las etiquetas se muestran ordenadas por su frecuencia o importancia, no está claro que variando el tamaño de la fuente se obtengan ventajas sobre el listado simple de los términos en orden de importancia. Quizás por esto, este tipo de ordenación de etiquetas es poco usada.

Hassan-Montero y Herrero-Solana (HM06) además señalan las siguientes restricciones que limitan la utilidad de la *tag cloud* como interfaz de recuperación visual:

- El método para seleccionar las etiquetas se basa exclusivamente en la frecuencia, lo que conlleva una alta densidad semántica. Los términos usados con mayor frecuencia son los peores discriminadores y unos pocos términos tienden a dominar la *tag cloud* junto con sus etiquetas relacionadas.
- Cuando la ordenación de las etiquetas dentro de la *tag cloud* es alfabética no se facilita el escaneo visual para inferir relación semántica entre las etiquetas.

En el experimento realizado por Sinclair y Cardew-Hall (Sin08) en el que se preguntó a los participantes si preferían una interfaz de búsqueda tradicional donde el usuario introduce las palabras clave o el uso de las *tag clouds*, la mayoría de los participantes afirmó preferir la interfaz tradicional, alegando que permitía mayor especificidad, aunque estos participantes no tenían ninguna experiencia anterior en etiquetado, por lo que su respuesta podía deberse a que el hecho de especificar la búsqueda les resultaba más familiar.

Evaluación de la Efectividad y del Diseño

- **Evaluación de la Efectividad**

Una *tag cloud* será efectiva si se compone de etiquetas significativas. Según Aouiche et al. (Aou09) para determinar si una *tag cloud* se compone de etiquetas significativas se calcula la entropía:

Sea t $\in$ T una etiqueta de una *tag cloud* T:

$$Entropy(T) = -\sum_{t \in T} p(t) log p(t) \tag{2.1}$$

donde

$$p(t) = \frac{weight(t)}{\sum_{t \in T} weight(t)} \tag{2.2}$$

La entropía cuantifica la disparidad de pesos entre etiquetas. Si ésta es baja, la *tag cloud* es significativa o efectiva, si es alta significa que los pesos de las etiquetas son uniformes lo que visualmente no es muy informativo.

- **Evaluaciones Realizadas sobre el Diseño**

Las etiquetas en la *tag cloud* pueden estar ordenadas alfabéticamente, por su frecuencia o según un determinado algoritmo. También pueden ordenarse semánticamente o el usuario puede especificar sus preferencias de *clustering*. Con respecto al diseño espacial, las palabras pueden representarse en líneas secuenciales, en forma de cubo, en forma de círculo, etc.

Un experimento realizado por Rivadeneira et al. (Riv07) muestra que las etiquetas representadas en el primer cuadrante de la *tag cloud* se recuerdan más por el usuario que las etiquetas representadas en el resto de cuadrantes y que la proximidad de las palabras no tiene efectos a la hora de recordarlas. Evidentemente, también se recuerdan más aquellas con mayor tamaño.

Bateman et al. (Bat08) desmienten el experimento realizado por Rivadeneira et al. y dicen que no hay efecto en la posición de las etiquetas dentro de la *tag cloud* a lo hora de recordarlas.

Otro experimento realizado por Halvey y Keane (Hal07), donde los participantes debían encontrar un país en una lista de 10 países, se demostró que la visualización en las listas era mucho más rápida que en un diseño espacial y que los listados alfabéticos fueron los más rápidos en todos los casos.

En el experimento realizado por Hearst y Rosner (Hea08), 7 de los 18 participantes no se dieron cuenta de que la *tag cloud* mostrada estaba organizada en orden alfabético, dos de los cuales eran programadores que usaban las *tag clouds* en sus propios sitios web y solían explorar con ellas.

Otros investigadores realizan diseños alternativos, como Hassan-Montero y Herrero-Solana (HM06) que proponen aplicar técnicas de *clustering* a la *tag cloud*. Para calcular la similaridad entre etiquetas utilizan la co-ocurrencia relativa entre ellas. Otros autores como Fujimura et al. (Fuj08) o Berlocher et al. (Ber08) utilizan la similaridad del coseno como medida de similaridad entre etiquetas. Pero ninguno de ellos realiza ningún experimento para evaluar el efecto de aplicar estas técnicas sobre los usuarios.

Quienes sí realizan una evaluación de los distintos diseños: semántico, aleatorio y alfabético, son Schrammel et al. (Sch09), resultando el diseño alfabético el mejor valorado por los usuarios.

Otra desventaja importante de los principales diseños de las *tag clouds* es que el vasto espacio en blanco entre las etiquetas las hace inapropiadas para dispositivos con pequeñas pantallas, como PDAs o teléfonos móviles (Kas07) y según muchos autores, si se comprime la nube para omitir estos espacios, el resultado es antiestético (Ver *tag cloud* 3 en figura (2.1)).

Viégas et al. (Vie09) proponen un diseño," Wordle" (Fei09), donde se omiten los espacios en blanco entre etiquetas y éstas pueden aparecer tanto en sentido vertical como horizontal o incluso diagonal (Ver *tag cloud* 5 en figura(2.1)).

Kaser y Lemire (Kas07) describen diferentes algoritmos para diferentes variaciones del diseño de la *tag cloud*.

Van-Ham et al. (VH09) introducen un nuevo diseño mediante una técnica para construir mapas de textos sin estructura a la que llaman "phrase net". En ella la unidad de análisis es la frase, es decir, las relaciones entre las palabras que son definidas en base a un modelo o al análisis sintáctico.

En la figura (2.1) se pueden ver algunos ejemplos del diseño de diferentes tipos de *tag clouds*

2.1.4. Las Etiquetas en la *Tag Cloud*

Métodos para el Establecimiento de las Etiquetas

- Cuando las etiquetas son asignadas libre y manualmente por los usuarios, hay que establecer un *ranking* para determinar cuáles de estas etiquetas formarán parte de la *tag cloud*.

  Normalmente suele emplearse un *ranking* basado exclusivamente en la frecuencia absoluta o relativa de las etiquetas.

  Knautz et al. (Kna99) establecieron una segunda alternativa para la construcción de este *ranking* que resultó ser mejor según los usuarios encuestados en su experimento. Esta segunda alternativa hace uso de la fórmula:

$$WDF \cdot ITF = \left[\frac{\log frec(t,b)+1}{\log L}\right] \cdot \left[\log\left(\frac{M}{m}\right)+1)\right] \qquad (2.3)$$

  donde *WDF* es la frecuencia dentro del documento que toma logaritmos de las ocurrencias relativas e *ITF* es la inversa de la frecuencia de cada etiqueta. *freq(t,b)* es la frecuencia con que la etiqueta *t* se asigna a la fuente de información *b*, *L* es número total de etiquetas de la fuente de información y

Tag Cloud 1

Tag Cloud 2

Tag Cloud 3

Tag Cloud 4

Tag Cloud 5

Tag Cloud 6

Figura 2.1: *Distintos diseños de Tag Clouds*

M el número de todas la etiquetas en la *folksonomía*, siendo *m* la ocurrencia de una etiqueta en el conjunto.

Sinclair y Cardew-Hall (Sin08) también utilizaron una segunda forma para determinar el tamaño de las etiquetas en función de su frecuencia mediante la siguiente fórmula:

$$TagSize = 1 + C \cdot \frac{\log(f_i - f_{min} + 1)}{\log(f_{max} - f_{min} + 1)} \tag{2.4}$$

donde f_i es la frecuencia de la marca, f_min y f_max son las frecuencias mínima y máxima respectivamente y C es una constante que determina el tamaño máximo del texto.

- Cuando las etiquetas son extraídas del texto normalmente habrá que realizar primero una limpieza para eliminar las palabras superfluas.

Kuo et al. (Kuo07), generaron una aplicación para resumir los resultados de las consultas realizadas sobre una base de datos biomédica. Esta aplicación, respondía con *tag clouds* extraídas de los resúmenes devueltos por las consultas. La generación de las etiquetas de estas *tag clouds* se llevó a cabo de la siguiente forma:

1. Se eliminan, para cada resumen, las palabras que no aportan información: “la”, “de”, “con”, “y”, etc., así como las puntuaciones y símbolos.
2. Se eliminan los sufijos, por ejemplo “funcional” se convierte en “func”, esto se hace aplicando el algoritmo de stemming de Porter. Y estas raíces son las que se usan como etiquetas.

Después de generar la lista de etiquetas que describe la respuesta de la consulta, se calcula la frecuencia relativa y la actualidad de cada etiqueta. Sólo se tienen en cuenta las etiquetas que tienen una frecuencia de al menos el 10 %. La actualidad se calcula como la media de la fecha de publicación de los resúmenes en los que aparece esa etiqueta .

Al hacer ’click’ con el ratón sobre una etiqueta aparece una lista de palabras que comparte el mismo prefijo con diferentes sufijos y un enlace al conjunto de resúmenes que contienen la etiqueta.

Evaluación de la Utilidad de las Etiquetas

Como se ha dicho, normalmente las etiquetas se seleccionan en base a su frecuencia y este método de selección conlleva que las *tag clouds* ofrezcan una imagen semánticamente homogénea, donde la mayoría de las etiquetas son similares unas a otras.

Para seleccionar las etiquetas que mejor caracterizan la colección de fuentes de información etiquetados se determina la utilidad de una etiqueta como:

- La capacidad de representar cada recurso comparada con otras etiquetas asignadas al mismo recurso.
- El volumen de fuentes de información cubiertas en comparación con otras etiquetas.

Se considera una *folksonomía* como un vector de fuentes de información $D_i = (d_{i0}, ..., d_{in})$, cada una caracterizada a través de una o más etiquetas $T_j = (t_{j0}, ..., t_{jm})$, ponderando de acuerdo al número de usuarios que han etiquetado. Supongamos d_{ij} representa la frecuencia con que se ha usado cada etiqueta T_j para describir el recurso D_i. Se define $F(T_j)$ como la utilidad de la etiqueta T_j como parte de la *tag cloud* como (HM06):

$$F(T_j) = \sum_{i=1}^{i=n} \left[\frac{\log d(ij)}{m^2} \right] \tag{2.5}$$

donde n es el número de fuentes de información descritas por T_j y m el número de etiquetas asignadas a un diferente recurso.

2.1.5. *Tag Cloud* Monotérmino y *Tag Cloud* Multitérmino

Muchos autores señalan, que para convertir los resultados obtenidos tras la minería de texto a resultados más comprensibles para el usuario y para soportar el análisis de texto, debe visualizarse este texto con distintos niveles de granularidad, permitiendo descubrir patrones o conjuntos de *itemsets* frecuentes.

Las etiquetas multitérmino son uno de los elementos más útiles en el etiquetado y proporcionan la capacidad de utilizar términos relacionados. Estas etiquetas multitérmino proporcionan penetración en el contenido del texto al permitir que los términos relacionados puedan ir juntos (Ejemplos de términos relacionados: inteligencia artificial, redes sociales, sistemas operativos, etc.).

Pero las interfaces o herramientas para extraer etiquetas multitérmino son más complejas y confusas de lo que deberían ser. (Ver (VW09)).

Supongamos la etiqueta “red social”. Cuando una herramienta referencia esta etiqueta debe estar mirando ambas partes (“red” y “social”) como un sólo conjunto, lo que tiene un significado distinto al de los términos de forma individual.

Un estorbo común en el etiquetado social es que en las etiquetas multitérmino, los términos aparecen conectados mediante guiones, puntos o escritos juntos, sin espacios, como si fueran una sola palabra, como por ejemplo en Delicious (Sch03). Esto rompe la construcción básica del usuario y son las herramientas las que deberían abrazar los métodos humanos de interacción y no los humanos las restricciones tecnológicas. Además de que no permite el escaneo visual.

Otro inconveniente en la introducción de caracteres extraños en las etiquetas, como guiones, es que se rompe el modelo conceptual.

Algunas herramientas, tratan de normalizar estos “multitérminos” para identificar *items* similares y relevantes. Esto resulta fácil cuando los componentes del “multitérmino” están escritos separados, pero requiere mucho trabajo cuando no lo están.

Como hemos dicho, la mayoría de las herramientas de minería de texto muestran la frecuencia de las palabras en el texto de forma aislada, lo que limita su utilidad y eficacia, ya que:

- Dificultan la identificación del contenido de la fuente de información (No permiten identificar conceptos relacionados)
- No permiten la búsqueda de patrones frecuentes
- No permiten la búsqueda de términos con más de una componente (multitérminos)
- No permiten comparar y contrastar las características de diferentes patrones de texto
- No permiten identificar expresiones cercanas o repeticiones de términos con pequeñas variaciones
- No facilitan información de contexto.

Todos estos inconvenientes se pueden solventar permitiendo el uso de “multitérminos” (ver (Don07)).

Panunzi et al. (ver (Pan06)), realizaron un estudio para evaluar la diferencia entre un técnica de extracción de palabras clave de un sólo término y una técnica que permitía extraer palabras clave de más de un término. Los resultados mostraron que las palabras clave complejas se consideraban más descriptivas y permitían la identificación del contenido del texto, siendo más adecuadas que las palabras simples.

Esta técnica de extracción de palabras de más de un término, consistía primero en calcular el peso o frecuencia de los nombres presentes en el texto. Estos nombres se consideraban potencialmente ambiguos con respecto a su semántica y al mismo tiempo se consideraban la “cabeza” del multitérmino. Para incrementar la predictibilidad en la identificación del contenido, a partir de estos nombres construían un *n-grama* de términos, de los cuales seleccionaban sólo los más relevantes a través de un filtro lingüístico que identificaba sólo posibles combinaciones de “multitérminos”. Estas combinaciones debían cumplir tres condiciones:

- El *n-grama* debe contener un nombre
- Un patrón bi-término aceptable es “nombre + nombre” o “nombre + adjetivo”, pero no “nombre + preposición”.
- El *n-grama* debe ocurrir más de una vez en el texto

En la lista de salida, se consideraban las palabras clave multitérmino y las palabras clave monotérmino, para producir una lista coherente y a cada palabra clave se le asociaba un peso.

Calcularon el peso de los “multitérminos” mediante una fórmula en que éste era proporcional al número de ocurrencias del *n-grama* y a la frecuencia de cada componente del “multitérmino” e inversamente proporcional a este último número de ocurrencias. (Ver fórmulas en Frantzi et al. (Fra00a)).

Este algoritmo, fue usado posteriormente por autores como Agili et al. (Agi08) para la extracción de palabras clave multitérmino.

Watters (Wat08) creó la herramienta “*Cloud Mine*” para su uso como asistente en la búsqueda web y proporcionando la capacidad de análisis del texto. Para ello, empleó también “multitérminos” en lugar de palabras simples como ocurre normalmente en la mayoría de las visualizaciones que podemos encontrar en forma

de *tag cloud*. Utilizó la herramienta "TerMine" (Fra00b), que es una herramienta gratuita que podemos encontrar en la web que incorpora métodos lingüísticos, estadísticos e información de contexto para la extracción de palabras clave multitérmino. Y demostró que los resultados empleando palabras clave multitérmino mejoraban los resultados en que se empleaban palabras clave simples, ya que proporcionaban contexto y orientación al usuario incrementando el nivel de significación proporcionado por los métodos de recuperación de información.

2.1.6. *Clustering* y Herencia en la *Tag Cloud*

Algoritmo de *Clustering*

Según el estudio de Kuo et al. (Kuo07) la forma en que mostraron la información con la aplicación para las consultas sobre la base de datos biomédica, fue ventajosa a la hora de representar información descriptiva, pero menos efectiva que el tradicional método de búsqueda a la hora de permitir a los usuarios descubrir relaciones entre conceptos. La solución que propusieron para esto fue el empleo de algoritmos de *clustering*.

Con el fin de inferir semántica o relaciones entre conceptos se podrían aplicar técnicas de *clustering* para la construcción de la *tag cloud*.

Knautz et al. (Kna99) observaron que los usuarios que tienen que reformular sus consultas utilizando argumentos de búsqueda adicionales, encuentran dificultad en el uso de los operadores booleanos. En concreto, el operador AND sería el más usado para combinar nuevos términos con los argumentos de búsqueda iniciales. Los *clusters* en la *tag cloud* proporcionan automáticamente ayuda para el uso de los operadores, ya que una consulta inicial ofrece una respuesta donde se presentan otros argumentos, haciendo "click" en cada término del *cluster*, se desarrolla automáticamente una lista de términos añadidos a la consulta inicial. Cuando los *clusters* vienen representados en forma de grafo, haciendo "click" con el ratón en cualquier eje entre dos términos o vértices, lo que se está haciendo es añadir o adicionar ambos términos a la consulta original.

En la visualización de las etiquetas en *clusters* realizada por Knautz et al., se presenta un estatus igual para todas las etiquetas, lo que da la oportunidad a las etiquetas recientes de ser visualizadas.

Begelman et al. (Beg06) también presentan un algoritmo de *clustering* para

obtener una medida de similaridad entre etiquetas, para posteriormente aplicar las técnicas de *clustering* en la *tag cloud*.

Para ello, lo primero que hacen es una transformación de la información en una representación numérica que pueda ser utilizada por el algoritmo. Esta transformación la realizan de forma manual.

Estimación de la Medida de Similaridad entre Etiquetas

La medida de similaridad puede obtenerse de múltiples formas como son: aplicando el coeficiente de Jaccard, la distancia euclídea, la similaridad del coseno, la similaridad de Tanimoto, la correlación de Pearson, la distancia de Hamming, etc. (Aou09). Aunque no hay grandes diferencias entre ellas, Knautz et al. (Kna99) descubrieron mediante un cuestionario, una pequeña preferencia en favor de la medida de similaridad del coseno, que se calcula mediante la siguiente ecuación:

$$\psi(A-B) = \frac{g}{\sqrt{ab}} \tag{2.6}$$

donde ψ representa el valor de coincidencia para dos etiquetas A y B, a simboliza las fuentes de información que contienen la etiqueta A, b las fuentes de información que contienen la etiqueta B y g las fuentes de información que contienen ambas etiquetas. Los valores oscilan entre 0 y 1, siendo 0 la falta de similaridad y 1 la similaridad máxima que puede existir entre dos etiquetas.

En el algoritmo de Begelman et al. (Beg06) para encontrar las etiquetas fuertemente relacionadas basado en el número de co-ocurrencias de cualquier par de etiquetas, se establece un punto de corte a partir del cual se decide si la co-ocurrencia es significativa. Esto se representa en forma de matriz, de modo que cada elemento de la matriz es la co-ocurrencia entre dos etiquetas.

Para determinar el punto de corte mencionado anteriormente, se realiza un gráfico de frecuencias para las co-etiquetas (se representa la frecuencia de co-ocurrencia de cada etiqueta con todas las demás), así como una representación de la primera y segunda derivadas de la frecuencia. Se examina el gráfico de la primera derivada para ver donde presenta su pico más alto, que coincide con el punto en que la segunda derivada cambia de positiva a negativa. Por último, se comprueba si el valor de este pico es suficientemente alto para ser establecido como el punto de corte.

El único parámetro que debe ser optimizado es la “altura mínima del pico”. A veces la distribución no presenta ningún pico “suficientemente alto” o no se

tienen suficientes datos para calcularlo, en ese caso, consideran las etiquetas que presentan la mayor co-ocurrencia como fuertemente relacionadas.

Haciendo esto para cada etiqueta en el espacio de etiquetas, se obtiene un grafo $G(V,E,W)$, donde los vértices V son las etiquetas, los ejes E son las relaciones entre etiquetas y W es una matriz simétrica que representa el peso de cada relación (co-ocurrencia).

Tras realizar todo esto, Begelman et al. consideran que un *cluster* es cualquier conjunto de etiquetas conectadas en el grafo (teniendo en cuenta sólo las relaciones o ejes con una co-ocurrencia o peso superior al punto de corte).

Por último, se sugiere aplicar el algoritmo descrito por Scott White en el caso de que aparezcan *clusters* muy grandes, para dividirlos en *clusters* de tamaño más pequeño.

Otros autores, como Hassan-Montero y Herrero-Solana (HM06) definen la co-ocurrencia relativa entre dos etiquetas como:

$$RC(A,B) = \frac{|A \cap B|}{|A \cup B|} \tag{2.7}$$

donde A y B son los conjuntos de las fuentes de información descritos por las 2 etiquetas y $RC(A,B)$ se conoce como el "Coeficiente de Jaccard". Esta medida es la que utilizan para estimar la similaridad entre etiquetas.

Técnicas de *Clustering:*

Existen numerosas técnicas de clustering para construir el grafo $G(V,E,W)$. Begelman et al. (Beg06) utilizan el algoritmo de bisección espectral y la función de modularidad combinadas con un algoritmo *greedy* recursivo.

La función de modularidad mide la calidad de un *cluster* particular de nodos en un grafo, por lo que se usa como una medida de la calidad de particionamiento.

El algoritmo basado en la bisección espectral se utiliza para biseccionar los vértices del grafo.

El algoritmo greedy actúa del siguiente modo:

- Se usa la bisección espectral para dividir el grafo en dos *clusters*.

- Se compara el valor de la función de modularidad del grafo original con el grafo partido. Si este valor es mayor para el grafo partido se acepta la partición, si no se rechaza.
- Se procede del mismo modo de forma recursiva para cada partición.

Hassan-Montero y Herrero-Solana (HM06) utilizan el algoritmo de bisección de las K-medias con el objetivo de clasificar las etiquetas en *clusters*, usando la función de similaridad del coseno para medir modelos de co-ocurrencia entre etiquetas.

En la *tag cloud* que presentan Hassan-Montero y Herrero-Solana, las etiquetas similares son "vecinas" horizontalmente y los *clusters* similares son "vecinos" verticalmente.

Knautz et al. (Kna99) afirman que son 3 las operaciones posibles a realizar después de clasificar las etiquetas mediante los valores de similaridad:

1. El método de enlace simple.- Se comienza con el par de etiquetas más similares que estén contenidas en un número mínimo de fuentes de información. Posteriormente se van añadiendo todas las etiquetas que co-ocurren con la primera etiqueta y luego todas las que co-ocurren con la segunda, considerando un valor límite de similaridad. Este paso se repite para cada nueva etiqueta hasta que no quede ninguna suelta por encima del valor límite.
2. El método de enlace completo.- Se comienza también con el par de etiquetas más similares y se añaden aquellas etiquetas que co-ocurren en la misma fuente de información.
3. El método de agrupación por la media.- Inicialmente se opera igual que con el método de enlazado simple, pero después de construir el *cluster* se calcula una media de la similaridad para eliminar de éste todas las etiquetas cuyo valor de similaridad esté por debajo de esta media.

Según un experimento realizado por estos mismos autores sobre los 3 métodos, el tercer método parecía ser el preferido por los usuarios.

Herencia entre Etiquetas

Hsieh et al. (Hsi06) propusieron un sistema que incorpora la herencia entre las etiquetas. Este método tuvo una gran llamada (número de fuentes de información recuperados en una consulta) debido a la incorporación del concepto "distancia",

pero perdió en precisión.

El sistema de Hsieh et al. incluye una interfaz de edición que permite a los usuarios subir o comentar fuentes de información, un generador utilizado para construir espacios de conceptos y otros módulos que se basan en este generador para proporcionar sus servicios:

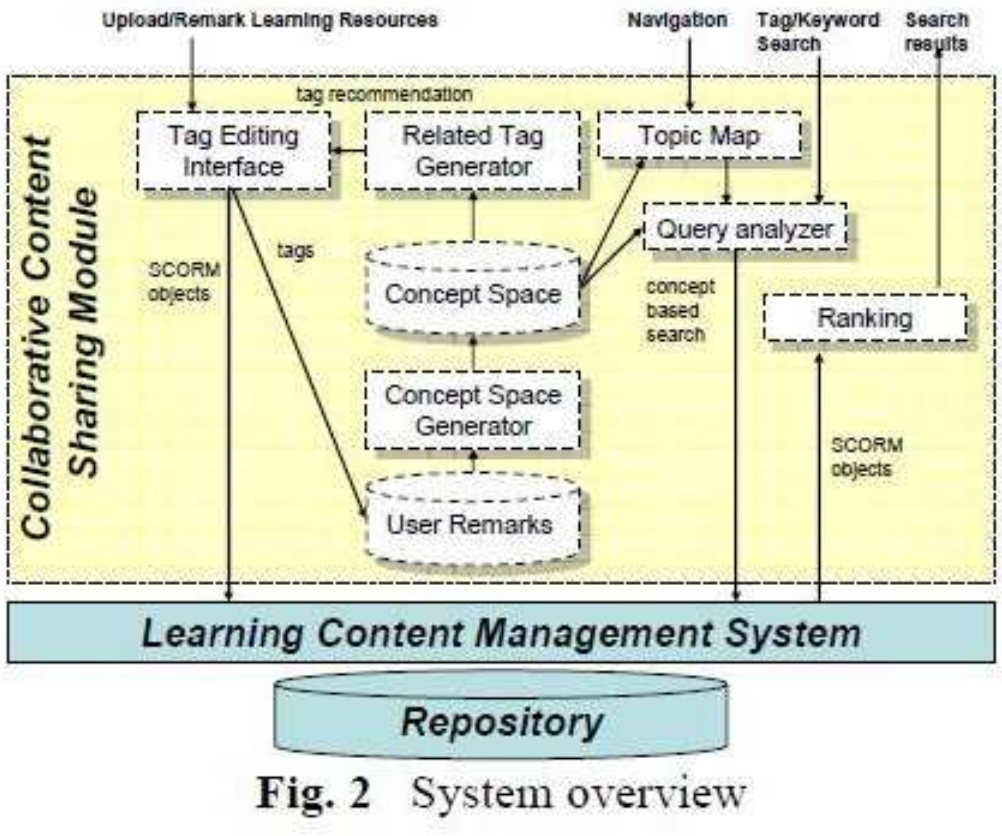

Figura 2.2: *Sistema de Hsieh et al. (Hsi06)*

Generador de Espacio de Conceptos

El generador de espacio de conceptos analiza el espacio de etiquetas y establece el concepto de herencia entre las etiquetas . Esto se realiza en los siguientes pasos:

1. Se usa un vector T_j para las etiquetas t_j que representa la relación entre éstas

y las fuentes de información. Cada elemento $T[i]$ denota el número de veces que la etiqueta se ha asignado al recurso *i*.

2. Se ordenan los vectores de etiquetas construidos en el paso 1 atendiendo a:

   a) El número de fuentes de información a las que se asigna una misma etiqueta.

   b) La suma total de veces que se ha asignado esa etiqueta si 2 etiquetas poseen el mismo valor para:

   $$\frac{|R_A \cap R_B|}{R_A} > \lambda \tag{2.8}$$

   Donde R_A son las fuentes de información descritas por la etiqueta *A* y R_B las fuentes de información descritas por la etiqueta *B*. *l* es un umbral entre 0 y 1 que define el límite inferior del concepto "distancia" en el concepto de herencia, esto es, si el valor de *l* está más cerca de 1, entonces la distancia entre las 2 etiquetas es corta. Si el valor de la ecuación (2.8) es verdadero, entonces la etiqueta *B* se considera un nodo padre de la etiqueta *A*. Se usa la similaridad del coseno para determinar la distancia.

   c) La primera vez que una etiquetas se usa si 2 etiquetas tienen el mismo valor para (2.8) y para *l*.

3. Se usa la lista ordenada del paso anterior para establecer la herencia.

En el ejemplo de la figura (2.3) se tienen 4 etiquetas y 3 fuentes de información. Las flechas que van de las etiquetas a las fuentes de información indican que las etiquetas son asignadas a las fuentes de información y el número encima de la flecha es el número de veces que son asignadas.

Generador de Etiquetas Relacionadas y Analizador de Consultas

Una vez construido el concepto de herencia se pueden realizar recomendaciones basadas en las etiquetas cercanas o de los nodos ancestros, descendientes y hermanos.

Enfoque *Clustering* para el Cálculo de la Herencia de una *Folksonomía.*

Nos basamos en el estudio realizado por Grahl et al. (Gra07) que realizan una clasificación interactiva del conjunto de etiquetas usando dos veces el algoritmo de las k-medias, seguida de otro algoritmo para descubrir los usuarios y fuentes

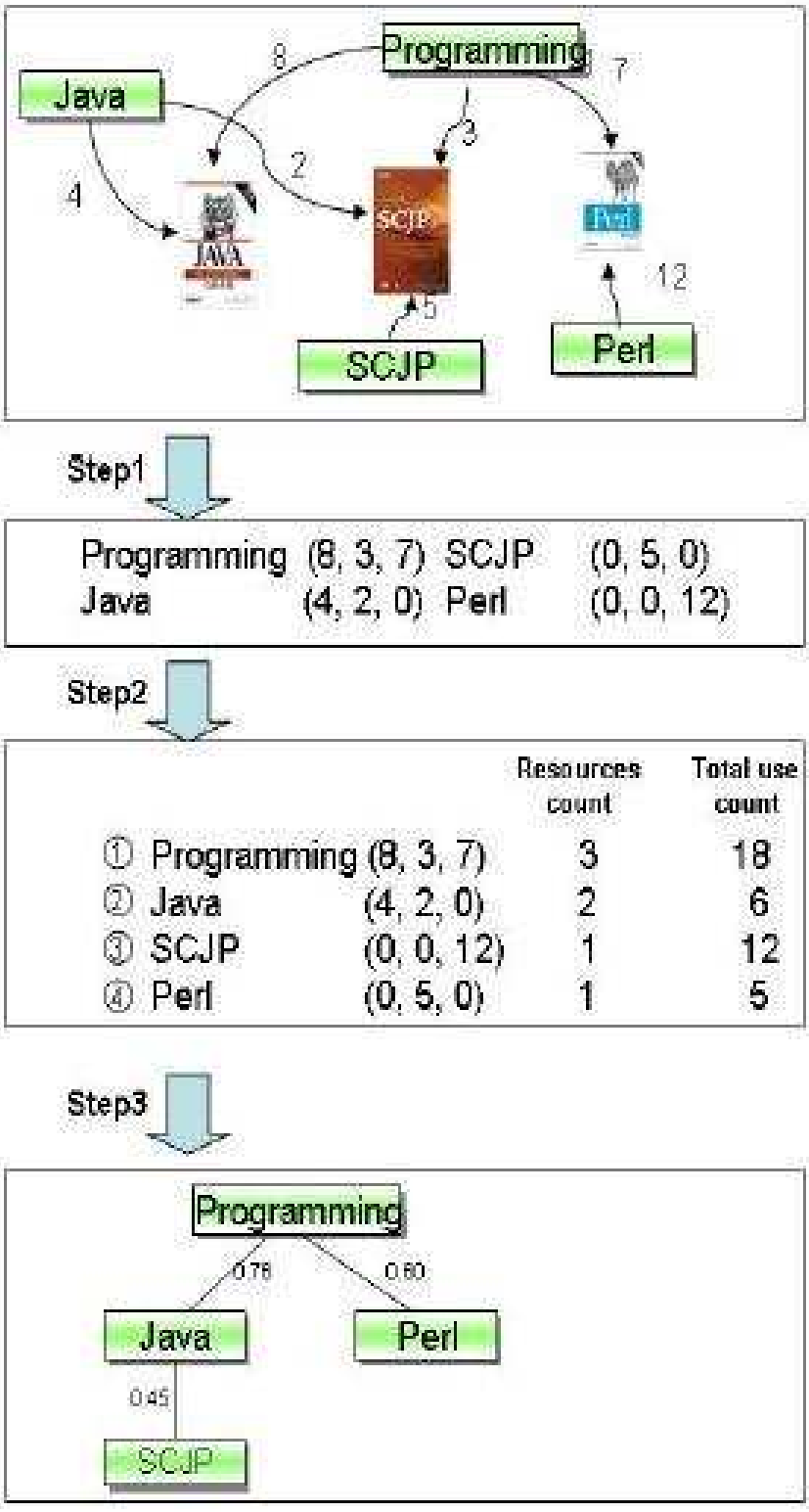

Figura 2.3: *Ejemplo de herencia (Hsi06)*

de información relacionados con los *clusters* hijos de la herencia generada.

La generación de la herencia entre *clusters* es completamente automática. El resultado es una herencia en tres niveles de conjuntos de etiquetas. Los *clusters* del nivel más bajo (y más detallado) son complementados por listas de los usuarios y fuentes de información relacionados.

No se ha proporcionado ninguna medida para la evaluación de la calidad de los *clusters* según este enfoque.

Construcción de los *Clusters*

Se utiliza el algoritmo de las k-medias para calcular los *clusters* con la medida de similaridad del coseno. Se trabaja con la matriz de co-ocurrencias entre etiquetas, por lo que las etiquetas son los objetos de este algoritmo.

Para calcular los usuarios y fuentes de información más relacionados con cada *cluster* de etiquetas se usa el algoritmo de *ranking* "Folkrank" presentado por Hotho et al. (Hot06). Dado un conjunto de las etiquetas, usuarios y fuentes de información preferentes de una *folksonomía*, "Folkrank" calcula una clasificación de temas que proporciona un orden descendiente en importancia de los elementos de la *folksonomía* con respecto a los elementos preferentes.

Primero se transforma el hipergrafo entre los conjuntos de etiquetas, usuarios y fuentes de información a un grafo ponderado, tripartito y sin dirigir. En este grafo se aplica una versión de "PageRank" que tiene en cuenta el peso de los ejes. Finalmente el algoritmo "Folkrank" calcula la clasificación por temas usando un enfoque diferencial.

Construcción de la Herencia Conceptual

Tras eliminar algunas palabras consideradas "spam", haber generado el espacio vector del conjunto de etiquetas y haber calculado los *clusters* con la máxima granulación posible, se extrae para cada *cluster* una etiqueta que hará de descriptora.

Estas etiquetas descriptoras son agrupadas otra vez, ocupando la capa media de la herencia conceptual.

Finalmente se calculan parejas de etiquetas como descriptores de estos *meta-clusters* ocupando el nivel más alto y general de la herencia.

Resumiendo, en la herencia final se tiene: las parejas de etiquetas en el primer nivel, las etiquetas descriptoras de cada pareja de etiquetas en el segundo nivel o nivel medio y los *clusters* de etiquetas relacionadas con cada etiquetas descriptora en el tercer o último nivel, el cual se completa con los usuarios y fuentes de información relacionadas con cada etiqueta.

2.1.7. *Data Cloud*

Definición y Generación

Según Koutrika et al. (Kou09a), (Kou09b), la *tag cloud* es útil para los propósitos de navegación y visualización sobre datos sin estructura porque resaltan los conceptos más significativos. Pero por otro lado, cuando la búsqueda se realiza sobre bases de datos estructuradas, es mejor realizarla por palabras clave, por lo que proponen un método para unir esta búsqueda con las capacidades de resumen y navegación de las *tag clouds* para ayudar a los usuarios a acceder a la base de datos. A la nube o *cloud* originada sobre datos estructurados, la denominan *data cloud*.

A través de la *data cloud* se resumen los resultados obtenidos con la búsqueda por palabras clave sobre los datos estructurados y se guía a los usuarios para que refinen estas búsquedas, para ello la *data cloud* presenta las palabras más significativas asociadas a los resultados de la búsqueda, permitiendo buscar en múltiples tablas de la base de datos.

En los sitios sociales, las etiquetas son asignadas manualmente, pero en el caso de las bases de datos estructuradas hay que categorizar los campos de texto, decidir como agregar las mismas palabras encontradas en campos diferentes, utilizar estructuras y estadísticas para soportar la búsqueda con nubes dinámicas, etc. Además, como las entidades tienen estructura, la posición del término afecta a su importancia, por lo que no serviría una nube basada únicamente en la frecuencia.

En IR, las unidades de información están bien definidas: son los documentos. En las bases de datos, sin embargo, la información conceptualmente se refiere a una sola entidad, pero puede encontrarse en diferentes relaciones, debido a la estructura de la base de datos y a su normalización.

Koutrika et al. modelan la base de datos D como una colección V de entidades de búsqueda. Una entidad de búsqueda es conceptualmente un objeto complejo con atributos $B_1 \ldots B_n$. Un atributo B_i puede ser atómico y estar almacenado en

una columna en la base de datos o puede estar compuesto por un objeto o lista de objetos que reúnen información para la búsqueda de la entidad *v*. La colección *V* puede entenderse como una "vista" que colecta y agrupa información relacionada con una entidad individual de las relaciones almacenadas en *D* y la representa como una sola unidad de información.

Una consulta *q* se formula como una conjunción de términos clave. Un término *k* puede ser una sola palabra o una frase. Dada una consulta *q* y una colección *V* definida sobre la base de datos *D*, la respuesta para *q* es el conjunto $Vq \subseteq V$ que contiene todas las entidades de *V* que contienen todos los términos de la consulta *q* al menos una vez.

Una cuestión muy importante es como añadir rangos a las entidades de búsqueda que se han ajustado al término clave. Pensando en las entidades de búsqueda como equivalentes a "documentos" podrían usarse los métodos de *ranking* de la IR estándar. Por ejemplo, se puede calcular el peso $tf \cdot idf$ de cualquier término de consulta *k* en cualquier entidad *v* en V_q. El término "frecuencia" tf puede calcularse usando la fórmula:

$$tf_{k,v} = \frac{\sum_{B \in v} n_B}{n_v} \tag{2.9}$$

donde n_B es el número de ocurrencias de *k* en un atributo *B* de *v* y n_v es el número de términos en *v*. La inversa de la frecuencia del documento idf para *k* es:

$$idf_k = \ln\left(\frac{N}{N_k}\right) \tag{2.10}$$

donde *N* es el número de entidades de búsqueda en la base de datos y N_k es el número de entidades que contienen *k*.

Con esto, se puede establecer una puntuación para *v* con referencia a la consulta *q* sumando los pesos $tf \cdot idf$ de todos los términos de consulta en *v*:

$$score(v,q) = \sum_{k \in q} tf_{k,v} \cdot idf_k \tag{2.11}$$

Con este enfoque no se tiene en cuenta la posición del término de consulta. Por ejemplo, si se busca en noticias, no tendría la misma puntuación que el término apareciera en el título de la noticia, en el desarrollo o en los comentarios. Para acatar esto podrían usarse pesos de posición.

Un peso de posición representa la significación de la ocurrencia del término dependiendo de su posición en el documento. Se puede transferir esta idea refinando la fórmula (2.11) usando atributos ponderados:

$$tf_{k,v} = \frac{\sum_{B \in v} \omega_B \cdot n_B}{n_B} \tag{2.12}$$

donde ω_B es el peso para el atributo B. Estos pesos pueden pre-asignarse a los atributos en la base de datos de forma manual o pueden determinarse automáticamente basándonos en un conjunto de reglas.

Construcción de la *Data Cloud.*

Para la generación de la *data cloud* se eliminan palabras como pronombres personales, preposiciones, etc.

En teoría la búsqueda se realiza a nivel de entidades. Las entidades de búsqueda son una abstracción útil, pero en la práctica se consume mucho tiempo si para generar el conjunto de entidades que se ajusta a una consulta se usa el índice invertido basado en las tuplas, ya que habría que recorrer todas las tuplas de este índice en las que se localice la palabra clave de la consulta. En lugar de esto, se usa un índice invertido basado en la entidad, donde cada ocurrencia de un término de consulta se enlaza a la entidad de búsqueda a la que conceptualmente pertenece.

Lo difícil es encontrar las mejores palabras para incluirlas en la *data cloud*. Para ésto se tienen varios enfoques (Kou09b):

1. Basado en popularidad.

$$score(k, q, V_q) = \sum_{v \in V_q} \sum_{B \in v} n_B \tag{2.13}$$

   Esencialmente, lo que se hace es medir la popularidad (número de veces que k co-ocurre en q) de los términos en los resultados de la consulta. Este es el enfoque típico en las *tag clouds*, pero como hemos dicho, no resulta bueno para las *data clouds* ya que no es útil para el propósito de refinamiento de la búsqueda.

2. Basado en la relevancia.

   Con este enfoque se seleccionan aquellos términos más relevantes para la consulta sobre las entidades de V_q. Se trata cada término candidato k como

una palabra de consulta y se calcula la similaridad entre k y cada entidad de respuesta v en V_q. Una puntuación alta significaría que el término y la entidad se ajustan por lo que la entidad sería un resultado relevante para el término k. Sumando la puntuación para todas las entidades en V_q encontraríamos la bondad de k para V_q:

$$score(k,q,V_q) = \sum_{v \in V_q} tf_{k,v} \cdot idf_k \tag{2.14}$$

Como vemos, también se ha tenido en cuenta el peso de k con respecto a su posición.

3. Dependiente de la consulta.

   La *data cloud* se genera sobre los resultados de la consulta, no sobre un subconjunto aleatorio de la base de datos. Para ésto, en el cómputo de las puntuaciones de los términos candidatos sólo se tiene en cuenta la consulta inicial, con lo que la información contenida en la *data cloud* estará más próxima a las necesidades del usuario:

$$score(k,q,V_q) = \sum_{v \in V_q} (tf_{k,v} \cdot idf_k) \cdot score(v,q) \tag{2.15}$$

En el experimento realizado por Koutrika et al. (Kou09b), se demostró que este último método era el más preciso y el menos preciso el del enfoque basado en la popularidad, considerando la precisión como el número de términos relevantes en función del número total de términos en la nube.

Data Clouds a través de Cubos OLAP

Aouiche et al. (Aou09) proponen un método para crear las *data clouds* que consiste en pasar las entradas de la base de datos a cubos OLAP y de ahí a *tag clouds.*

Un cubo OLAP contiene un conjunto no vacío de dimensiones y otro de medidas, normalmente se derivan de una tabla de hechos donde cada dimensión y medida es una columna y todas las filas (o hechos) contienen tuplas con dimensión disjunta (Ver figura (2.4))

El cubo de datos soporta las siguientes operaciones:

- *Slice* (Cortar en rodajas) → Cuando solamente se está interesado en algunos atributos.

Dimensions				Measures	
location	time	salesman	product	cost	profit
Montreal	March	John	shoe	100$	10 $
Montreal	December	Smith	shoe	150$	30 $
Quebec	December	Smith	dress	175$	45 $
Ontario	April	Kate	dress	90$	10 $
Paris	March	John	shoe	100$	20 $
Paris	March	Marc	table	120$	10 $
Paris	June	Martin	shoe	120$	5 $
Lyon	April	Claude	dress	90$	10 $
New York	October	Joe	chair	100$	10 $
New York	May	Joe	chair	90$	10 $
Detroit	April	Jim	dress	90$	10 $

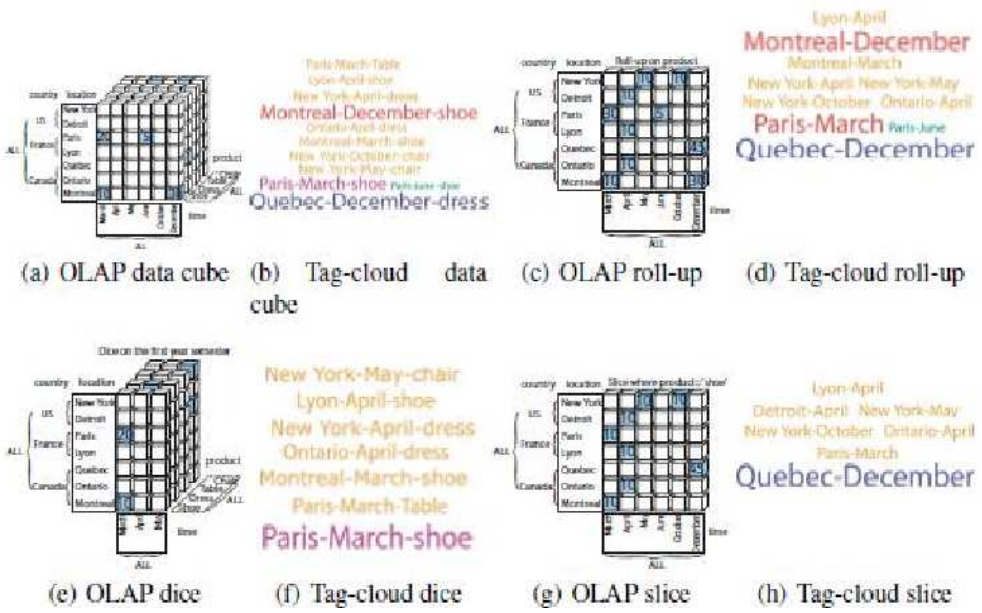

Figura 2.4: *Ejemplo de* Tag Clouds *a partir de Cubos OLAP (Aou09)*

- *Dice* (Cortar en dados) → Cuando se está interesado en un rango de los valores de algunos atributos.
- *Roll-up* (Enrollar) → Cuando se agregan los valores de los atributos.
- *Drill-down* (Desglosar o subdividir) → Es la operación contraria a la anterior.

Aouiche et al. obtienen las etiquetas o *tags* como términos composición de 3 palabras, donde cada palabra es una dimensión del cubo OLAP y los términos tienen un grosor de fuente proporcional a su importancia, evaluada esta importancia

mediante las columnas medida.

Desde que el cubo de datos soporta las operaciones indicadas arriba, las *tag clouds* también las soportarán ya que se generan de los nuevos *cuboids* obtenidos tras aplicar las operaciones al cubo de datos. En aquellos casos en que la operación haga pasar al cubo de un estado en que posee una dimensión mayor a otro estado en que posee otra dimensión menor (como sería agregar los valores de 2 atributos), también disminuirán las palabras presentes en el término composición.

La generación de la *tag cloud* a partir del cubo OLAP según el método propuesto por Aouiche et al. es sencilla. Como solamente pueden representarse un número *k* moderado de etiquetas, se buscarán las *k* celdas con mayores medidas en el cubo OLAP. Otra forma de hacerlo es estableciendo un límite para la frecuencia o este caso la medida, a partir del cual incluiremos la etiqueta en la *tag cloud*. En este último tipo de *tag cloud* el número de etiquetas presentes es variable.

Aouiche et al. establecen, después de realizar una serie de tests, que el número *k* no debe superar 150 etiquetas.

Intuitivamente, un paralepípedo que represente las mayores medidas proporcionará una *tag cloud* razonable.

Aouiche et al. también calculan medidas de similaridad entre etiquetas. Para ello el usuario especifica las dimensiones de *clustering* y se construye un paralepípedo con esas dimensiones. Por ejemplo si se elige la dimensión “País” en el ejemplo de arriba, Montreal-Abril y Toronto-Marzo se considerarían cercanos por lo que se incluirían en el mismo *cluster*. A los paralepípedos se les aplicaría alguna de las medidas de similaridad, como por ejemplo la similaridad del coseno.

2.1.8. Conclusiones

- **Referentes a la identificación del contenido**

  Resulta difícil comparar los tamaños de fuente y algunos trabajos señalan que las *tag clouds* son peores a la hora de reconocer palabras que una simple lista vertical ordenada alfabéticamente (Riv07).

  La mayor parte de las veces la visualización de la *tag cloud* se realiza manteniendo un orden alfabético entre las etiquetas mientras que los usuarios

ignoran este hecho según el estudio realizado por Hearst y Rosner (Hea08).

Diversos trabajos apuntan, que los problemas referentes a la identificación del contenido, se solventarían en gran medida con el uso de componentes multitérmino.

- **Referentes a la semántica**

  Las *tag clouds* han sido y son muy criticadas. Uno de los motivos es que los términos populares no tienen por qué ser los términos relevantes, lo que en ocasiones puede dificultar la búsqueda.

  Cuando sobre la *tag cloud* no se emplean técnicas de *clustering*, no es posible descubrir relaciones entre conceptos.

- **Referentes a la teoría**

  Desde una perspectiva tradicional sorprende la rápida adopción de las *tag clouds*, ya que conllevan algunos problemas teóricos, como es el hecho de no estar definidas matemáticamente.

  En algunos casos incluso, las *tag clouds* se usan con fines analíticos, como por ejemplo, examinar las diferencias entre discursos políticos, buscando patrones en el texto, lo que puede llevar a concepciones erróneas.

- **Referentes al método**

  No existe un método estándar para la extracción de etiquetas de una *tag cloud*, aunque en este trabajo se han presentado diversas técnicas de extracción, selección, *clustering* y herencia de etiquetas utilizadas por algunos autores. A pesar de eso, cada uno usa su propia técnica.

- **Referentes a su funcionalidad**

  Bar-Ilan et al. (BI08) sugieren orientar al usuario que va a asignar las etiquetas con una lista de descriptores, ya que demuestran que es más útil el

> etiquetado estructurado que el etiquetado libre.
>
> A pesar de las deficiencias teóricas, las *tag clouds* se han convertido en una herramienta de análisis.
>
> Se ha demostrado que las *tag clouds* son mejores para explorar que cuando la búsqueda es específica y que el uso de la *tag cloud* conlleva realizar más consultas para una búsqueda que el método tradicional por palabras clave (Sin08)
>
> Hearst y Rosner (Hea08) sugirieron que el propósito de las *tag clouds* era meramente social, ya que provee una atmósfera amigable y una forma de entrar en un sitio complejo. Aunque otros estudios demuestran que los usuarios prefieren especificar los términos de búsqueda a usar un sistema basado en etiquetado (Sin08).
>
> La Web 2.0 tiende a atraer a millones de usuarios que contribuyen a su contenido, por lo que las *tag clouds* pueden actuar como espejo de grupos o individuos, lo que las convierte en algo divertido en lugar de serio (Vie08).
>
> Cuando las *tag clouds* se usan fuera de sitios sociales suelen actuar como retratos de individuos antes que de grupos. Por ejemplo, una persona que sube el texto de 20 *blogs* que ha leído para crear una galería de instantáneas verbales. Cada nube se acompaña de un comentario, lo que revela la personalidad del *bloggero*.
>
> Viégas y Wattenberg (Vie08) afirman que las *tag clouds* son una técnica de lenguaje de la comunidad que funciona en la práctica aunque no en la teoría.

Se desconoce porque se siguen utilizando las *tag clouds* a pesar de que no proporcionan beneficios cuantificables y que los usuarios son incapaces de reconocer la forma en que las entradas se organizan en la visualización.

Sin embargo, la creciente demanda de *tag clouds* indica que existe una clase importante de datos que los usuarios quieren visualizar: el texto no estructurado.

En este trabajo, se pretende solventar gran parte de los inconvenientes de la *tag cloud*, permitiendo que las etiquetas presentes en esta, puedan estar formadas

por componentes multitérmino, en la línea que se apunta en la subsección (2.1.5).

Para ello, introduciremos a continuación la estructura-AP, haciendo un breve repaso a sus antecedentes, a los conjuntos-AP y a algunas de las propiedades de estos conjuntos y estructura.

En el siguiente capítulo, se establecerán los modelos matemáticos de la estructura-AP ponderada y la estructura multitérmino ponderada.

Empezaremos definiendo los *itemsets* ponderados que compondrán los conjuntos-AP ponderados, para definir posteriormente la estructura-AP ponderada y algunas de sus propiedades.

A continuación se definirán los conjuntos monotémino y multitérmino y se comentarán algunas propiedades de los conjuntos y estructuras multitérmino.

Posteriormente se definirá un modelo para la estructura monotérmino y multitérmino ponderada. Para esta última se establecerá la definición de secuencia de elementos ponderada o *item-seq* ponderada.

Finalmente se analizarán algunas propiedades de la estructura multitérmino ponderada y el acopliento de conjuntos multitérmino con estructuras multitérmino, considerando índices para la medida de este acoplamiento.

2.2. Estructura-AP

La estructura-AP nace del hecho de que la falta de estructura de los atributos textuales hace difícil su procesamiento automático para manejarlos de forma masiva.

La estructura-AP se basa en la transformación del atributo textual en una estructura intermedia que permita su representación de forma más estructurada. Dicha estructura está basada en el concepto de *itemset* frecuente y sus propiedades, llamado conjunto-AP.

Se utiliza esta forma de representación, ya que asumimos como hipótesis que los *itemsets* frecuentes mantienen la semántica subyacente en los atributos textuales, ya que mantienen agrupados los términos más relevantes. Una razón adicional para utilizar la estructura de *itemsets* frecuentes como base para nuestro modelo es que los algoritmos que se encargan de la obtención de estos *itemsets* son bien conocidos y han sido implementados en distintas variantes y están altamente optimizados.

La estructura-AP es un modelo multidimensional que facilita el procesamiento de la semántica básica que puede obtenerse de atributos textuales, con la ayuda de una estructura de almacenamiento. Esta representación estructurada pasa por una limpieza previa de los datos y por un proceso posterior de minería de textos. El resultado de este proceso conduce al concepto de estructura-AP.

Es importante señalar que los conceptos y operaciones pertenecientes a dicha representación estructurada y expuestos a continuación son referencias de la tesis doctoral del Dr. Sandro Martínez Folgoso (Mar08).

2.2.1. Concepto de Estructura-AP y Operaciones Asociadas

En esta subsección se comienza la exposición de las definiciones matemáticas bases para la representación formal de los datos. Inicialmente se establece la definición y propiedades de los conjuntos que tienen la propiedad “a priori” (Agr94) (llamados conjuntos-AP), y luego la definición de la estructura subyacente en los textos que es la de estructura-AP (Mar06), donde se captura la semántica básica que encierran los textos.

Aquí solamente van a darse algunas de las definiciones y propiedades, para más información consultar (Mar08).

Definición y Propiedades de los Conjuntos-AP

Definición 1.- Conjunto-AP

Sean $X = \{x_1 ... x_n\}$ un conjunto referencial de items y $R \subseteq P(X)$ un conjunto de itemsets frecuentes, siendo $P(X)$ las partes de X. Se dice que R es un conjunto-AP sí y sólo sí:

1.

$$\forall Z \in R \Rightarrow P(Z) \subseteq R \tag{2.16}$$

2. $\exists Y \in R$ *tal que* :

a)

$$card(Y) = max_{Z \in R}(card(Z))y$$
$$\nexists Y' \in R \mid card(Y') = card(Y) \tag{2.17}$$

b)

$$\forall Z \in R; Z \subseteq Y \tag{2.18}$$

El conjunto Y de cardinal maximal que caracteriza el conjunto-AP se denomina *conjunto generador de R*. Notaremos $R = g(Y)$, es decir $g(Y)$ será el conjunto-AP con conjunto generador Y. Llamaremos *Nivel de* $g(Y)$ al cardinal de Y. Obviamente, los conjuntos-AP de nivel 1 son los elemento de X, se considera el conjunto vacío $\emptyset$ como el conjunto-AP de nivel cero.

Ejemplo 1.- Conjunto-AP

Sea $X = \{rosa, blanco, negro, azul, rojo, verde, amarillo, violeta\}$
Sea $R = (\{azul\}, \{amarillo\}, \{violeta\}, \{azul, amarillo\}, \{azul, violeta\}, \{amarillo, violeta\}, \{azul, amarillo, violeta\})$.
Entonces el conjunto generador de R es $Y = (\{azul, amarillo, violeta\})$.

Como se observa en el ejemplo anterior y teniendo en cuenta la definición de conjunto-AP, el conjunto generador $Y = \{azul, amarillo, violeta\}$ se corresponde con el conjunto-AP de mayor cardinalidad y que incluye a su vez, todas las combinaciones presentes en R, tal y como se puede observar en la figura (2.5).

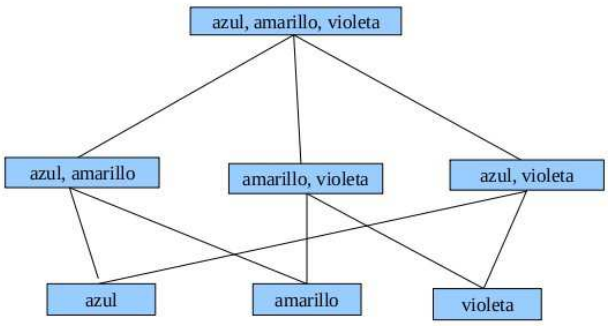

Figura 2.5: *Retículo del Conjunto-AP*

A continuación se dan algunas operaciones relacionadas con el conjunto-AP definido. Dichas operaciones serán utilizadas en la definición de otras operaciones como la obtención de la estructura global de conocimiento, que encierra la semántica básica de los datos procesados. Se comienza por la operación que verifica si un conjunto-AP está incluido en otro.

Definición 2.- Inclusión de Conjuntos-AP

Sea $R = g(Y)$ *y* $S = g(Y')$ *dos conjuntos-AP con el mismo conjunto referencial de items:*

$$R \subseteq S \Leftrightarrow Y \subseteq Y' \tag{2.19}$$

En la definición se puede apreciar que, los conjuntos-AP R y S estarán incluidos unos en otros si alguno de los dos conjuntos generadores Y o Y' está contenido uno en otro.

A continuación se introduce una operación importante en el contexto en el que se plantea este modelo, que es el *subconjunto-AP inducido* por un conjunto determinado. Esta operación se encargará de obtener el conjunto-AP particular que se genera, al intersecar el retículo global del conjunto-AP con un conjunto dado.

Definición 3.- Subconjunto-AP Inducido

Sea $R = g(Y)$ *e* $Y' \subseteq X$ *diremos que S es el* subconjunto-AP inducido por Y' *si y sólo si:*

$$S = g(R \bigcap Y') \tag{2.20}$$

Se puede apreciar en la definición que el subconjunto-AP inducido se obtiene de hacer la intersección de los conjuntos generadores de R y el conjunto Y'. Ésta intersección nunca será vacía porque el conjunto Y' es un subconjunto del conjunto referencial X.

Definición y Propiedades de la Estructura-AP

Los conceptos de conjunto-AP establecidos se usan para definir la estructura de información que se construye cuando se calculan itemsets frecuentes. Hay que tener en cuenta que estas estructuras se obtienen de forma constructiva, generando inicialmente itemsets con cardinal igual a 1; luego éstos se combinan para obtener los de cardinal 2, y así sucesivamente hasta obtener itemsets con cardinal maximal, con un soporte mínimo fijado. Por tanto, la estructura final es la de un conjunto de conjuntos-AP, que formalmente se define como estructura-AP (MB08).

Definición 4.- Estructura-AP

Sea $X = \{x_1 ... x_n\}$ *un conjunto referencial de items y* $S = \{A, B, ...\} \subseteq P(X)$ *un conjunto de itemsets frecuentes, tal que:*

$$\forall A, B \in S; A \not\subseteq B, B \not\subseteq A$$

Llamaremos estructura-AP del generador S, $T = g(A, B, ...)$, *al conjunto de conjuntos-AP cuyos conjuntos generadores son* $A, B, ...$

De la definición anterior queda claro entonces que: la estructura-AP no es más que una colección de conjuntos-AP. Tal como se definió, los conjuntos generadores de la estructura-AP $A, B, ...$ no pueden estar contenidos unos en otros, utilizando para esta interpretación la definición de conjunto-AP incluido que se dio anteriormente (ver ecuación 2.2.1). Entonces, la estructura-AP quedará constituida por todos los conjuntos generadores que se obtengan de las combinaciones de X presentes, dentro de todas las posibles ($P(X)$).

Hay que hacer notar que cualquier estructura-AP es un retículo de subconjuntos cuyos extremos superiores son sus conjuntos generadores. La figura (2.6) muestra una estructura-AP global que se define como $g(\{azul, amarillo, violeta\}, \{violeta, negro\})$. Se dan a continuación algunas definiciones y propiedades pertenecientes a estas nuevas estructuras.

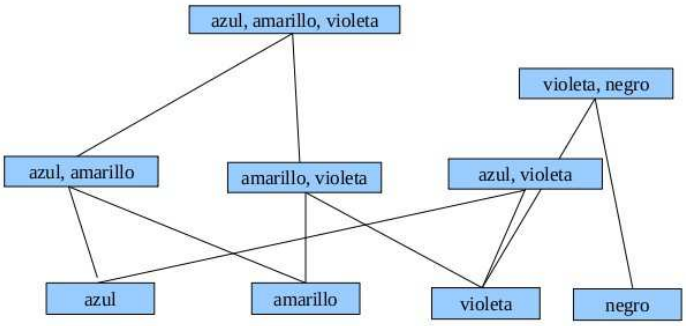

Figura 2.6: *Estructura-AP global*

Definición 5.- Inclusión de Estructuras-AP

Sean T_1, T_2, dos estructuras-AP con el mismo conjunto referencial de items:

$$\begin{aligned} T_1 \subseteq T_2 \Leftrightarrow \forall\ & R \text{ conjunto-AP de } T_1 \\ & \exists\ S \text{ conjunto-AP de } T_2 \mid R \subseteq S \end{aligned} \tag{2.21}$$

De esta definición se puede interpretar que para que una estructura-AP T_1 esté contenida en otra estructura-AP T_2, todos los conjuntos generadores de T_1 tienen que aparecer incluidos en alguno de los conjuntos generadores de T_2.

A continuación se introduce una importante operación sobre esta estructura-AP, la operación subestructura-AP inducida. Ésta no es más que la estructura-AP resultante de intersecar una estructura-AP cualquiera con un conjunto dado.

Definición 6.- Subestructura-AP Inducida

Sea la estructura-AP $T = g(A_1, A_2, ..., A_n)$ con conjunto referencial de items X y $Y \subseteq X$. Definiremos la estructura-AP de T inducida por Y *como:*

$$T' = T \bigwedge Y = g(B_1, B_2, ..., B_m) \tag{2.22}$$

donde

$$\begin{aligned} \forall B_i \in \{B_1, ..., B_m\} \quad \Rightarrow \quad & \exists A_j \in \{A_1, A_2, ..., A_n\} \\ tal\ que \qquad & B_i = A_j \bigcap Y \end{aligned} \tag{2.23}$$

$$\begin{array}{rcl} \forall A_j \in \{A_1,...,A_n\} & \Rightarrow & \exists B_i \in \{B_1,B_2,...,B_m\} \\ tal\ que & & A_j \bigcap Y \subseteq B_i \end{array} \quad (2.24)$$

Está claro que T' es la estructura-AP generada por aquellas intersecciones de Y con los conjuntos generadores T, que no están en contradicción con la definición de estructura-AP. El siguiente ejemplo expone estas ideas.

Ejemplo 2.- Subestructura-AP Inducida

Sea $X = \{rosa, blanco, negro, azul, rojo, verde, amarillo, violeta\}$
Sea T = g({azul, verde},{amarillo, azul},{rojo, blanco, negro},
{blanco, violeta, rosa}),
Sea Y = {rojo, blanco, violeta}

$$\Rightarrow T \bigwedge Y = g(\{rojo, blanco\}, \{blanco, violeta\}).$$

Acoplamiento de Conjuntos de Términos con las Estructuras-AP

En esta sección, serán establecidas las definiciones necesarias para consultar la base de datos, donde se cuenta con la estructura-AP como tipo de dato. La idea es que el usuario expresará sus requerimientos como conjuntos de términos, para ser consultados sobre los atributos textuales en la base de datos. Dado que dichos atributos estarán representados por sus estructuras-AP particulares, algunos tipos de acoplamientos tienen que ser dados para satisfacer las consultas sobre dichas estructuras.

Para hacerlo, contamos con dos enfoques distintos, que definimos a continuación.

Definición 7.- Acoplamiento Fuerte

Sea la estructura-AP $T = g(A_1, A_2, ..., A_n)$ *con conjunto referencial de items X y* $Y \subseteq X$. *Se define el acoplamiento fuerte entre Y y T como la operación lógica:*

$$Y \bigodot T = \begin{cases} verdadero\ si & \exists A_i \in \{A_1, A_2, ..., A_n\} \\ & / Y \subseteq A_i \\ falso & en\ otro\ caso \end{cases} \quad (2.25)$$

Definición 8.- Acoplamiento Débil

Sea la estructura-AP $T = g(A_1, A_2, ..., A_n)$ con conjunto referencial de items X y $Y \subseteq X$. Se define el acoplamiento débil entre Y y T como la operación lógica:

$$Y \bigoplus T = \begin{cases} \textit{verdadero si} & \exists A_i \in \{A_1, A_2, ..., A_n\} \\ & / Y \bigcap A_i \neq \emptyset \\ \textit{falso} & \textit{en otro caso} \end{cases} \tag{2.26}$$

Estas definiciones se pueden complementar dando alguna medida o índice que cuantifique estos acoplamientos. La idea es considerar que el acoplamiento de un conjunto grande de términos tendrá un índice mayor que uno con un menor número de términos; adicionalmente, si algún conjunto de términos se acopla con más de un conjunto generador, éste tendrá un índice mayor que el de otro, que sólo se acopla con un sólo conjunto. Pueden establecerse dos índices de acoplamiento, con lo que tenemos las definiciones siguientes.

Definición 9.- Índice de Acoplamiento Fuerte (Débil)

Sea la estructura-AP $T = g(A_1, A_2, ..., A_n)$ con conjunto referencial de items X y $Y \subseteq X$, se define el índice de acoplamiento fuerte (débil) entre Y y T como sigue:

$$\forall A_i \in \{A_1, A_2, ..., A_n\} \textit{se denota } m_i(Y) = card(Y \bigcap A_i)/card(A_i), \tag{2.27}$$

$$S = \{i \in \{1, ..., n\} | Y \subseteq A_i\}, \tag{2.28}$$

$$W = \{i \in \{1, ..., n\} | Y \bigcap A_i \neq \emptyset\} \tag{2.29}$$

Entonces se define el índice de acoplamiento fuerte y débil entre Y y T como:

$$\textbf{Índice fuerte} = S(Y|T) = \sum_{i \in S} m_i(Y)/n \tag{2.30}$$

$$\textbf{Índice débil} = W(Y|T) = \sum_{i \in W} m_i(Y)/n \tag{2.31}$$

Cumpliendo:

$$\forall Y \text{ y } T \text{ , } S(Y|T) \in [0,1] \text{ , } W(Y|T) \in [0,1] \text{ y } W(Y|T) \geq S(Y|T) \tag{2.32}$$

Capítulo 3

PROPUESTA TEÓRICA DE ESTRUCTURA-AP PONDERADA Y ESTRUCTURA MULTITÉRMINO PONDERADA

Con el propósito de representar la información en forma de *tag cloud*, lo que, como ya se ha visto, presenta grandes ventajas en la identificación del contenido de la información representada, la navegación a través de ésta, la depuración de la consulta, etc. y aprovechar la funcionalidad de la estructura-AP, se hace necesario ponderar la estructura-AP, para poder representar posteriormente los *itemsets* que la componen con distintos tamaños de fuente según su peso o frecuencia. Para ello definiremos lo que llamamos la "Estructura-AP Ponderada".

El principal inconveniente de la estructura-AP, es que no existe una relación de orden entre los términos que componen los *itemsets*, por lo que los resultados de las consultas representados mediante esta estructura, podrían no ser del todo precisos o devolver más información de la que se solicita. Para solventar este inconveniente, definiremos lo que llamamos la "Estructura Multitérmino", que, al igual que la estructura-AP, deberá ser ponderada con el fin de visualizarla en forma de *tag cloud*, creando lo que llamamos la "Estructura Multitérmino Ponderada".

Analizaremos las ventajas e incovenientes que poseen ambas estructuras en el ejemplo práctico de comparación del capítulo siguiente.

Empezaremos definiendo la estructura-AP ponderada.

3.1. Estructura-AP Ponderada

3.1.1. Definición de Estructura-AP Ponderada

Para definir las estructura-AP ponderada, empezaremos introduciendo los *itemsets* frecuentes ponderados:

Definición 10.- *Itemset* ponderado de un Conjunto-AP

Sea R = g(Y) un conjunto-AP con conjunto referencial de items X. Diremos que $\widetilde{I_t} \subseteq Y$ *es un itemset ponderado del conjunto Y si:*

$$\widetilde{I_t} = (I_t, \omega_t) \tag{3.1}$$

donde I_t es un conjunto de *items* y ω_t es el peso de éste, que en nuestro caso será igual a su frecuencia de aparición en el texto ($\omega_t \in \mathbb{N}$).

El peso o frecuencia de los itemsets de mayor grado, será menor o igual que el peso o frecuencia de los itemsets de grado menor.

$$Si\ I_1 \subseteq I_2 \Rightarrow \omega_1 \geq \omega_2 \tag{3.2}$$

Ejemplo 3.- *Itemsets* ponderados

Sea X $= \{$*negro, rosa, verde, azul, amarillo, naranja, violeta*$\}$
Sea R $= g(Y) = \{$*rosa, negro, violeta*$\}$

$$\begin{aligned} \Rightarrow \quad & \widetilde{I_1}(R) = \{rosa, (8)\}, \widetilde{I_2}(R) = \{negro, (10)\}, \widetilde{I_3}(R) = \{violeta, (16)\} \\ & \widetilde{I_4}(R) = \{rosa, negro, (8)\}, \widetilde{I_5}(R) = \{rosa, violeta, (6)\}, \\ & \widetilde{I_6}(R) = \{negro, violeta, (9)\}, \widetilde{I_7} = \{rosa, negro, violeta, (5)\} \end{aligned}$$

En este ejemplo se han extraído todos los posibles itemsets del conjunto R. El dígito al final de cada una de ellos indicaría su frecuencia de aparición en un texto hipotético, es decir, su peso.

Las estructuras-AP ponderadas son estructuras-AP (ver subsección(2.2.1)) compuestas por conjuntos-AP ponderados, siendo un conjunto-AP ponderado aquel que se compone solamente de *itemsets* ponderados.

Definición 11.- Estructura-AP Ponderada

Sea $X = \{x_1 ... x_n\}$ *un conjunto referencial de items y* $\widetilde{S} = \{\widetilde{A}, \widetilde{B}, ...\} \subseteq P(X)$ *un conjunto de itemsets frecuentes ponderados, tal que:*

$$\forall A, B \in S; A \not\subseteq B, B \not\subseteq A$$

Llamaremos estructura-AP ponderada del generador $\widetilde{S}$, $\widetilde{T} = g(\widetilde{A}, \widetilde{B}, ...)$, *al conjunto de conjuntos-AP cuyos conjuntos generadores son* $\widetilde{A}, \widetilde{B}, ...$

Nota.- El conjunto-AP generador $\widetilde{A}$ puede expresarse como $\widetilde{g}(A)$

Ejemplo 4.- Estructura-AP Ponderada

$$\begin{aligned}
Sea\ \widetilde{A} &= \widetilde{g}(\{negro, azul, amarillo\}) \\
&= (\{negro, azul, amarillo, (3)\}, \{negro, azul, (4)\}, \{azul, amarillo, (3)\}, \\
&\quad \{negro, amarillo(3)\}, \{negro, (7)\}, \{azul, (5)\}, \{amarillo, (4)\})\ \ y \\
Sea\ \widetilde{B^k} &= \widetilde{g}(\{negro, verde\} \\
&= (\{negro, verde, (2)\}, \{negro, (7)\}, \{verde, (3)\}),\ \ entonces: \\
\widetilde{T^k} &= \widetilde{g}(\{negro, azul, amarillo\}, \{negro, verde\}) \\
&= (\{negro, azul, amarillo, (3)\}, \{negro, azul, (4)\}, \{azul, amarillo, (3)\}, \\
&\quad \{negro, amarillo, (3)\}, \{negro, verde, (2)\}, \{negro, (7)\}, \{azul, (5)\}, \\
&\quad \{amarillo, (4)\}, \{verde, (3)\}
\end{aligned}$$

Visualización de la Estructura-AP Ponderada en forma de *Tag Cloud*.

En la figura (3.1) podemos ver la *tag cloud* de la estructura-AP ponderada del ejemplo anterior.

3.1.2. Propiedades de la Estructura-AP Ponderada

Inclusión de Estructuras-AP Ponderadas

Definición 12.- Inclusión de Estructuras-AP Ponderadas

Sean $\widetilde{T_1}$ *y* $\widetilde{T_2}$ *dos estructuras-AP ponderadas con el mismo conjunto referencial de items* X. *Diremos que* $\widetilde{T_1}$ *está incluida en* $\widetilde{T_2}$ $(\widetilde{T_1} \subseteq \widetilde{T_2})$ *si y sólo si:*

$$\begin{aligned}
&\widetilde{T_1} \subseteq \widetilde{T_2} \Leftrightarrow \forall\ \widetilde{R}\ conjunto\text{-}AP\ ponderado\ de\ \widetilde{T_1}, \\
&\exists\ \widetilde{S}\ conjunto\text{-}AP\ ponderado\ de\ \widetilde{T_2},\ tal\ que\ \widetilde{R} \subseteq \widetilde{S}
\end{aligned} \tag{3.3}$$

Figura 3.1: *Representación de la tag cloud de la estructura-AP del ejemplo 4*

Para aplicar esta propiedad, nos servimos de la ecuación (2.2.1) de la definición de inclusión de estructuras-AP que se vio en el capítulo anterior.

Subestructura-AP Ponderada Inducida

Definición 13.- Subestructura-AP Ponderada Inducida

Sea la estructura-AP ponderada $\widetilde{T} = g(\widetilde{A_1}, \widetilde{A_2}, ..., \widetilde{A_n})$ *con conjunto referencial de items* X *e* $Y \subseteq X$. *Definiremos la sub-estructura-AP ponderada de* T, $\widetilde{T'}$ *inducida por* Y *como:*

$$\widetilde{T'} = \widetilde{T} \wedge Y = g(\widetilde{B_1}, \widetilde{B_2}, ..., \widetilde{B_m}) \tag{3.4}$$

donde

$$\begin{aligned} \forall\, \widetilde{B_i} \in \{\widetilde{B_1}, ..., \widetilde{B_m}\} \quad &\Rightarrow \quad \exists\, \widetilde{A_j} \in \{\widetilde{A_1}, ..., \widetilde{A_n}\} \\ tal\ que \quad &\quad B_i = A_j \cap Y \end{aligned} \tag{3.5}$$

$$\begin{aligned} \forall\, \widetilde{A_j} \in \{\widetilde{A_1}, ..., \widetilde{A_n}\} \quad &\Rightarrow \quad \exists\, \widetilde{B_i} \in \{\widetilde{B_1}, ..., \widetilde{B_n}\} \\ tal\ que \quad &\quad A_j \cap Y \subseteq B_i \end{aligned} \tag{3.6}$$

$\widetilde{T'}$ es la estructura-AP generada por el acoplamiento de Y con los conjuntos generadores de $\widetilde{T}$. Para ello, se hace la intersección de cada *itemset* de Y con todos los *itemsets* generados por $(A_1, A_2, ..., A_n)$ de $\widetilde{T}$. Ver la intersección entre conjuntos en (Mar08).

En los conjuntos B_i de $\widetilde{T'}$ irán sólo los *itemsets* que no estén completamente incluidos en otro conjunto de B_i, ya que si no serían redundantes y la estructura-AP resultante no estaría constituida únicamente por conjuntos maximales.

Nota.- No tiene sentido que el conjunto-AP Y sea ponderado, ya que normalmente este conjunto representará los elementos introducidos en la consulta, por lo que el peso de los *itemsets* tras la intersección será el mismo que tuvieran en $\widetilde{T}$.

Aclararemos estas ideas con el siguiente ejemplo.

Ejemplo 5.- Subestructura-AP Ponderada Inducida

$$
\begin{aligned}
&\textit{Sea } X = \{\textit{negro, rosa, verde, azul, amarillo, naranja, violeta}\}\\
&\quad \textit{Sea } \widetilde{T} \;=\; \widetilde{g}(\{rosa, verde, amarillo\}, \{negro, amarillo\})\\
&\qquad\qquad =\; (\{rosa, verde, amarillo, (4)\}, \{rosa, verde, (6)\}, \{verde,\\
&\qquad\qquad\quad amarillo, (4)\}, \{rosa, amarillo, (7)\}, \{negro, amarillo, (2)\},\\
&\qquad\qquad\quad \{rosa, (7)\}, \{verde, (8)\}, \{amarillo, (7)\}, \{negro, (3)\}\\
&\quad \textit{Sea } \widetilde{Y} \;=\; g(\{rosa, negro, amarillo\})\\
&\Rightarrow \widetilde{T} \wedge Y \;=\; \widetilde{g}(\{rosa, amarillo\}, \{negro, amarillo\})\\
&\qquad\qquad =\; (\{rosa, amarillo, (7)\}, \{negro, amarillo, (2)\}, \{rosa, (7)\},\\
&\qquad\qquad\quad \{negro, (3)\}, \{amarillo, (7)\})
\end{aligned}
$$

El peso de los *itemsets* en la subestructura-AP ponderada inducida es el que tenían en la estructura-AP ponderada original. Como el peso es igual a la frecuencia, se recuperarán tantas entradas con la consulta como entradas haya en la base de datos conteniendo los elementos de la consulta. El resultado de la consulta sería la subestructura-AP ponderada inducida, el conjunto-AP contendría los términos de la consulta y la estructura-AP ponderada representaría la información contenida en la base de datos.

Superestructura-AP Ponderada Inducida

Definición 14.- Superestructura-AP Ponderada Inducida

Sea la estructura-AP ponderada $\widetilde{T} = g(\widetilde{A_1}, \widetilde{A_2}, ..., \widetilde{A_n})$ *con conjunto referencial de items* X *e* $Y \subseteq X$. *Definiremos la superestructura-AP ponderada de* T, $\widetilde{T'}$ *inducida por* $\widetilde{Y}$ *como:*

$$\widetilde{T'} = \widetilde{T} \vee \widetilde{Y} = g(\widetilde{B_1}, \widetilde{B_2}, ..., \widetilde{B_n}) \tag{3.7}$$

donde

$$\begin{array}{rcl} \forall\, \widetilde{B_i} \in \{\widetilde{B_1},...,\widetilde{B_m}\} & \Rightarrow & \exists\, \widetilde{A_j} \in \{\widetilde{A_1},...,\widetilde{A_n}\} \\ tal\ que & & B_i = A_j \cup Y \end{array} \tag{3.8}$$

$$\begin{array}{rcl} \forall\, \widetilde{A_j} \in \{\widetilde{A_1},...,\widetilde{A_n}\} & \Rightarrow & \exists\, \widetilde{B_i} \in \{\widetilde{B_1},...,\widetilde{B_n}\} \\ tal\ que & & A_j \cup Y \subseteq B_i \end{array} \tag{3.9}$$

donde

$$\omega_{I_t(\widetilde{T'})} = \begin{cases} \omega_{I_t(\widetilde{T})}\ si\ I_t(\widetilde{T'}) \in \widetilde{T}, \\ \omega_{I_t(\widetilde{Y})}\ si\ I_t(\widetilde{T'}) \in \widetilde{Y} \\ \omega_{I_t(\widetilde{T})} + \omega_{I_t(\widetilde{Y})}\ si\ I_t(\widetilde{T'}) \in \widetilde{T}, \widetilde{Y} \end{cases} \tag{3.10}$$

donde $\omega_{I_t(\widetilde{T'})}$ es el peso de los *itemsets* en $\widetilde{T'}$, $\omega_{I_t(\widetilde{Y})}$ es el peso de los *itemsets* generadas por $\widetilde{Y}$ y $\omega_{I_t(\widetilde{T})}$ es el peso de los *itemsets* en $\widetilde{T}$ generadas por A_j^k.

En este caso, los *itemsets* que estaban presentes tanto en $\widetilde{T}$ como en $\widetilde{Y}$, aparecerán en $\widetilde{T'}$ con peso igual a la suma de los pesos que presentaran en $\widetilde{T}$ y en $\widetilde{Y}$. Los *itemsets* que estuvieran sólo en $\widetilde{T}$ o $\widetilde{Y}$ conservarán su peso.

Para ver unión entre conjuntos consultar (Mar08).

Aquí sí tiene sentido que el conjunto-AP sea ponderado puesto que lo que estamos haciendo es añadir información a la base de datos.

Veamos ésto con un ejemplo.

Ejemplo 6.- Superestructura-AP Ponderada Inducida

Sea $X = \{$*negro, rosa, verde, azul, amarillo, naranja, violeta*$\}$

$$\begin{array}{rcl} Sea\ \widetilde{T} & = & \widetilde{g}(\{verde, amarillo\}) \\ & = & (\{verde, amarillo, (4)\}, \{verde, (8)\}, \{amarillo, (7)\}), \\ Sea\ \widetilde{Y} & = & \widetilde{g}(\{verde, negro\}) \\ & = & (\{verde, negro, (3)\}, \{verde, (4)\}, \{negro, (3)\}) \\ \Rightarrow \widetilde{T} \vee \widetilde{Y} & = & \widetilde{g}(\{verde, amarillo, negro\}) \\ & = & (\{verde, amarillo, negro, (1)\}, \{verde, amarillo, (4)\} \\ & = & \{verde, negro, (3)\}, \{amarillo, negro, (0)\}, \{verde, (12)\} \\ & = & \{amarillo, (7)\}, \{negro, (3)\} \end{array}$$

El peso de los nuevos itemsets que se generan y que no estaban en $\widetilde{T}$ ni en $\widetilde{Y}$ se calculará de nuevo, en base al texto, contabilizando la frecuencia de estos itemsets.

Unión de Estructuras-AP Ponderadas

Definición 15.- Unión de Estructuras-AP Ponderadas

Sean $\widetilde{T_1} = g(\widetilde{A_1}, \widetilde{A_2}, ..., \widetilde{A_n})$ y $\widetilde{T_2} = g(\widetilde{B_1}, \widetilde{B_2}, ..., \widetilde{B_m})$ dos estructuras-AP ponderadas, se define la unión como:

$$S = \widetilde{T_1} \cup \widetilde{T_2} = g(\widetilde{C_1}, \widetilde{C_2}, ..., \widetilde{C_l}) \tag{3.11}$$

verificando:

$$\begin{aligned} &\forall \widetilde{C_i} \in \{\widetilde{C_1}, ..., \widetilde{C_l}\};\ \exists\, \widetilde{A_p} \in \{\widetilde{A_1}, ..., \widetilde{A_n}\} \\ &\quad y\ \widetilde{B_q} \in \{\widetilde{B_1}, ..., \widetilde{B_m}\}\ /\ \widetilde{C_i} = \widetilde{A_p} \cap \widetilde{A_q} \end{aligned} \tag{3.12}$$

donde

$$\omega_{I_t(S)} = \begin{cases} \omega_{I_t(\widetilde{T_1})}\ si\ I_t(S) \in \widetilde{T_1}, \\ \omega_{I_t(\widetilde{T_2})}\ si\ I_t(S) \in \widetilde{T_2} \\ \omega_{I_t(\widetilde{T_1})} + \omega_{I_t(\widetilde{T_2})}\ si\ I_t(S) \in \widetilde{T_1}, \widetilde{T_2} \end{cases} \tag{3.13}$$

La unión de estructuras-AP ponderadas representaría la unión de la información contenida en dos bases de datos diferentes.

Ejemplo 7.- Unión de Estructuras-AP Ponderadas

Sea $X = \{$negro, rosa, verde, azul, amarillo, naranja, violeta$\}$
Sea $\widetilde{T_1} = \widetilde{g}(Y), Y =$($\{$rosa, verde$\}$,$\{$negro$\}$)
Sea $\widetilde{T_2} = \widetilde{g}(Y'), Y' =$($\{$naranja, verde$\}$,$\{$rosa$\}$)

$$\Rightarrow \widetilde{T_1} \cup \widetilde{T_2} = \widetilde{g}(\{rosa, verde, naranja\}, \{negro, naranja, verde\}, \{negro, rosa\})$$

Los pesos se calcularían igual que en el ejemplo anterior.

Intersección de Estructuras-AP Ponderadas

Definición 16.- Intersección de Estructuras-AP Ponderadas

Sean $\widetilde{T_1} = g(\widetilde{A_1}, \widetilde{A_2}, ..., \widetilde{A_n})$ *y* $\widetilde{T_2} = g(\widetilde{B_1}, \widetilde{B_2}, ..., \widetilde{B_m})$ *dos estructuras-AP ponderadas, se define la intersección de* $\widetilde{T_1}$ *y* $\widetilde{T_2}$ *como:*

$$S = \widetilde{T_1} \cap \widetilde{T_2} = g(\widetilde{C_1}, \widetilde{C_2}, ..., \widetilde{C_l}) \tag{3.14}$$

verificando:

$$\begin{aligned} &\forall \widetilde{C_i} \in \{\widetilde{C_1}, ..., \widetilde{C_l}\};\ \exists\, \widetilde{A_p} \in \{\widetilde{A_1}, ..., \widetilde{A_n}\} \\ &\quad y\ \widetilde{B_q} \in \{\widetilde{B_1}, ..., \widetilde{B_m}\}\ /\ \widetilde{C_i} = \widetilde{A_p} \cap \widetilde{A_q} \end{aligned} \tag{3.15}$$

donde

$$\omega_{I_t(S)} = \omega_{I_t(\widetilde{T_1})} + \omega_{I_t(\widetilde{T_2})} \tag{3.16}$$

El peso de los *itemsets* de la intersección entre $\widetilde{T_1}$ y $\widetilde{T_2}$ será la suma del peso que tuvieran en $\widetilde{T_1}$ y $\widetilde{T_2}$ respectivamente. La intersección entre dos estructuras-AP ponderadas representa los resultados comunes a la consulta que habría en ambas bases de información, por lo que se sumaría la frecuencia de los *itemsets* en ambas estructuras.

Ejemplo 8.- Intersección de Estructuras-AP Ponderadas

Sea $X = \{$*negro, rosa, verde, azul, amarillo, naranja, violeta*$\}$

$$\begin{aligned} \textit{Sea } \widetilde{T_1} &= \widetilde{g}(\{rosa, verde, amarillo\}, \{negro, amarillo\}) \\ &= (\{rosa, verde, amarillo, (4)\}, \{rosa, verde, (6)\}, \{verde, \\ &\quad amarillo, (4)\}, \{rosa, amarillo, (5)\}, \{negro, amarillo, (2)\}, \\ &\quad \{rosa, (7)\}, \{verde, (8)\}, \{amarillo, (7)\}, \{negro, (3)\} \\ \textit{Sea } \widetilde{T_2} &= \widetilde{g}(\{rosa, negro, amarillo\}) \\ &= (\{rosa, negro, amarillo, (3)\}, \{rosa, negro, (4)\}, \{negro, ama- \\ &\quad rillo, (3)\}, \{rosa, amarillo, (4)\}, \{rosa, (5)\}, \{negro, (6)\}, \\ &\quad \{amarillo, (5)\}) \\ \Rightarrow \widetilde{T_1} \cap \widetilde{T_2} &= \widetilde{g}(\{rosa, amarillo\}, \{negro, amarillo\}) \\ &= (\{rosa, amarillo, (9)\}, \{negro, amarillo, (5)\}, \{rosa, (12)\}, \\ &\quad \{negro, (9)\}, \{amarillo, (12)\}) \end{aligned}$$

3.2. Estructura Multitérmino Ponderada

La estructura multitérmino es un modelo multidimensional similar a la estructura-AP, pero que discrimina entre términos según el orden que éstos presenten en el texto.Al igual que la estructura-AP, esta representación pasará por una limpieza previa de los datos y por un proceso posterior de minería de textos.

Para introducir la estructura multitérmino ponderada, empezaremos definiendo la estructura multitérmino y para introducir ésta, los conjuntos multitérmino. Además,puesto que, la estructura monotérmino es la que más se ve en la web representada en forma de *tag cloud* monotérmino (a pesar de carecer de un modelo matemático subyacente), nos parece importante darle una definición y también a los conjuntos monotérmino, para ver similitudes y diferencias con los conjuntos y estructura multitérmino.

3.2.1. Definición de Conjunto Monotérmino y Conjunto Multitérmino

Empezaremos definiendo los componentes monotérmino y multitérmino de los conjuntos monotérmino y multitérmino, para entender como se generan estos conjuntos.

Componente Monotérmino

Definición 18.- Componente Monotérmino

Sea $X = \{x_1, x_2, ..., x_n\}$ un conjunto referencial de items. $P(X)$ es el conjunto de las partes de X o de las posibles combinaciones de elementos de X. Diremos que $y \in P(X)$ es una componente monotérmino si:

$$y = (x_i) \quad \forall i \in [1, n] \tag{3.17}$$

El cardinal de una componente es igual a la cantidad de elementos del conjunto X que hay en la componente, por lo que el cardinal de las componentes monotérmino es igual a uno $(card(y) = 1)$.

Conjunto Monotérmino

Definición 19.- Conjunto Monotérmino

Sea $X = \{x_1, x_2, ..., x_n\}$ un conjunto referencial de items. $P(X)$ es el conjunto de las partes de X. Se dice que $R \subseteq P(X)$ conjunto de itemsets frecuentes es un conjunto monotérmino si:

$$\begin{aligned} \exists \quad & y \in R \quad t.q. \quad y = (x_i) \quad para\ cualquier \quad i \in [1,n] \\ \nexists \quad & z \in R \quad t.q. \quad card(z) \geq card(y) \end{aligned} \tag{3.18}$$

Componente *k*-término o Multitérmino

Definición 20.- Componente *k*-término o Multitérmino

Sea $X = \{x_1, x_2, ..., x_n\}$ un conjunto referencial de items. $P(X)$ es el conjunto de las partes de X. Diremos que $y \in P(X)$ es una componente k-término o multitérmino si consta de k elementos diferentes del conjunto X, con $2 \leq k \leq n$.

"y" componente k*-término o multitérmino si y sólo si:*

$$card(y) = k \tag{3.19}$$

Conjunto Multitérmino

Definición 21.- Conjunto Multitérmino

Sea $X = \{x_1, x_2, ..., x_n\}$ un conjunto referencial de items y $R \subseteq P(X)$ un conjunto de itemsets frecuentes, siendo $P(X)$ las partes de X. Diremos que R es un conjunto multitérmino si:

1.

$$\begin{aligned} \forall Z \quad = \quad & z_1, z_2, ..., z_k \in R \Rightarrow \\ \Rightarrow \quad & \begin{cases} (z_1, z_2, ..., z_{k-1}) \in R \\ (z_2, z_3, ..., z_k) \in R \end{cases} para \quad 2 \leq k \leq n \end{aligned} \tag{3.20}$$

2. $\exists Y \in R$ *tal que:*

a*)*

$$card(Y) = max_{\,Z \in R}\,(card(Z)) \quad y$$
$$\nexists\, Y' \in R \mid card(Y') = card(Y) \tag{3.21}$$

b*)*

$$\forall Z \in R; Z \subseteq Y \tag{3.22}$$

El conjunto Y de máxima cardinalidad caracteriza al conjunto multitérmino y será llamado conjunto generador de R. Se denotará $R = g(Y)$ y significa que $g(Y)$ será el conjunto multitérmino con conjunto generador Y.

Se denotará grado de $g(Y)$ al cardinal de Y, es decir, a la cantidad de elementos del conjunto generador Y. Los conjuntos multitérmino de grado uno son, como hemos visto, conjuntos monotérmino y son los elementos de X. El conjunto multitérmino de grado cero es el conjunto vacío $\emptyset$.

Ejemplo 9.- Conjunto Multitérmino

Sea $X = \{$*negro, rosa, verde, azul, amarillo, naranja, violeta*$\}$.
Sea $R = \{$*(negro, azul, amarillo, rosa), (negro, azul, amarillo), (azul, amarillo, rosa), (negro, azul), (azul, amarillo), (amarillo, rosa), (negro), (azul), (amarillo), (rosa)*$\}$.

R sería un conjunto multitérmino de X, siendo el conjunto generador Y el conjunto de mayor cardinalidad, Y = (negro,azul,amarillo,rosa), estando todos los demás conjuntos incluidos en éste y cumpliendo (3.20):

$$\forall Z = (z_1, z_2, ..., z_k) \in R \Rightarrow \begin{cases} (z_1, z_2, ..., z_{k-1}) \in R \\ (z_2, z_3, ..., z_k) \in R \end{cases} \; para \quad 2 \leq k \leq n$$

Para ver mejor que se cumple esta propiedad, en la figura (3.2) se representa el retículo de este conjunto multitérmino.

Vemos que el conjunto Y de mayor cardinalidad es la raíz del árbol y en las hojas aparecen los elementos de los que se compone el conjunto Y. En los niveles intermedios están los conjuntos con grado intermedio entre el grado 1 y el grado del conjunto generador, de forma escalonada.

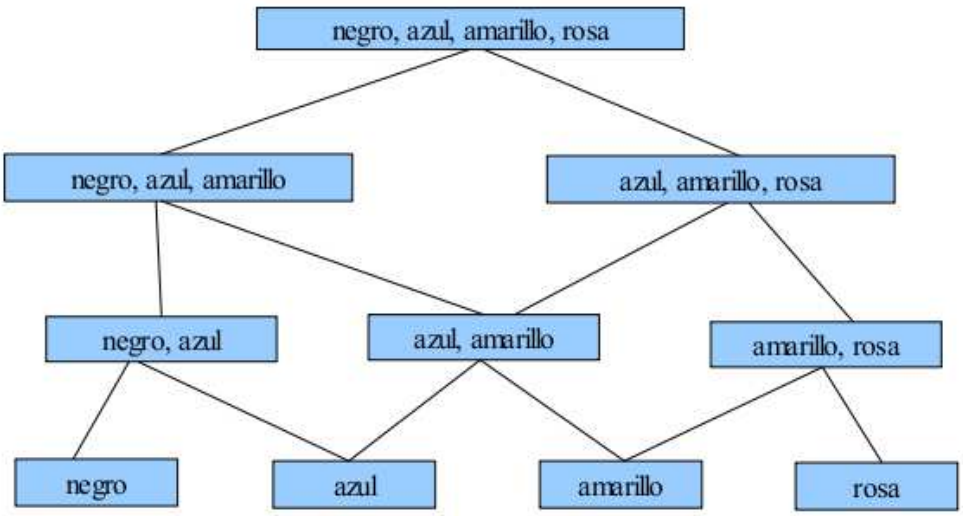

Figura 3.2: *Retículo del Conjunto Multitérmino Y={negro, azul, amarillo, rosa}*

Los conceptos de conjunto monotérmino y multitérmino establecidos se usan para definir la estructura de información que se construye en base a éstos o estructura monotérmino/multitérmino. Para ello se calculan las secuencias frecuentes de elementos, a las que llamaremos a partir de ahora *"item-seqs" (item-sequencies)* frecuentes. La estructura multitérmino, al igual que la estructura-AP, se obtiene de forma constructiva, generando inicialmente las *item-seqs* frecuentes con cardinal igual a uno, según un soporte mínimo establecido; luego éstas se combinan para obtener las de cardinal dos, y así sucesivamente hasta obtener las *item-seqs* frecuentes con cardinal maximal, como veremos más adelante en la subsección (3.3.2), donde se explica con este fin, una pequeña modificación del algoritmo APriori (3.3.1). Por tanto, la estructura final es la de un conjunto de conjuntos monotémino o multitérmino, que formalmente se define como estructura monotérmino o estructura multitérmino según corresponda.

La definición de estas estructuras se introduce a continuación para comprender el funcionamiento de algunas de las operaciones entre conjuntos que darán como resultado una estructura. Posteriormente se hablará de la estructura monotérmino y multitérmino ponderadas.

3.2.2. Definición de Estructura Monotérmino y Estructura Multitérmino

Definición 22.- Estructura Monotérmino

Una estructura monotérmino es una estructura obtenida a partir de conjuntos monotérmino, es decir, será un conjunto de conjuntos monotérmino o teniendo en cuenta que los conjuntos monotérmino son los elementos, un conjunto de elementos.

Sea $X = \{x_1, x_2, ..., x_n\}$ *un conjunto referencial de items y* $S = \{A^1, B^1, ...\} \subseteq P(X)$ *un conjunto de item-seqs frecuentes de grado uno, tal que:*

$$A^1 \neq B^1$$

Llamaremos estructura monotérmino generada por S, $E^1 = g(A^1, B^1, ...)$ *al conjunto de conjuntos monotérmino cuyos conjuntos generadores son* $A^1,\ B^1, ...$

Ejemplo 10.- Estructura Monotérmino

Sea $X = \{$*negro, rosa, verde, azul, amarillo, naranja, violeta*$\}$
Sea $A^1 =$ *(negro)*
Sea $B^1 =$ *(verde)*
Sea $C^1 =$ *(naranja)*
Sea $D^1 =$ *(violeta)*
$\Rightarrow E^1 = g(A^1, B^1, C^1, D^1) = \{$*(negro), (verde), (naranja), (violeta)*$\}$

Definición 23.- Estructura Multitérmino

La estructura multitérmino es una estructura generada a partir de conjuntos multitérmino.

Sea $X = \{x_1, x_2, ..., x_n\}$ *un conjunto referencial de items y* $S = \{A^k, B^k, ...\} \subseteq P(X)$ *un conjunto de item-seqs frecuentes de grado mayor o igual que uno (*$k \geq 1$*), siendo P(X) las partes de X y* $A^k, B^k, ...$ *conjuntos multitérmino tales que:*

$$\forall\ A^k, B^k \in S;\ A^k \nsubseteq B^k, B^k \nsubseteq A^k \text{ y } B^k \neq A^k$$

Llamaremos estructura multitérmino generada por S, $E^k = g(A^k, B^k, ...)$*, al conjunto de conjuntos multitérmino cuyos conjuntos generadores son* $A^k, B^k, ...$

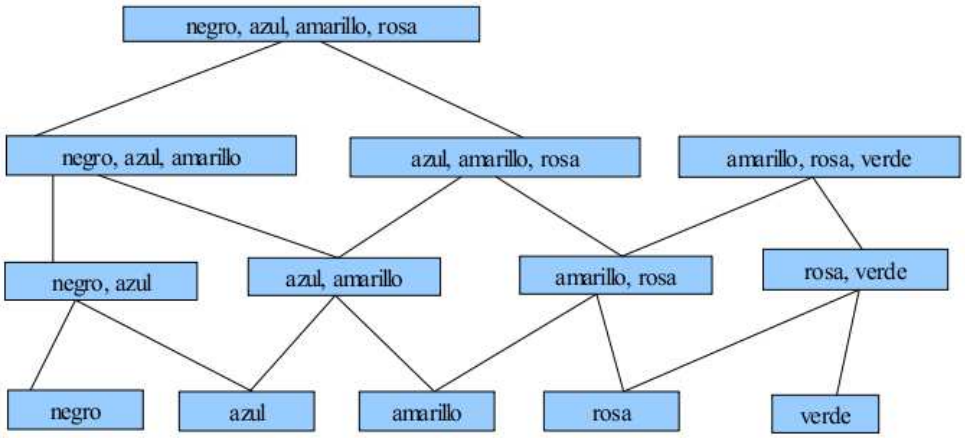

Figura 3.3: *Estructura Multitérmino*

Ejemplo 11.- Estructura Multitérmino

En el retículo de la figura (3.3) podemos ver la estructura multitérmino formada por los conjuntos multitérmino:

A^k = *g({negro, azul, amarillo, rosa}) y*
B^k = *g({amarillo, rosa, verde}), o lo que es lo mismo:*

$$E^k = g(\{negro, azul, amarillo, rosa\},\{amarillo, rosa, verde\})$$

Ninguno de estos dos conjuntos A^k y B^k está completamente incluido en el otro, aunque compartan elementos comunes.

3.2.3. Algunas Propiedades de los Conjuntos y Estructuras Multitérmino

Inclusión de Conjuntos Multitérmino

Definición 24.- Inclusión de Conjuntos Multitérmino

Sea $R = g(Y)$ un conjunto multitérmino con conjunto referencial de items $X = \{x_1, .., x_n\}$.
Sea $S = g(Y')$ otro conjunto multitérmino con el mismo conjunto referencial X.

Se dice que R está incluido en S si y sólo si:

$$\begin{array}{l} R \subseteq S \Leftrightarrow \ \exists i, j \in [1,k],\ k \in [1,n] / \\ Y = (y_i,...,y_j),\ Y' = (y'_1,..,y'_i,..,y'_j,...,y'_k) / \\ (y'_i,..,y'_j) = (y_i,..,y_j) \end{array} \tag{3.23}$$

Es decir, un conjunto *R* estará incluído en otro conjunto *S*, cuando exista una secuencia de elementos exactamente igual en el conjunto generador de *S* al conjunto generador de *R*.

Ejemplo 12.- Inclusión de Conjuntos Multitérmino

$Sea\ X = \{negro, rosa, verde, azul, amarillo, naranja, violeta\}$
$Sea\ R = g(Y),\ Y = \{azul, verde, negro, amarillo\}$
$Sea\ S = g(Y'),\ Y' = \{verde, negro, amarillo\}$
$Sea\ T = g(Y''),\ Y'' = \{azul, negro, amarillo\}$

$$\Rightarrow S \subseteq R,\ T \nsubseteq R$$

Veamos:

R = {*(azul, verde, negro, amarillo), (azul, verde, negro), (verde, negro, amarillo), (azul, verde), (verde, negro), (negro, amarillo), (azul), (verde), (negro), (amarillo)*}

S ={*(verde, negro, amarillo), (verde, negro), (negro, amarillo), (verde), (negro), (amarillo)*}

T = {*(azul, negro, amarillo), (azul, negro), (negro, amarillo), (azul), (negro), (amarillo)*}

Todos los elementos del conjunto S están dentro del conjunto R, pero no todos los elementos del conjunto T están dentro de R, como es el caso del elemento (azul, negro), que está en T, pero no está en R, a pesar de que todos los elementos del conjunto generador de T, Y'', aparecen en el conjunto generador de R, Y.

Acoplamiento de Conjuntos Multitérmino

Para definir el acoplamiento estableceremos previamente la definición formal de secuencia de elementos.

Definición 25.- Secuencia de Elementos o *item-seq* de un Conjunto Multitérmino

Sea R = g(Y) un conjunto multitérmino con conjunto referencial de items X, donde $Y = (y_1,...,y_i,..,y_j,..,y_k)$. *Diremos que* $\alpha_t \subseteq Y$ *es una secuencia de elementos del conjunto Y si:*

$$\exists i,j \in [1,k] \; tal \; que \; \alpha_t = (y_i,..,y_j) \tag{3.24}$$

con $t \in [1, \sum_1^k k]$ donde $\sum_1^k k \geq 3$, ya que $\sum_1^k k$ es el número máximo de secuencias o *item-seqs* que se pueden formar con los elementos del con junto Y y el número mínimo de *item-seqs* que se pueden formar con un conjunto multitérmino será 3 (para los conjuntos multitérmino de grado 2).

Todos los elementos de cada secuencia α_t aparecerán en Y y conservarán el mismo orden.

Ejemplo 13.- Secuencia de Elementos o *item-seq* de un Conjunto Multitérmino

Sea $X = \{negro, rosa, verde, azul, amarillo, naranja, violeta\}$
Sea $R = g(Y) = \{azul, verde\}$
Sea $S = g(Y) = \{verde, amarillo, rosa\}$

$$\Rightarrow \quad \alpha_1(R) = \{azul\}, \alpha_2(R) = \{verde\}, \alpha_3(R) = \{azul.verde\}$$

$$\begin{aligned} \Rightarrow \quad & \alpha_1(S) = \{verde\}, \alpha_2(S) = \{amarillo\}, \alpha_3(S) = \{rosa\}, \\ & \alpha_4(S) = \{verde, amarillo\}, \alpha_5(S) = \{amarillo, rosa\} \\ & \alpha_6(S) = \{verde, amarillo, rosa\} \end{aligned}$$

Definición 26.- Acoplamiento de Conjuntos Multitérmino

Sean R y S conjuntos multitérmino con conjunto referencial X. $R = g(Y)$, *donde* $Y = (y_1,..,y_k)$. $S = g(Y')$, *donde* $Y' = (y'_1,..,y'_{k'}) \forall \; k,k' \in [1,n]$.

Se define el acoplamiento entre R y S $(R \frown S)$ *como la estructura multitérmino* $M = g(\beta_a, \beta_b, ...)$, *tal que:*

1.

$$\forall\ \beta_a \in \{\beta_1,...,\beta_s\} \quad \Rightarrow \quad \exists\ \alpha_t, \alpha'_{t'} \in Y, Y'\ respectivamente, \\ tal\ que \quad \beta_a = \alpha_t = \alpha'_{t'} \tag{3.25}$$

2.

$$\forall\ \alpha_t \in Y = \alpha'_{t'} \in Y' \Rightarrow \exists\ \beta_a \in \{\beta_1,..,\beta_n\}\ tal\ que\ \alpha_t \subseteq \beta_a\ \alpha'_{t'} \subseteq \beta_a \tag{3.26}$$

donde α_t es una item-seq o secuencia de elementos de Y y $\alpha'_{t'}$ es una secuencia de elementos de Y'.

Es decir, el acoplamiento es igual a la estructura generada por las *item-seqs* coincidentes en Y e Y', eliminando aquellas que no sean maximales.

Ejemplo 14.- Acoplamiento de Conjuntos Multitérmino

Sea $X = \{$negro, rosa, verde, azul, amarillo, naranja, violeta$\}$
Sea $R = g(Y)$, $Y = \{$rosa, verde, azul, negro, amarillo, naranja$\}$
Sea $S = g(Y')$, $Y' = \{$rosa, verde, violeta, amarillo, naranja$\}$

Acoplamiento o item-seqs coincidentes:

$\beta_a = (y_1, y_2) = (y'_1, y'_2) =$ *(rosa, verde)*
$\beta_b = (y_4, y_5) = (y'_4, y'_5) =$ *(amarillo, naranja)*
$\Rightarrow M = R \frown S = g(\{rosa, verde\}, \{amarillo, naranja\}) = \{(rosa, verde),$
$(amarillo, naranja), (rosa), (verde), (amarillo), (naranja)\}$

Unión de Conjuntos Multitérmino

Definición 27.- Unión de Conjuntos Multitérmino

Sea $R = g(Y)$ y $S = g(Y')$. Se define la unión de R y S como la estructura multitérmino U generada por Y e Y'

$$U = g(R \cup S) = g(Y, Y') \tag{3.27}$$

Ejemplo 15.- Unión de Conjuntos Multitérmino

Sea $X = \{$negro, rosa, verde, azul, amarillo, naranja, violeta$\}$
Sea $R = g(Y), Y = \{$azul, verde, negro, amarillo$\}$

Sea $S = g(Y')$, $Y' = \{$*rosa, verde, negro, azul*$\}$

$g(Y \cup Y') =$ *g(*$\{$*azul, verde, negro, amarillo*$\}$, $\{$*rosa, verde, negro, azul*$\}$*)=* $\{$*(azul, verde, negro, amarillo), (rosa, verde, negro, azul), (azul, verde, negro), (verde, negro, amarillo), (rosa, verde, negro), (verde, negro, azul), (azul, verde), (verde, negro), (negro, amarillo), (rosa, verde), (negro, azul), (azul), (verde), (negro), (amarillo), (rosa)*$\}$

Subestructura Multitérmino Inducida

Definición 28.- Subestructura Multitérmino Inducida

Sea $R = g(Y)$ *e* $Y' \subseteq X$*, diremos que S es la subestructura multitérmino inducida por* Y' *si y sólo si:*

$$S = g(Y \frown Y') \tag{3.28}$$

es decir, S es una estructura multitérmino inducida por Y' si y sólo si S está generada por las *item-seqs* coincidentes en Y e Y', o dicho de otra forma, por el acoplamiento entre Y e Y' (ver ecuaciones (3.25) y (3.26)).

Ejemplo 16.- Subestructura Multitérmino Inducida

Sea $X = \{$*negro, rosa, verde, azul, amarillo, naranja, violeta*$\}$
Sea $R = g(Y)$, $Y = \{$*rosa, verde, azul, negro, amarillo, naranja*$\}$
Sea $Y' = \{$*rosa, violeta, negro, amarillo*$\}$

Acoplamiento o secuencias coincidentes:
$\beta_a = (y_1) = (y'_1) =$ *(rosa)*
$\beta_b = (y_4, y_5) = (y'_3, y'_4) =$ *(negro, amarillo)*
$\Rightarrow$ $S = g(Y \frown Y') =$ *g(*$\{$*rosa*$\}$, $\{$*negro,amarillo*$\}$*) =* $\{$*(negro,amarillo), (negro), (amarillo), (rosa)*$\}$

Superestructura Multitérmino Inducida

Definición 29.- Superestructura Multitérmino Inducida

Sea $R = g(Y)$ *e* $Y' \subseteq X$ *diremos que V es la superestructura multitérmino inducida por* Y' *si y sólo si:*

$$V = g(Y \cup Y') \tag{3.29}$$

La estructura V será la generada por la unión de Y e Y', siendo la unión la que se ha definido anteriormente (ver ecuación (3.27)).

Ejemplo 17.- Superestructura Multitérmino Inducida

Sea $X = \{$*negro, rosa, verde, azul, amarillo, naranja, violeta*$\}$
Sea $R = g(Y)$, $Y = \{$*rosa, verde*$\}$
Sea $Y' = \{$*rosa, violeta, negro*$\}$
$\Rightarrow$ $V =$ *g(*$\{$*rosa, verde*$\}$,$\{$*rosa, violeta, negro*$\}$*)*

3.2.4. Definición de Estructura Monotérmino y Estructura Multitérmino ponderadas

Para definir las estructuras monotérmino y multitérmino ponderadas, empezaremos introduciendo las *item-seqs* frecuentes ponderadas:

Definición 30.- Secuencia de Elementos o *item-seq* ponderada de un Conjunto Multitérmino

Sea R = g(Y) un conjunto multitérmino con conjunto referencial de items X. Diremos que $\widetilde{\alpha}_t \subseteq Y$ *es una item-seq ponderada del conjunto Y si:*

$$\widetilde{\alpha}_t = (\alpha_t, \omega_t) \tag{3.30}$$

donde α_t es la secuencia definida en la ecuación (3.24) y ω_t es el peso de ésta. Como las secuencias serán extraídas del texto, el peso representa la frecuencia de aparición de la *item-seq* en él y:

El peso o frecuencia de las item-seqs de mayor grado, será menor o igual que el peso o frecuencia de las item-seqs de grado menor.

$$Si\ \alpha_1 \subseteq \alpha_2 \Rightarrow \omega_1 \geq \omega_2 \tag{3.31}$$

Ejemplo 18.-*Item-seq* ponderada de un Conjunto Multitérmino

Sea $X = \{$*negro, rosa, verde, azul, amarillo, naranja, violeta*$\}$
Sea $R = g(Y) = \{$*rosa, negro, violeta*$\}$

$$\Rightarrow \quad \widetilde{\alpha_1}(R) = \{rosa, (8)\}, \widetilde{\alpha_2}(R) = \{negro, (5)\}, \widetilde{\alpha_3}(R) = \{violeta, (5)\}$$
$$\widetilde{\alpha_4}(R) = \{rosa, negro, (3)\}, \widetilde{\alpha_5}(R) = \{negro, violeta, (4)\}$$
$$\widetilde{\alpha_6}(R) = \{rosa, negro, violeta, (2)\}$$

En este ejemplo se han extraído todas las posibles item-seqs del conjunto R. El dígito al final de cada una de ellas indicaría su frecuencia de aparición en un texto hipotético, es decir, su peso.

Estructura Monotérmino Ponderada

Las estructuras monotérmino ponderadas, son estructuras monotérmino (ver sección (3.2.2)) en las que cada conjunto monotérmino de la estructura estará formado por una *item-seq* ponderada de grado uno. Estos conjuntos monotérmino se llamarán a su vez conjuntos monotérmino ponderados.

Definición 31.- Estructura Monotérmino Ponderada

Sea $X = \{x_1, x_2, ..., x_n\}$ *un conjunto referencial de items y* $\widetilde{S} = \{\widetilde{A^1}, \widetilde{B^1}, ...\} \subseteq P(X)$ *un conjunto de item-seqs frecuentes ponderadas de grado uno, tal que:*

$$\widetilde{A^1} \neq \widetilde{B^1}$$

Llamaremos estructura monotérmino generada por $\widetilde{S}$, $\widetilde{E^1} = g(\widetilde{A^1}, \widetilde{B^1}, ...)$ *al conjunto de conjuntos monotérmino ponderados cuyos conjuntos generadores son* $\widetilde{A^1}$, $\widetilde{B^1}, ...$

Cada conjunto $\widetilde{A^1} = (A^1, (\omega_{A^1}))$ donde A^1 es el conjunto monotérmino y ω_{A^1} el peso o la frecuencia de ese conjunto.

Ejemplo 19.- Estructura Monotérmino Ponderada

Sea $X = \{$*negro, rosa, verde, azul, amarillo, naranja, violeta*$\}$
Sea $\widetilde{A^1} = (A^1, (\omega_{A^1}))$= *(negro, (5))*
Sea $\widetilde{B^1} = (B^1, (\omega_{B^1}))$= *(verde, (2))*
Sea $\widetilde{C^1} = (C^1, (\omega_{C^1}))$= *(naranja, (2))*
Sea $\widetilde{D^1} = (D^1, (\omega_{D^1}))$= *(violeta, (8))*

$$\begin{aligned} \Rightarrow \widetilde{E^1} &= g(\widetilde{A^1},\widetilde{B^1},\widetilde{C^1},\widetilde{D^1}) \\ &= \{(negro,(5)),(verde,(2)),(naranja,(2)),(violeta,(8))\} \end{aligned}$$

Visualización de la Estructura Monotérmino Ponderada en forma de *Tag Cloud*.

En la figura (3.4) podemos ver la *tag cloud* de la estructura monotérmino ponderada del ejemplo anterior.

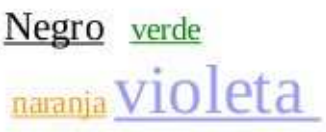

Figura 3.4: *Representación de la tag cloud de la estructura monotérmino del ejemplo 13*

Estructura Multitérmino Ponderada

Las estructuras multitérmino ponderadas son estructuras multitérmino (ver sección(3.2.2)) compuestas por conjuntos multitérmino ponderados, siendo un conjunto multitérmino ponderado aquel que se compone solamente de *item-seqs* ponderadas.

Definición 32.- Estructura Multitérmino Ponderada

Sea $X = \{x_1, x_2, ..., x_n\}$ *un conjunto referencial de items y* $\widetilde{S} = \{\widetilde{A^k}, \widetilde{B^k}, ...\} \subseteq P(X)$ *un conjunto de item-seqs frecuentes ponderadas de grado mayor o igual que uno (*$k \geq 1$*), siendo P(X) las partes de X y* $\widetilde{A^k}, \widetilde{B^k}, ...$ *conjuntos multitérmino ponderados tales que:*

$$\forall\, A^k, B^k \in S;\;\; A^k \nsubseteq B^k,\;\; B^k \nsubseteq A^k \;\; y \;\; B^k \neq A^k$$

Llamaremos estructura multitérmino ponderada generada por $\widetilde{S}$, $\widetilde{E^k} = g(\widetilde{A^k}, \widetilde{B^k}, ...)$, *al conjunto de conjuntos multitérmino cuyos conjuntos generadores son* $\widetilde{A^k}, \widetilde{B^k}, ...$

Nota.- El conjunto multitérmino generador $\widetilde{A^k}$ puede expresarse como $\widetilde{g}(A^k)$

Ejemplo 20.- Estructura Multitérmino Ponderada

$$\begin{aligned}
Sea\ \widetilde{A^k} &= \widetilde{g}(\{negro, azul, amarillo\}) \\
&= (\{negro, azul, amarillo, (3)\}, \{negro, azul, (4)\}, \{azul, amarillo, (3)\}, \\
&\quad \{negro, (7)\}, \{azul, (5)\}, \{amarillo, (4)\})\ \ y \\
Sea\ \widetilde{B^k} &= \widetilde{g}(\{negro, verde\} \\
&= (\{negro, verde, (2)\}, \{negro, (7)\}, \{verde, (3)\}),\ \ entonces: \\
\widetilde{E^k} &= \widetilde{g}(\{negro, azul, amarillo\}, \{negro, verde\}) \\
&= (\{negro, azul, amarillo, (3)\}, \{negro, azul, (4)\}, \{azul, amarillo, (3)\}, \\
&\quad \{negro, verde, (2)\}, \{negro, (7)\}, \{azul, (5)\}, \{amarillo, (4)\}, \{verde, (3)\}
\end{aligned}$$

Visualización de la Estructura Multitérmino Ponderada en forma de *Tag Cloud*.

En la figura (3.5) podemos ver la *tag cloud* de la estructura multitérmino ponderada del ejemplo anterior.

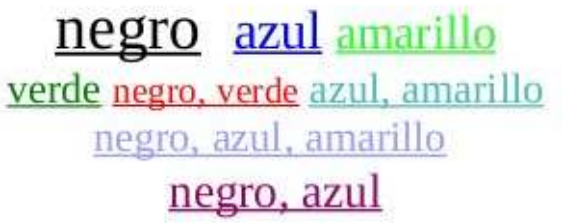

Figura 3.5: *Representación de la tag cloud de la estructura multitérmino del ejemplo 20*

3.2.5. Propiedades de la Estructura Multitérmino Ponderada

Inclusión de Estructuras Multitérmino Ponderadas

Definición 33.- Inclusión de Estructuras Multitérmino Ponderadas

Sean $\widetilde{E_1^k}$ y $\widetilde{E_2^k}$ dos estructuras multitérmino ponderadas con el mismo conjunto referencial de items X. Diremos que $\widetilde{E_1^k}$ está incluida en $\widetilde{E_2^k}$ ($\widetilde{E_1^k} \subseteq \widetilde{E_2^k}$) si y sólo si:

$$\widetilde{E_1^k} \subseteq \widetilde{E_2^k} \Leftrightarrow \forall\, \widetilde{R}\ conjunto\ multitermino\ ponderado\ de\ \widetilde{E_1^k},$$
$$\exists\, \widetilde{S}\ conjunto\ multitermino\ ponderado\ de\ \widetilde{E_2^k},\ tal\ que\ \widetilde{R} \subseteq \widetilde{S} \qquad (3.32)$$

Para aplicar esta propiedad, nos servimos de la ecuación (3.23) de la definición de inclusión de conjuntos multitérmino que se vio en la subsección(3.2.3).

Ejemplo 21.- Inclusión de Estructuras Multitérmino Ponderadas

Sea X = {negro, rosa, verde, azul, amarillo, naranja, violeta}
$\widetilde{E_1^k} = \widetilde{g}$ *({rosa, amarillo, naranja},{azul, amarillo}, {verde})*
$\widetilde{E_2^k} = \widetilde{g}$ *({rosa, amarillo}, {verde})*
$\widetilde{E_3^k} = \widetilde{g}$ *({rosa, naranja}, {amarillo})*

$$\Rightarrow \widetilde{E_3^k} \nsubseteq \widetilde{E_1^k},\ pero\ \widetilde{E_2^k} \subseteq \widetilde{E_1^k}$$

La tercera estructura multitérmino ponderada $\widetilde{E_3^k}$ no está incluida en $\widetilde{E_1^k}$ porque el conjunto {*(rosa, naranja)*}, no está incluido en ninguno de los conjuntos de $\widetilde{E_1^k}$, ya que aunque sus elementos aparecen en el primer conjunto de $\widetilde{E_1^k}$, no son elementos adyacentes.

Subestructura Multitérmino Ponderada Inducida

Definición 34.- Subestructura Multitérmino Ponderada Inducida

Sea la estructura multitérmino ponderada $\widetilde{E^k} = g(\widetilde{A_1^k}, \widetilde{A_2^k}, ..., \widetilde{A_n^k})$ con conjunto referencial de items X e $Y \subseteq X$. Definiremos la sub-estructura multitérmino ponderada $\widetilde{E^{k'}}$ inducida por Y como:

$$\widetilde{E^{k'}} = \widetilde{E^k} \wedge Y = g(\widetilde{B_1^k}, \widetilde{B_2^k}, ..., \widetilde{B_m^k}) \qquad (3.33)$$

donde

$$\begin{array}{rcl} \forall\, \widetilde{B_i} \in \{\widetilde{B_1},...,\widetilde{B_m}\} & \Rightarrow & \exists\, \widetilde{A_j} \in \{\widetilde{A_1},...,\widetilde{A_n}\} \\ tal\ que & & B_i = A_j \frown Y \end{array} \tag{3.34}$$

$$\begin{array}{rcl} \forall\, \widetilde{A_j} \in \{\widetilde{A_1},...,\widetilde{A_n}\} & \Rightarrow & \exists\, \widetilde{B_i} \in \{\widetilde{B_1},...,\widetilde{B_n}\} \\ tal\ que & & A_j \frown Y \subseteq B_i \end{array} \tag{3.35}$$

$\widetilde{E^{k'}}$ es la estructura multitérmino generada por el acoplamiento de Y con los conjuntos generadores de $\widetilde{E^k}$. Para ello, se hace el acoplamiento de cada *item-seq* de Y con todas las *item-seqs* generadas por $(A_1^k, A_2^k, ..., A_n^k)$ de $\widetilde{E^k}$, tal como vimos en las ecuaciones (3.25) y (3.26). En los conjuntos B_i^k de $\widetilde{E^{k'}}$ irán sólo las *item-seqs* que no estén completamente incluidas en otro conjunto de B_i^k, ya que si no serían redundantes y la estructura multitérmino resultante no estaría constituida únicamente por conjuntos maximales.

Nota.- No tiene sentido que el conjunto multitérmino Y sea ponderado, ya que normalmente este conjunto representará los elementos introducidos en la consulta, por lo que el peso de las *item-seqs* tras el acoplamiento será el mismo que tuvieran en $\widetilde{E^k}$.

Ilustraremos estas ideas con el siguiente ejemplo.

Ejemplo 22.- Subestructura Multitérmino Ponderada Inducida

Sea $X = \{$*negro, rosa, verde, azul, amarillo, naranja, violeta*$\}$

$$\begin{array}{rcl} Sea\ \widetilde{E^k} & = & \widetilde{g}(\{rosa, verde, amarillo\}, \{negro, amarillo\}) \\ & = & (\{rosa, verde, amarillo, (4)\}, \{rosa, verde, (6)\}, \{verde, \\ & & amarillo, (4)\}, \{negro, amarillo, (2)\}, \{rosa, (7)\}, \{verde, (8)\}, \\ & & \{amarillo, (7)\}, \{negro, (3)\} \\ Sea\ \widetilde{Y} & = & g(\{rosa, negro, amarillo\}) \\ \Rightarrow \widetilde{E^K} \frown Y & = & \widetilde{g}(\{rosa\}, \{negro, amarillo\}) \\ & = & (\{rosa, (7)\}, \{negro, amarillo, (2)\}, \{negro, (3)\}, \{amarillo, (7)\}) \end{array}$$

El peso de las *item-seqs* en la sub-estructura multitérmino ponderada inducida es el que tenían en la estructura multitérmino ponderada original. Como el peso es igual a la frecuencia, se recuperarán tantas entradas con la consulta como entradas haya en la base de datos conteniendo los elementos de la consulta. El resultado de la consulta sería la sub-estructura multitérmino ponderada inducida, el conjunto multitérmino contendría los términos de la consulta y la estructura multitérmino ponderada representaría la información contenida en la base de datos.

Superestructura Multitérmino Ponderada Inducida

Definición 35.- Superestructura Multitérmino Ponderada Inducida

Sea la estructura multitérmino ponderada $\widetilde{E^k} = g(\widetilde{A_1^k}, \widetilde{A_2^k}, ..., \widetilde{A_n^k})$ *con conjunto referencial de items* X *e* $Y \subseteq X$*. Definiremos la super-estructura multitérmino ponderada* $\widetilde{E^{k'}}$ *inducida por* $\widetilde{Y}$ *como:*

$$\widetilde{E^{k'}} = \widetilde{E^k} \vee \widetilde{Y} = g(\widetilde{A_1^k}, \widetilde{A_2^k}, ..., \widetilde{A_n^k}, \widetilde{Y}) \tag{3.36}$$

donde

$$\omega_{\alpha_t(\widetilde{E^{k'}})} = \begin{cases} \omega_{\alpha_t(\widetilde{E^k})} \text{ si } \alpha_t(\widetilde{E^{k'}}) \in \widetilde{E^k}, \\ \omega_{\alpha_t(\widetilde{Y})} \text{ si } \alpha_t(\widetilde{E^{k'}}) \in \widetilde{Y} \\ \omega_{\alpha_t(\widetilde{E^k})} + \omega_{\alpha_t(\widetilde{Y})} \text{ si } \alpha_t(\widetilde{E^{k'}}) \in \widetilde{E^k}, \widetilde{Y} \end{cases} \tag{3.37}$$

donde $\omega_{\alpha_t(\widetilde{E^{k'}})}$ es el peso de las *item-seqs* en $\widetilde{E^{k'}}$, $\omega_{\alpha_t(\widetilde{Y})}$ es el peso de las *item-seqs* generadas por $\widetilde{Y}$ y $\omega_{\alpha_t(\widetilde{E^k})}$ es el peso de las *item-seqs* en $\widetilde{E^k}$ generadas por A_j^k.

En este caso, las *item-seqs* que estaban presentes tanto en $\widetilde{E^k}$ como en $\widetilde{Y}$, aparecerán en $\widetilde{E^{k'}}$ con peso igual a la suma de los pesos que presentaran en $\widetilde{E^k}$ y en $\widetilde{Y}$. Las *item-seqs* que estuvieran sólo en $\widetilde{E^k}$ o $\widetilde{Y}$ conservarán su peso.

Al igual que en superestructura-AP ponderada inducida, aquí sí tiene sentido que el conjunto multitérmino sea ponderado puesto que lo que estamos haciendo es añadir información a la base de datos.

Ejemplo 23.- Superestructura Multitérmino Ponderada Inducida

Sea $X = \{$*negro, rosa, verde, azul, amarillo, naranja, violeta*$\}$

$$\begin{aligned}
\text{Sea } \widetilde{E^k} &= \widetilde{g}(\{rosa, verde, amarillo\}) \\
&= (\{rosa, verde, amarillo, (4)\}, \{rosa, verde, (6)\}, \{verde, amarillo, \\
&\quad (4)\}, \{rosa, (7)\}, \{verde, (8)\}, \{amarillo, (7)\}), \\
\text{Sea } \widetilde{Y} &= \widetilde{g}(\{rosa, verde, negro\}) \\
&= (\{rosa, verde, negro, (2)\}, \{rosa, verde, (3)\}, \{verde, negro, (3)\}, \\
&\quad \{rosa, (5)\}, \{verde, (4)\}, \{negro, (3)\}) \\
\Rightarrow \widetilde{E^K} \vee \widetilde{Y} &= \widetilde{g}(\{rosa, verde, amarillo\}, \{rosa, verde, negro\})
\end{aligned}$$

$$\begin{aligned}
&= (\{rosa, verde, amarillo(4)\}, \{rosa, verde, negro, (2)\}, \\
&\quad \{rosa, verde, (9)\}, \{verde, amarillo, (4)\}, \{verde, negro, (3)\} \\
&\quad \{rosa, (12)\}, \{verde, (12)\}, \{amarillo, (7)\}, \{negro, (3)\})
\end{aligned}$$

Como vemos, la *item-seq (verde)* aparecía tanto en $\widetilde{E^k}$ como en $\widetilde{Y}$ y su peso era ocho y cuatro respectivamente, por lo que el peso de la *item-seq verde* en la super-estructura multitérmino ponderada inducida es doce.

Propiedad 1

Sea la estructura multitérmino ponderada $\widetilde{E^k} = g(\widetilde{A_1^k}, \widetilde{A_2^k}, ..., \widetilde{A_n^k})$ *con conjunto referencial de items X e Y,* $Y' \subseteq X$*. Entonces:*

$$(\widetilde{E^k} \wedge Y) \wedge Y' = \widetilde{E^k} \wedge (Y \frown Y') \tag{3.38}$$

$$(\widetilde{E^k} \vee \widetilde{Y}) \vee \widetilde{Y'} = \widetilde{E^k} \vee (\widetilde{Y} \cup \widetilde{Y'}) \tag{3.39}$$

Ejemplo 24.-

Sea $X = \{$*negro, rosa, verde, azul, amarillo, naranja, violeta*$\}$
Sea $\widetilde{E^k} = \widetilde{g}(Y)$, $Y =$*(*$\{$*rosa, verde, azul, violeta*$\}$, $\{$*negro, amarillo*$\}$,$\{$*naranja, amarillo*$\}$*)*
Sea $Y' =$*(*$\{$*rosa, violeta, negro, amarillo*$\}$*)*
Sea $Y'' =$ *(*$\{$*azul, naranja, violeta, negro, rosa*$\}$*)*

$\Rightarrow (\widetilde{E^k} \wedge Y') \wedge Y'' = \widetilde{g}(\{rosa\},\{violeta\},\{negro,amarillo\}) \wedge \widetilde{g}(\{azul,naranja,$ $violeta,negro,rosa\}) = \widetilde{g}(\{rosa\},\{violeta\},\{negro\})$

Por otro lado:

$\Rightarrow \widetilde{E^k} \wedge (Y' \frown Y'') = \widetilde{g}(\{rosa,verde,violeta\},\{negro,amarillo\},\{naranja,$ $amarillo\}) \wedge \widetilde{g}(\{rosa\},\{violeta,negro\}) = \widetilde{g}(\{rosa\},\{violeta\},\{negro\})$

Ejemplo 25.-

Sea $X = 1,2,...,9$
Sea $\widetilde{E^k} = \widetilde{g}(Y), Y = (\{1,2,5\},\{2,3\})$
Sea $\widetilde{Y'} = (\{1,5\})$
Sea $\widetilde{Y''} = (\{5,3\})$

$\Rightarrow (\widetilde{E^k} \vee \widetilde{Y'}) \vee \widetilde{Y''} = \widetilde{g}(\{1,2,5\},\{1,5\},\{2,3\}) \vee \widetilde{g}(\{5,3\}) =$
$= \widetilde{g}\{(1,2,5),(1,5),(2,3),(5,3)\}$

Por otro lado:

$\Rightarrow (\widetilde{Y'} \cup \widetilde{Y''}) = \widetilde{g}(\{1,5\}),\{5,3\})$
$\Rightarrow \widetilde{E^k} \vee (\widetilde{Y'} \cup \widetilde{Y''}) = \widetilde{g}(\{1,2,5\},\{2,3\}) \vee \widetilde{g}(\{1,5\},\{5,3\}) =$
$= \widetilde{g}(\{1,2,5\},\{2,3\},\{1,5\},\{5,3\})$

Unión de Estructuras Multitérmino Ponderadas

Definición 36.- Unión de Estructuras Multitérmino Ponderadas

Sean $\widetilde{E_1^k} = g(\widetilde{A_1^k},\widetilde{A_2^k},...,\widetilde{A_n^k})$ *y* $\widetilde{E_2^k} = g(\widetilde{B_1^k},\widetilde{B_2^k},...,\widetilde{B_m^k})$ *dos estructuras multitérmino ponderadas, se define la unión como:*

$$S = \widetilde{E_1^k} \cup \widetilde{E_2^k} = g(\widetilde{A_1^k},\widetilde{A_2^k},...,\widetilde{A_n^k},\widetilde{B_1^k},\widetilde{B_2^k},...,\widetilde{B_m^k}) \tag{3.40}$$

verificando:

$$\omega_{\alpha_t(S)} = \begin{cases} \omega_{\alpha_t(\widetilde{E_1^k})} \text{ si } \alpha_t(S) \in \widetilde{E_1^k}, \\ \omega_{\alpha_t(\widetilde{E_2^k})} \text{ si } \alpha_t(S) \in \widetilde{E_2^k} \\ \omega_{\alpha_t(\widetilde{E_1^k})} + \omega_{\alpha_t(\widetilde{E_2^k})} \text{ si } \alpha_t(S) \in \widetilde{E_1^k},\widetilde{E_2^k} \end{cases} \tag{3.41}$$

La unión de estructuras multitérmino ponderadas representaría la unión de la información contenida en dos bases de datos diferentes.

Ejemplo 26.- Unión de Estructuras Multitérmino Ponderadas

Sea $X = \{$*negro, rosa, verde, azul, amarillo, naranja, violeta*$\}$
Sea $\widetilde{E_1^k} = \widetilde{g}(Y), Y =$*(*$\{$*rosa, verde*$\}$,$\{$*negro*$\}$*)*
Sea $\widetilde{E_2^k} = \widetilde{g}(Y'), Y' =$*(*$\{$*naranja, verde*$\}$,$\{$*rosa*$\}$*)*

$$\Rightarrow \widetilde{E_1^k} \cup \widetilde{E_2^k} = \widetilde{g}(\{rosa, verde\}, \{naranja, verde\}, \{negro\})$$

Como se puede ver, no se ha incluido el conjunto *(rosa),* ya que éste forma parte de otro conjunto maximal, el $\{$*rosa, verde*$\}$.

Acoplamiento de Estructuras Multitérmino Ponderadas

Definición 37.- Acoplamiento de Estructuras Multitérmino Ponderadas

Sean $\widetilde{E_1^k} = g(\widetilde{A_1^k}, \widetilde{A_2^k}, ..., \widetilde{A_n^k})$ *y* $\widetilde{E_2^k} = g(\widetilde{B_1^k}, \widetilde{B_2^k}, ..., \widetilde{B_m^k})$ *dos estructuras multitérmino ponderadas, se define el acoplamiento de* $\widetilde{E_1^k}$ *y* $\widetilde{E_2^k}$ *como el conjunto generado por:*

$$S = \widetilde{E_1^k} \frown \widetilde{E_2^k} = g(\widetilde{C_1^k}, \widetilde{C_2^k}, ..., \widetilde{C_h^k}) \tag{3.42}$$

verificando:

$$\begin{aligned} \forall \widetilde{C_i^k} \in \{\widetilde{C_1^k}, ..., \widetilde{C_h^k}\};\ \exists\, \widetilde{A_p} \in \{\widetilde{A_1^k}, ..., \widetilde{A_n^k}\} \\ y\ \widetilde{B_q} \in \{\widetilde{B_1^k}, ..., \widetilde{B_m^k}\}\ /\ \widetilde{C_i^k} = \widetilde{A_p^k} \frown \widetilde{A_q^k} \end{aligned} \tag{3.43}$$

donde

$$\omega_{\alpha_t(S)} = \omega_{\alpha_t(\widetilde{E_1^k})} + \omega_{\alpha_t(\widetilde{E_2^k})} \tag{3.44}$$

El peso de las *item-seqs* del acoplamiento entre $\widetilde{E_1^k}$ y $\widetilde{E_2^k}$ será la suma del peso que tuvieran en $\widetilde{E_1^k}$ y $\widetilde{E_2^k}$ respectivamente. El acoplamiento entre dos estructuras multitérmino ponderadas representa los resultados comunes a la consulta que habría en ambas bases de datos, por lo que se sumaría la frecuencia de las *item-seqs* en ambas bases.

Ejemplo 27.- Acoplamiento de Estructuras Multitérmino Ponderadas

Sea $X = \{negro, rosa, verde, azul, amarillo, naranja, violeta\}$

$$\begin{aligned}
Sea\ \widetilde{E_1^k} &= \widetilde{g}(\{rosa, verde, amarillo\}, \{negro, amarillo\}) \\
&= (\{rosa, verde, amarillo, (4)\}, \{rosa, verde, (6)\}, \{verde, \\
&\quad amarillo, (4)\}, \{negro, amarillo, (2)\}, \{rosa, (7)\}, \{verde, (8)\}, \\
&\quad \{amarillo, (7)\}, \{negro, (3)\} \\
Sea\ \widetilde{E_2^k} &= \widetilde{g}(\{rosa, negro, amarillo\}) \\
&= (\{rosa, negro, amarillo, (3)\}, \{rosa, negro, (4)\}, \{negro, ama- \\
&\quad rillo, (3)\}, \{rosa, (5)\}, \{negro, (6)\}, \{amarillo, (5)\}) \\
\Rightarrow \widetilde{E^K} \frown \widetilde{E_2^k} &= \widetilde{g}(\{rosa\}, \{negro, amarillo\}) \\
&= (\{rosa, (12)\}, \{negro, amarillo, (5)\}, \{negro, (9)\}, \{amarillo, \\
&\quad (12)\})
\end{aligned}$$

No incluimos la *item-seq (verde)* porque existe otro conjunto maximal que la contiene, la *item-seq (rosa, verde).*

Propiedad 2

Sean $\widetilde{E_1^k} = g(\widetilde{A_1^k}, \widetilde{A_2^k}, ..., \widetilde{A_n^k})$ *y* $\widetilde{E_2^k} = g(\widetilde{B_1^k}, \widetilde{B_2^k}, ..., \widetilde{B_m^k})$ *dos estructuras multitérmino ponderadas con el mismo conjunto referencial de items* X *e* $Y \subseteq X \Rightarrow$

$$Y \wedge (\widetilde{E_1^k} \frown \widetilde{E_2^k}) = (Y \wedge \widetilde{E_l^k}) \frown (Y \wedge \widetilde{E_2^k}) \tag{3.45}$$

Ejemplo 28.-

Sea $X = \{negro, rosa, verde, azul, amarillo, naranja, violeta\}$
Sea $\widetilde{E_1^k} = \widetilde{g}(Y'), Y' = (\{rosa, verde, azul\}, \{negro\})$
Sea $\widetilde{E_2^k} = \widetilde{g}(Y''), Y'' = (\{naranja, verde, negro\}, \{rosa, verde\})$
Sea $Y = (\{azul, verde, negro\})$

$\Rightarrow \widetilde{E_1^k} \frown \widetilde{E_2^k} = \widetilde{g}(\{negro\}, \{rosa, verde\})$
$\Rightarrow Y \wedge (\widetilde{E_1^k} \frown \widetilde{E_2^k}) = \widetilde{g}(\{negro\}, \{verde\})$

Por otro lado:

$\Rightarrow (Y \wedge \widetilde{E_1^k}) = \widetilde{g}(\{azul\},\{verde\},\{negro\})$
$\Rightarrow (Y \wedge \widetilde{E_2^k}) = \widetilde{g}(\{verde,negro\})$
$\Rightarrow (Y \wedge \widetilde{E_1^k}) \frown (Y \wedge \widetilde{E_2^k}) = \widetilde{g}(\{verde\},\{negro\})$

Propiedad 3

Sean $\widetilde{E_1^k} = g(\widetilde{A_1^k},\widetilde{A_2^k},...,\widetilde{A_n^k})$ *y* $\widetilde{E_2^K} = g(\widetilde{B_1^k},\widetilde{B_2^k},...,\widetilde{B_m^k})$ *dos estructuras multitérmino ponderadas con el mismo conjunto referencial de items* X *e* $Y \subseteq X \Rightarrow$

$$Y \wedge (\widetilde{E_1^k} \cup \widetilde{E_2^k}) = (Y \wedge \widetilde{E_1^k}) \cup (Y \wedge \widetilde{E_2^k}) \tag{3.46}$$

Ejemplo 29.-

Sea $X = \{negro, rosa, verde, azul, amarillo, naranja, violeta\}$
Sea $\widetilde{E_1^k} = \widetilde{g}(Y'), Y' = (\{rosa,verde,violeta\},\{negro\})$
Sea $\widetilde{E_2^k} = \widetilde{g}(Y''), Y'' = (\{naranja,verde,negro,violeta\},\{rosa,verde\})$
Sea $Y = (\{azul,verde,negro\})$

$\Rightarrow \widetilde{E_1^k} \cup \widetilde{E_2^k} = \widetilde{g}(\{rosa,verde,violeta\},\{naranja,verde,negro,violeta\})$
$\Rightarrow Y \wedge (\widetilde{E_1^k} \cup \widetilde{E_2^k}) = \widetilde{g}(\{verde,negro\})$

Por otro lado:

$\Rightarrow (Y \wedge \widetilde{E_1^k}) = \widetilde{g}(\{verde\},\{negro\})$
$\Rightarrow (Y \wedge \widetilde{E_2^k}) = \widetilde{g}(\{verde,negro\})$
$\Rightarrow (Y \wedge \widetilde{E_1^k}) \cup (Y \wedge \widetilde{E_2^k}) = \widetilde{g}(\{verde,negro\})$

3.2.6. Consultando la Base de Datos: Acoplamiento de Conjuntos Multitérmino con Estructuras Multitérmino.

Acoplamientos Fuerte y Débil

Definición 38.- Acoplamiento Fuerte

Sea la estructura multitérmino $E_1^k = g(A_1^k,A_2^k,...,A_n^k)$ *con conjunto referencial de items* X *e* $Y \subseteq X$. *Se define el acoplamiento fuerte entre* Y *y* E^k *como la opera-*

ción lógica:

$$Y \odot E^k = \begin{cases} verdadero & si\ \exists A_i^k \in \{A_1^k, ..., A_n^k\} / Y \subseteq A_i^k \\ falso & en\ otro\ caso \end{cases} \tag{3.47}$$

El acoplamiento fuerte será, por lo tanto, el que tendrá un conjunto Y con la estructura multitérmino E^k cuando dicho conjunto aparezca completamente incluido en alguno de los conjuntos generadores de E^k.

Definición 39.- Acoplamiento Débil

Sea la estructura multitérmino $E^k = g(A_1^k, A_2^k, ..., A_n^k)$ con conjunto referencial de items X e $Y \subseteq X$. Se define el acoplamiento débil entre Y y E^k como la operación lógica:

$$Y \oplus E^k = \begin{cases} verdadero & si\ \exists A_i^k \in \{A_1^k, ..., A_n^k\} / Y \frown A_i^k \neq \emptyset \\ falso & en\ otro\ caso \end{cases} \tag{3.48}$$

El acoplamiento débil será, por lo tanto, el que tendrá un conjunto Y con la estructura multitérmino E^k cuando dicho conjunto aparezca parcialmente incluido en alguno de los conjuntos generadores de E^k, es decir, el acoplamiento de Y con E^k sea distinto de cero.

Ejemplo 30.-Acoplamientos Fuerte y Débil

Sea $X = \{negro,\ rosa,\ verde,\ azul,\ amarillo,\ naranja,\ violeta\}$
Sea $E_1^k = g(Y), Y = (\{rosa, verde, violeta\}, \{negro, amarillo\}, \{naranja, azul\})$
Sea $Y' = (\{verde, violeta\})$
Sea $Y'' = (\{violeta, rosa, verde\})$

$$\Rightarrow \begin{cases} Y' \odot E^k & = & \text{verdadero} \\ Y'' \odot E^k & = & \text{falso} \\ Y'' \oplus E^k & = & \text{verdadero} \end{cases}$$

Vemos que Y' está completamente incluido en el primer conjunto generador de E^k, por lo tanto tendrá un acoplamiento fuerte con E^k. Y'' en cambio, sólo está incluido parcialmente en éste primer conjunto, aunque aparezcan incluidos sus tres elementos. Esto ocurre porque el conjunto multitérmino *(rosa, verde, violeta)* no es igual que el conjunto multitérmino *(violeta, rosa, verde)*.

Cálculo de la Bondad de Acoplamiento: Índice de Acoplamiento Fuerte y Débil

Este acoplamiento puede calcularse por medio de 2 formas:

1. Cálculo de índice por el promedio: Se calcula teniendo en cuenta todos los conjuntos multitérmino que componen la estructura multitérmino, sumando el grado de acoplamiento de cada conjunto y dividiendo entre el número de conjuntos.

2. Cálculo de índice por el máximo: Para su cálculo sólo se tiene en cuenta el conjunto multitérmino perteneciente a la estructura multitérmino con el que mejor se acopla el conjunto de términos buscado, que seŕa el conjunto donde la cardinalidad del acoplamiento sea mayor. Esta cardinalidad se calcula sumando la cardinalidad de todos los emparejamientos resultantes tras el acoplamiento del conjunto multitérmino con la estructura multitérmino.

Sea la estructura multitérmino $E^k = g(A_1^k, A_2^k, ..., A_m^k)$ con conjunto refencial de items X e $Y \subseteq X$. Cada conjunto A_i^k puede expresarse como un conjunto de secuencias o *item-seqs*: $A_i^k = (\alpha_1, \alpha_2, ...)$. LLamaremos δ_{ij} al valor del cardinal de la *item-seq* α_j del acoplamiento del conjunto Y con A_i^k:

$$\delta_{ij} = card(Y|\alpha_{j(A_i^k)}) \tag{3.49}$$

Y γ_i al valor del cardinal de A_i^k:

$$\gamma_i = card(A_i^k) \tag{3.50}$$

Calculemos la probabilidad de que el valor del cardinal de la *item-seq* α_j del acoplamiento de Y con A_i^k sea δ_{ij}.

Aplicando la probabilidad de Laplace,

$$p(\delta_{ij}) = \frac{\text{casos favorables}}{\text{casos posibles}} \tag{3.51}$$

Con “casos posibles” nos referimos al número de posibles combinaciones de los elementos de A_i^k independientemente del cardinal de las *item-seqs* del acoplamiento de Y con estos elementos. Con “casos favorables” nos referimos al número de posibles combinaciones de los elementos de A_i^k en donde el cardinal de la *item-seq* α_j del acoplamiento de Y con A_i^k es δ_{ij}.

Para averiguar el número de casos posibles o posibles combinaciones de los elementos de A_i^k, calculamos las permutaciones del cardinal de A_i^k:

$$\text{casos posibles} = \gamma_i! \tag{3.52}$$

Para averiguar el número de casos favorables también calculamos las permutaciones, pero en este caso no serán las permutaciones de todos los elementos de A_i^k, ya que queremos obtener sólo el número de posibles combinaciones donde el cardinal de la *item-seq* α_j del acoplemiento de A_i^k con Y sea δ_{ij}.

El acoplamiento de dos conjuntos es igual a las *item-seqs* coincidentes en los dos conjuntos. Como dentro de cada una de estas secuencias o *item-seqs*, el orden de los elementos es inalterable, cada *item-seq* se contará como un sólo elemento para el cálculo las combinaciones de A_i^k.

Con lo que el número de casos favorables será el número de posibles combinaciones de los elementos de A_i^k menos el cardinal de la *item-seq* α_j del acoplamiento de Y con A_i^k más uno (porque cada *item-seq* cuenta como un elemento):

$$\text{casos favorables} = [\gamma_i - (\delta_{ij} - 1)]! = (\gamma_i - \delta_{ij} + 1)! \tag{3.53}$$

Con lo que:

$$p(\delta_{ij}) = \frac{[\gamma_i - (\delta_{ij} - 1)]!}{\gamma_i!} = \frac{(\gamma_i - \delta_{ij} + 1)!}{\gamma_i!} \tag{3.54}$$

Ya tenemos la probabilidad de que el cardinal de la *item-seq* α_j del acoplamiento de Y con A_i^k sea δ_{ij}. Lo que pretendemos con esto es ponderar estas *item-seqs* "fruto" del acoplamiento de forma que los emparejamientos de *item-seqs* de mayor longitud (mayor cardinal) reciban más peso que los emparejamientos de longitud menor (menor cardinal), de modo que un emparejamiento de tres elementos, reciba más peso que tres emparejamientos de un sólo término.

Como $p(\delta_{ij})$ es menor cuanto mayor sea el cardinal de α_j, multiplicaremos el cardinal de α_j por la inversa de la $p(\delta_{ij})$.

Definición 40.-Cálculo del Índice de Acoplamiento Fuerte

Se define el acoplamiento fuerte entre Y y E^k como $S(Y|E^k)$ donde:

$$\forall A_i^k \in \{A_1^k, ..., A_n^k\},\ \forall \alpha_j \in \{\alpha_1, ..., \alpha_m\}, \quad m_{ij}(Y) = \frac{\delta_{ij}\ \frac{\gamma_i!}{(\gamma_i - \delta_{ij}+1)!}}{\gamma_i} \tag{3.55}$$

$$S = \{i \in \{1,...,n\}, j \in \{1,...,m\} |\ Y \subseteq A_i^k,\ \alpha_j \subseteq A_i^k\} \tag{3.56}$$

Con lo que se define:

1. *Índice débil por el promedio*

$$S(Y|E^k) = \frac{\sum_{i,j \in S} m_{ij}(Y)}{\sum_{i \in S} \gamma_i!} \tag{3.57}$$

2. *Índice débil por el máximo*

$$S(Y|E^k) = max\left(\frac{\sum_j m_i(Y)}{\gamma_i!}\right);\ i,j\ \in S \tag{3.58}$$

Definición 41.- Cálculo del Índice de Acoplamiento Débil

Se define el acoplamiento débil entre Y y E^k como $W(Y|E^k)$ donde:

$$\forall A_i^k \in \{A_1^k,...,A_n^k\},\ \forall \alpha_j \in \{\alpha_1,...,\alpha_m\}, \quad m_{ij}(Y) = \frac{\delta_{ij}\ \frac{\gamma_i!}{(\gamma_i - \delta_{ij}+1)!}}{\gamma_i} \tag{3.59}$$

$$W = \{i \in \{1,...,n\}, j \in \{1,...,m\} |\ Y \frown \widetilde{A_i^k} \neq \emptyset,\ \alpha_j \subseteq A_i^k\} \tag{3.60}$$

Con lo que se define:

1. *Índice débil por el promedio*

$$W(Y|E^k) = \frac{\sum_{i,j \in W} m_{ij}(Y)}{\sum_{i \in W} \gamma_i!} \tag{3.61}$$

2. *Índice débil por el máximo*

$$W(Y|E^k) = max\left(\frac{\sum_j m_i(Y)}{\gamma_i!}\right);\ i,j \in W \tag{3.62}$$

Ejemplo 31.- Cálculo del Índice de Acoplamiento Fuerte

Sea $E^k = (\{1,2,3,4\},\{2,3,1\},\{3,5,4\})$
Sea $Y = (\{2,3,1\})$

1. *Cálculo de Índice de Acoplamiento Fuerte por el Promedio.*

$$\begin{aligned} S(Y|E^k) &= \frac{\frac{3\left(\frac{3!}{(3-3+1)!}\right)}{3}}{3!} \\ &= \frac{3!}{3!} = 1 \end{aligned}$$

2. *Cálculo del Índice de Acoplamiento Fuerte por el Máximo.*

$$S(Y|E^k) = max(1) = 1$$

Ejemplo 32.- Cálculo del Índice de Acoplamiento Débil

Sea $E^k = (\{1,2,3,4\},\{2,3,1\},\{3,5,4\})$
Sea $Y = (\{2,3,1\})$

1. *Cálculo de Índice de Acoplamiento Débil por el Promedio.*

$$\begin{aligned} W(Y|E^k) &= \frac{\frac{2\left(\frac{4!}{(4-2+1)!}\right)}{4} + \frac{1\left(\frac{4!}{(4-1+1)!}\right)}{4} + \frac{3\left(\frac{3!}{(3-3+1)!}\right)}{3} + \frac{1\left(\frac{3!}{(3-1+1)!}\right)}{3}}{4!+3!+3!} \\ &= \frac{2+{}^1/_4+3!+{}^{2!}/_3}{4!+3!+3!} \\ &= \frac{2+0{,}25+6+0{,}67}{36} \\ &= 0{,}25 \end{aligned}$$

2. *Cálculo del Índice de Acoplamiento Débil por el Máximo.*

$$W(Y|E^k) = \left(\frac{2+{}^1/_4}{4!}, \frac{3!}{3!}, \frac{{}^2/_3}{3!}\right) = 1$$

3.3. Método de extracción: Algoritmos APriori y APriori Modificado Para la Generación de la Estructura-AP y Estructura Multitérmino.

3.3.1. Algoritmo APriori para la extracción de la estructura-AP

Un *itemset* frecuente es un *itemset* cuyo soporte es mayor que un soporte mínimo especificado por el usuario (Lo denotamos L_k, donde k es la longitud del *itemset*).

Un *itemset* candidato es un *itemset* potencialemente frecuente (Lo denotamos C_k, donde k es la longitud del *itemset*).

Algoritmo APriori

El algoritmo APriori ((How09),(Agr95)) puede utilizarse para generar todos los *itemsets* frecuentes:

- Paso 1
  1. Generar los *itemsets* candidatos en C_1
  2. Almacenar los *itemsets* frecuentes en L_1
- Paso k
  1. Generar los *itemsets* candidatos en C_k a partir de los *itemsets* frecuentes en L_{k-1}
     a) Unir $L_{k-1}p$ con $L_{k-1}q$ como sigue:
     insert into C_k
     select $p.item_1, p.item_2, ..., p.item_{k-1}, q.item_{k-1}$
     from $L_{k-1}p, L_{k-1}q$
     where $p.item_1 = q.item_1, ..., p.item_{k-2} = q.item_{k-2}, p.item_{k-1} \neq q.item_{k-1}$
     b) Generar todos los *itemsets* candidatos en C_k a partir de la unión de los *itemsets* en L_{k-1}
     c) Eliminar todos los *itemsets* candidatos en C_k donde algún subconjunto no sea un *itemset* frecuente en L_{k-1}
     d) Eliminar todos los *itemsets* candidatos en C_k que contengan los mismos elementos cambiados de orden

2. Escanear las transacciones de la base de datos para determinar el soporte de cada *itemset* candidato en C_k
3. Almacenar los *itemsets* frecuentes en L_k

3.3.2. Modificación del Algoritmo APriori para la Extracción de la Estructura Multitérmino

Existen numerosos algoritmos para la generación de secuencias frecuentes en el texto, algunos de los cuales podemos verlos recogidos en el capítulo 8 de (Tan06).

El algoritmo que proponemos aquí en muy similar al algoritmo APriori. La única diferencia estriba en la forma de generar las *item-seqs* en C_k a partir de las *item-seqs* de L_{k-1} y en que aquellas *item-seqs* candidatas que tengan los mismos elementos en distinto orden, se considerarán *item-seqs* candidatas diferentes, por lo tanto no se eliminará ninguna de ellas.

También habrá que tener en cuenta que el cálculo del soporte se realizará de forma diferente para los *itemsets* que para las *item-seqs*, ya que para los primeros se realiza contando las tuplas en las que aparezcan todos los elementos contenidos en el *itemset* y para las *item-seqs* se contarán sólo aquellas tuplas en las que los elementos mantengan el mismo orden que en la *item-seq*. En ambos casos, se dividirá después por el número total de tuplas:

- Paso 1
  1. Generar las *item-seqs* candidatas en C_1
  2. Almacenar las *item-seqs* frecuentes en L_1
- Paso k
  1. Generar las *item-seqs* candidatas en C_k a partir de las *item-seqs* frecuentes en L_{k-1}
     a) Unir $L_{k-1}p$ con $L_{k-1}q$ como sigue:
     insert into C_k
     select $p.item_1, p.item_2, ..., p.item_{k-1}, q.item_{k-1}$
     from $L_{k-1}p, L_{k-1}q$
     where $p.item_2 = q.item_1, ..., p.item_{k-1} = q.item_{k-2}$
     b) Generar todas las *item-seqs* candidatas en C_k a partir de la unión de las *item-seqs* en L_{k-1}

   c) Eliminar todas las *item-seqs* candidatas en C_k donde alguna subsecuencia no sea una *item-seq* frecuente en L_{k-1}

2. Escanear las transacciones de la base de datos para determinar el soporte de cada *item-seq* candidata en C_k
3. Almacenar las *item-seqs* frecuentes en L_k

Veremos el funcionamiento de estos algoritmos paso a paso en el ejemplo práctico del siguiente capítulo.

Capítulo 4

EJEMPLO DE GENERACIÓN, VISUALIZACIÓN Y COMPARACIÓN DE LA ESTRUCTURA MONOTÉRMINO, LA ESTRUCTURA-AP Y LA ESTRUCTURA MULTITÉRMINO

Supongamos que se extraen los siguientes titulares de noticias relacionadas con el empleo de una página de Internet (ver tabla (4.1))

Se realiza la limpieza de datos, mediante la cual se eliminan términos irrelevantes que no aportan información como las preposiciones, conjunciones, determinantes, etc. También se tendría un diccionario de sinónimos y acrónimos, mediane el cual se sustituirían por una única palabras todas las palabras similares o de igual significado.

En la tabla (4.2) podemos ver como quedaría el texto tras la limpieza de datos.

Tras esta limpieza, se tendrían los conjuntos de términos o *itemsets* que aparecen en la tabla (4.3).

	TITULARES
1	Faltan funcionarios en las oficinas de empleo
2	Cuestiones sobre el empleo de los funcionarios
3	Las oficinas de funcionarios vacías
4	Los funcionarios cuestionan su empleo
5	Las oficinas de empleo abarrotadas
6	Cuestionario en las oficinas de empleo
7	El empleo de los funcionarios el mejor calificado
8	Disminución de sueldo a los funcionarios
9	Disminución de empleo
10	Los funcionarios no van a la oficina de empleo
11	Cuestiones sobre el sueldo de los funcionarios
12	Los funcionarios se abarrotan frente a la oficina de empleo
13	Los funcionarios no están siempre en la oficina
14	Disminución de empleo y sueldo en 2010
15	Cuestionario sobre el sueldo de los funcionarios

Tabla 4.1: *Muestra de titulares relacionados con el empleo*

n	1^{er} item	2^{o} item	3^{er} item	4^{o} item
1	faltar	funcionarios	oficina	empleo
2	cuestionar	empleo	funcionarios	
3	oficina	funcionarios	vaciar	
4	funcionarios	cuestionar	empleo	
5	oficina	empleo	abarrotar	
6	cuestionar	oficina	empleo	
7	empleo	funcionarios	mejor	calificar
8	disminuir	sueldo	funcionarios	
9	disminuir	empleo		
10	funcionarios	ir	oficina	empleo
11	cuestionar	sueldo	funcionarios	
12	funcionarios	abarrotar	oficina	empleo
13	funcionarios	estar	oficina	
14	disminuir	empleo	sueldo	2010
15	cuestionar	sueldo	funcionarios	

Tabla 4.2: *Titulares tras la Limpieza de Datos*

4.1. Generación de la Estructura Monotérmino

Para generar la estructura monotérmino o estructura tradicional que vemos representada en la web en forma de *tag cloud*, lo que se hace es contar el número de ocurrencias de cada palabra en el texto, o lo que es lo mismo, calcular la frecuencia absoluta (F_i) de cada término (ver tabla (4.4)) y la estructura monotérmino estará compuesta por aquellos términos que se consideren frecuentes según un soporte mínimo establecido.

n	Itemsets
1	{*faltar, funcionarios, oficina, empleo*}
2	{*cuestionar, empleo, funcionarios*}
3	{*oficina, funcionarios, vaciar*}
4	{*funcionarios, cuestionar, empleo*}
5	{*oficina, empleo, abarrotar*}
6	{*cuestionar, oficina, empleo*}
7	{*empleo, funcionarios, mejor, calificar*}
8	{*disminuir, sueldo, funcionarios*}
9	{*disminuir, empleo*}
10	{*funcionarios, ir, oficina, empleo*}
11	{*cuestionar, sueldo, funcionarios*}
12	{*funcionarios, abarrotar, oficina, empleo*}
13	{*funcionarios, estar, oficina*}
14	{*disminuir, empleo, sueldo, 2010*}
15	{*cuestionar, sueldo, funcionarios*}

Tabla 4.3: *Conjunto de* Itemsets *tras la Limpieza*

n	Palabra	F_i	$f_i(\%) = F_i/N$
1	faltar	1	2’04
2	funcionarios	11	22’45
3	oficina	7	14’29
4	empleo	10	20’41
5	cuestionar	5	10’20
6	vaciar	1	2’04
7	abarrotar	2	4’08
8	mejor	1	2’04
9	calificar	1	2’04
10	disminuir	3	6’12
11	sueldo	4	8’16
12	ir	1	2’04
13	estar	1	2’04
14	2010	1	2’04

Tabla 4.4: *Frecuencia Absoluta y Relativa de cada Término en el Texto*

Aunque casi todos los sitios web emplean la frecuencia absoluta para determinar cuando una palabra aparecerá el la *tag cloud*, nosotros hemos calculado también la frecuencia relativa f_i en tanto por ciento, siendo N el número total de palabras en el texto, en nuestro caso $N = 49$.

Los generadores de *tag cloud* que encontramos en la web ((Ste06), (Nab09), etc.) normalmente permiten al usuario especificar la frecuencia absoluta mínima

que desean que tengan los términos que aparezcan visualizados en la *tag cloud* generada. Tao et al. (Tao03) apuestan por la utilización de un peso para determinar el soporte, en lugar de la frecuencia absoluta. Este peso lo calculan como una frecuencia relativa. Nosotros encontramos más útil el uso de la frecuencia relativa que el de la absoluta, ya que sólo teniendo en cuenta el número de palabras totales o extensión del texto podremos estimar la frecuencia absoluta de cada término como alta o baja.

Suponemos que se considera que un término es relevante cuando su frecuencia relativa es superior al 5 % ($f_i(\%) > 5\%$), lo que equivale a una frecuencia absoluta superior a 2'45 ($F_i > 2'45$). Como la frecuencia absoluta es un número entero tomaremos los términos cuya frecuencia absoluta sea igual o mayor que 3 ($F_i \geq 3$).

Con lo que lo términos presentes en la estructura monotérmino serían los siguientes:

Palabra (x)	F_i
funcionarios	11
oficina	7
empleo	10
cuestionar	5
disminuir	3
sueldo	4

Tabla 4.5: *Términos en la Estructura Monotérmino*

El cardinal de esta estructura monotérmino es:

$$card(E^1) = 6$$

Y la estructura monotérmino ponderada que se representará posteriormente en forma de *tag cloud* se expresaría de la siguiente forma:

$$\begin{aligned} \Rightarrow \widetilde{E^1} &= g(\widetilde{A^1}, \widetilde{B^1}, \widetilde{C^1}, \widetilde{D^1}, \widetilde{E^1}, \widetilde{F^1}) \\ &= \{(fucionarios, (11)), (empleo, (10)), (oficina, (7)), (cuestionar, (5) \\ &\quad , (sueldo, (4)), (disminuir, (3))\} \end{aligned}$$

En la figura (4.1) se puede ver esta visualización en forma de *tag cloud*. Los distintos tamaños de fuente indican la frecuencia de los términos.

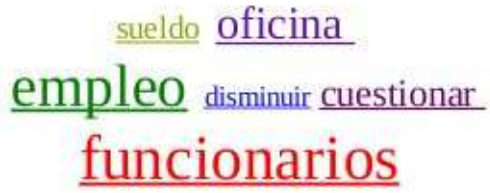

Figura 4.1: *Visualización en Forma de Tag Cloud de la Estructura Monotérmino*

4.2. Generación de la Estructura-AP

La tabla (4.6) nos será de utilidad para la generación de la estructura-AP. Las filas corresponden a los *itemsets* generados en la tabla (4.3). Las columnas representan todas las palabras que aparecen en el texto. Un "1" indica que la palabra de la columna correspondiente está incluída en el *itemset* de la fila correspondiente.

Itemsets	falt	func	ofic	empl	cues	vaci	abar	mejo	cali	dism	suel	ir	esta	2010
1	1	1	1	1										
2		1		1	1									
3		1	1			1								
4		1		1	1									
5			1	1			1							
6			1	1	1									
7		1		1				1	1					
8		1								1	1			
9				1						1				
10		1	1	1								1		
11		1			1						1			
12		1	1	1			1							
13		1	1										1	
14				1						1	1			1
15		1			1						1			

Tabla 4.6: *Pertenencia de Cada Palabra a los* Itemsets *Generados*

Para la generar los conjuntos-AP que conformarán la estructura-AP, haremos uso del algoritmo APriori comentado anteriormente. A continuación podemos ver, como funcionaría paso a paso este algoritmo.

Algoritmo APriori paso a paso para la generación de los *itemsets* frecuentes para formar los conjuntos-AP

Para establecer cuando un *itemset* es frecuente estableceremos un soporte mínimo del 20 %. El soporte de un *itemset A* en *D*, siendo *D* el conjunto de todos

los *itemsets*, se calcula como el % de ocurrencias de A en D (How09).

- 1^{er} **paso.-** Se generan los *itemsets* candidatos en C_1 o *itemsets* candidatos de grado 1.

C_1	**Itemset(x)**	F_i	$Supp(x)(\%) = F_i/n$
1	{*faltar*}	1	6'67
2	{*funcionarios*}	11	73'33
3	{*oficina*}	7	46'67
4	{*empleo*}	10	66'67
5	{*cuestionar*}	5	33'33
6	{*vaciar*}	1	6'67
7	{*abarrotar*}	2	13'33
8	{*mejor*}	1	6'67
9	{*calificar*}	1	6'67
10	{*disminuir*}	3	20
11	{*sueldo*}	4	26'67
12	{*ir*}	1	6'67
13	{*estar*}	1	6'67
14	{*2010*}	1	6'67

Tabla 4.7: *Algoritmo APriori en Fase* C_1. Itemsets *Candidatos.*

donde n es igual al número total de transacciones, $n = 15$ (ver tabla(4.3)).

A continuación, de entre estos *itemsets* candidatos, se seleccionan los *itemsets* frecuentes en lo que se conoce como fase L_1. Los *itemsets* frecuentes serán aquellos cuyo soporte es igual o superior al 20 % ($Supp(x) \geq 20\,\%$). (Ver tabla (4.8))

L_1	**Itemset(x)**	F_i	$Supp(x)(\%) = F_i/n$
1	{*funcionarios*}	11	73'33
2	{*oficina*}	7	46'67
3	{*empleo*}	10	66'67
4	{*cuestionar*}	5	33'33
5	{*disminuir*}	3	20
6	{*sueldo*}	4	26'67

Tabla 4.8: *Algoritmo APriori en Fase* L_1. Itemsets *Frecuentes.*

- 2^{o} **paso.-** Se generan los *itemsets* candidatos en C_2 o *itemsets* candidatos de grado 2. Para ello, combinamos entre sí todos los *itemsets* frecuentes en L_1 y eliminamos aquellos en que aparezcan los mismos elementos pero en distinto orden. Para calcular la frecuencia absoluta F_i de los *itemsets* así generados, es cuando haremos uso de la tabla (4.6), seleccionando los elementos de los que se compone cada *itemset* y contando el número de veces en

que esos elementos “co-ocurren”. Una vez que conozcamos F_i podremos calcular el soporte.

C_2	**Itemset(x)**	F_i	$Supp(x)(\%) = F_i/n$
1	{*funcionarios, oficina*}	5	33’33
2	{*funcionarios, empleo*}	6	40
3	{*funcionarios, cuestionar*}	4	26’67
4	{*funcionarios, disminuir*}	1	6’67
5	{*funcionarios, sueldo*}	3	20
6	{*oficina, empleo*}	5	33’33
7	{*oficina, cuestionar*}	1	6’67
8	{*oficina, disminuir*}	0	0
9	{*oficina, sueldo*}	0	0
10	{*empleo, cuestionar*}	3	20
11	{*empleo, disminuir*}	2	13’33
12	{*empleo, sueldo*}	1	6’67
13	{*cuestionar, disminuir*}	1	6’67
14	{*cuestionar, sueldo*}	2	13’33
15	{*disminuir, sueldo*}	2	13’33

Tabla 4.9: *Algoritmo APriori en Fase* C_2. Itemsets *Candidatos.*

Nos damos cuenta, que en la tabla de *itemsets* candidatos en la fase C_2 (4.9), está por ejemplo el *itemset* {*funcionarios, oficina*}, pero no el {*oficina, funcionarios*}. Esto es porque se han eliminado los *itemsets* con los mismos elementos pero en orden diferente.

A continuación, se seleccionan de entre estos *itemsets* candidatos los *itemsets* frecuentes en lo que se denomina la fase L_2. (Ver tabla (4.10)).

L_2	**Itemset(x)**	F_i	$Supp(x)(\%) = F_i/n$
1	{*funcionarios, oficina*}	5	33’33
2	{*funcionarios, empleo*}	6	40
3	{*funcionarios, cuestionar*}	4	26’67
5	{*funcionarios, sueldo*}	3	20
6	{*oficina, empleo*}	5	33’33
10	{*empleo, cuestionar*}	3	20

Tabla 4.10: *Algoritmo APriori en Fase* L_2. Itemsets *Frecuentes.*

- 3^{er} **paso.-** Se generan los *itemsets* candidatos en C_3 o *itemsets* candidatos de grado 3. Para ello, se combinan entre sí todos los *itemsets* frecuentes en L_2 que tengan algún k-1 *items* en común, en este caso, que tengan algún elemento en común y se eliminan los que resulten con los mismos elementos y orden diferente. Podemos ver los *itemsets* candidatos en la tabla (4.11)).

C_3	**Itemset(x)**	F_i	$Supp(x)(\%) = F_i/n$
1	{*funcionarios, oficina, empleo*}	3	20
2	{*funcionarios, oficina, cuestionar*}	0	0
3	{*funcionarios, oficina, sueldo*}	0	0
4	{*funcionarios, empleo, cuestionar*}	2	13'33
5	{*funcionarios, empleo, sueldo*}	0	0
6	{*funcionarios, cuestionar, sueldo*}	1	6'67
7	{*oficina, empleo, cuestionar*}	1	6'67

Tabla 4.11: *Algoritmo APriori en Fase* C_3. Itemsets *Candidatos.*

De los *itemsets* candidatos en C_3 se seleccionan los *itemsets* frecuentes. Esto podemos verlo en la tabla (4.12).

C_3	**Itemset(x)**	F_i	$Supp(x)(\%) = F_i/n$
1	{*funcionarios, oficina, empleo*}	3	20

Tabla 4.12: *Algoritmo APriori en Fase* L_3. Itemsets *Frecuentes.*

Como vemos, en L_3 ya sólo tenemos un *itemsets* frecuente, que al no poder combinarlo con ningún otro no se tendría ningún *itemsets* candidato en C_4 y habríamos acabado.

Una vez generados todos los *itemsets* frecuentes, estudiaríamos cuáles son los conjuntos-AP que contienen esos *itemsets*. Evidentemente, el *itemset* de grado superior, que es el *itemset* de grado tres, sería uno de los conjuntos-AP, ya que no hay ningún otro que lo contenga. Luego, bajaríamos de grado, para ver qué *itemsets* de grado 2 no están contenidos en el conjunto-AP de grado 3 y aquellos que no estuvieran contenidos, serían a su vez, conjuntos-AP e igual para los *itemsets* de grado 1. En total, tendríamos los siguientes conjuntos-AP:

$$A = \{funcionarios, oficina, empleo\}$$
$$B = \{funcionarios, cuestionar\}$$
$$C = \{funcionarios, sueldo\}$$
$$D = \{empleo, cuestionar\}$$
$$E = \{disminuir\}$$

Con lo que la estructura-AP sería la generada por todos esos conjuntos-AP:

$$T = g(A, B, C, D, E)$$

$$T = \{\{funcionarios, oficina, empleo\}, \{funcionarios, oficina\}, \{funciona-rios, empleo\}, \{oficina, empleo\}, \{funcionarios, cuestionar\}, \{funcio-narios, sueldo\}, \{empleo, cuestionar\}, \{funcionarios\}, \{oficina\}, \{em-pleo\}, \{cuestionar\}, \{sueldo\}, \{disminuir\}\}$$

Y el cardinal de la estructura-AP es $card(T) = 13$

Con el fin de comparar la estructura-AP con la estructura multitérmino, lo que nos interesa sería la estructura-AP ponderada (junto con la frecuencia de cada *itemset*):

$$\widetilde{T} = g(\widetilde{A}, \widetilde{B}, \widetilde{C}, \widetilde{D}, \widetilde{E})$$

$$\widetilde{T} = \{\{funcionarios, oficina, empleo, (3)\}, \{funcionarios, oficina, (5)\}, \{funcionarios, empleo, (6)\}, \{oficina, empleo, (5)\}, \{funcionarios, cuestionar, (4)\}, \{funcionarios, sueldo, (3)\}, \{empleo, cuestionar, (3)\}, \{funcionarios, (11)\}, \{oficina, (7)\}, \{empleo(10)\}, \{cuestionar, (5)\}, \{sueldo, (4)\}, \{disminuir, (3)\}\}$$

En la figura (4.2) podemos ver cómo quedaría esta estructura-AP ponderada visualizada en forma de *Tag Cloud*.

Figura 4.2: *Visualización en Forma de Tag Cloud de la Estructura-AP*

4.3. Generación de la Estructura Multitérmino

Modificación del algoritmo APriori para generar las *item-seqs* frecuentes de los conjuntos multitérmino.

Para la generación de las *item-seqs* frecuentes que compondrán los conjuntos multitérmino, se hará una modificación del algoritmo APriori empleado para la generación de los *itemsets* frecuentes que componían los conjuntos-AP como se comentó en el capítulo anterior.

- 1^{er} **paso.-** Se generan las *item-seqs* candidatas en C_1 o *item-seqs* candidatas de grado 1 y de entre todas los *item-seqs* candidatas generadas, se seleccionan las *item-seqs* frecuentes, que serán las *item-seqs* en L_1. Estableceremos un soporte mínimo igual al 20 % para determinar cuándo una *item-seq* es frecuente.

  Como el soporte es el mismo que establecimos para la generación de los *itemsets* frecuentes de los conjuntos-AP, en este primer paso, tanto C_1 como L_1 coincidirán con las tablas C_1 y L_1 generadas para los conjuntos-AP (ver tablas (4.7) y (4.8)).

- 2^{o} **paso.-** A partir de estas *item-seqs* frecuentes generadas en L_1, buscamos las *item-seqs* frecuentes de grado 2.

  La tabla que tendríamos, sería la misma tabla que para los *itemsets* candidatos en fase C_2 (4.9), pero con el doble de elementos, es decir, por cada *itemset* de la tabla (4.9) tendríamos una *item-seq* con los mismos elementos del *itemset* en el mismo orden y otra *item-seq* con los elementos en orden inverso.

  Como esta tabla resulta muy extensa y muchas de esas uniones resultantes no estarían en el texto y tendrían un soporte igual a cero, nosotros vamos a examinar las adyacencias de las *item-seqs* en L_1 con otras *item-seqs* en L_1 cuya unión tenga un soporte distinto de cero, que son las que incluimos en la tabla (4.13).

  Para ello, hemos tomado la primera *item-seq* frecuente de grado 1, en nuestro caso $\{funcionarios\}$ y hemos buscado todas sus adyacencias con las demás *item-seqs* frecuentes de grado 1. Después haríamos lo mismo para $\{oficina\}$ y así sucesivamente.

  Como examinaremos las adyacencias de todas las *item-seqs* frecuentes, sólo será necesario examinar las adyacencias a la derecha.

Con lo que las *item-seqs* candidatas en C_2 son los que vemos en la tabla (4.14)

Adyacencias de item-seq(x)	n	Item-seqs generadas a partir de x
Adyacencias a la derecha de {funcionarios}	1	{**funcionarios**,oficina}
	2	{**funcionarios**, cuestionar}
Adyacencias a la derecha de {oficina}	3	{**oficina**, empleo}
	4	{**oficina**, funcionarios}
Adyacencias a la derecha de {empleo}	5	{**empleo**, funcionarios}
	6	{**empleo**, sueldo}
Adyacencias a la derecha de {cuestionar}	7	{**cuestionar**, empleo}
	8	{**cuestionar**, oficina}
	9	{**cuestionar**, sueldo}
Adyacencias a la derecha de {disminuir}	10	{**disminuir**, empleo}
	11	{**disminuir**, sueldo}
Adyacencias a la derecha de {sueldo}	12	{**sueldo**, funcionarios}

Tabla 4.13: *Adyacencias de las* Item-seqs *de Grado 1*

C_2	Item-seq(x)	F_i	$Supp(x)(\%) = F_i/n$
1	{*funcionarios, oficina*}	3	20
2	{*funcionarios, cuestionar*}	1	6’67
3	{*oficina, empleo*}	5	33’33
4	{*oficina, funcionarios*}	1	6’67
5	{*empleo, funcionarios*}	1	6’67
6	{*empleo, sueldo*}	1	6’67
7	{*cuestionar, empleo*}	2	13’33
8	{*cuestionar, oficina*}	1	6’67
9	{*cuestionar, sueldo*}	2	13’33
10	{*disminuir, sueldo*}	1	6’67
11	{*disminuir, empleo*}	2	13’33
12	{*sueldo, funcionarios*}	3	20

Tabla 4.14: *Algoritmo APriori Modificado para Conjuntos Multitérmino en Fase C_2.*

Y las *item-seqs* frecuentes las que aparecen en la tabla (4.15)

- 3^{er} **paso.-** Para construir las *item-seqs* candidatas en C_n, buscamos las *subitem-seqs* de grado n-2 comunes en las *item-seqs* frecuentes de grado n-1. Así, para construir las *item-seqs* candidatas en C_3 buscamos las *subitem-seqs* comunes de grado 1 en las *item-seqs* de L_2 si las hubiera. En las dos primeras *item-seqs* la *subitem-seq* común es {oficina}, y respetando el orden de adyacencia, el único conjunto tritérmino que se genera de la combinación

C_2	Item-seq(x)	F_i	$Supp(x)(\%) = F_i/n$
1	{*funcionarios, oficina*}	3	20
2	{*oficina, empleo*}	5	33'33
3	{*sueldo, funcionarios*}	3	20

Tabla 4.15: *Algoritmo APriori Modificado para Conjuntos Multitérmino en Fase L_2.*

de ambas *item-seqs* es {*funcionarios, oficina, empleo*}. Luego buscaríamos la *subitem-seq* común entre la primera y la tercera *item-seq* y luego entre la segunda y la tercera. Esto es lo mismo que mirar las adyacencias a la derecha de las *item-seqs* de L_2. Con los que en C_3 tenemos las *item-seqs* de la tabla (4.16).

C_3	Item-seq(x)	F_i	$Supp(x)(\%) = F_i/n$
1	{*funcionarios, oficina, empleo*}	2	13'33
2	{*sueldo, funcionarios, oficina*}	0	0

Tabla 4.16: *Algoritmo APriori Modificado para Conjuntos Multitérmino en Fase C_3.*

Y vemos que, como ninguna *item-seq* es frecuente en C_3, no tendremos ninguna *item-seq* en L_3 y habremos terminado.

Una vez generadas las *item-seqs* frecuentes, estudiaríamos cuáles son los conjuntos multitérmino que contienen esas *item-seqs*. Tendríamos los siguientes conjuntos multitérmino:

$$A^2 = \{funcionarios, oficina\}$$
$$B^2 = \{oficina, empleo\}$$
$$C^2 = \{sueldo, funcionarios\}$$
$$D^1 = \{cuestionar\}$$
$$E^1 = \{disminuir\}$$

Y la estructura multitérmino ponderada sería la generada por todos esos conjuntos multitérmino ponderados:

$$\widetilde{E^k} = g(\widetilde{A^2}, \widetilde{B^2}, \widetilde{C^2}, \widetilde{D^1}, \widetilde{E^1})$$

El cardinal de la estructura multitérmino es $card(E^k) = 9$

$$\widetilde{E^k} = \{\{funcionarios, oficina, (3)\}, \{oficina, empleo, (5)\}, \{sueldo, funcionarios, (3)\}, \{funcionarios, (11)\}, \{oficina, (7)\}, \{empleo, (10)\}, \{cuestionar, (5)\}, \{sueldo, (4)\}, \{disminuir, (3)\}$$

En la figura (4.3) podemos ver visualizada en forma de *tag cloud* esta estructura multitérmino.

Figura 4.3: *Visualización en Forma de* Tag Cloud *de la Estructura Multitérmino*

4.4. Comparación de la Estructura Monotérmino, la Estructura-AP y la Estructura Multitérmino.

4.4.1. Diferencias entre la Estructura-AP y la Estructura Multitérmino

De la tabla (4.17) podemos deducir las siguientes diferencias entre la estructura-AP y la estructura multitérmino:

- Vemos que, aunque la estructura-AP y la estructura multitérmino tengan el mismo número de conjuntos generadores, el grado de estos conjuntos para la estructura-AP es superior, es decir, la estructura-AP incluye un conjunto de grado 3 que no lo incluye la estructura multitérmino. En consecuencia, hay muchas *item-seqs* que se pierden en la estructura multitérmino con respecto a la estructura-AP. Además, al perder los conjuntos de mayor grado, se pierde precisión en las consultas.

	EST. MONOTÉRMINO	ESTRUCTURA-AP	EST. MULTITÉRMINO
Notación	E^1	$T = g(A,B,C,D,E)$	$E^K = g(A^k,B^k,C^k,D^k,E^k)$
Generadores	No tiene	$A = \{funcionarios, oficina, empleo\}$ $B = \{funcionarios, cuestionar\}$ $C = \{funcionarios, sueldo\}$ $D = \{empleo, cuestionar\}$ $E = \{disminuir\}$	$A = \{funcionarios, oficina\}$ $B = \{oficina, empleo\}$ $C = \{sueldo, funcionarios\}$ $D = \{cuestionar\}$ $E = \{disminuir\}$
Cardinal	6	13	9
Itemsets o *Item-seqs* ponderados/as	$\{funcionarios, (11)\}$ $\{empleo, (10)\}$ $\{oficina, (7)\}$ $\{cuestionar, (5)\}$ $\{sueldo, (4)\}$ $\{disminuir, (3)\}$	$\{funcionarios, oficina, empleo, (3)\}$ $\{funcionarios, oficina, (5)\}$ $\{funcionarios, empleo, (6)\}$ $\{oficina, empleo, (5)\}$ $\{funcionarios, cuestionar, (4)\}$ $\{funcionarios, sueldo, (3)\}$ $\{empleo, cuestionar, (3)\}$ $\{funcionarios, (11)\}$ $\{empleo, (10)\}$ $\{oficina, (7)\}$ $\{cuestionar, (5)\}$ $\{sueldo, (4)\}$ $\{disminuir, (3)\}$	$\{funcionarios, oficina, (3)\}$ $\{oficina, empleo, (5)\}$ $\{sueldo, funcionarios, (3)\}$ $\{funcionarios, (11)\}$ $\{empleo, (10)\}$ $\{oficina, (7)\}$ $\{cuestionar, (5)\}$ $\{sueldo, (4)\}$ $\{disminuir, (3)\}$

Tabla 4.17: *Comparación de la Estructura Monotérmino, la Estructura-AP y la Estructura Multitérmino*

- El cardinal de la estructura-AP es mayor que el de la estructura multitérmino. Es por esto, que tendremos mayor número de *itemsets* en la visualización de la estructura-AP que de *item-seqs* en la de la estructura multitérmino.
- El soporte y la frecuencia de las *item-seqs* de la estructura multitérmino siempre es menor o igual que le soporte de los *itemsets* de la estructura-AP, por lo que un mismo conjunto de término podrá verse con mayor tamaño en la visualización de la estructura-AP que en la visualización de la estructura multitérmino.
- Vemos, por ejemplo, que el *itemset* $\{funcionarios, sueldo\}$ aparece en este orden de elementos en la estructura-AP, pero en orden inverso de elementos en la *item-seq* {sueldo, funcionarios} de la estructura multitérmino. Esto es porque la estructura multitérmino tiene en cuenta el orden en que aparecen los elementos en el texto y la estructura-AP no tiene en cuenta este orden, por lo que el orden en que aparecen los elementos dentro de los *itemsets* de la estructura-AP es indiferente. En el texto, el orden en que aparecen

los elementos del *itemset* $\{funcionarios, sueldo\}$ con más frecuencia, es primero "sueldo" y luego "funcionarios", es por esto que ese será el orden en que aparezcan estos elementos dentro de las *item-seqs* de la estructura multitérmino.

Si un usuario introdujera en la consulta los elementos de la *item-seq* {*sueldo, funcionarios*} de la estructura multitérmino en orden inverso, es decir, primero "funcionarios" y luego "sueldo", la respuesta estaría vacía, mientras que si lo hiciera a través de la estructura-AP, el sistema le devolvería 3 entradas. Esto tendría sus ventajas y sus inconvenientes:

- Ventaja principal: La consulta produce más resultados
- Inconveniente principal: Los resultados podrían ser menos precisos

- **Visualización** (ver figura (4.4)

Figura 4.4: *Visualización de las Tres Estructuras en Forma de Tag Cloud*

- **¿Cuándo será mejor utilizar la estructura-AP y cuándo la estructura multitérmino?**

  - Utilizaremos la estructura multitérmino para consultas donde el orden es importante y, por lo tanto, queramos obtener los términos en la respuesta en el mismo orden que los hemos introducido en la consulta. Por ejemplo, se utilizaría la estructura multitérmino si se está buscando entradas relacionadas con la oficina de empleo, pero no nos interesan las relacinadas con empleo en la oficina, empleo de oficina, etc.

    Mediante la estructura multitérmino la búsqueda es más restrictiva y visualmente, los término en la nube aparecen menos aglomerados al ser menores en número, por lo que es más fácil identificarlos.

- Utilizaremos la estructura-AP cuando queremos que no importe el orden en que hemos introducido los términos en la consulta y que el sistema nos devuelva todas las entradas que los contengan, independientemente del orden que aparezcan. Sería el caso por ejemplo de la búsqueda de personas, donde nos da igual que el sistema devuelva las entradas que contengan el nombre primero y luego los apellidos o primero los apellidos primero y luego el nombre.

  La visualización de la estructura-AP en forma de nube presenta mayor número de términos que la estructura multitérmino, por lo que será más indicada cuando se esté más interesado en explorar el entorno que en realizar un búsqueda concreta o cuando queremos que se nos sugieran términos de búsqueda o palabras asociadas.

4.4.2. ¿Cómo Mejoran la Estructura-AP y la Estructura Multitérmino a la Estructura Monotérmino?

Tanto la estructura-AP como la estructura multitérmino ofrecen incuestinables mejoras sobre la estructura monotérmino. Citaremos las más evidentes:

- Recuperan más información.
- Ofrecen más cantidad de sugerencias de búsqueda y exploración.
- Permiten identificar relaciones entre conceptos y sugieren términos relacionados con el término o términos de la consulta.
- Al permitir componentes multitérmino, facilitan la discriminación entre conceptos.
- Es más fácil identificar el contenido, debido a los componentes multitérmino. Pensemos por ejemplo en términos como inteligencia artificial, red social, bases de datos, sistemas operativos, etc. Estos términos tienen distinto significado cuando sus elementos van juntos a cuando van de forma independiente, por lo que, si no se permiten componentes multitérmino, podría no idenficarse el contenido de la información que se muestra o llevar a confusión, ya que estos elementos no tendrían porqué aparecer juntos en la representación, a pesar de ser componentes de un sólo término.
- Están definidas matemáticamente.
- Hemos explicado un método estándar para su generación.

4.4.3. Cálculo de los Índices de Acoplamiento Fuerte y Débil de un Conjunto con una Estructura-AP y con una Estructura Multitérmino. Comparación

Supongamos ahora que un usuario realiza una consulta a la base de datos. Para ver cuál de las dos estructuras (estructura-AP y estructura multitérmino) ofrecería mejores resultados a la consulta, calcularemos los índices de acoplamiento fuerte y débil.

Sea X conjunto referencial de items, X = {sueldo, empleo, cuestionar, funcionarios, oficina, disminuir}.

Sea la tag cloud multitérmino $E^k = g(A_1^k, A_2^k, .., A_n^k)$ *con conjunto referencial de items X. Sea* $T = g(A_1, A_2, .., A_m)$ *una estructura-AP con conjunto referencial de items X e* $Y \subseteq X$ *un conjunto-AP o un conjunto multitérmino según proceda.*

En nuestro ejemplo:

$$T = g(\{funcionarios, oficina, empleo\}, \{funcionarios, cuestionar\}, \{funcionarios, sueldo\}, \{empleo, cuestionar\}, \{disminuir\})$$

$$E^k = g(\{funcionarios, oficina\}, \{oficina, empleo\}, \{sueldo, funcionarios\}, \{cuestionar\}, \{disminuir\})$$

E Y sería el conjunto de términos introducidos en la consulta.

1. **Índice de acoplamiento fuerte**

   El índice de acoplamiento fuerte es una medida del grado de acoplamiento fuerte del conjunto Y con la estructura-AP o la estructura multitérmino, es decir, el grado con que ese conjunto se ajusta al dominio activo de la base de datos, entendiendo que el dominio activo viene representado por la estructura-AP o la estructura multitérmino según corresponda. Cuando calculamos el índice de acoplamiento fuerte lo que estamos calculando es el grado en que todos los términos del conjunto Y (o consulta) aparecen en la base de datos, o dicho de otro modo, la exhaustividad de la respuesta de forma precisa.

a) Índice de acoplamiento fuerte de Y con la Estructura-AP por el promedio y por el máximo.

$$T = g(\{funcionarios, oficina, empleo\}, \{funcionarios, cuestionar\}, \{funcionarios, sueldo\}, \{empleo, cuestionar\}, \{disminuir\})$$

- Supongamos que $Y = (empleo)$. Ver tabla (4.18):

Índ. de aco. fuerte por el promedio	Índ. de aco. fuerte por el máximo
$S(Y\|T) = \frac{1/3+1/2}{5} = \frac{5}{30} = 0'17$	$S(Y\|T) = max\left(\frac{1}{3}, \frac{1}{2}\right) = \frac{1}{2} = 0'5$

Tabla 4.18: *Índice de Acoplamiento Fuerte de T con $Y = (empleo)$*

- Supongamos que $Y = (funcionarios, sueldo)$. Ver tabla (4.19):

Índ. de aco. fuerte por el promedio	Índ. de aco. fuerte por el máximo
$S(Y\|T) == \frac{1}{5} = 0'2$	$S(Y\|T) = max(1) = 1$

Tabla 4.19: *Índice de Acoplamiento Fuerte de T con $Y = (funcionarios, sueldo)$*

- Supongamos que $Y = (funcionarios, oficina)$. Ver tabla (4.20)

Índ. de aco. fuerte por el promedio	Índ. de aco. fuerte por el máximo
$S(Y\|T) = \frac{2/3}{5} = \frac{2}{15} = 0'13$	$S(Y\|T) = max\left(\frac{2}{3}\right) = \frac{3}{3} = 0'67$

Tabla 4.20: *Índice de Acoplamiento Fuerte de T con $Y = (funcionarios, oficina)$*

b) Índice de acoplamiento fuerte de Y con la Estructura Multitérmino por el promedio y por el máximo.

$$E^k = g(\{funcionarios, oficina\}, \{oficina, empleo\}, \{sueldo, funcionarios\}, \{cuestionar\}, \{disminuir\})$$

- Supongamos que $Y = (empleo)$. Ver tabla (4.21)

Índ. de aco. fuerte por el promedio	Índ. de aco. fuerte por el máximo
$S(Y\|E^k) = \frac{1/2}{2!} = \frac{1}{4} = 0'25$	$S(Y\|E^k) = max\left(\frac{1}{4}\right) = 0'25$

Tabla 4.21: *Índice de Acoplamiento Fuerte de* E^k *con* $Y = (empleo)$

- Supongamos que $Y = (funcionarios, sueldo)$. Ver tabla (4.22):

Índ. de aco. fuerte por el promedio	Índ. de aco. fuerte por el máximo
$S(Y\|E^k) = \frac{0}{5} = 0$	$S(Y\|E^k) = max(0) = 0$

Tabla 4.22: *Índice de Acoplamiento Fuerte de* E^k *con* $Y = (funcionarios, sueldo)$

- Supongamos que $Y = (funcionarios, oficina)$. Ver tabla (4.23)

Índ. de aco. fuerte por el promedio	Índ. de aco. fuerte por el máximo
$S(Y\|E^k) = \frac{2}{2} = 1$	$S(Y\|E^k) = max(1) = 1$

Tabla 4.23: *Índice de Acoplamiento Fuerte de* E^k *con* $Y = (funcionarios, oficina)$

c) Comparación del índice de acoplamiento fuerte del conjunto Y con T y E^k. Ver tabla (4.24)

El dominio de la estructura-AP es distinto del dominio de la estructura multitérmino y además el índice fuerte se calcula de forma diferente para una y otra estructura, ya que para la estructura multitérmino y con el fin de ponderar el orden de los términos, se calcula teniendo en cuenta todas las combinaciones posibles de los elementos dentro de las *item-seqs* generadoras, por lo que una comparación directa del índice fuerte para ambas estructuras no tiene mucho sentido.

	Estructura-AP T		Estructura Multitérmino E^k	
Índice acoplamiento fuerte	**Promedio**	**Máximo**	**Promedio**	**Máximo**
Y=(empleo)	0’17	0’5	0’25	0’25
Y=(funcionarios, sueldo)	0’2	1	0	0
Y=(funcionarios, oficina)	0’13	0’67	1	1

Tabla 4.24: *Comparación del Índice de Acoplamiento Fuerte de T y E^k para Varios Conjuntos Y*

En el primer y tercer caso, el índice de acoplamiento fuerte es mayor para la estructura multitérmino que para la estructura-AP. Esto sucede porque el dominio de la estructura multitérmino en realidad es más pequeño que el dominio de la estructura-AP y en un dominio más pequeño, el acoplamiento de un determinado conjunto, puede ser mayor que al acoplarlo con un dominio más grande.

En el segundo caso, el índice de acoplamiento fuerte con la estructura multitérmino es cero para el promedio y para el máximo. Esto es porque los elementos del conjunto Y aparecen en una *item-seq* generadora de la estructura multitérmino pero en orden inverso al que aparecen en Y, luego no existe ninguna *item-seq* generadora que se acople de forma exacta con el conjunto Y en E^k, donde, como ya sabemos, el orden importa.

En todos los casos, el índice de acoplamiento fuerte por el máximo es mayor que el índice de acoplamiento fuerte por el promedio.

2. **Índice de acoplamiento débil**

El índice de acoplamiento débil es una medida del grado de acoplamiento débil del conjunto Y con la estructura-AP o la estructura multitérmino, es decir, el grado con que ese conjunto se ajusta al dominio activo de la base de datos, entendiendo que el dominio activo viene representado por la estructura-AP o la estructura multitérmino. Cuando calculamos el índice de acoplamiento débil lo que estamos calculando es el grado en que los términos del conjunto Y (o consulta) aparecen en la base de datos, aunque al ser débil no tienen que aparecer exactamente todos los términos de la consulta, si no que bastaría con que aparecieran algunos de ellos. Este índice también mide la exhaustividad de la respuesta pero sin que esta respuesta sea del todo precisa.

a) Índice de acoplamiento débil de Y con la Estructura-AP por el promedio y por el máximo.

$$T = g(\{funcionarios, oficina, empleo\}, \{funcionarios, cuestionar\}, \{funcionarios, sueldo\}, \{empleo, cuestionar\}, \{disminuir\})$$

- Supongamos que $Y = (empleo)$. Ver tabla (4.25):

Índ. de aco. débil por el promedio	**Índ. de aco. débil por el máximo**
$W(Y\|T) = \frac{1/3+1/2}{5} = 0'17$	$W(Y\|T) = max\left(\frac{1}{3}, \frac{1}{2}\right) = \frac{1}{2} = 0'5$

Tabla 4.25: *Índice de Acoplamiento Débil de T con $Y = (empleo)$*

- Supongamos que $Y = (funcionarios, sueldo)$. Ver tabla (4.26):

Índ. de aco. débil por el promedio	**Índ. de aco. débil por el máximo**
$W(Y\|T) = \frac{1/3+1/2+1}{5} = 0'37$	$W(Y\|T) = max\left(\frac{1}{3}, \frac{1}{2}, 1\right) = 1$

Tabla 4.26: *Índice de Acoplamiento Débil de T con $Y = (funcionarios, sueldo)$*

- Supongamos que $Y = (funcionarios, oficina)$. Ver tabla (4.27)

Índ. de aco. débil por el promedio	**Índ. de aco. débil por el máximo**
$W(Y\|T) = \frac{2/3+1/3+1/3}{5} = 0'27$	$W(Y\|T) = max\left(\frac{1}{3}, \frac{1}{2}, \frac{1}{3}\right) = \frac{2}{3} = 0'67$

Tabla 4.27: *Índice de Acoplamiento Débil de T con $Y = (funcionarios, oficina)$*

b) Índice de acoplamiento débil de Y con la Estructura Multitérmino por el promedio y por el máximo.

$$E^k = g(\{funcionarios, oficina\}, \{oficina, empleo\}, \{sueldo, funcionarios\}, \{cuestionar\}, \{disminuir\})$$

- Supongamos que $Y = (empleo)$. Ver tabla (4.28)

Índ. de aco. débil por el promedio	Índ. de aco. débil por el máximo
$W(Y\|E^k) = \frac{1/2}{2!} = \frac{1}{4} = 0'25$	$W(Y\|E^k) = max\left(\frac{1}{4}\right) = \frac{1}{4} = 0'25$

Tabla 4.28: *Índice de Acoplamiento Débil de* E^k *con* $Y = (empleo)$

- Supongamos que $Y = (funcionarios, sueldo)$. Ver tabla (4.29):

Índ. de aco. débil por el promedio	Índ. de aco. débil por el máximo
$W(Y\|E^k) = \frac{1/2+(1/2+1/2)}{2!+2!} = \frac{3}{8} = 0'37$	$W(Y\|E^k) = max\left(\frac{1}{4}, \left(\frac{1}{4}+\frac{1}{4}\right)\right) = \frac{1}{2} = 0'5$

Tabla 4.29: *Índice de Acoplamiento Débil de* E^k *con* $Y = (funcionarios, sueldo)$

- Supongamos que $Y = (funcionarios, oficina)$. Ver tabla (4.30)

Índ. de aco. débil por el promedio	Índ. de aco. débil por el máximo
$W(Y\|E^k) = \frac{2+1/2+1/2}{2!+2!+2!} = \frac{3}{6} = 0'5$	$W(Y\|E^k) = max\left(1, \frac{1}{4}, \frac{1}{4}\right) = 1$

Tabla 4.30: *Índice de Acoplamiento Débil de* E^k *con* $Y = (funcionarios, oficina)$

c) Comparación del índice de acoplamiento débil del conjunto Y con T y E^k. Ver tabla (4.31)

Vemos como todos los índices están comprendidos entre cero y uno. Y también se cumple en todos los casos que el índice por el máximo es mayor o igual que el índice por el promedio.

El dominio de la estructura-AP es distinto del dominio de la estructura multitérmino y además los índices se calculan de forma diferente para una y otra estructura, ya que para la estructura multitérmino y

	Estructura-AP T		Estructura Multitérmino E^k	
Índice acoplamiento débil	**Promedio**	**Máximo**	**Promedio**	**Máximo**
Y=(empleo)	0'17	0'5	0'25	0'25
Y=(funcionarios, sueldo)	0'37	1	0'37	0'5
Y=(funcionarios, oficina)	0'27	0'67	0'5	1

Tabla 4.31: *Comparación del Índice de Acoplamiento Débil de T y E^k para Varios Conjuntos Y*

con el fin de ponderar el orden de los términos, se calcula teniendo en cuenta todas las combinaciones posibles de los elementos dentro de los conjuntos generadores, por lo que una comparación directa del índice débil para ambas estructuras no tiene mucho sentido.

Destacar que en la estructura multitérmino el índice máximo para la segunda *item-seq*, $Y=\{funcionarios, sueldo\}$, no es 1 porque la estructura multitérmino distingue para $\{funcionarios, sueldo\}$ y $\{sueldo, funcionarios\}$ y además este segundo es el orden en que aparecen estos elementos en la *item-seq* generadora que los contiene.

Para $Y = \{funcionarios, oficina\}$ el índice máximo para la estructura multitérmino es 1 y para la estructura-AP 0'67, esto es porque en el dominio de la estructura multitérmino existe una *item-seq* generadora que es igual que el conjunto Y, por lo que este acoplaría perfectamente, mientras que en la estructura-AP el conjunto generador con el que acopla el conjunto Y se compone de tres términos.

3. Comparación de los Índices Fuerte y Débil

   Por último, compararemos los índices fuerte y débil para la estructura-AP y la *tag cloud* multitérmino (ver tablas (4.32) y (4.33)).

	Ind.Acoplamiento fuerte		Ind. Acoplamiento débil	
Estructura-AP T	**Promedio**	**Máximo**	**Promedio**	**Máximo**
Y=(empleo)	0'17	0'5	0'17	0'5
Y=(funcionarios, sueldo)	0'2	1	0'37	1
Y=(funcionarios, oficina)	0'13	0'67	0'27	0'67

Tabla 4.32: *Comparación de los Índices de Acoplamiento Fuerte y Débil para T para Varios Conjuntos Y*

Tanto para la estructura-AP como para la estructura multitérmino se cumple:

	Ind.Acoplamiento fuerte		Ind. Acoplamiento débil	
Estructura Multitérmino E^k	**Promedio**	**Máximo**	**Promedio**	**Máximo**
Y=(empleo)	0'25	0'25	0'25	0'25
Y=(funcionarios, sueldo)	0	0	0'37	0'5
Y=(funcionarios, oficina)	1	1	0'5	1

Tabla 4.33: *Comparación de los Índices de Acoplamiento Fuerte y Débil para E^k para Varios Conjuntos Y*

- Todos los índices están comprendidos entre 0 y 1.
- El índice por el máximo es mayor o igual en todos los casos que el índice por el promedio.
- El índice de acoplamiento débil por el máximo es mayor o igual para todos los casos que el índice de acoplamiento fuerte por el máximo.
- El índice de acoplamiento débil por el promedio es mayor o igual para todos los casos que el índice de acoplamiento fuerte por el promedio para la estructura-AP. Sin embargo, hay un caso en la estructura multitérmino en el que el índice de acoplamiento débil por el promedio es menor que el índice de acoplamiento fuerte por el promedio y es el caso de Y=$\{$*funcionarios, oficina*$\}$. Esto ocurre porque el conjunto Y se acopla perfectamente con una de los *item-seq* generadoras de la estructura multitérmino, por lo que el índice de acoplamiento fuerte se calcula teniendo en cuenta solamente este conjunto generador. En el cálculo del índice de acoplamiento débil se tienen en cuenta todos los conjuntos generadores con los que el conjunto Y se acopla parcialmente, es por esto que el grado de acoplamiento con el total de conjuntos resulta menor. Esto nos lleva a darnos cuenta, que sólo tiene sentido calcular el índice de acoplamieno débil cuando Y no se acopla completamente con ninguno de los conjuntos generadores, ya que para eso tenemos el índice de acoplamiento fuerte.

Capítulo 5

CONCLUSIONES Y TRABAJOS FUTUROS

En este trabajo se han presentado dos alternativas a la estructura monotérmino representada en forma *tag cloud* encontrada en la Web y que presenta numerosos inconvenientes como podemos ver en la subsección (1.1) y que son descritos con profundidad a lo largo de este trabajo.

Estas alternativas son la estructura-AP y la estructura multitérmino. Ambas solventan los principales inconvenientes de la estructura monotérmino y la visualización en forma de *tag cloud* multitérmino solventa los problemas de la visualización de ambas en forma de *tag cloud* monotérmino. El uso de una u otra estructura, dependerá de ciertos requerimientos del usuario, como pueden ser el orden de los términos en la consulta, lo que apuntaría hacia el uso de la estructura multitérmino o una mayor exhaustividad de la respuesta, para lo que sería más recomendable el uso de la estructura-AP.

La gran ventaja de la estructura-AP y la estructura multitérmino sobre la estructura monotérmino de debe a que permiten el uso de términos con múltiples componentes o “multitérminos”, lo que permite la identificación del contenido de las fuentes de información sobre las cuales se construyen y ofrecen una mayor semántica:

- Son más descriptivas,
- facilitan la inferencia de relaciones entre conceptos
- ayudan en la discriminación de los términos, etc.

Además, ambas estructuras han sido definidas matemáticamente y es posible generarlas con dos métodos estándar que también han sido definidos y cuyo funcionamiento ha sido explicado al emplearlos en el ejemplo de la sección (4).

En este ejemplo, se ha realizado una comparativa de estas dos estructuras para poder discernir con mayor profundidad sus similitudes y diferencias y a su vez, han sido comparadas con la estructura monotérmino, con lo que hemos puesto de manifiesto las ventajas cuantificables que se aprecian sobre ésta.

Son innumerables los trabajos futuros a los que da pie esta breve introducción a la problemática del tratamiento semántico de la información recuperada de internet con fines de consulta y a la solución que se ofrece.

Destacaremos la realización de una propuesta teórica de la estructura-AP ponderada con mayor profundidad de la que se ha empleado en este trabajo, analizando un mayor número de propiedades, estudiando la asociatividad, conmutatividad, etc. de estas estructuras.

De igual forma, quedaría pendiente ampliar la teoría de la estructura multitérmino, estudiando todas las propiedades inherentes a ésta.

Otro posible estudio, sería el del papel de la ponderación en el cálculo de los índices de acoplamiento.

Aunque aquí se ha utilizado una modificación del algoritmo APriori para generar las *item-seqs* frecuentes, sería conveniente el uso de otros algoritmos de generación de secuencias, comprobando su funcionamiento en variadas colecciones de texto, para comparar estos algoritmos y ver cuál de ellos resulta más eficiente.

También se plantea variar los criterios para la visualización de la estructura-AP y la estructura multitérmino en forma de *tag cloud*, criterios tales como:

- El orden alfabético de los términos.
- El color de éstos
- Su dirección horizontal y/o vertical dentro de la *tag cloud*, etc.

con el fin de ver qué visualización es la más atractiva y cuál facilita más la identificación del contenido.

En esta línea, se propone calcular el tamaño de los términos dentro de la *tag cloud* multitérmino mediante alguna fórmula donde se tenga en cuenta la frecuencia, pero también la longitud de éstos, ya que está demostrado que se confunde la longitud del término con el tamaño de la fuente, lo que a veces dificulta la identificación de los términos más relevantes.

Por otro lado, podría calcularse el peso de los términos en la línea que apuntaban Koutrika et al. (Kou09a), dándole distinta puntuación al término según éste aparezca en un atributo u otro de la base de datos. Por ejemplo, en una base de datos de noticias, no será igual de relevante un término que aparezca en el "titulares" a otro que aparezca en "comentarios".

Como trabajo futuro, se plantea también, la consideración de mecanismos de *clustering* y herencia dentro de la visualización de los términos de las *tag clouds* multitérmino.

Se propone como trabajo futuro realizar el experimento de Aouiche et al. (Aou09), que pasaron la información de bases de datos a cubos OLAP y de ahí a *tag clouds* con "multitérminos".

También podría considerarse el uso de etiquetas difusas dentro de la *tag cloud* multitérmino y la aplicación del análisis formal de conceptos.

Como tareas más inmediatas se pretende llevar a cabo la implementación de un sistema que manipule el texto tal como se ha explicado, a través de estructuras-AP y estructuras multitérmino, dándole la opción al usuario de especificar si quiere que se haga con una u otra estructura y lo visualice en forma de *tag cloud* multitérmino, así como la comprobación de su funcionamiento sobre una base de datos creada en Oracle a partir de las noticias de "Digg" (Ros04).

Bibliografía

[30097] 300Sites. http://www.300sites.com/, 1997.

[Agi08] Agili, A., Fabbri, M., Panunzi, A., y Zini, M. Integration of a Multilingual Keyword Extractor in a Document Management System. En *Proceedings of the 6th International Language Resources and Evaluation, LREC*. 2008.

[Agr94] Agrawal, R. y Srikant, R. Fast algorithms for mining association rules. En *Proceeding of the 20th International Conference in Very Large Data Bases, VLDB*, tomo 1215, páginas 487–499. Citeseer, 1994.

[Agr95] Agrawal, R. y Srikant, R. Mining sequential patterns. En *Proceedings of the Eleventh International Conference on Data Engineering*, páginas 3–14. 1995.

[Aou09] Aouiche, K., Lemire, D., y Godin, R. Web 2.0 OLAP: From Data Cubes to Tag Clouds. *Web Information Systems and Technologies*, páginas 51–64, 2009.

[Bat08] Bateman, S., Gutwin, C., y Nacenta, M. Seeing Things in the Clouds: the Effect of Visual Features on Tag Cloud Selections. En *Proceedings of the 19th ACM Conference on Hypertext and Hypermedia*, páginas 193–202. ACM, 2008.

[Beg06] Begelman, G., Keller, P., y Smadja, F. Automated Tag Clustering: Improving Search and Exploration in the Tag Space. En *Collaborative Web Tagging Workshop at WWW2006*. Citeseer, 2006.

[Ber08] Berlocher, I., Lee, K., y Kim, K. TopicRank: Bringing Insight to Users. En *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, páginas 703–704. ACM, 2008.

[BI08] Bar-Ilan, J., Shoham, S., Idan, A., Miller, Y., y Shachak, A. Structured Versus Unstructured Tagging: a Case Study. *Online Information Review*, 32:635–647, 2008.

[Bie05] Bielenberg, K. y Zacher, M. *Groups in Social Software: Utilizing Tagging to Integrate Individual Contexts for Social Navigation*. Tesis Doctoral, Department of Science in Digital Media. University of Bremen, Alemania, 2005.

[Cla09] Clark, J. WordCloud. http://neoformix.com/, 2009.

[Don07] Don, A., Zheleva, E., Gregory, M., Tarkan, S., Auvil, L., Clement, T., Shneiderman, B., y Plaisant, C. Discovering Interesting Usage Patterns in Text Collections: Integrating Text Mining with Visualization. En *Proceedings of the 16th ACM Conference on Information and Knowledge Management*, páginas 213–222. ACM, 2007.

[D'S07] D'Souza, P. RawSugar. http://www.rawsugar.com/, 2007.

[Fei09] Feinberg, J. Wordle. http://www.wordle.net/, 2009.

[For01] Money Makes the World Go 'Round'. *Fortune Magazine*, 2001. Http://money.cnn.com/magazines/fortune/.

[Fra00a] Frantzi, K., Ananiadou, S., y Mima, H. Automatic Recognition of Multi-Word Terms:. the C-Value/NC-Value Method. *International Journal on Digital Libraries*, 3:115–130, 2000.

[Fra00b] Frantzi, K., Ananiadou, S., y Mima, H. TerMine. Automatic Recognition of Multi-Word Terms. http://www.nactem.ac.uk/software/termine, 2000.

[Fuj08] Fujimura, K., Fujimura, S., Matsubayashi, T., Yamada, T., y Okuda, H. Topigraphy: Visualization for Large-Scale Tag Clouds. En *Proceeding of the 17th International Conference on World Wide Web*, páginas 1087–1088. ACM, 2008.

[Gra07] Grahl, M., Hotho, A., y Stumme, G. Conceptual Clustering of Social Bookmarking Sites. En *Proceedings of I-KNOW*, tomo 7, páginas 5–7. 2007.

[Hal07] Halvey, M.J. y Keane, M.T. An assessment of tag presentation techniques. En *Proceedings of the 16th international conference on World Wide Web*, páginas 1314–1315. ACM, 2007.

[Hea08] Hearst, M.A. y Rosner, D. Tag clouds: Data Analysis Tool or Social Signaller? En *Hawai International Conference on System Sciencies (HICSS)*, páginas 160–169. IEEE Computer Society, 2008.

[Hey06] Heymann, P. y Garcia-Molina, H. Collaborative Creation of Communal Hierarchical Taxonomies in Social Tagging Systems. *Infolab: Technical Report. University of Stanford*, 2006.

[HM06] Hassan-Montero, Y. y Herrero-Solana, V. Improving Tag-Clouds as Visual Information Retrieval Interfaces. En *International Conference on Multidisciplinary Information Sciences and Technologies*, páginas 25–28. Citeseer, 2006.

[Hot06] Hotho, A., Jaschke, R., Schmitz, C., y Stumme, G. Information Retrieval in Folksonomies: Search and Ranking. *The Semantic Web: Research and Applications*, páginas 411–426, 2006.

[How09] Howard, H. Knowledge Discovery in Databases. *Online Notes. Computer Science. University of Regina*, 2009.

[Hsi06] Hsieh, W.T., Lai, W.S., y Chou, S.C.T. A Collaborative Tagging System for Learning Resources Sharing. *Current Developments in Technology-Assisted Education*, 2:1364–1368, 2006.

[Kas07] Kaser, O. y Lemire, D. Tag-Cloud Drawing: Algorithms for Cloud Visualization. *Proceedings of the Workshop on Tagging and Metadata for Social Information Organization (WWW2007)*, 2007.

[Kna99] Knautz, K., Soubusta, S., y Stock, W.G. Tag Clusters as Information Retrieval Interfaces. En *Hawai International Conference on System Sciencies (HICSS)*, páginas 1–10. IEEE Computer Society, 1899.

[Kou09a] Koutrika, G., Zadeh, Z.M., y Garcia-Molina, H. CourseCloud: Summarizing and Refining Keyword Searches over Structured Data. En *Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology*, páginas 1132–1135. ACM, 2009.

[Kou09b] Koutrika, G., Zadeh, Z.M., y Garcia-Molina, H. Data Clouds: Summarizing Keyword Search Results over Structured Data. En *Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology*, páginas 391–402. ACM, 2009.

[Kuo07] Kuo, B.Y.L., Hentrich, T., Good, B.M., y Wilkinson, M.D. Tag Clouds for Summarizing Web Search Results. En *Proceedings of the 16th International Conference on World Wide Web*, páginas 1204–1205. ACM, 2007.

[Lud04] Ludicorp. Flickr. http://www.flickr.com/, 2004.

[Mar06] Marin, N., Martin-Bautista, MJ, Prados, M., y Vila, MA. Enhancing Short Text Retrieval in Databases. *Flexible Query Answering Systems*, páginas 613–624, 2006.

[Mar08] Martínez, S. *Una solución semántica al tratamiento de atributos textuales en un Modelo Relacional Orientado a Objetos: Implementación en Software Libre*. Tesis Doctoral, Departamento de Ciencias de la Computación e Inteligencia Artificial. Universidad de Granada, 2008.

[MB08] Martín-Bautista, M., Vila, MA, y Martínez-Folgoso, S. A New Semantic Representation for Short Texts. En *Data Warehousing and Knowledge Discovery*, tomo 5182, páginas 347–356. 2008.

[Mil76] Milgram, S. y Jodelet, D. Psychological Maps of Paris. *Environmental psychology*, páginas 104–124, 1976.

[Nab09] Nabru, GmbH y CoKG. Tag Cloud Generator. http://www.tagcloud-generator.com/, 2009.

[Pan06] Panunzi, A., Marco, F., y Massimo, M. Integrating Methods and LRs for Automatic Keyword Extraction from Open Domain Texts. En *Proceedings of the 5th International Language Resources and Evaluation (LREC)*, páginas 1917–1920. 2006.

[Pap10] Papadopoulos, S., Kompatsiaris, Y., y Vakali, A. A Graph-Based Clustering Scheme for Identifying Related Tags in Folksonomies. *Data Warehousing and Knowledge Discovery*, páginas 65–76, 2010.

[Riv07] Rivadeneira, AW, Gruen, D.M., Muller, M.J., y Millen, D.R. Getting our Head in the Clouds: Toward Evaluation Studies of Tagclouds. En *Proceedings of the Computer/Human Interaction (CHI)*, páginas 998–1001. ACM, 2007.

[Ros04] Rose, K. Digg. http://digg.com/, 2004.

[Sch03] Schachter, J. Delicious. http://delicious.com/, 2003.

[Sch06] Schmitz, P. Inducing Ontology from Flickr Tags. En *Collaborative Web Tagging Workshop at WWW2006*, páginas 210–214. Citeseer, 2006.

[Sch09] Schrammel, J., Leitner, M., y Tscheligi, M. Semantically Structured Tag Clouds: An Empirical Evaluation of Clustered Presentation Approaches. En *Proceedings of the 27th International Conference on Human Factors In Computing Systems*, páginas 2037–2040. ACM, 2009.

[Sha05] Shaw, B. Utilizing folksonomy: Similarity metadata from the del. icio. us system. *Project Proposal.*, 2005. Http://www.metablake.com/web folk/web-project.pdf.

[Sie09] Siegel, S. Semantic Cloud. http://semanticcloud.rieskamp.info/, 2009.

[Sin08] Sinclair, J. y Cardew-Hall, M. The folksonomy tag cloud: When is it useful? *Journal of Information Science*, 34:15–30, 2008.

[Ste06] Steinbock, D. TagCrowd: Create your own Tag Cloud from any Text. http://tagcrowd.com/, 2006.

[Tan06] Tan, P.N., Steinbach, M., y Kumar, V. *Introduction to Data Mining*. Pearson Addison Wesley Boston, 2006.

[Tao03] Tao, F., Murtagh, F., y Farid, M. Weighted Association Rule Mining using weighted support and significance framework. En *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, páginas 661–666. ACM, 2003.

[VdB09] Van den Broeck, W. y Cattuto, C. netr.it. http://www.netr.it/, 2009.

[VH09] Van-Ham, F., Wattenberg, M., y Viégas, F. Mapping Text with Phrase Nets. *IEEE Transaction on Visualization and Computer Graphics*, 15:1169–1176, 2009.

[VI05] Von-Isenburg, M. Hub Log. http://hublog.hubmed.org/archives/001049.html, 2005.

[Vie07] Viegas, F. y Wattenberg, M. ManyEyes. http://manyeyes.alphaworks.ibm.com/manyeyes/, 2007.

[Vie08] Viegas, F. y Wattenberg, M. TIMELINES Tag clouds and the case for vernacular visualization. *Interactions*, 15:49–52, 2008.

[Vie09] Viegas, F., Wattenberg, M., y Feinberg, J. Participatory Visualization with Wordle. *IEEE Transactions on Visualization and Computer Graphics*, 15:1137–1144, 2009.

[VW05] Vander-Wall, T. Folksonomy presentation at Online Information. 2005.

[VW09] Vander-Wall, T. Off de Top: Folksonomy Entries. http://www.vanderwal.net/about.php, 2009.

[Wat08] Watters, D. y Chicago, IL. Meaningful Clouds: Towards a Novel Interface for Document Visualization. *Online Notes. University of Chicago*, 2008.

[Wu06] Wu, X., Zhang, L., y Yu, Y. Exploring Social Annotations for the Semantic Web. En *Proceedings of the 15th International Conference on World Wide Web*, páginas 426–435. ACM, 2006.

[Xex09] Xexéo, G., Morgado, F., y Fiuza, P. Automatically Generated Tag Clouds. *XXIV Simpósio Brasileiro de Banco de Datos*, 2009.

[Yu06] Yu, T.Z. y Chien, L.F. *Automatic Organization of User-Generated Tags from the Web*. Tesis Doctoral, National Taiwan University, 2006.