UNIVERSIDAD DE GRANADA

TRATAMIENTO SEMÁNTICO DE INFORMACIÓN TEXTUAL EN BASES DE DATOS

Tesis doctoral presentada por Úrsula Torres Parejo

dentro del Programa de Doctorado en Tecnologías de la Información y la Comunicación

Dirigida por: Dr. Miguel Delgado Calvo-Flores

y Dra. Mª Amparo Vila Miranda

Editor: Editorial de la Universidad de Granada
Autor: Úrsula Torres Parejo
D.L.: En trámite
ISBN: En trámite

UNIVERSIDAD DE GRANADA

TRATAMIENTO SEMÁNTICO DE INFORMACIÓN TEXTUAL EN BASES DE DATOS

Tesis doctoral presentada por Úrsula Torres Parejo

dentro del Programa de Doctorado en Tecnologías de la Información y la Comunicación

Dirigida por: Dr. Miguel Delgado Calvo-Flores

y Dra. Mª Amparo Vila Miranda

El doctorando | El director | El director

Granada, Octubre de 2013

El doctorando Úrsula Torres Parejo y los directores de la tesis Miguel Delgado Calvo-Flores y MªAmparo Vila Miranda Garantizamos, al firmar esta tesis doctoral, que el trabajo ha sido realizado por el doctorando bajo la dirección de los directores de la tesis y hasta donde nuestro conocimiento alcanza, en la realización del trabajo, se han respetado los derechos de otros autores a ser citados, cuando se han utilizado sus resultados o publicaciones.

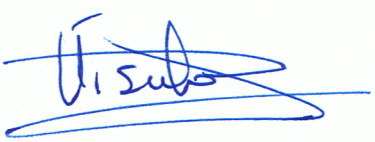
En Granada, a 02 de Diciembre de 2013.

Directores de la Tesis

Fdo: Miguel Delgado Calvo-Flores

Fdo: Mª Amparo Vila Miranda

Doctorando



Fdo: Úrsula Torres Parejo

A mis padres
A Raúl

Agradecimientos

Cuando inicié mis estudios universitarios no me planteaba donde me llevaría ese camino. Al echar la vista atrás me doy cuenta del largo recorrido por una travesía llena de desafíos, logros, frustraciones... durante la que he ido formándome académica y personalmente, orientando el rumbo de mi vida con cada pequeña decisión. Estoy orgullosa de haber llegado al punto en que me encuentro, no ha sido sencillo y desde luego no lo hubiera conseguido sin el apoyo de todas las personas que me rodean y a las que quiero hacer llegar mi más sincero agradecimiento.

En primer lugar quiero agradecer a Miguel y Amparo la magnífica oportunidad que me dieron de realizar con ellos la tesis doctoral, así como su confianza y apoyo en las decisiones más difíciles. De no ser por ellos yo no habría llegado tan lejos, ya que me han proporcionado todos los medios necesarios para progresar, dándome los más sabios consejos con total altruismo. Gracias Miguel por ser tan bueno, por tu cercanía desde el primer momento en que te conocí, por tu forma de ver la vida y tu manera de tratar a los demás. Amparo, en ti tengo un ejemplo constante de esfuerzo y humanidad, gracias por iluminarme con tantas buenas ideas, por encontrar la alegría en el trabajo y saber contagiarla a los demás. Hay personas que nos cambian la vida y a las que nunca podremos devolverles una porción del bien que nos producen. Recibid al menos este profundo agradecimiento.

En segundo lugar quiero hacer una agradecimiento especial a Jesús, ya que con su inestimable colaboración y ayuda ha sido posible la finalización de este trabajo. Gracias por ser tan buen compañero, tan trabajador, tan generoso, por no quejarte nunca e inculcarme parte de

tu perfeccionismo y rigurosidad. Gracias por tus valiosos consejos, por comprenderme y apoyarme. Siempre estaré en deuda contigo.

A todos los miembros del Departamento de Ciencias de de la Computación e Inteligencia Artificial de la Universidad de Granada y a mis compañeros becarios, en especial a Rita.

A mis colegas del Departamento de Estadística e Investigación Operativa de la Universidad de Cádiz, que me hicieron sentir en casa desde el primer momento, especialmente a Miguel Ángel, que siempre me ha facilitado las cosas y a Inma, por su buena disposición, por ser tan afable, trabajadora y comprensiva.

El pilar básico donde todo comienza es la familia. Es tanto lo que tengo que agradecer a mis padres que no sé por donde empezar. Ellos han sido personas especialmente pacientes y nunca han perdido la esperanza en mí, a pesar de haberles dado más de una preocupación. Si no fuera por ellos, yo no hubiera encontrado la motivación para llegar hasta aquí. Ellos me han enseñado a ser valiente, a actuar con predisposición, a saber afrontar los problemas y hallar la satisfacción en el trabajo de cada día. Gracias mamá por estar siempre ahí, dándome ánimos y haciéndome ver pequeñas todas las dificultades, por ser tan comprensiva, cariñosa y tan llena de bondad. Gracias papá por tantas inquietudes que has despertado en mí desde pequeña, que me han hecho valorar la vida y descubrir el placer en las cosas sencillas, por preocuparte tanto, por querer siempre lo mejor para mí. Sin duda os debo a los dos todos mis logros y mi felicidad.

El siguiente agradecimiento es para Raúl, mi compañero de penas y alegrías, que siempre ha sabido ver lo mejor de mí. Es por él que quiero ser mejor persona cada día. Él me da la estabilidad y fuerza que necesito, la seguridad para avanzar. Gracias por poner luz en mis días, endulzarme el camino, compartir conmigo mis sueños y todas mis aflicciones. Tú le das sentido a todo lo que hago y consigues que no me venga abajo. Tú haces que sonría en los momentos más sombríos.

Al resto de mi familia, especialmente a mi hermano, por estar ahí cuando le necesito, a mi tía Aurora, por alegrarnos siempre, a mi tío Antonio, por ser tan bueno con nosotros y a mis suegros, Germán y Antonia, por ser personas tan discretas y con tan buen corazón.

Por último, pero no menos importante, quiero agradecer a mis amigos que estén siempre conmigo. Son ellos quienes me dan plenitud y ponen en mi vida un toque de color, especialmente gracias a Vane, Sandra, Maribel, Elena, Jorge y David. Vuestra amistad es mi mayor tesoro y fortaleza.

diciembre 2013

Este trabajo ha sido posible gracias al soporte de la Junta de Andalucía, bajo el proyecto de investigación “Un sistema para la movilización del Conocimiento contenido en una Base de Datos poco estructurada. Aplicación a los textos de una Base de Historias Clínicas (P07 TIC 02786)”.

Índice general

Índice de figuras

Índice de tablas

CAPÍTULO

1

Introducción

El problema que nos lleva a la realización de esta memoria de tesis doctoral se plantea a continuación, en la Sección 1.1. Los objetivos que nos proponemos alcanzar con respecto a este problema y a su solución, se exponen en la Sección 1.2. Para alcanzar estos objetivos se ha realizado el trabajo que se desarrolla en los siguientes capítulos, los cuales están organizados según se detalla en la Sección 1.3.

1.1 Planteamiento del Problema

Con la llegada de las nuevas tecnologías a nuestra sociedad, es cada vez mayor la cantidad datos y conocimiento disponible para los usuarios. Un procesamiento adecuado de esta información es fundamental para poder hacer uso de ella y realizar tareas complejas como filtrado o clasificación.

Cuando la información está estructurada, este procesamiento resulta sencillo, pero la complicación se va incrementando cuanto mayor es la falta de estructura en las fuentes de conocimiento.

El exceso de información y la dificultad de procesar los datos no estructurados son inconvenientes presentes en los sistemas de bases de datos desde hace mucho

tiempo. Este problema se acentúa con el creciente uso de Internet, pero el núcleo de la cuestión continúa siendo el mismo. Mucha información que se acumula no llega a transformarse en información útil para el usuario, debido a que las formas de procesarla y visualizarla no son lo suficientemente eficientes para presentarla de forma que pueda interpretarse y consultarse de manera satisfactoria.

El hecho de no procesar correctamente la información textual de los atributos de una base de datos, se traduce en que habrá una parte muy útil de ésta que quedará inaccesible y también en que el sistema devolverá al usuario más información de la que le solicita y de forma desorganizada. Aunque los atributos textuales suelen ser objeto frecuente de consulta, la mayoría de los sistemas procesan estas consultas basándose en la búsqueda de coincidencias sintácticas y obviando cualquier tipo de semántica asociada al texto [Cn09].

Supongamos, por ejemplo, una base de datos que contiene información sobre noticias recuperadas de Internet, donde cada atributo de la base de datos contiene un elemento de las noticias; título, subtítulo, categoría, autor, resumen, etc. El primer inconveniente que encuentra un usuario a la hora de consultar, es que puede no tener conocimiento alguno acerca del contenido de la base de datos y en un escenario ideal, donde sí conociera al menos una parte de esta información, se encontraría con un segundo inconveniente, el de que sus consultas no siempre recuperan información precisa de acuerdo con sus necesidades. Para aminorar el primer inconveniente, se hace indispensable una visualización previa que resuma de forma óptima el contenido de la información y para que la información recuperada sea más precisa, es necesario tener presente la semántica asociada a los términos introducidos con la consulta.

Una de las soluciones aportadas en Internet para mermar estos inconvenientes viene de la mano de los sistemas basados en etiquetado (*tagging*), que permiten al usuario categorizar las fuentes de información mediante las denominadas etiquetas (*tags*), con el fin de poder recuperarlas con posterioridad. A su vez, estos sistemas han popularizado una herramienta de visualización de texto denominada "*Tag Cloud*", de la que hablaremos en profundidad en la Sección 2.1 y en la que se muestran las etiquetas asignadas por los usuarios con mayor frecuencia, para de esta forma representar el contenido de la información etiquetada y navegar a través de ésta.

La *tag cloud* es una herramienta muy conocida, sencilla, llamativa y fácil de usar, por lo que nos parece una buena idea emplearla para visualizar el contenido de la información textual en las bases de datos. La potencia de las *tag clouds* reside en el esfuerzo colaborativo de los usuarios que etiquetan y clasifican las fuentes de información. La frecuencia en el uso de una determinada etiqueta para una particular fuente de información, confiere a esa etiqueta cierta precisión como elemento de clasificación. En bases de datos, este tipo de etiquetado colectivo no es fácilmente aplicable, por lo que la forma de obtener la *tag cloud* es a partir de etiquetas extraídas del propio texto a través de algún método que les permita conservar su semántica.

Por otro lado, tanto los sistemas basados en *tagging* como la propia *tag cloud*, han sido ampliamente criticados en la literatura debido a numerosas deficiencias que se verán con detalle en las Secciones 2.1.1 y 2.1.4.

La mayoría de las deficiencias que acaecen a la *tag cloud* se deben al uso exclusivo de monotérminos en las etiquetas. Resumimos estos defectos a continuación:

- **Con respecto a la identificación del contenido.** La *tag cloud* muchas veces lleva a concepciones erróneas sobre el contenido de la información, debido a la ambigüedad de los términos frecuentes o populares que son los que aparecen representados, a pesar de ser en numerosas ocasiones los que menos discriminan.

- **Con respecto a la semántica.** No permite inferir relaciones entre los conceptos representados, siendo necesaria la aplicación de algún mecanismo adicional, como *clustering*.

- **Con respecto a la teoría.** Aunque algunos autores han presentado un modelo formal para describirla de forma matemática (ver [Xex09]), este modelo no se ajusta a la *tag cloud* tradicional que encontramos en la Web. La falta de una definición estándar acarrea numerosos problemas teóricos, a pesar de lo cual no es difícil ver el uso de la *tag cloud* con fines analíticos, como en comparación de textos.

- **Con respecto al método.** Tampoco existe un método estándar para su generación.

Con la intención de encontrar una solución que ofrezca al usuario un escenario donde pueda ver representado el contenido de la información y acceder a ésta de forma precisa, partimos de la siguiente **hipótesis**:

Se puede obtener una forma de representación del conocimiento que mantenga la semántica de los atributos textuales en una base de datos. Dicha forma de representación, conocida como forma intermedia, podría implementarse como un Tipo de Dato Abstracto (TDA) que permita manejar los atributos de la base de datos y obtener una estructura global de conocimiento.

A su vez, esta forma de representación podría visualizarse a través de una *tag cloud* multitérmino, que favorecería la identificación del contenido de la información al discriminar el significado de los términos debido al empleo de términos relacionados en una misma etiqueta y que, por esta misma razón, estaría dotada de semántica. Como ésta proviene de la forma intermedia, estaría definida matemáticamente y poseería un método estándar de generación, con lo que se solventarían los defectos achacados a este tipo de visualización.

1.2 Objetivos

El objetivo principal de este trabajo es establecer una metodología de extracción semántica de información en textos no estructurados en bases de datos, sirviéndonos para ello de formas matemáticas intermedias, que permitan el uso de multitérminos y que puedan presentarse al usuario en forma de *tag cloud*. Esta visualización debe facilitar la identificación del contenido del texto representado y servir para diversas tareas, como para sugerir términos de búsqueda, para la exploración y la consulta. Este proceso queda reflejado en el esquema gráfico de la Figura 1.1.

Es deseable que el diseño de esta *tag cloud* disminuya los inconvenientes que se han visto en la tradicional, compuesta únicamente por monotérminos. Por otro lado, se pretende que la información recuperada gane en precisión gracias al establecimiento de la semántica.

Para alcanzar estos objetivos, hemos pensado en una estructura que fue desarrollada en el curso de otra investigación en nuestro grupo y que conocemos como estructura-AP [MF08, MB08, MB06].

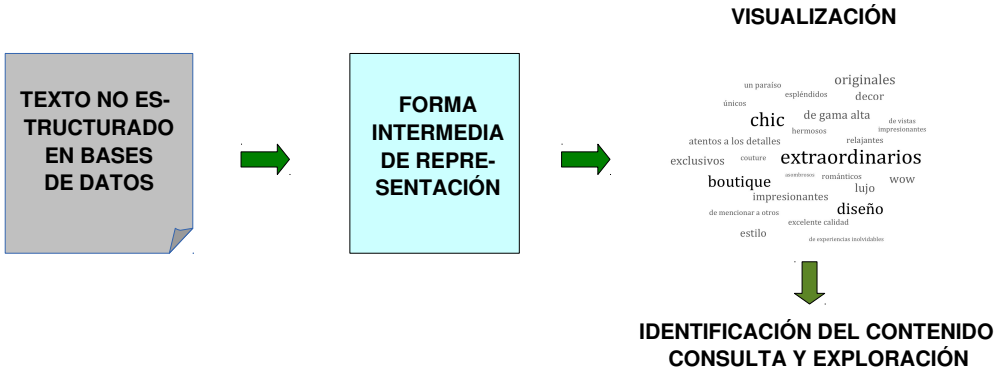


Figura 1.1: Esquema gráfico del proceso global

La estructura-AP tiene la propiedad de mantener la semántica del texto, al permitir que los términos relacionados puedan permanecer unidos. Se hablará de ella en la Sección 2.2.

La idea es obtener la estructura-AP y representarla a través de una *tag cloud*, aprovechando así el éxito que esta visualización tiene en la Web debido a su atractivo diseño, además, como ésta será una *tag cloud* semántica, multitérmino, que vendrá dada por una estructura matemática, solventará los principales inconvenientes de la *tag cloud* monotérmino, que es la que usualmente vemos empleada. Aunque cada vez es más frecuente encontrar representaciones donde las etiquetas son multitérmino, éstas carecen de modelo matemático y de método estándar de generación, suelen proceder de etiquetas compuestas que han utilizado los usuarios para marcar alguna fuente de información.

Para poder representar la estructura-AP a través de una *tag cloud*, donde cada etiqueta tiene distinto tamaño según su frecuencia de aparición en el texto, se deben ponderar los componentes de la estructura según la frecuencia de éstos. Creamos así lo que conocemos como “Estructura-AP Ponderada (Estructura WAP)”.

El problema con la estructura-AP es que no discrimina según el orden de los términos en el texto ni según el tipo de adyacencia que presenten los unos con los otros. Al introducir éstos en la estructura-AP creamos la “Estructura-AP Ordenada (Estructura APO)”, que también deberá ser ponderada para facilitar su visualización a través de una *tag cloud*. Esto nos lleva a la “Estructura-AP Ordenada

Ponderada (Estructura WAPO)".

Resumiendo, los objetivos que se plantean de forma global son:

- Establecer un procedimiento de extracción semántica de información en textos no estructurados en bases de datos.
- Ofrecer una visualización en la que el contenido quede representado y que ayude en las tareas de exploración y consulta.
- Introducir semántica en la *tag cloud* tradicional, así como una fundamentación teórica y una metodología de obtención.
- Conseguir una mayor precisión en la consulta.
- Dotar de orden y de ponderación a la estructura-AP.

Para alcanzar estos objetivos generales nos fijamos los siguientes objetivos específicos o tareas a realizar:

1. **Estudiar los antecedentes del problema planteado y de su solución**. Principalmente los antecedentes de la *tag cloud* como herramienta de visualización y de la estructura-AP como forma matemática intermedia.
2. **Establecer las definiciones formales de las estructuras APO y WAPO**, que son las que introducen orden y ponderación en la estructura-AP. Éstas representarán el atributo textual como TDA y nos servirán para definir todas las operaciones que pueden realizarse sobre éste.
3. **Estudiar métodos alternativos para la generación de la forma intermedia**. Hasta ahora se ha empleado el algoritmo Apriori [Agr94] para la generación de la estructura-AP. Para las estructura APO y WAPO debe realizarse una modificación de éste y examinar otros métodos alternativos como es la generación a través de implicaciones frecuentes.
4. **Plantear una metodología y la arquitectura del sistema para obtener el TDA de un atributo textual**. Detallar todas las tareas de preprocesamiento que se deberán realizar sobre el texto hasta obtener la *tag cloud* de la forma intermedia de representación y las herramientas empleadas en cada etapa.

5. **Visualizar la forma intermedia a través de una *tag cloud***. Mejorar con ésta a la *tag cloud* tradicional en los siguientes aspectos:

   - **Identificación del contenido.** Facilitarla con la ayuda de los componentes multitérmino.
   - **Semántica.** Matizar el significado de un término, en principio ambiguo, mediante sus términos relacionados y posibilitar la identificación de relaciones entre conceptos. Pensemos por ejemplo en el término "sistema", utilizado en multitud de contextos diferentes. Con el uso de etiquetas multitérmino en la *tag cloud*, podremos tener una etiqueta que sea "sistema operativo" y otra que sea "sistema nervioso". De esta forma estamos matizando en cada caso el significado de "sistema" según su contexto, infiriéndole semántica y permitiendo relacionar conceptos como "sistema" y "operativo" o "sistema" y "nervioso".
   - **Fundamentación teórica.** Dotar a la *tag cloud* de una base matemática sólida, lo que se consigue gracias a la forma matemática intermedia. De esta manera podrá utilizarse, entre otras cosas, con fines analíticos.
   - **Metodología de obtención.** Ésta debe ser aplicable sobre distintos formatos de texto corto.

6. **Validación del procedimiento planteado**. Por último, comprobar que la metodología descrita cumple las expectativas esperadas mediante la realización de experimentos sobre distintos conjuntos de datos, el cálculo de métricas y encuestas que nos permitan conocer la opinión de los usuarios.

1.3 Contenido de la Memoria

La memoria se organiza de la siguiente forma:

- En el Capítulo 2 veremos los **Antecedentes** tanto de la *tag cloud* como de la estructura-AP.

  Comenzaremos hablando de los sistemas basados en *tagging* como precursores de la *tag cloud* tal como hoy la conocemos. Luego veremos las distintas

terminologías con que ésta se denomina, dependiendo del método de extracción y otras variantes. Haremos una revisión bibliográfica de la *tag cloud* y comentaremos sus características, ventajas e incovenientes, aspectos sobre el diseño, análisis de los métodos existentes para la extracción de etiquetas, etc. Posteriormente se hablará de la *tag cloud* monotérmino "versus" multitérmino, veremos en qué aspectos la multitérmino mejora a la monotérmino. Veremos un tipo especial de *tag clod*: la *data cloud*, propuesta por [Kou09a] para datos estructurados. Repasaremos otros procesos que tratan de asignar semántica a las etiquetas, como *clustering*, herencia y ontologías y terminaremos mencionando otras apariencias con que puede encontrarse a la *tag cloud*.

Posteriormente estableceremos la definición formal de estructura-AP, para lo cual será necesario introducir el concepto de conjunto-AP y algunas de sus operaciones. También se darán algunas de las propiedades y operaciones de la estructura-AP y terminaremos con el acoplamiento de conjuntos de términos con estructuras. En el escenario que nos ocupa, estos conjuntos representan los requerimientos de los usuarios en la consulta y la estructura-AP el TDA del atributo textual consultado.

- En el Capítulo 3 estableceremos nuestra **Propuesta Teórica**, definiendo los modelos matemáticos de la estructura APO y la estructura WAPO. Además, introduciremos un tercer modelo, la estructura WAP (estructura-AP Ponderada), con el fin de poder visualizar también la estructura-AP a través de una *tag cloud*.

  Esta última se construye ponderando la estructura-AP, por lo que empezaremos definiendo los *itemsets* ponderados que compondrán los conjuntos-AP ponderados o conjuntos WAP, para definir posteriormente la estructura WAP y sus principales operaciones.

  Para definir la estructura APO, tendremos que empezar distinguiendo entre un componente monotérmino y un componente multitérmino. Será necesario definir la estructura monotérmino, que será la que subyace bajo una *tag cloud* monotérmino. La importancia de este tipo de *tag cloud* reside en que es la

que se utiliza con mayor frecuencia y por lo tanto, la usaremos con fines comparativos.

La estructura APO se compone de conjuntos multitérmino, que son conjuntos cuyas componentes son multitérmino y además tienen establecida una relación de orden. Los conoceremos como secuencias-AP o AP-Seqs.

Cuando estas AP-Seqs estén ponderadas, compondrán la estructura WAPO. Veremos las operaciones y propiedades más importantes y estableceremos unos índices para cuantificar los acoplamientos de las AP-Seqs con la estructura APO, operación que reviste de especial importancia en la consulta.

Terminaremos con un ejemplo práctico en el que, a partir de unas pocas tuplas, generaremos las estructuras-AP, monotérmino y APO, con sus respectivas ponderaciones y las visualizaremos mediante *tag clouds*, comentando paso a paso los métodos de generación utilizados, realizando una comparación y detallando el proceso de consulta con las estructuras-AP y APO.

- En el Capítulo 4 veremos distintas formas de **Generación de las Estructuras**: a partir del algoritmo Apriori [Agr94] y a partir de implicaciones frecuentes [Blu87].

  Para hacerlo de esta segunda forma, es preciso contar con un índice invertido completo, cuya definición estableceremos previamente junto a la de lista invertida.

  Daremos otras definiciones previas a la de implicación y desarrollaremos un ejemplo para ilustrar cómo se obtienen las estructuras a partir de implicaciones frecuentes.

  Terminaremos con unas conclusiones sobre cuándo es mejor emplear el algoritmo Apriori y cuándo generar la estructura a partir de implicaciones.

- En el Capítulo 5 recorreremos el camino **Del Atributo Textual a la Tag Cloud**.

  Empezaremos explicando de modo general la metodología a seguir para la representación semántica de textos no estructurados, haciendo especial hincapié en el preprocesamiento de los datos, tanto sintáctico como semántico.

Veremos qué herramientas se han utilizado para estos preprocesamientos, para la generación de formas intermedias y de la *tag cloud*.

Finalizaremos detallando las etapas de postprocesamiento y visualización.

- En el Capítulo 6 realizaremos una **Evaluación Experimental** de la metodología propuesta.

  Presentaremos un primer experimento sobre una base de datos electrónica de artículos científicos de la revista "Security and Communication Network", de Wiley [1]. Seleccionamos esta revista porque ofrece en su página web una *tag cloud* con la que comparar la que obtenemos empleando nuestra metodología.

  Esta comparación se hará a través del cálculo de métricas y de una encuesta de usabilidad.

  Realizaremos un segundo experimento sobre una base de datos de historias clínicas, la cual posee características diferentes a la de artículos científicos, por ejemplo, está en español y es más extensa y desorganizada.

  Generaremos una *tag cloud* que evaluaremos también a través de métricas y de una encuesta a usuarios expertos para conocer el grado de satisfacción con la herramienta.

  Terminaremos con otros experimentos en los que se generan *tag clouds* de diferentes conjuntos de datos y de las que se evalúa su cobertura, solapamiento y balance.

- Por último, en el Capítulo 7, estableceremos algunas conclusiones llevadas a cabo a partir de este trabajo y realizaremos una serie de propuestas para trabajos futuros.

[1] http://onlinelibrary.wiley.com/

CAPÍTULO

2

Antecedentes

La idea de este capítulo es presentar los antecedentes de nuestro trabajo, para ello haremos un estudio profundo de los antecedentes de la *tag cloud* como forma de visualización, desde su procedencia hasta nuestros días, en los que encontramos numerosos tipos diferentes de *tag clouds*. Empezaremos hablando de los sistemas basados en *tagging* como precursores de esta herramienta, posteriormente analizaremos sus características, sus aspectos negativos, sus aportaciones, las *tag clouds* construidas a partir de bases de datos estructuradas, los distintos métodos que existen hasta el momento para aplicar semántica a las etiquetas y las grandes ventajas de las componentes multitérmino.

No menos importante es la introducción de los antecedentes de la estructura-AP, como forma de representación intermedia del texto a visualizar, ya que de esta estructura parten las extensiones teóricas que se proponen en el Capítulo 3 y sobre las que se basa este trabajo.

La idea es obtener, a partir de los atributos textuales de una base de datos, las extensiones referidas de la estructura-AP y visualizarlas a través de una *tag cloud* para identificar el contenido de estos atributos y para facilitar las tareas de exploración y búsqueda, entre otras.

La estructura-AP y sus extensiones se componen de conjuntos de *itemsets* frecuentes y conjuntos de secuencias frecuentes. Veremos los antecedentes a los algoritmos de obtención de estos conjuntos y secuencias frecuentes en el Capítulo 4. Nos parece más apropiado verlo ahí, debido a que en ese capítulo se exponen los algoritmos utilizados para la generación de las estructuras.

Por lo tanto, en la Sección 2.1 se hará el estudio de los antecedentes de la *tag cloud* y en la Sección 2.2 revisaremos los antecedentes de la estructura-AP. Finalmente, en la Sección 2.3 haremos un resumen, obtendremos algunas conclusiones y justificaremos el tipo de *tag cloud* a utilizar en nuestro trabajo.

2.1 Antecedentes de la Tag Cloud

Empezaremos hablando de los sistemas basados en etiquetado (*tagging*) como precursores de la *tag cloud*, tal como se conoce actualmente.

2.1.1 Sistemas Basados en Etiquetado (*Tagging*)

Una folksonomía es el resultado del etiquetado libre y personal de información y objetos 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 (*tagging*) permiten a los usuarios categorizar las fuentes de información web mediante etiquetas (*tags*), que son palabras clave (*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 las 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 esa fuente de información.

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, que es lo que llamamos 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*) e 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 una nueva fuente de información sea indexada 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 con-

sulta actúan como etiquetas y el usuario es el que discrimina entre las fuentes de información recuperadas.

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] y puede ser representada de distintas formas, tales como tensores o hipergrafos [Mar12].

Marinho et al. [Mar12] definen la folksonomía como una estructura relacional $F := (U, R, T, Y)$ donde:

- U, R y T son conjuntos finitos disjuntos no vacíos, cuyos elementos son los usuarios, las fuentes de información y las etiquetas respectivamente.
- Y es el conjunto de relaciones ternarias entre ellos; $Y \subseteq U \times R \times T$, cuyos elementos son las asignaciones de etiquetas por los usuarios a las fuentes de información.
- Un *post* corresponde al conjunto de etiquetas asignadas por los usuarios para una determinada fuente de información, esto es, la tríada $(u, r, T_{u,r})$ con $u \in U$, $r \in R$ y siendo el conjunto no vacío $T_{u,r} := \{t \in T | (u, r, t) \in Y\}$

Lo que se considere como fuente de información dependerá del tipo de sistema.

Yager y Reformat [Yag10] modelan la folksonomía a través de conjuntos difusos. Para ello, entienden que una fuente de información puede ser un conjunto difuso sobre las etiquetas, ya que una única fuente de información estará marcada con un número determinado de etiquetas, cada una de las cuales tendrá un grado de pertenencia a dicha fuente de información. Este grado de pertenencia se calcula en base al número de veces que la etiqueta se asigna a la fuente de información. A su vez, las etiquetas también pueden representarse como conjuntos difusos sobre las fuentes de información. Esto traería una nueva dimensión a la *tag cloud*, la dimensión de "grado", proporcionando una forma de formalizar la imprecisión. Una vez que las relaciones entre etiquetas y fuentes de información se han expresado a través de conjuntos difusos, pueden aplicarse técnicas basadas en lógica difusa para razonar acerca de la similaridad entre las etiquetas o las fuentes de información.

La *tag cloud* es una visualización usada por los sistemas basados en *tagging* como interfaz de recuperación visual de información, que además de representar el conocimiento, permite la navegación [HM06] a través de éste. La *tag cloud* traslada el vocabulario emergente de una folksonomía en una herramienta social de navegación [Sin08].

Cuando un usuario pulsa con el ratón sobre una etiqueta en la *tag cloud*, 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 normalmente 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 frecuencia de estos conceptos, la actualidad e idealmente las asociaciones entre ellos. Cada etiqueta se representa con diferente tamaño y color (el tamaño de las etiquetas suele determinar su frecuencia y el color algún otro atributo como su actualidad, por ejemplo puede utilizarse rojo brillante para la más reciente y gris oscuro para la más antigua [Kuo07]), aunque la mayoría de las veces el color se usa de forma arbitraria, simplemente para hacer el diseño más llamativo.

Sinclair y Cardew-Hall [Sin08] realizaron 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.

Hsieh y Cho [Hsi12] presentan un trabajo donde afirman que el etiquetado social, a pesar de ser incontrolado y añadir información superflua, es suficiente para encontrar términos similares a la consulta como sugerencias para la búsqueda. A pesar de presentar la popularidad y la falta de relaciones entre etiquetas, tomando éstas como entidades, como inconvenientes, estos se aminoran a través de estrategias basadas en pesos y probabilidades.

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

- **Aspectos Positivos**

  Una gran ventaja del etiquetado aparece cuando no tenemos claramente definida la información que necesitamos, 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.

  Un aspecto positivo de los sistemas basados en *tagging* es que pueden beneficiarse de la personalización para mejorar la consistencia de las etiquetas usadas en la comunidad y la efectividad de una consulta. La personalización puede llevarse a cabo combinando las preferencias del usuario con la opinión general de los demás usuarios dentro del entorno colaborativo [Wan10]. Esta personalización incluye tareas comunes como la recomendación.

  La recomendación reduce el tiempo consumido por los usuarios. Las etiquetas recomendadas pueden ser filtradas por los usuarios o por las fuentes de información. Si volvemos a la definición de Marinho en al. [Mar12] de folksonomía, tendríamos que para un usuario $u \in U$ y una etiqueta $t \in T$, el

sistema de recomendación predicirá las etiquetas en $T_u \setminus \{t\}$ relacionadas con t.

Golub et al. [Gol09] realizan un estudio en el que demuestran que el etiquetado social combinado con sugerencias del lenguaje colaborativo facilita la producción de etiquetas a usar, produce consistencia e incrementa el número de puntos de acceso en la recuperación de la información.

Zubiaga [Zub12] combina el etiquetado social con la enciclopedia colaborativa "Wikipedia" y muestra como se mejora con ello la navegación y la búsqueda, ya que las etiquetas permiten recuperar los artículos más populares y a su vez hacen de filtro y de pivote en la exploración cuando se buscan etiquetas relacionadas.

Este mismo autor resume las ventajas del etiquetado del siguiente modo:

- Proporciona nuevos términos no existentes
- Proporciona nuevos caminos en la navegación a través de etiquetas
- Ayuda a mejorar la búsqueda
- Permite descubrir los artículos más populares

- **Aspectos Negativos**

  La primera dificultad que encuentran los sistemas basados en *tagging* aparece 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 estos aspectos, la búsqueda a través de una *tag cloud* puede estar muy limitada, aunque puede mejorar si se le aplican técnicas de *clustering* [Beg06].

  Los diseñadores de los sistemas basados en *tagging* no establecen ninguna diferencia entre etiquetar personas y etiquetar objetos. Por ejemplo, la mayoría de las etiquetas que se usan sobre uno mismo, son aquellas que mejoran la percepción de los demás, es decir, no suelen usarse únicamente como mera

presentación, por lo que no se debería usar la misma tecnología en el tratamiento entre unas y otras etiquetas [Rab12].

Por otro lado, la forma que tiene los sistemas basados en etiquetado de ordenar los resultados de la consulta, puede no ser relevante, ya que muchos sistemas ordenan estos resultados basándose en actualidad con que las fuentes de información han sido etiquetadas o el número de usuarios que las 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, queda mucho por descubrir sobre la efectividad y utilidad de las folksonomías.

Evaluación de la Utilidad de la Folksonomía

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

1. La especificidad término-etiqueta → Número de fuentes de información descritas por una etiqueta.

2. La exhaustividad indexación-etiqueta → Número de etiquetas asignadas a una fuente de información.

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 recuperación y alta precisión, por 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.

Al mismo tiempo, la exhaustividad en etiquetado suele ser bastante baja, ya que en el 90 % de los casos los usuarios asignan menos de 5 etiquetas a cada fuente de información.

La baja especificidad y exhaustividad son razonables considerando que el bajo coste cognitivo del etiquetado es uno de los principales factores de su popularidad.

Una vez vistos los antecedentes referentes a los sistemas basados en *tagging* y las folksonomías, pasamos a ver los de la *tag cloud* que, como hemos dicho, es la visualización que usan estos sistemas como interfaz de recuperación visual. Es en el uso de esta herramienta en el que basamos nuestro interés, ya que la utilizaremos para identificar el contenido textual de las bases de datos y como asistente para la consulta y la exploración de la información.

2.1.2 Diferentes Terminologías para la *Tag Cloud*

Aunque la mayoría de los autores y usuarios usan 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 casi todos los autores usan la terminología *tag cloud* de manera indistinta para referirse a ambos conceptos.

La *tag cloud* es la visualización emergente de los sistemas basados en *tagging*. *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 en los sistemas que lo permiten. 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 [Vié08]).

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 [1], Wordle [2] , Neoformix [3], TagCrowd [4] o Tag Cloud Generator [5].

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*. Generalmente la terminología *text cloud* se utiliza para referirse a una visualización de la frecuencia de las palabras que conforman un texto específico, presentada en forma de lista ponderada. Esta técnica es conocida por su uso para analizar las palabras más usadas 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.

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*. También se hablará de la *data cloud* tal como la definieron Koutrika et al. [Kou09a] y de los aspectos clásicos de la *tag cloud* en el sentido estricto del término.

[1] http://manyeyes.alphaworks.ibm.com/manyeyes/
[2] http://www.wordle.net/
[3] http://neoformix.com/
[4] http://tagcrowd.com/
[5] http://www.tagcloud-generator.com/

2.1.3 Revisión Bibliográfica

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 visualizadas 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 [For01] 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.

En 2002 *Flickr* [1], 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 [Vié08].

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 los nodos simbolizan las etiquetas y los ejes, las relaciones de similaridad entre nodos.

[1] http://www.flickr.com/

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.

Simon et al. [Sim10] en 2010 presentaron un sistema basado en la Web para anotar viejos mapas digitalizados usando metadatos bibliográficos y referencias geográficas asociadas al mapa. Este sistema permitía a su vez que los usuarios crearan enlaces a fuentes de información relacionadas en otras bases de datos, con lo que era una herramienta colaborativa y a la vez automática.

En 2011, continuando la misma idea, Hahmann y Burghardt [Hah11] usan una técnica que se sirve de *tag clouds* sobre mapas cartográficos y que denominan "Maple". La idea de esta técnica es representar sobre las distintas secciones del mapa, contenido verbal adicional, es decir, palabras extraídas de la semántica contenida en las características de esa sección: restaurantes, rutas, servicios, etc.

En este mismo año, Kim et al. [Kim11] construyen un grafo similar al de Shaw [Sha05] mediante una técnica que denominan "WordBridge", en la que, tanto los nodos como los enlaces entre estos están formados por *tag clouds*. Usando esta visualización se puede deducir no sólo las relaciones que existen entre entidades y sus conexiones, si no también la naturaleza de estas relaciones a través del uso de palabras clave representativas en nodos y ejes.

En la actualidad, numerosos sitios web como *Delicious* [1], presentan un sistema de etiquetado caracterizado porque permite que cualquier usuario pueda etiquetar cualquier fuente de información web de forma ciega, es decir, sin 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 visualización para términos extraídos del texto en lugar de para etiquetas asignadas libremente, mediante un mismo criterio de selección: la frecuencia.

[1] http://delicious.com/

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 [Vié08] criticaron la *tag cloud* cuando se usa con fines analíticos, proponiendo la *word cloud* como herramienta de análisis alternativa.

Estos mismos autores en 2009 [Vié09] , presentan una herramienta de visualización de texto en forma de nube, donde los términos se extraen directamente del texto, es decir, presentan una herramienta para la generación de *word clouds*. Esta herramienta, "Wordle", actualmente es muy popular y usada por gran número de usuarios.

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

En 2010, Cui et al. [Cui10] exponen un método para representar *word clouds* de forma dinámica. Lo que hacen es visualizar un conjunto de *word clouds* en diferentes instantes de tiempo, resaltando los cambios en el contenido para apreciar como evoluciona el vocabulario a través del tiempo.

En 2012, algunos autores como Hsieh y Cho [Hsi12] y Zubiaga [Zub12] experimentan extrayendo una *tag cloud* de una base de datos con los artículos de la enciclopedia colaborativa "Wikipedia", demostrando que el uso de la *tag cloud* mejora la navegación y la búsqueda en "Wikipedia".

Actualmente son muchas 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 *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.

A partir de 2010, empezamos a encontrar un mayor número de autores que emplean etiquetas multitérmino en diversos trabajos. Es el caso de Kaptein y Marx [Kap10], que resumen el contenido de discursos políticos a través de *word clouds*. Estos autores facilitan que en la *word cloud* aparezcan etiquetas "bitérmino", alegando que la interpretación del contenido es más sencilla a partir de bitérminos que usando únicamente palabras aisladas, ya que los bitérminos matizan el contexto. Otro caso es el de Choudhury y Breslin [Cho10], que describen un sistema para anotar y recuperar vídeos a través de etiquetas semánticas. Esta semántica la proporcionan también a través de etiquetas bitérmino, explicando que son más precisas y descriptivas que las de un sólo término.

Otros autores, como Durao et al. [Dur12] apuestan por agrupar las palabras relacionadas en la misma línea dentro de la *tag cloud* mediante mecanismos de *clustering*, lo que sería innecesario si se permite el uso de multitérminos.

Revisión bibliográfica sobre *clustering* y herencia en la *tag cloud*

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*. Esto mismo se estudia en una tesis presentada en la Universidad de Taiwan [Yu06].

Ese mismo año, Hsieh et al. [Hsi06] presentaron el concepto de herencia entre etiquetas, que permitía obtener mayor información para la recomendación y la búsqueda. Esta herencia, es introducida a su vez por otros autores 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 más adelante).

Papadopoulos et al. [Pap10] estudiaron alternativas para la aplicación de estas técnicas, como es la construcción de un esquema gráfico basado en *clustering* para la identificación de etiquetas relacionadas.

Trattner et al. [Tra11] abordan algunos problemas sobre la navegabilidad de las *tag clouds* mediante un modelo de generación basado en herencia que se sirve de contenido web estructurado.

Y podemos ver un enfoque para la creación de *clusters* interpretables y reconfigurables en Balachandran et al [Bal12], mientras Durao et al. [Dur12] proponen un algoritmo espectral de *clustering*.

Hoy en día, numerosos sitios emplean técnicas de *clustering* y/o herencia en sus páginas, por ejemplo *Flirckr* utiliza *clusters* que proporcionan etiquetas populares agrupadas junto a sus etiquetas relacionadas, *HubLog* [1] permite a los usuarios la exploración entre las etiquetas relacionadas de *Delicious*, *Netr.it* [2] construye una red, extensible de forma manual, de las co-ocurrencias de las etiquetas personales de cada usuario de *Flirckr* y *Semantic Cloud* [3] 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 debido a la alta dimensionalidad de los datos y el gran número de etiquetas a ser agrupadas. Este problema empeora cuando los *clusters* no contienen descriptores, ya que son difícilmente interpretables por el usuario. Es por esto que autores como Morik et al. [Mor12] continúan estudiando mecanismos de *clustering* que mejoren los actuales y proponen una técnica de *clustering* consistente en la optimización de una función multi-objetivo que evalúa conjuntamente las métricas de cobertura, solapamiento y simplicidad (ver distintas métricas en Sección 2.1.4).

[1] http://hublog.hubmed.org/
[2] http://www.netr.it/
[3] http://semanticcloud.rieskamp.info/

2.1.4 Características de la *Tag Cloud*

Funciones

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.

Además las *tag clouds* sirven para asistir a los usuarios, que en el momento de usar el sistema de búsqueda, no tengan definidas claramente sus necesidades, facilitando la expresión de la necesidad en aquellas situaciones en las que el usuario no sea capaz de formular la consulta, pero sí reconocerla entre el conjunto de posibles consultas representadas por cada una de las etiquetas. Así mismo, en las *tag clouds* que se generan a partir de una consulta, se le puede ofrecer al usuario una guía visual que le permita el refinamiento de dicha consulta [HM10].

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 *Delicious*, 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 con

mayor tamaño o más importantes. Además permite que se representen simultáneamente al menos tres dimensiones: las palabras, su importancia relativa y cualquier tipo de orden, como por ejemplo el orden alfabético. Aunque con el uso del color y otras estrategias, podrían representarse muchas más.

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.

Por otro lado, la tarea de la búsqueda y la exploración a través de bases de datos, es una tarea realizada únicamente por expertos que conocen el lenguaje de consulta y están familiarizados con el esquema de la base de datos. La *tag cloud* puede facilitar la búsqueda y la exploración en una base de datos a través de su visualización, por lo que puede ser usada fácilmente por usuarios sin experiencia [Leo11].

- **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 una fuente de información 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 visualización). En un 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, entendida como el grado en que las etiquetas presentes se solapan semánticamente entre sí, ya que los términos usados con mayor frecuencia son los que menor valor de discriminación poseen [Sal75]. En otras palabras, las *tag clouds* tienden a estar dominadas por unas pocas temáticas diferentes.

- 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 un 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.

Helic et al. [Hel10, Hel11] resaltan otros inconvenientes con respecto a la navegabilidad a través de *tag clouds* provenientes de folksonomías:

- La utilidad de las *tag clouds* para la navegación es sensible a la adopción del lenguaje de los sistemas basados en *tagging*.
- Hay más etiquetas con altas frecuencias que fuentes de información con altas frecuencias, con lo que la distribución de frecuencias domina la distribución de fuentes de información y la navegación está centralizada.
- Limitar la alta frecuencia de las etiquetas con tamaños "fuera de rango", convierte el sistema en vulnerable a la fragmentación, lo que destruye la navegación a través de *tag clouds*.

Trattner et al. [Tra11] dan una solución a estos inconvenientes a través de un modelo de navegación basado en herencia para la generación de las *tag clouds*. A pesar de disminuir considerablemente con este modelo las búsquedas fallidas a través de *tag clouds*, éste porcentaje continua siendo elevado, un 27 % de las búsquedas no llegan a obtener el resultado esperado en el experimento realizado por estos investigadores.

Evaluación sobre la Utilidad de la *Tag Cloud*

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 ∈ T una etiqueta de una *tag cloud* T:

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

donde

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

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.

Venetis et al. [Ven11] definen un conjunto de métricas para capturar las propiedades estructurales de la *tag cloud*, con el fin de evaluar la utilidad de ésta cuando se pretende resumir con ella los resultados de la consulta y ayudar a la navegación a través de estos resultados. Parten definiendo la *tag cloud* del siguiente modo:

Sea C un conjunto de objetos y T un conjunto de etiquetas. Sea $q \in T$ un conjunto consulta. Se define una *tag cloud* S como un subconjunto de etiquetas en T para una particular consulta q.

Sea $C_q \subseteq C$ el conjunto de resultados obtenidos con la consulta, $C_t \subseteq C$ el conjunto de objetos recuperados con cada etiqueta t en S y A_q el conjunto asociado a una etiqueta bajo la consulta q, $A_q \subseteq C_q$. Se define el siguiente conjunto de métricas para S:

1. **Cobertura.** Nos da la fracción de C_q cubierta por S:

$$cov(S) = \frac{|\bigcup_{t \in S} A_q(t)|_{s,q}}{|C_q|_{s,q}} \tag{2.3}$$

   Esta métrica toma valores entre 0 y 1. Si está cerca de 0, entonces S cubre pocos objetos en C_q, si por el contrario está cerca de 1, cubrirá muchos objetos en C_q.

2. **Solapamiento.** Diferentes etiquetas en S pueden estar asociadas al mismo objeto en C_q. Con esta métrica se captura la extensión de tales redundancias:

$$over(S) = avg_{t_1,t_2 \in S, t_1 \neq t_2} \left\{ \frac{|A_q(t_1) \cap A_q(t_2)|_{1-s,q}}{min_{i \in \{1,2\}} \{|A_q(t_i)|_{1-s,q}\}} \right\} \tag{2.4}$$

   Esta métrica también toma valores en el intervalo [0,1]. Si está cerca de 0, la intersección de los conjuntos de asociación es muy pequeña, lo que revela resultados distintos resultados distintos de la consulta con las etiquetas t_1 y t_2, es decir, existirá poco solapamiento. Si por el contrario, el valor de está métrica está cerca de 1, el solapamiento será alto y las etiquetas no son muy distintas una de otra.

3. **Cohesividad.** Mide la cercanía de los objetos en cada conjunto de asociación A_q de S en función de las relaciones entre estos objetos.

4. **Relevancia.** Se define como el solapamiento entre C_q y C_t. Se calcula como la fracción de resultados en C_t que también están en C_q.

5. **Popularidad.** Una etiqueta en S es popular en C_q si está asociada con numerosos objetos en C_q.

6. **Independencia.** Dos etiquetas en S serán más independientes cuánto menos similares sean entre sí los objetos que recuperan. La métrica en este caso es similar a la de cohesividad, pero esta última se calcula para cada par de conjuntos de asociación diferentes de las etiquetas en S.

7. **Balance.** Una *tag cloud* S será balanceada cuando las etiquetas en S representen un número similar de objetos en C_q. Para calcular el balance se hace uso de la siguiente ecuación:

$$bal(S) = \frac{\text{mín}_{t_i \in S}\{|A_q(t_i)|\}}{\text{máx}_{t_j \in S}\{|A_q(t_j)|\}} \tag{2.5}$$

   .

Estas métricas pueden calcularse en función de las preferencias de cada uno para evaluar la efectividad de una *tag cloud* específica. Aquí se han especificado las fórmulas de las métricas más relevantes, que serán las que usemos en el Capítulo 6. El resto de fórmulas pueden verse en [Ven11]. Durao et al. [Dur12] usan estas métricas destacando como más importantes la de cobertura, de solapamiento y la de relevancia, que las formulan como sigue a continuación:

- **Cobertura:** $cov(S) = \frac{|S|}{|C_q|}$ donde $0 < cov(s) < 1$. Cuanto más próximo esté este valor a 1, mayor será la cobertura.
- **Solapamiento:** $over(S) = avg_{t_i \neq t_j} \frac{|t_i \cap t_j|}{|C_q|}$ donde $t_i, t_j \in S$ donde $0 < over(S) < 1$. Cuanto más próximo esté este valor a 1, mayor será el solapamiento, es decir, la redundancia de las dos etiquetas.
- **Relevancia:** $rel(S) = avg_{t \in S} \frac{|C_t \cup C_q|}{|C_t|}$ donde $0 < rel(S) < 1$. Cuanto más próximo esté este valor a 1, mayor será la relevancia de S para la consulta q.

Skoutas y Alrifai [Sko11] proponen además la métrica **"Selectividad"** que mide el número de objetos filtrados en una *tag cloud* cuando se selecciona una etiqueta. Cuanto mayor sea el número de objetos filtrados sin relación con la etiqueta seleccionada, mayor será el valor de esta métrica.

Morik et al. [Mor12] ponen de manifiesto que algunas de estas métricas están en conflicto las unas con las otras, como es el caso de la cobertura y el solapamiento, ya que al aumentar la cobertura, también aumenta el solapamiento, que nos interesa que sea mínimo. Esto dificulta las tareas de *clustering* que realizan estos autores, por lo que proponen la optimización de una función multiobjetivo a la hora de configurar los *clusters* que proporcione un valor óptimo simultáneo para todas las métricas consideradas.

El Diseño

Existen diversos diseños para la *tag cloud*, diseños donde las etiquetas están ordenadas alfabéticamente, por su frecuencia o según un determinado algoritmo, diseños donde están ordenadas semánticamente o el usuario puede especificar sus preferencias de *clustering*, diseños espaciales donde las etiquetas pueden representarse en líneas secuenciales, en forma de cubo, en forma de círculo, etc.

En un experimento realizado por Halvey y Keane [Hal07] en el que 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.

Además, como afirman Morville y Rosenfeld [Mor06], el criterio de ordenación alfabética sólo es útil cuando el usuario se encuentra realizando una búsqueda por elementos sintácticos conocidos.

Otros investigadores, como Hassan-Montero y Herrero-Solana [HM06], realizan diseños alternativos, proponiendo la aplicación de técnicas de *clustering* a la *tag cloud*. Aunque en el estudio posterior, en el que realizan una evaluación mediante la técnica "eye-tracking" [HM10], estos mismos autores nos dicen que la agrupación semántica no supone una mejora en términos de eficiencia en las tareas de localización visual de las etiquetas. Para calcular la similaridad entre etiquetas utilizan la co-ocurrencia relativa entre éstas.

En un diseño similar, Fujimura et al. [Fuj08] y Berlocher et al. [Ber08] utilizan la similaridad del coseno como medida de similaridad entre etiquetas, pero ninguno de estos autores realiza un experimento para evaluar el efecto de aplicar estas técnicas de diseño sobre los usuarios.

Sin embargo, es fácil encontrar diversas evaluaciones de distintos diseños, incluyendo estos últimos, considerados diseños semánticos. Por ejemplo, Schrammel et al. [Sch09] realizaron un experimento para evaluar los diseños alfabético, aleatorio y semántico, resultando el diseño alfabético el mejor valorado por los usuarios, mientras que Lohmann et al. [Loh09] comparan los diseños alfabético, semántico, circular (con las etiquetas más populares en el centro del círculo) y secuencial alfabético sin ponderar, encontrando que el mejor diseño depende de las intenciones que tenga el usuario y el diseñador, por ejemplo, para encontrar una etiqueta específica, resultó ser mejor el diseño alfabético, para encontrar las etiquetas más populares, el circular y para encontrar las etiquetas relacionadas con algún tema determinado, el semántico.

Visto el gran número de evaluaciones realizadas sobre el diseño de las *tag cloud* que nos ofrecen conclusiones distintas, lo más razonable es pensar que, efectivamente, un diseño será mejor que otro dependiendo de las intenciones del usuario o de la tarea que se le pida realizar en el experimento.

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*.

Una 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. [Vié09] proponen un diseño," Wordle", 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). Posteriormente, Koh et al. [Koh10] lo mejoran, creando lo que denominan "ManyWordle". La diferencia principal con Wordle es que ManyWordle permite que los usuarios manipulen el diseño indicando la tipografía, la posición, la orientación y el color de las etiquetas o la *tag cloud*, así como la composición de esta última (ver *tag cloud* 6 en Figura 2.1).

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

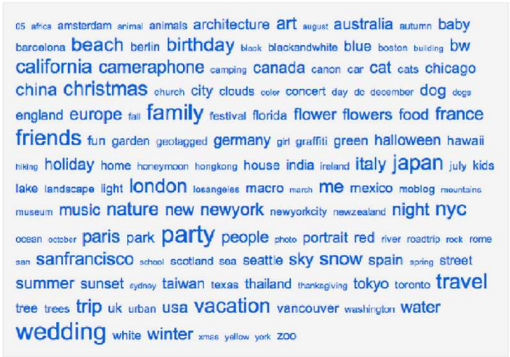
Con respecto a la localización de las etiquetas dentro del diseño de la *tag cloud*, un experimento realizado por Rivadeneira et al. [Riv07] muestra que las etiquetas representadas en el primer cuadrante de la *tag cloud* se recuerdan más por el usuario que las etiquetas representadas en el resto de 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], sin embargo, afirman que son las zonas centrales las de mayor influencia visual.

2. ANTECEDENTES

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*

Hassan-Montero et al. [HM10] realizan un estudio mediante *eye-tracking* o seguimiento visual, donde destacan como punto influyente en la exploración visual las etiquetas de mayor tamaño y, dependiendo de la parte de la visualización en que estén situadas estas etiquetas, se obtienen diferentes patrones de escaneo visual, lo que explicaría las diferencias entre los dos trabajos anteriores.

2.1.5 Las Etiquetas en la *Tag Cloud*

Métodos para el Establecimiento de las Etiquetas

Distinguiremos entre distintos métodos dependiendo de si las etiquetas se extraen del texto o se asignan manualmente por los usuarios.

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

  Algunos autores, han propuesto técnicas de SVD (Singular Value Decomposition) para la selección de las mejores candidatas [Pro08] y de detección automática de variantes sintácticas mediante funciones de similitud [Ast09].

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

  Knautz et al. [Kna10] 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:

$$WDF \cdot ITF = \left[\frac{\log(frec(t,b)) + 1}{\log L}\right] \cdot \left[\log\left(\frac{M}{m}\right) + 1)\right] \tag{2.6}$$

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

Skoutas y Alrifai [Sko11] realizaron un estudio donde compararon las distintas estrategias de selección de etiquetas representadas en la *tag cloud*, resumiéndolas en las siguientes:

1. **Estrategias basadas en la frecuencia.**
   a) **Estrategias basadas en una puntuación proporcional a la frecuencia.**
   b) **Estrategias basadas en una puntuación del tipo $WDF \cdot ITF$.** Según esta estrategia, se les da una mayor puntuación a las etiquetas que están asignadas a una única fuente de información.
   c) **Estrategias basadas en una puntuación obtenida del grafo.** Las etiquetas con mayor puntuación son aquellas que ocurren conjuntamente con otras etiquetas, ya que son las que proporcionan información de contexto.
2. **Estrategias basadas en la diversidad.** En los enfoques basados exclusivamente en la frecuencia, la utilidad de cada etiqueta se calcula de forma independiente, sin tener en cuenta el resto de etiquetas en la *tag cloud*, lo que puede conllevar que algunos objetos estén sobrerepresentados mientras que otros no tengan representación. Surgen por esto las siguientes estrategias:
   a) **Estrategias basadas en la diversidad.** El objetivo de esta estrategia es seleccionar las etiquetas con menor similaridad. Una etiqueta tendrá una alta utilidad si no hay otras en la nube con las que tenga alta similaridad.
   b) **Estrategias basadas en la novedad.** Con esta estrategia se da mayor puntuación a las etiquetas asociadas a las fuentes de información a las que no se puede acceder mediante otras etiquetas.
3. **Estrategias basadas en un *ranking* de agregación.** Estas estrategias consideran otro aspecto a tener en cuenta en la selección de etiquetas, el orden en que éstas se asignan a las fuentes de información, asumiendo que existe correlación entre ese orden y la relevancia de la etiqueta para la fuente de información marcada.

Con posterioridad a esta propuesta de estrategias para la selección de etiquetas, Skoutas y Alrifai [Sko11] realizaron un experimento con un conjunto de datos extraídos de *Flickr* para probar qué estrategia funcionaba mejor, para ello, se estudiaron las métricas: cobertura, solapamiento y diversidad. Así mismo, también se evaluó el coste de navegación, entendiendo por éste la suma de todas las acciones que realiza el usuario hasta llegar a la fuente de información requerida.

Los resultados obtenidos fueron que la estrategia basada únicamente en la frecuencia absoluta/relativa, funcionaba bastante bien como método de selección de etiquetas, mientras que la estrategia basada en un cálculo de la frecuencia del tipo $WDF \cdot ITF$ daba unos resultados bajos para todas las métricas, excepto la de solapamiento. La estrategia basada en el cálculo de la frecuencia según el grafo tampoco mejoraba el valor de las métricas obtenidas para la frecuencia absoluta/relativa teniendo además un mayor coste computacional.

Sin embargo, las estrategias basadas en la diversidad, daban los mejores resultados para todas las métricas, siendo la mejor la estrategia basada en la novedad, especialmente por la métrica de cobertura, el bajo coste computacional, el bajo coste de navegación y por ofrecer las etiquetas más adecuadas para la recomendación.

Las estrategias basadas en el *ranking* de agregación, funcionaron mejor que las basadas en frecuencias, pero peor que las basadas en la diversidad.

- **Cuando las etiquetas se extraen del texto**, generalmente habrá que realizar primero una limpieza para eliminar las palabras vacías de significado y los signos de puntuación.

  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 considerando los siguientes pasos:

1. Eliminar, para cada resumen, las palabras que no aportan información: “la”, “de”, “con”, “y”, etc., así como las puntuaciones y símbolos.
2. Eliminar los sufijos aplicando el algoritmo de stemming de Porter y usar las raíces de las palabras como etiquetas.

Después de generar la lista de etiquetas que describe la respuesta de la consulta, calcularon la frecuencia relativa y la actualidad de cada etiqueta. Sólo tuvieron en cuenta las etiquetas que tenían una frecuencia de al menos el 10 %. Calcularon la actualidad como la media de la fecha de publicación de los resúmenes en los que aparece la etiqueta en cuestión.

Al pulsar con el ratón sobre una etiqueta, aparecía una lista de palabras que compartía el mismo prefijo con diferentes sufijos y un enlace al conjunto de resúmenes que contienen la etiqueta.

En otros casos, se usan alternativas a la frecuencia de las etiquetas para considerar éstas como aptas para estar en la *tag cloud*, como por ejemplo algún tipo de probabilidad como la calculada por Kaptein y Marx [Kap10].

Establecimiento del tamaño de las etiquetas

Sinclair y Cardew-Hall [Sin08] utilizaron esta forma para determinar el tamaño de las etiquetas en función de su frecuencia:

$$TagSize = 1 + C \cdot \frac{\log\left(f_i - f_{min} + 1\right)}{\log\left(f_{max} - f_{min} + 1\right)} \tag{2.7}$$

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.

Evaluación de la utilidad de las etiquetas

Sin conocer la motivación de los usuarios a la hora de etiquetar una fuente de información, es difícil predecir la utilidad de esa etiqueta [Str10], ya que la utilidad será distinta si se pretende categorizar una fuente de información (para encontrarla luego) o describirla (útil en la extracción de conocimiento de las folksonomías), por lo que es conveniente distinguir previamente entre categorización y descripción.

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 mejores de estas etiquetas para caracterizar la colección de fuentes de información etiquetadas, puede determinarse la utilidad de una etiqueta como:

- La capacidad de representar cada fuente de información comparada con otras etiquetas asignadas a la misma fuente de información.
- El volumen de fuentes de información cubiertas en comparación con otras etiquetas.

Si consideramos 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 que d_{ij} representa la frecuencia con que se ha usado cada etiqueta T_j para describir la fuente de información D_i, Hassan-Montero y Herrero-Solana [HM06] definen la utilidad $F(T_j)$ de la etiqueta T_j como parte de la *tag cloud* como:

$$F(T_j) = \sum_{i=1}^{i=n} \left[\frac{\log d(ij)}{m^2} \right] \tag{2.8}$$

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

2.1.6 *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 nos posibilitan penetrar en el contenido del texto al permitir que los términos relacionados puedan ir juntos. Ejemplos de términos relacionados son: inteligencia artificial, redes sociales, sistemas operativos, etc.

Pensemos por ejemplo en 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.

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

Un estorbo común en el etiquetado social es que en las etiquetas multitérmino, los términos aparecen conectados mediante guiones, puntos o escritos sin espacios, como si fueran una sola palabra. Un ejemplo de creación de etiquetas multitérmino en las *tag clouds* usando guiones para separar los términos es el que tenemos en Ammari et al. [Amm12]. 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 de esta forma, no se favorece la comprensión del contenido mediante el simple 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. Por ejemplo, en Choudhury y Breslin [Cho10] podemos ver una heurística mediante la que se intenta separar los multitérminos que se escriben sin espacios. También la herramienta de Google *did you mean* [1] posee un excelente mecanismo para separar palabras de dos términos que se escriben sin espacio, pero falla cuando estas palabras compuestas son de más de dos términos.

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

[1] http://code.google.com/intl/en/apis/ajaxsearch/

- 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 [Don07].

Panunzi et al. [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 (ver fórmulas en Frantzi et al. [Fra00]).

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, herramienta que proporciona a su vez 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" [1], que es una herramienta gratuita que podemos encontrar en la Web y que fue creada por los mismos autores mencionados anteriormente, Frantzi et al. [Fra00]. TerMine incorpora métodos lingüísticos, estadísticos e información de contexto para la extracción de palabras clave multitérmino. Watters [Wat08] demostró que los resultados empleando palabras clave multitérmino mejoraban los resultados en que se empleaban palabras clave monotérmino, ya que proporcionaban contexto y orientación al usuario incrementando el nivel de significación proporcionado por los métodos de recuperación de información.

Esto mismo volvió a demostrarse en 2010 por Kaptein y Marx [Kap10], sólo que los multitérminos utilizados por estos autores sólo permitían un máximo de dos términos. En la *tag cloud* representaban tanto monotérminos como bitérminos, pero omitiendo aquellos monotérminos que ya estaban incluidos en alguno de los bitérminos. En el experimento que realizaron se prefirió este modelo al compuesto únicamente por monotérminos o bitérminos.

[1] http://www.nactem.ac.uk/software/termine/

2.1.7 Las *Tag Cloud* en las Bases de Datos (*Data Cloud*)

Dada la flexibilidad de las *tag clouds* en términos de representación de información junto con la simplicidad de la navegación a través de éstas, es natural que los investigadores de bases de datos consideren explotar el concepto de *tag cloud* para afrontar el remanente problema de la usabilidad de las bases de datos.

El uso del lenguaje de consulta requiere que el usuario conozca, no sólo este lenguaje, sino también el esquema de la base de datos. Para permitir que los usuarios puedan visualizar los datos de forma natural, se hace necesaria una representación del contenido a alto nivel, como un esquema visual para la exploración y la consulta [Leo11].

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* estos autores resumen los resultados obtenidos con la búsqueda por palabras clave sobre los datos estructurados y guían 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, para representar el contenido completo de la base de datos, 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 independientemente del campo de la base de datos en el que se encuentre la palabra.

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.

Definición y Generación de *Data Clouds*

Koutrika et al. [Kou09a] 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 $B_1...B_n$. 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 debería tener 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* de Koutrika et al.[Kou09a].

Para la generación de la *data cloud* que proponen estos autores, primero 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 esto 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 de los términos en los resultados de la consulta, considerando la popularidad como el número de veces que k co-ocurre con todos los términos de q. Este es el enfoque típico en las *tag clouds*.

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:

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

   En esta fórmula 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. Por esto, con este enfoque, 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.

La *data cloud* de Koutrika et al. [Kou09a, Kou09b] depende de la construcción de entidades en la base de datos. Un enfoque más general, en el que no se necesita

información específica de dominio, es el dado por Leone et al. [Leo11] para la creación *tag clouds* sincronizadas, para explorar conjuntamente el esquema y los datos.

Las etiquetas en la *data cloud* de Leone et al. [Leo11] representan el valor de los atributos. Pulsando en una etiqueta, se seleccionan los elementos con ese valor de atributo. En el caso de las bases de datos orientadas a objetos, el resultado sería una colección de objetos, mientras que en el caso de las bases de datos relacionales, el resultado sería una colección de tuplas. La fuente de datos se define como un conjunto de colecciones de datos, estas colecciones pueden ser clases o conjuntos de objetos en las bases de datos orientadas a objetos o relaciones en las bases de datos relacionales.

Leone et al. [Leo11] permiten que el usuario comience explorando la base de datos, bien introduciendo una expresión de consulta dependiente de la base de datos o seleccionando una o más etiquetas en la *data cloud*. Esta *data cloud* es una composición de nubes (*sub-clouds*). En la primera nube aparece representado el esquema, esto es, el nombre de los atributos tal como se han definido. Pulsando sobre una etiqueta de esta "nube esquema" se obtiene la *tag cloud* correspondiente a los valores de ese atributo, lo que constituye la segunda componente de la visualización. En la tercera componente se visualiza un segundo atributo, que se refleja sobre la *tag cloud* con distintos colores según el valor de éste.

Sin embargo, los atributos que toman un largo rango de valores, no son muy apropiados para este tipo de visualización y Leone et al. [Leo11] prefieren que se haga la consulta de estos atributos en términos del lenguaje SQL (*Structured Query Language*). Además, en una encuesta que realizaron para comprobar la usabilidad de la *data cloud*, sólo el 50 % de los participantes tuvo en cuenta el color de las etiquetas para resolver las tareas planteadas.

2.1.8 **Procesos de Asociación Semántica a las Etiquetas en las Folksonomías.**

García-Silva et al. [GS12] realizan una revisión de los enfoques más relevantes existentes para asociar semántica a las etiquetas con el propósito de darles significado (enfoques basados en *clustering*, en ontologías o en ambas), identificando

una serie de pasos comunes en todos los enfoques y proporcionando un punto de vista unificado en todos ellos. Por lo general, el proceso de asociación semántica se desarrolla los siguientes pasos:

1. **Selección y limpieza de datos**: Para la selección se usan filtros como frecuencia, características léxicas, lenguaje, etc.

2. **Identificación de contexto**: Por contexto se entiende el conjunto de elementos tenidos en cuenta para averiguar el significado de las etiquetas. Esta noción de contexto se aplica en las etiquetas de dos formas; la primera es examinando las demás etiquetas que se han usado para categorizar la misma fuente de información y la segunda, es examinar las etiquetas que el usuario ha usado conjuntamente para marcar una fuente. Además, en la noción de contexto va incluida la información lingüística. sinónimos y otras variaciones morfológicas.

3. **Desambigüación**. Se lleva a cabo usando recursos semánticos externos como WordNet.

4. **Identificación semántica**: Se realiza mediante el acoplamiento de las etiquetas con entidades semánticas, usando ontologías predefinidas.

Tras revisar los enfoques más importantes y las deficiencias y aportaciones de cada uno de ellos, García-Silva et al. [GS12], ofrecen algunas recomendaciones a llevar a cabo en cada una de las cuatro tareas anteriores del proceso de asociación semántica:

1. Selección y limpieza de datos

   a) Eliminar las *stop-words* comunes como pronombres, artículos, etc. (Cantador et al. [Can08])

   b) Procesar las palabras mal escritas con mecanismos como *Google did you mean* [1] o *Yahoo spelling suggestion service* [2] (Specia y Motta [Spe07])

[1] http://code.google.com/intl/en/apis/ajaxsearch/

[2] http://developer.yahoo.com/search/boss/boss_guide/Spelling_Suggest.html

c) Los acronismos, abreviaciones y nombres propios se pueden tratar con Wikipedia. (Cantador et al. [Can08])

d) Filtrar las palabras sin significado. (Specia y Motta [Spe07], Angeletou et al. [Ang08], Cantador et al. [Can08], Giannakidou et al. [Gia08])

e) Identificar las variaciones morfológicas de las etiquetas e unificarlas en un único término. (Giannakidou et al. [Gia08])

f) Permitir etiquetas de palabras compuestas (Cantador et al.[Can08])

g) Aumentar la cobertura de los enfoques basados en ontologías, generando varias representaciones de las palabras compuestas, es decir, usando varios caracteres para concatenar las palabras (Angeletou et al. [Ang08])

2. Identificación de contexto

   a) Mejorar la contextualización teniendo en cuenta los diferentes niveles de información contenidos en las folksonomías.

   b) Incrementar el contexto de una etiqueta usando las etiquetas que coocurren con ésta al etiquetar la misma fuente de información. (García-Silva et al. [GS09])

   c) Combinar el uso de enfoques estadísticos con el uso de enfoques basados en ontologías.

3. Desambigüación
   Usar WordNet como repositorio inicial y Wikipedia y DBpedia como repositorios complementarios para aumentar la cobertura.

4. Identificación semántica
   Crear herencias a partir de la información de la folksonomía que pueden ser comparadas con herencias ontológicas, por ejemplo como hacen Heymann y García-Molina [Hey06].

Mecanismos de *Clustering*

Begelman et al. [Beg06] presentaron un algoritmo de *clustering* para obtener una medida de similaridad entre etiquetas, para posteriormente agruparlas en la *tag cloud*.

Este algoritmo se basa en el número de co-ocurrencias de cualquier par de etiquetas. Para encontrar aquellas fuertemente relacionadas 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.

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. [Beg06] 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.

Durao et al. [Dur12] añaden a la co-ocurrencia entre etiquetas en la forma que la calculan Begelman et al. [Beg06] un cálculo de similaridad. Una vez que se establece el punto de corte a partir del cual la co-ocurrencia entre dos etiquetas se considera significativa y las etiquetas relacionadas se identifican, calculan entre estas etiquetas la medida de similaridad del coseno y posteriormente, se construye el grafo $G(V, E, W)$. En este caso, W es una matriz de pesos representando la medida de similaridad de cada co-ocurrencia. Para construir los *clusters* a partir del grafo G utilizan el algoritmo espectral de *clustering*, que consiste en dividir el grafo de forma que los pesos "*intra-clusters*" sean altos (similaridades) y los pesos "*inter-clusters*" sean bajos.

Otros muchos autores han estudiado medidas de similaridad. Hassan-Montero y Herrero-Solana [HM06] definen la co-ocurrencia relativa entre dos etiquetas como:

$$RC(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{2.16}$$

donde A y B son los conjuntos de las fuentes de información descritos por las dos etiquetas y $RC(A, B)$ se conoce como el "Coeficiente de Jaccard". Y esta es

la medida que utilizan para estimar la similaridad entre etiquetas.

Herencia entre Etiquetas

Hsieh et al. [Hsi06] propusieron un sistema que incorpora la herencia entre las etiquetas. Con este método consiguieron una gran recuperación de información debido a la incorporación del concepto "distancia", pero perdieron en precisión.

Trattner et al. [Tra11] construyen un modelo de navegación a través de *tag clouds* basado en herencia. En este modelo, al generar la *tag cloud* de una categoría padre, las etiquetas de todas las categorías hijo se añaden de forma recursiva.

Mediante un experimento que realizan para comprobar la eficiencia de este sistema basado en herencia, obtienen que las búsquedas fallidas a través de *tag clouds* han disminuido de un 89 % a un 27 %.

Ontologías derivadas de folksonomías

La necesidad de semántica en una folksonomía surge de la idea de conceptualización. Sin estructura en los datos y sin control semántico, se hace precisa una forma de acceder y exponer la información representada para permitir la interoperación a través de fuentes heterogéneas. Uno de los mecanismos para conseguir esto es a través de una ontología que provea la semántica apropiada.

Otra motivación para el uso de las ontologías es poder enlazar conjuntos de datos diferentes marcados con etiquetas similares.

Diseñar una ontología de etiquetas es un intento para proveer una conceptualización común del significado de la información contenida en las folksonomías, proporcionando una forma estandarizada para recolectar, interpretar y usar los datos etiquetados. Según esto, Gruber [Gru07] propone una ontología para las folksonomías que puede actuar como una infraestructura para construir un ecosistema de fuentes de información etiquetadas, servicios, agentes y herramientas.

Pero las ontologías sobre etiquetas existentes, están limitadas a expresiones varias de las folksonomías. Para rehusar las etiquetas en distintas plataformas, se requieren otras tecnologías más eficientes, como la Web Semántica.

Kim et al. [Kim10] abordan este problema mediante la ontología "SCOT".

2.1.9 Otros Aspectos de la *Tag Cloud*

***Data Clouds* a través de Cubos OLAP (*On-Line Analytical Processing*)**

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.

El cubo de datos soporta las siguientes operaciones:

- *Slice* (Cortar en rodajas) → Cuando solamente se está interesado en algunos atributos.
- *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. [Aou09] obtienen las etiquetas o *tags* como términos composición de tres 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.

Dado 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 una dimensión menor (como sería agregar los valores de dos 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 estos autores 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 en

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.

Interfaces Visuales Alternativas

Como se ha visto, la *tag cloud* puede servir como un buen complemento en diferentes actividades del proceso de búsqueda de información, pero a raíz de los aspectos negativos descritos en la Sección 2.1.4, o simplemente fruto de la experimentación, han surgido multitud de interfaces visuales alternativas, la mayoría de las cuales se han ido viendo, como el caso de una *tag cloud* en la que las etiquetas aparezcan agrupadas en *clusters*, los grafos de etiquetas o las *tag clouds* representadas sobre mapas cartográficos.

Una interfaz visual muy utilizada son los diagramas arbóreos. Para aplicar este tipo de visualización es preciso tener las etiquetas ordenadas según una relación de herencia. En un grafo con estructura de árbol, los nodos representan conceptos (etiquetas) y las líneas de unión entre nodos las relaciones de herencia entre los conceptos. En algunos casos, los ejes pueden representar otro tipo de relaciones, como en [Sha05].

Cuando la dimensionalidad es muy alta, no resulta sencillo buscar una información concreta en un grafo arbóreo. Una alternativa a este problema es la propuesta por Di Caro et al. [DC08], que consiste en visualizar la estructura arbórea a través de una lista.

Los diagramas de Venn son otra forma para representar las etiquetas agrupadas en *clusters*. Cada *cluster* puede verse como un conjunto de etiquetas, de forma que los conjuntos pueden superponerse los unos a los otros, quedando en la zona superpuesta las etiquetas que comparten, como si se tratase de un diagrama de Venn [Che09].

Las metáforas topográficas son utilizadas por autores como Fujimura et al. [Fuj08] y son una solución para representar *tag clouds* a gran escala, es decir, con un gran número de etiquetas (cinco mil o más). Se trata de superponer diferentes *tag clouds* sobre una imagen topográfica.

Existen otras muchas visualizaciones, como por ejemplo la de Bielenberg y Zacher [Bie05] que, como ya se dijo en la Sección 2.1.3, proponen un diseño circular en el que el tamaño de fuente de las etiquetas y su distancia al centro representa su

importancia; las más cercanas al centro serán las más importantes y las que tengan mayor tamaño o la de Kerr [Ker06], que presenta un diseño con forma de órbita en el que cada etiqueta primaria se sitúa en el centro de una órbita y sus etiquetas relacionadas en las bandas circundantes.

2.2 Antecedentes de la Estructura-AP

La estructura-AP nace de la necesidad de procesar automáticamente, de forma masiva, los textos cortos en una colección de documentos o los atributos textuales de una base de datos con falta de estructura. Se basa en la transformación del atributo textual en una forma intermedia, que permite su representación de manera más estructurada. Dicha estructura está basada en el concepto de *itemset* frecuente y sus propiedades [Mar06]. Se genera de forma constructiva, extrayendo primero las palabras frecuentes y combinándolas para originar los *itemsets* candidatos de nivel dos, compuestos por dos palabras; los que resulten frecuentes entre estos *itemsets* de nivel dos, volverán a combinarse entre sí para obtener los *itemsets* candidatos de nivel tres y así sucesivamente hasta obtener los *itemsets* frecuentes maximales o de nivel máximo.

Se utiliza esta forma de representación debido a que se asume como hipótesis que los *itemsets* frecuentes conservan la semántica subyacente en los atributos textuales, ya que permiten que los términos más relevantes puedan mantenerse agrupados. Una razón adicional para utilizar la estructura de *itemsets* frecuentes [MB08] como base para nuestro modelo, es que los algoritmos que se encargan de la obtención de estos *itemsets* son bien conocidos, han sido implementados en distintas variantes y están altamente optimizados.

La estructura-AP es un modelo matemático 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 texto. 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 [MF08].

2.2.1 Concepto de Estructura-AP y Operaciones Asociadas

En esta sección se exponen las definiciones matemáticas básicas para la representación formal de los datos. Inicialmente se establece la definición y propiedades de los conjuntos que tienen la propiedad conocida como "a priori" [Agr94], llamados **"Conjuntos-AP"**, a partir de los cuales se define la de **"Estructura-AP"** [Mar06], que no es más que un conjunto de conjuntos-AP. Esta estructura captura la semántica básica del texto.

Aquí solamente van a darse algunas de las definiciones y propiedades, para más información consultar [MF08].

Definición y Propiedades de los Conjuntos-AP

Definición 2.2.1. Conjunto-AP
Sea $X = \{x_1...x_n\}$ 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.

$$\forall\, Z \in R \Rightarrow P(Z) \subseteq R \tag{2.17}$$

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

$$\begin{aligned} &a)\ card(Y) = \ max_{Z \in R}(card(Z))\ y \\ &\quad \nexists\, Y' \in R \mid card(Y') = card(Y) \\ &b)\ \forall\, Z \in R; Z \subseteq Y \end{aligned} \tag{2.18}$$

El conjunto Y de cardinal maximal que caracteriza el conjunto-AP se denomina *conjunto generador de* R. La notación $R = g(Y)$ indica que $g(Y)$ será el conjunto-AP generado por 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 2.2.1. Conjunto-AP

Sea $X = \{lluvia, viento, sol, tormenta, nube, calor, brisa, nieve\}$

Sea $R = (\{nube\}, \{lluvia\}, \{viento\}, \{nube, lluvia\}, \{nube, viento\}, \{lluvia, viento\}, \{nube, lluvia, viento\})$.

Entonces el conjunto generador de R es $Y = (\{nube, lluvia, viento\})$.

Como se observa en el ejemplo anterior y teniendo en cuenta la definición de conjunto-AP, el conjunto generador $Y = \{nube, lluvia, viento\}$ 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 (2.2).

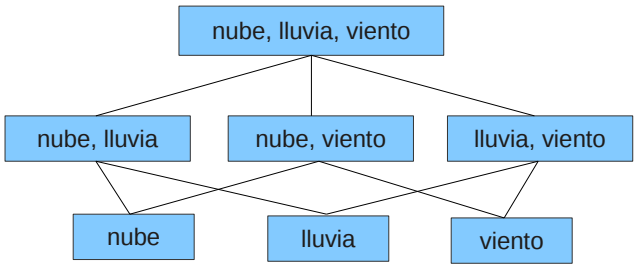


Figura 2.2: Conjunto-AP

A continuación se dan algunas operaciones relacionadas con el conjunto-AP definido. Dichas operaciones serán, a su vez, 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 2.2.2. Inclusión de Conjuntos-AP

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

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

Por la definición podemos apreciar que un conjunto-AP está incluido en otro, si el conjunto generador del primero está incluido en el conjunto generador del se-

gundo.

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 otro conjunto dado.

Definición 2.2.3. 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:*

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

Por la definición podemos apreciar que el subconjunto-AP inducido se obtiene de hacer la intersección de los conjuntos generadores de R con el conjunto Y'.

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 en una base de datos transaccional obtenida de los textos, donde los *items* son términos. Hay que tener en cuenta que estas estructuras se obtienen de forma constructiva según el algoritmo Apriori, generando inicialmente *itemsets* de nivel uno y luego combinándolos para obtener los de nivel dos y así sucesivamente hasta obtener *itemsets* con cardinal maximal, para un soporte mínimo [Agr94, Agr95]. Por tanto, la estructura final es la de un conjunto de conjuntos-AP, que formalmente se define como "Estructura-AP" [MB08].

Definición 2.2.4. 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:*

$$\forall A, B \in S \,;\, A \not\subseteq B \,,\, B \not\subseteq A \tag{2.21}$$

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

De esta definición se deduce 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 2.2.2 de conjunto-AP incluido que se dio anteriormente. 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 $(P(X))$.

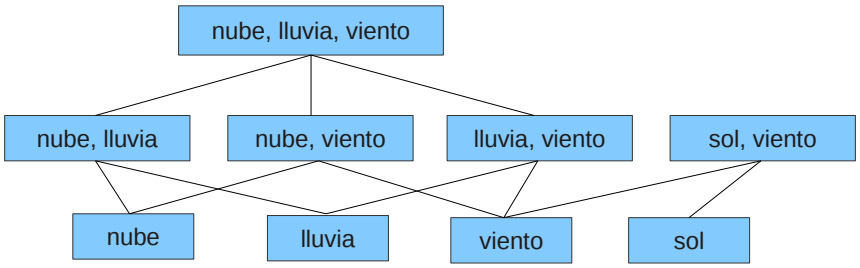


Figura 2.3: Estructura-AP global

Hay que hacer notar que cualquier estructura-AP es un retículo de subconjuntos cuyos extremos superiores son sus conjuntos generadores. La Figura (2.3) muestra una estructura-AP global que se define como $g(\{nube, lluvia, viento\}, \{sol, viento\})$. Se dan a continuación algunas definiciones y propiedades pertenecientes a estas nuevas estructuras.

Definición 2.2.5. Inclusión de Estructuras-AP

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

$$\begin{aligned} &T_1 \subseteq T_2 \Leftrightarrow \forall\ R \textit{ conjunto-AP de } T_1 \\ &\exists\ S \textit{ conjunto-AP de } T_2 \mid R \subseteq S \end{aligned} \tag{2.22}$$

De esta definición se interpreta 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 la 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. Esta operación reviste especial importancia porque es la que permite representar una tupla dada de la base de datos como tipo de dato abstracto.

Definición 2.2.6. 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:*

$$T' = T \bigwedge Y = g(B_1, B_2, ..., B_m) \tag{2.23}$$

donde

$$\forall\, B_i \in \{B_1, ..., B_m\} \Rightarrow \exists\, A_j \in \{A_1, A_2, ..., A_n\} \quad tal\ que\ B_i = A_j \bigcap Y \tag{2.24}$$

$$\forall\, A_j \in \{A_1, ..., A_n\} \Rightarrow \exists\, B_i \in \{B_1, B_2, ..., B_m\} \quad tal\ que\ A_j \bigcap Y \subseteq B_i \tag{2.25}$$

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

Ejemplo 2.2.2. Subestructura-AP Inducida

Sea $X = \{lluvia, viento, sol, tormenta, nube, calor, brisa, nieve\}$

Sea $T = g(\{brisa, sol\}, \{calor, sol\}, \{lluvia, viento, tormenta\}$
$\{viento, nube, nieve\})$,

Sea $Y = \{lluvia, viento, nube\}$

$$\Rightarrow T \bigwedge Y = g(\{lluvia, viento\}, \{viento, nube\}).$$

2.2.2 Acoplamiento de las Estructuras-AP con Conjuntos de Términos

En esta sección se establecen 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 exprese 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, es necesario definir algunos tipos de acoplamientos para satisfacer las consultas sobre dichas estructuras.

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

Definición 2.2.7. 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:*

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

Definición 2.2.8. 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:*

$$Y \bigoplus T = \begin{cases} \textit{verdadero si} & \exists\, A_i \in \{A_1, A_2, ..., A_n\} \\ & /Y \bigcap A_i \neq \emptyset \\ \textit{falso} & \textit{en otro caso} \end{cases} \tag{2.27}$$

Estas definiciones se pueden complementar dando alguna medida o índice que cuantifique estos acoplamientos. La idea es considerar que el acoplamiento que devuelva mayor número de términos, tendrá un índice mayor que uno que devuelva menor número de términos; adicionalmente, si algún conjunto de términos se acopla con un mayor número de conjuntos generadores, éste tendrá un índice mayor que otro que se acople con un número menor de conjuntos generadores. Se establecen dos índices distintos de acoplamiento, el "Índice de Acoplamiento Fuerte" y el "Índice de Acoplamiento Débil".

Definición 2.2.9. Índice de Acoplamiento Fuerte (Débil)
Sea la estructura-AP $T = g(A_1, A_2, ..., A_n)$ *con conjunto referencial de items* X *y* $Y \subseteq X$. *Entonces* $\forall A_i \in \{A_1, A_2, ..., A_n\}$ *se denota:*

$$m_i(Y) = \frac{card(Y \bigcap A_i)}{card(A_i)}, \tag{2.28}$$

$$S = \{i \in \{1, ..., n\} | Y \subseteq A_i\},\ y \tag{2.29}$$

$$W = \{i \in \{1, ..., n\} | Y \bigcap A_i \neq \emptyset\} \tag{2.30}$$

A partir de lo cual se define el índice de acoplamiento fuerte y débil entre Y y T como:

1. *Índice de acoplamiento por el promedio:*

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

$$\textbf{Índice débil} = W(Y|T) = \sum_{i \in W} m_i(Y)/n \tag{2.32}$$

2. *Índice de acoplamiento por el máximo:*

$$\textbf{Índice fuerte} = S(Y|T) = max(m_i(Y));\ i\ \in\ S \tag{2.33}$$

$$\textbf{Índice débil} = W(Y|T) = max(m_i(Y));\ i\ \in\ W \tag{2.34}$$

cumpliendo:

$$\forall Y\ y\ T\ ,\ S(Y|T) \in [0, 1]\ ,\ W(Y|T) \in [0, 1]\ y\ W(Y|T) \geq S(Y|T) \tag{2.35}$$

2.3 Resumen y Conclusiones

Son muchos autores los que han criticado las *tag clouds* por varias razones: por no facilitar información descriptiva [BI08], por representar los términos más populares cuando proceden de folksonomías (siendo estos normalmente los menos relevantes [Sin08]) o por representar los términos más frecuentes cuando se extraen del texto (que son los que menos poder de discriminación poseen), quedando así la visualización dominada por unos pocos temas y dando lugar al solapamiento semántico [HM06].

Resumimos las críticas realizadas a la *tag cloud* en los siguientes puntos:

- **Respecto a la identificación del contenido**

  Normalmente, las etiquetas en la *tag cloud* son palabras aisladas, es decir, monotérminos y al no facilitar ninguna información de contexto, muchas veces no es fácil saber qué concepto representan exactamente.

- **Respecto a la semántica**

  Los términos más frecuentes o populares suelen ser términos ambiguos, no descriptivos, por ejemplo: sistema, conjunto, tipo..., sin embargo, éstos son los términos que aparecen representados en la *tag cloud*.

  Además, sin el empleo de técnicas de *clustering*, no es posible descubrir relaciones entre conceptos cuando éstos están expresados a través de palabras de un sólo término.

- **Respecto a la fundamentación teórica**

  Desde una perspectiva tradicional, sorprende la rápida adopción de las *tag clouds*, ya que conllevan serios 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.

- **Respecto a la metodología de extracción**

  Aunque en este trabajo se han presentado diversas técnicas de extracción, selección, *clustering* y herencia, no existe una metodología estandarizada para la extracción de las etiquetas de una *tag cloud*.

A pesar de toda esta crítica, el uso de las *tag clouds* crece de forma continuada y cada vez es mayor el número de sitios web que las añaden a sus páginas. Esto se debe principalmente a que ofrecen una atmósfera amigable para acceder a un sitio complejo [Hea08].

Ésta es la razón principal por la que proponemos el uso de este tipo de visualización para consultar los atributos textuales de una base de datos y representar el contenido de estos atributos.

Además, las tareas de exploración y búsqueda en bases de datos suelen realizarse por expertos que conocen el lenguaje de consulta y están familiarizados con el esquema de la base de datos [Leo11]. A través de una *tag cloud*, estas tareas pueden ser realizadas por cualquier usuario sin experiencia.

Hemos visto que algunos autores como Koutrika et al. [Kou09a, Kou09b] ya utilizan las *tag clouds* sobre bases de datos, pero no lo hacen para representar el contenido textual de los atributos ni para asistir al usuario en la exploración y en la consulta, si no que únicamente las emplean para resumir los resultados de las búsquedas realizadas a partir de palabras clave. Además, estas búsquedas se realizan sobre datos estructurados, siendo la falta de estructura (normalmente presente en los atributos textuales) el principal problema para manejar la información en las bases de datos.

Esta falta de estructura es la que nos lleva a buscar formas intermedias de representación del texto, matemáticamente definidas, mediante las cuales dotar a estos atributos de estructura y de semántica.

En este capítulo se han visto los antecedentes de la estructura-AP, como forma matemática intermedia. Esta estructura es una buena candidata para representar el texto conservando su semántica, ya que permite que los términos relacionados puedan permanecer unidos. Sin embargo, no discrimina según el orden o la adyacencia de los términos en el texto ni está ponderada para poder representarla en forma de *tag cloud*. En el Capítulo 3 corregiremos estas deficiencias de la estructura-AP, estableciendo la propuesta teórica de este trabajo.

La representación de la forma matemática intermedia que proponemos (basada en la estructura-AP) a través de una *tag cloud*, hace innecesaria la aplicación de mecanismos de *clustering* en la visualización, ya que esta forma intermedia permite el uso de multitérminos, que conservan unidas las palabras relacionadas, aportando así semántica al matizar el significado de un término con sus términos relacionados, facilitando información de contexto y permitiendo la identificación del contenido [Don07]. Además, como nos dicen Hassan-Montero et al. [HM10] en su estudio mediante la técnica "eye-tracking", la agrupación semántica mediante técnicas de *clustering* no supone una mejora en términos de eficiencia en la localización visual de las etiquetas.

Mediante la forma intermedia de representación se favorece el uso de palabras compuestas [Can08], pero los multitérminos no son los únicos componentes en esta representación, ya que la estructura-AP se compone tanto de multitérminos como de monotérminos, lo que produce una lista coherente [Pan06] y es lo preferido por los usuarios, como apuntaron Kaptein y Marx [Kap10] en su experimento.

El criterio para decidir cuándo una etiqueta (*itemset*) estará representada en la forma intermedia y consecuentemente, en la *tag cloud*, es un criterio basado en frecuencia. Este criterio es el que se aplica normalmente cuando las etiquetas se extraen del texto [Kuo07], ya que los criterios de diversidad y de *ranking* de agregación [Sko11], son sólo aplicables cuando son los usuarios los que asignan las etiquetas a las fuentes de información.

El posicionamiento de las etiquetas dentro de la *tag cloud*, lo realizaremos de forma aleatoria, ya que según demostraron Hassan-Montero et al. [HM10], lo que afecta a los distintos patrones de escaneo visual de las *tag clouds* es donde estén situadas las etiquetas de mayor tamaño y no que las etiquetas estén en un cuadrante determinado de la visualización o que tengan algún tipo de orden.

Por otro lado, no vamos a considerar establecer ninguna herencia entre las etiquetas, ya que lo que principalmente pretendemos con este método para la consulta sobre los atributos textuales, es ganar en precisión y no perderla, como ocurriría con la incorporación de herencia [Hsi06].

Mediante la forma intermedia de representación, la *tag cloud* quedaría definida matemáticamente y la generación de esta forma intermedia sería el método para la extracción de sus etiquetas. Uniendo esto a la semántica aportada por los componentes multitérmino y a la facilidad de identificación de contenido que aportan estos componentes, habríamos solventado las principales deficiencias de la *tag cloud* convencional [TP12].

CAPÍTULO

3

Propuesta Teórica

La falta de estructura en los atributos textuales dificulta el procesamiento automático de éstos. Como se ve en [MF08], los conjuntos-AP proveen al texto de una estructura matemática, dotada de semántica, que facilita su procesamiento: la estructura-AP. En la Sección 2.2 se ha revisado su definición y principales operaciones.

Cuando se procesan los atributos textuales de una base de datos con esta técnica, se obtienen dos tipos de estructuras diferentes: una que define el dominio general del atributo textual y un conjunto de subestructuras (una para cada tupla) que representan el contenido de las tuplas.

Estas estructuras pueden formularse mediante un retículo que posee propiedades estadísticas. En el caso de la estructura de dominio, este retículo resume el contenido textual del atributo, representando con cada una de sus operaciones asociadas las distintas acciones que podemos ejercer sobre los atributos de la base de datos.

En definitiva, la estructura-AP facilita el procesamiento semántico del texto, ya que asumimos como hipótesis que los *itemsets* frecuentes de los que se compone conservan la semántica textual al permitir que los términos más relevantes puedan permanecer unidos.

Otra razón adicional para el uso de la estructura-AP es que los algoritmos usados para su obtención son bien conocidos y están altamente optimizados.

Como vimos en el Capítulo 2, la representación de la información a través de una *tag cloud* ayuda en la identificación del contenido, en la navegación a través de éste, en la depuración de la consulta, etc. Y la estructura-AP aporta semática y una base matemática que permite procesar el texto de forma más eficiente. Si ponderamos esta estructura según la frecuencia con que se repiten los *itemsets* en el texto, es posible visualizarla a través de una *tag cloud*, aprovechando así las ventajas de ambas.

El principal inconveniente que presenta la estructura-AP es que no contempla una relación de orden entre los términos que componen los *itemsets*, por lo que los resultados de las consultas realizadas mediante esta estructura, podrían no ser precisos o devolver más información de la que se solicita. Para solventar este inconveniente y establecer la ponderación, se hace necesario desarrollar algunas extensiones de la estructura-AP.

En la Sección 3.1, definiremos lo que llamamos la **"Estructura-AP Ponderada"** o **"Estructura WAP"** por sus siglas en inglés (*Weighted AP-Structure*), en la que se introduce la ponderación. La "**Estructura-AP Ordenada (Estructura APO) o Estructura Multitérmino**", que establece el orden en la estructura-AP, se define en la Sección 3.2 y, al igual que esta última, debe ser ponderada con el fin de visualizarla en forma de *tag cloud*, creando lo que llamamos la "**Estructura-AP Ordenada Ponderada"** o **"Estructura WAPO"** por sus siglas en inglés (*Weighted AP-Ordered Structure*) y cuya definición podemos ver en la Sección 3.3.

El esquema de la Figura 3.1 clarifica la relación entre la estructura-AP y estas extensiones.

En la Sección 3.4 veremos los tipos de consultas que podemos realizar sobre los atributos textuales de la base de datos dependiendo de si queremos que todos los términos introducidos sean considerados para la recuperación de información ("**Acoplamiento Fuerte**") o si nos conformamos con que al menos alguno lo sea ("**Acoplamiento Débil**"). Y se establecerán unas medidas que nos permitirán medir la bondad de estos tipos de acoplamientos.

Ilustraremos como se generan estas estructuras con el ejemplo práctico de la Sección 3.5 y las compararemos entre sí y con la "**Estructura Monotérmino**", que

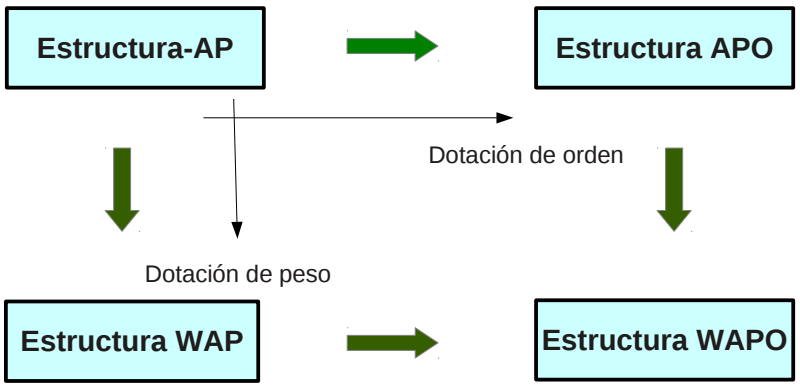


Figura 3.1: Esquema de las extensiones propuestas para la estructura-AP

es la estructura que normalmente vemos representada en Internet con forma de *tag cloud*. Se pondrán varios ejemplos de posibles consultas sobre un atributo ficticio con el fin de ilustrar los cálculos de las medidas de bondad de los acoplamientos fuerte y débil y hacer una comparación entre éstos.

Terminaremos exponiendo un resumen y algunas de las conclusiones obtenidas en la Sección 3.6.

3.1 Estructura WAP

Para definir la estructura WAP, es necesario introducir el concepto de **"*item-set* ponderado"** y a partir de éste, el concepto de **"Conjunto-AP ponderado"** o **"Conjunto WAP"** por sus siglas en inglés (*Weighted AP-Set*).

3.1.1 Definición y Propiedades de los Conjuntos WAP

Definición 3.1.1. Itemset ponderado

Sea $X = \{x_1, x_2, ...\}$ *un conjunto referencial de items. Sea* $I_t \subseteq X$ *un conjunto de items. Diremos que* $\widetilde{I}_t$ *es un itemset ponderado de* X *si y sólo si:*

$$\widetilde{I_t} = [I_t, \omega_t] \qquad \forall t = 1, 2, ... \tag{3.1}$$

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

Propiedad 3.1.1.

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 para un mismo conjunto referencial X

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

Ejemplo 3.1.1. Itemsets ponderados
Sea $X = \{lluvia, viento, sol, tormenta, nube, calor, brisa, nieve\}$
Sea $R = g(Y) = g(\{calor, brisa, sol\})$

$$\begin{aligned} \Rightarrow \quad & \widetilde{I_1}(R) = [\{calor\}, (8)], \widetilde{I_2}(R) = [\{brisa\}, (10)], \widetilde{I_3}(R) = [\{sol\}, (16)] \\ & \widetilde{I_4}(R) = [\{calor, brisa\}, (8)], \widetilde{I_5}(R) = [\{calor, sol\}, (6)], \\ & \widetilde{I_6}(R) = [\{brisa, sol\}, (9)], \widetilde{I_7} = [\{calor, brisa, sol\}, (5)] \end{aligned}$$

En el Ejemplo 3.1.1 se han extraído todos los posibles *itemsets* del conjunto R. El dígito al final de cada uno de ellos indicaría el número de veces que aparecen los *items* que componen el *itemset* en un texto hipotético. Esta frecuencia es lo que identificaremos con el peso del *itemset*.

Si existen valoraciones asociadas a los *itemsets* de la estructura-AP, de forma que cada $\widetilde{I_t}$ pueda considerarse un *itemset* ponderado verificando la Propiedad 3.1.1., tendremos la estructura WAP, la cual se compone de conjuntos WAP.

Definición 3.1.2. Conjunto WAP
Sea R un conjunto-AP, diremos que $\widetilde{R}$ es un conjunto WAP si y sólo si:

$$R = \{I_1, I_2, ...\} \Rightarrow \widetilde{R} = \{\widetilde{I_1}, \widetilde{I_2}, ...\} \tag{3.3}$$

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

Propiedades y Operaciones de los conjuntos WAP

Para entender bien las operaciones de la estructura WAP es necesario introducir la inclusión de conjuntos WAP y el concepto de Subconjunto WAP Inducido.

Definición 3.1.3. Inclusión de Conjuntos WAP
Sean $\widetilde{R} = \widetilde{g}(Y)$ *y* $\widetilde{S} = \widetilde{g}(Y')$ *dos conjuntos WAP con el mismo conjunto referencial de items* X*:*

$$\widetilde{R} \subseteq \widetilde{S} \Leftrightarrow Y \subseteq Y' \tag{3.4}$$

siendo

$$\omega_{I_t}(\widetilde{R}) = \omega_{I_t}(\widetilde{S}) \ \ \forall\, t = 1, 2, ... \tag{3.5}$$

Es decir, el peso de los *itemsets* será el mismo en $\widetilde{R}$ y en $\widetilde{S}$, ya que viene dado por la frecuencia en que éstos aparecen en el texto y ambos proceden del mismo texto.

Ejemplo 3.1.2. Inclusión de Conjuntos WAP
Sea $X = \{lluvia, viento, sol, tormenta, nube, calor, brisa, nieve\}$
Sea $\widetilde{R} = \widetilde{g}(\{viento, lluvia\}) = ([\{viento, lluvia\}, (3)], [\{viento\}, (4)], [\{lluvia\}, (7)])$
Sea $\widetilde{S} = \widetilde{g}(\{viento, sol, lluvia\}) = ([\{viento, lluvia, sol\}, (2)], [\{viento, lluvia\}, (3)], [\{lluvia, sol\}, (5)], [\{viento, sol\}, (4)], [\{viento\}, (4)], [\{lluvia\}, (7)], [\{sol\}, (6)])$

$$\Rightarrow \widetilde{R} \subseteq \widetilde{S}$$

Los *itemsets* comunes en $\widetilde{R}$ y $\widetilde{S}$ deben tener el mismo peso, como se muestra en el Ejemplo 3.1.2.

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

Definición 3.1.4. Subconjunto WAP Inducido
Sea $\widetilde{R} = \widetilde{g}(Y)$ diremos que $\widetilde{S}$ es el subconjunto WAP inducido por Y'si y sólo si:

$$\widetilde{S} = g(\widetilde{R} \cap Y') \tag{3.6}$$

siendo

$$\omega_{I_t}(\widetilde{S}) = \omega_{I_t}(\widetilde{R}) \tag{3.7}$$

No tiene sentido que el conjunto Y' sea ponderado, ya que normalmente este conjunto representará a una tupla o bien a los elementos introducidos en la consulta.

El peso de los *itemsets* tras la intersección, será el mismo que tuvieran en $\widetilde{R}$, ya que ésta representa los *itemsets* en $\widetilde{R}$ que se recuperan con Y'.

Ver la intersección entre conjuntos-AP en [MF08].

Pasemos a ver la definición de estructura WAP, que representará el dominio activo del atributo a partir del cual se construye.

La estructura WAP, al igual que la estructura-AP, se obtiene de forma constructiva, por la generación inicialmente de los *itemsets* con cardinalidad igual a uno; seguidamente, combinando éstos para obtener los de cardinalidad igual a dos y así sucesivamente hasta obtener los *itemsets* de máxima cardinalidad, según un soporte mínimo establecido.

Por consiguiente, la estructura final será un conjunto de conjuntos WAP, por lo que se define como sigue.

3.1.2 Definición y Operaciones de la Estructura WAP

Definición 3.1.5. Estructura WAP
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:

$$\forall \widetilde{A}, \widetilde{B} \in \widetilde{S} \; ; \; \widetilde{A} \not\subseteq \widetilde{B} \; , \; \widetilde{B} \not\subseteq \widetilde{A} \tag{3.8}$$

Al conjunto de conjuntos WAP: $\widetilde{T} = g(\widetilde{A}, \widetilde{B}, ...)$, lo llamaremos "Estructura WAP con conjunto generador $\widetilde{S}$".

Nota.- La ponderación de los *itemsets* en $\widetilde{S}$, vendrá dada por la frecuencia con que éstos aparezcan en el texto, independientemente de si un *itemset* se encuentra en un sólo conjunto generador o en más de uno.

Ejemplo 3.1.3. Estructura WAP

$$\begin{aligned}
\text{Sea } \widetilde{A} &= \widetilde{g}(\{calor, lluvia, viento\}) \\
&= ([\{calor, lluvia, viento\}, (3)], [\{calor, lluvia\}, (4)], [\{calor, viento\}, (3)], \\
&\quad [\{lluvia, viento\}, (6)], [\{calor\}, (10)], [\{lluvia\}, (6)], [\{viento\}, (7)]) \text{ y} \\
\text{Sea } \widetilde{B} &= \widetilde{g}(\{calor, sol\} \\
&= ([\{calor, sol\}, (5)], [\{calor\}, (10)], [\{sol\}, (5)]), \text{ entonces :} \\
\widetilde{T} &= \widetilde{g}(\{calor, lluvia, viento\}, \{calor, sol\}) \\
&= ([\{calor, lluvia, viento\}, (3)], [\{calor, lluvia\}, (4)], [\{calor, viento\}, (3)], \\
&\quad [\{lluvia, viento\}, (6)], [\{calor, sol\}, (5)], [\{calor\}, (10)], [\{lluvia\}, (6)], \\
&\quad [\{viento\}, (7)], [\{sol\}, (5)]
\end{aligned}$$

Visualización de la Estructura WAP en forma de *Tag Cloud*.

En la Figura 3.2 podemos ver la *tag cloud* de la estructura WAP del Ejemplo 3.1.3.

Propiedades y Operaciones de la Estructura WAP

La primera propiedad que definiremos será la inclusión de estructuras WAP, para ello nos serviremos de la Definición 3.1.3 de inclusión de conjuntos WAP.

Definición 3.1.6. Inclusión de Estructuras WAP
Sean $\widetilde{T}_1$ y $\widetilde{T}_2$ dos estructuras WAP 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:

$$\begin{aligned}
\widetilde{T}_1 \subseteq \widetilde{T}_2 \Leftrightarrow \forall\ \widetilde{R} \textit{ conjunto-WAP de } \widetilde{T}_1 \Rightarrow \exists\ \widetilde{S} \textit{ conjunto WAP de } \widetilde{T}_2, \\
tal\ que\ \widetilde{R} \subseteq \widetilde{S}\ (\textit{análogamente } R \subseteq S)
\end{aligned} \tag{3.9}$$

siendo

$$\omega_{I_t}(\widetilde{T}_1) = \omega_{I_t}(\widetilde{T}_2)\ \ \forall\ t = 1, 2, ... \tag{3.10}$$

Figura 3.2: Estructura WAP

De esta definición se deduce que para que una estructura WAP $\widetilde{T}_1$ esté incluida en otra estructura WAP $\widetilde{T}_2$, todos los conjuntos generadores de $\widetilde{T}_1$ deben aparecer incluidos en alguno de los conjuntos generadores de $\widetilde{T}_2$.

El peso de los *itemsets* será el mismo en $\widetilde{T}_1$ y $\widetilde{T}_2$, ya que ambas estructuras proceden del mismo texto.

A continuación se introduce una de las operaciones más importantes que se pueden definir sobre la estructura WAP, la operación de subestructura WAP inducida, que es la estructura resultante de realizar la intersección de la estructura WAP con cualquier conjunto dado. Esta operación revestía especial importancia con la estructura-AP porque era la que permitía encontrar la representación del tipo de dato abstracto (TDA) de cada tupla de la base de datos. En la estructura WAP, al introducir la ponderación, la importancia fundamental de esta operación viene dada porque además representa los resultados obtenidos mediante una consulta y gracias al peso, es posible saber el número exacto de tuplas que se recuperan.

Definición 3.1.7. Subestructura WAP Inducida
Sea la estructura WAP $\widetilde{T} = g(\widetilde{A_1}, \widetilde{A_2}, ..., \widetilde{A_n})$ *con conjunto referencial de items* X *e* $Y \subseteq X$. *Definiremos la subestructura WAP de* T, $\widetilde{T'}$ *inducida por* Y *como:*

$$\widetilde{T'} = \widetilde{T} \wedge Y = g(\widetilde{B_1}, \widetilde{B_2}, ..., \widetilde{B_m}) \tag{3.11}$$

donde

$$\forall\, \widetilde{B_i} \in \{\widetilde{B_1}, ..., \widetilde{B_m}\} \Rightarrow \exists\, \widetilde{A_j} \in \{\widetilde{A_1}, ..., \widetilde{A_n}\} \quad tal\ que\ B_i = A_j \cap Y \tag{3.12}$$

$$\forall\, \widetilde{A_j} \in \{\widetilde{A_1}, ..., \widetilde{A_n}\} \Rightarrow \exists\, \widetilde{B_i} \in \{\widetilde{B_1}, ..., \widetilde{B_m}\} \quad tal\ que\ A_j \cap Y \subseteq B_i \tag{3.13}$$

siendo

$$\omega_{I_t}(\widetilde{T'}) = \omega_{I_t}(\widetilde{T}) \tag{3.14}$$

$\widetilde{T'}$ es la estructura WAP 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 $\widetilde{T}$.

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

No tiene sentido que el conjunto Y sea ponderado, como vimos en la Definición 3.3.5.

Aclararemos estas ideas con el Ejemplo 3.1.4.

Ejemplo 3.1.4. Subestructura WAP Inducida

Sea $X = \{lluvia, viento, sol, tormenta, nube, calor, brisa, nieve\}$

$$\begin{aligned}
Sea\, \widetilde{T} &= \widetilde{g}(\{viento, sol, nube\}, \{lluvia, nube\}) \\
&= ([\{viento, sol, nube\}, (4)], [\{viento, sol\}, (6)], [\{sol, nube\}, (4)], \\
&\quad [\{viento, nube\}, (7)], [\{lluvia, nube\}, (2)], [\{viento\}, (7)], \\
&\quad [\{sol\}, (8)], [\{nube\}, (7)], [\{lluvia\}, (3)] \\
Sea\, \widetilde{Y} &= g(\{viento, lluvia, nube\}) \\
\Rightarrow \widetilde{T} \wedge Y &= \widetilde{g}(\{viento, nube\}, \{lluvia, nube\}) \\
&= ([\{viento, nube\}, (7)], [\{lluvia, nube\}, (2)], [\{viento\}, (7)], \\
&\quad [\{lluvia\}, (3)], [\{nube\}, (7)])
\end{aligned}$$

El peso de los *itemsets* en la subestructura WAP inducida es el que tenían en la estructura WAP original. Esta primera representa el resultado obtenido con la consulta, el conjunto-AP los términos introducidos en la consulta (por lo que no estaría ponderado) y la estructura WAP la información contenida en el atributo textual consultado.

Se define ahora la operación superestructura WAP inducida, que es la que permite obtener la estructura WAP generada por la unión de un conjunto dado con una estructura WAP determinada.

Definición 3.1.8. Superestructura WAP Inducida
Sea la estructura WAP $\widetilde{T} = g(\widetilde{A_1}, \widetilde{A_2}, ..., \widetilde{A_n})$ *con conjunto referencial de items* X *e* $Y \subseteq X$. *Definiremos la superestructura WAP de* T, $\widetilde{T'}$ *inducida por* $\widetilde{Y}$ *como:*

$$\widetilde{T'} = \widetilde{T} \vee \widetilde{Y} = g(\widetilde{B_1}, \widetilde{B_2}, ..., \widetilde{B_m}) \tag{3.15}$$

donde

$$\forall\, \widetilde{B_i} \in \{\widetilde{B_1}, ..., \widetilde{B_m}\} \Rightarrow \exists\, \widetilde{A_j} \in \{\widetilde{A_1}, ..., \widetilde{A_n}\} \quad tal\ que\ B_i = A_j \cup Y \tag{3.16}$$

$$\forall\, \widetilde{A_j} \in \{\widetilde{A_1}, ..., \widetilde{A_n}\} \Rightarrow \exists\, \widetilde{B_i} \in \{\widetilde{B_1}, ..., \widetilde{B_m}\} \quad tal\ que\ A_j \cup Y \subseteq B_i \tag{3.17}$$

con

$$\omega_{I_t(\widetilde{T'})} = \begin{cases} \omega_{I_t(\widetilde{T})} & si\ I_t(\widetilde{T'}) \in (\widetilde{T} - \widetilde{Y}), \\ \omega_{I_t(\widetilde{Y})} & si\ I_t(\widetilde{T'}) \in (\widetilde{Y} - \widetilde{T}), \\ \omega_{I_t(\widetilde{T})} + \omega_{I_t(\widetilde{Y})} - \omega_{I_t(\widetilde{T} \cap \widetilde{Y})} & si\ I_t(\widetilde{T'}) \in \widetilde{T}, \widetilde{Y} \end{cases} \tag{3.18}$$

En este caso lo que hacemos es añadir información al atributo textual, representado por $\widetilde{T}$. Esta información que añadimos se representa por $\widetilde{Y}$. En este proceso podría ocurrir que algunas de las entradas que se agregan nuevas al atributo ya se encuentren en éste previamente, por lo que para hallar el peso de los *itemsets*

coincidentes en $\widetilde{T}$ y en $\widetilde{Y}$, hay que tener en cuenta que algunos pueden provenir de entradas comunes o repetidas en ambos. Por lo tanto, la ponderación de estos *itemsets* coincidentes se calcula como la suma de los pesos que presentan en los conjuntos originales menos el peso de los que provenientes de entradas comunes. Los *itemsets* que estuvieran sólo en $\widetilde{T}$ $(\widetilde{T} - \widetilde{Y})$ o sólo en $\widetilde{Y}$ $(\widetilde{Y} - \widetilde{T})$ conservarán su peso.

Aquí sí tiene sentido que el conjunto Y sea ponderado, ya que no representa los términos introducidos con la consulta, si no información que se añade al atributo textual.

Ver unión entre conjuntos-AP en [MF08].

Ejemplo 3.1.5. Superestructura WAP Inducida
Sea $X = \{lluvia, viento, sol, tormenta, nube, calor, brisa, nieve\}$

$$\begin{aligned}
Sea\,\widetilde{T} &= \widetilde{g}(\{sol, nube\}) \\
&= ([\{sol, nube\}, (4)], [\{sol\}, (8)], [\{nube\}, (7)]), \\
Sea\,\widetilde{Y} &= \widetilde{g}(\{sol, lluvia\}) \\
&= ([\{sol, lluvia\}, (3)], [\{sol\}, (4)], [\{lluvia\}, (3)]) \\
\Rightarrow \widetilde{T} \vee \widetilde{Y} &= \widetilde{g}(\{sol, nube, lluvia\}) \\
&= ([\{sol, nube, lluvia\}, (1)], [\{sol, nube\}, (4)] \\
&\quad [\{sol, lluvia\}, (3)], [\{nube, lluvia\}, (0)], [\{sol\}, (10)], \\
&\quad [\{nube\}, (7)], [\{lluvia\}, (3)])
\end{aligned}$$

Vemos que el *itemset* $\{sol\}$ estaba ya presente en $\widetilde{T}$ con un peso igual a 8. Como en $\widetilde{Y}$ el peso de $\{sol\}$ es 4 y en $\widetilde{T}'$ es 10, se deduce que 2 elementos que contabilizaban para el peso de este *item* en $\widetilde{Y}$ ya estaban presentes en $\widetilde{T}$, por lo que no se han añadido como nuevos y por lo tanto, no se suman en el cálculo del peso de $\{sol\}$ en $\widetilde{T}'$.

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 función de su frecuencia en el texto.

Definición 3.1.9. Intersección de Estructuras WAP
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 WAP, se*

define la intersección de $\widetilde{T}_1$ *y* $\widetilde{T}_2$ *como:*

$$S = \widetilde{T}_1 \cap \widetilde{T}_2 = g(\widetilde{C_1}, \widetilde{C_2}, ..., \widetilde{C_l}) \tag{3.19}$$

verificando:

$$\begin{array}{l} \forall \widetilde{C_i} \in \{\widetilde{C_1}, ..., \widetilde{C_l}\};\ \exists\ \widetilde{A_p} \in \{\widetilde{A_1}, ..., \widetilde{A_n}\} \\ y\ \widetilde{B_q} \in \{\widetilde{B_1}, ..., \widetilde{B_m}\}\ /\ C_i = A_p \cap A_q \end{array} \tag{3.20}$$

siendo

$$\omega_{I_t(S)} \leq \omega_{I_t(\widetilde{T_1})}, \omega_{I_t(\widetilde{T_2})} \tag{3.21}$$

La intersección entre dos estructuras WAP representa las entradas comunes en dos atributos textuales de la base de datos, por lo que el peso de los *itemsets* en ésta se calculará sobre el texto, en función de la frecuencia que presenten en el conjunto intersección y en todo caso, será menor igual que el que tuvieran en sus estructuras de procedencia.

Ejemplo 3.1.6. Intersección de Estructuras WAP
Sea $X = \{lluvia, viento, sol, tormenta, nube, calor, brisa, nieve\}$

$$\begin{aligned} Sea\ \widetilde{T}_1 &= \widetilde{g}(\{viento, sol, nube\}, \{lluvia, nube\}) \\ &= ([\{viento, sol, nube\}, (4)], [\{viento, sol\}, (6)], [\{sol, nube\}, (4)], \\ &\quad [\{viento, nube\}, (5)], [\{lluvia, nube\}, (2)], [\{viento\}, (7)], \\ &\quad [\{sol\}, (8)], [\{nube\}, (7)], [\{lluvia\}, (3)] \\ Sea\ \widetilde{T}_2 &= \widetilde{g}(\{viento, lluvia, nube\}) \\ &= ([\{viento, lluvia, nube\}, (3)], [\{viento, lluvia\}, (4)], \\ &\quad [\{lluvia, nube\}, (3)], [\{viento, nube\}, (4)], [\{viento\}, (5)], \\ &\quad [\{lluvia\}, (6)], [\{nube\}, (5)]) \\ \Rightarrow \widetilde{T}_1 \cap \widetilde{T}_2 &= \widetilde{g}(\{viento, nube\}, \{lluvia, nube\}) \\ &= ([\{viento, nube\}, (3)], [\{lluvia, nube\}, (2)], [\{viento\}, (4)], \\ &\quad [\{lluvia\}, (3)], [\{nube\}, (4)]) \end{aligned}$$

A continuación se define la unión de estructuras WAP, que representa la unión de dos atributos textuales diferentes.

Definición 3.1.10. Unión de Estructuras WAP
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 WAP, se define la unión como:*

$$S = \widetilde{T_1} \cup \widetilde{T_2} = g(\widetilde{C_1}, \widetilde{C_2}, ..., \widetilde{C_l}) \tag{3.22}$$

verificando:

$$i) \quad \forall \widetilde{C_i} \in \{\widetilde{C_1}, ..., \widetilde{C_l}\} \Rightarrow (\ \exists\ \widetilde{A_j}/\widetilde{C_i} = \widetilde{A_j})\ o\ (\exists\ \widetilde{B_l}/\widetilde{C_i} = \widetilde{B_l}) \tag{3.23}$$

$$ii) \quad \forall\ \widetilde{A_j} \in \{\widetilde{A_1}, ..., \widetilde{A_n}\} \Rightarrow \exists\ \widetilde{C_i}/\widetilde{A_j} \subseteq \widetilde{C_i} \tag{3.24}$$

$$iii) \quad \forall\ \widetilde{B_l} \in \{\widetilde{B_1}, ..., \widetilde{B_m}\} \Rightarrow \exists\ \widetilde{C_i}/\widetilde{B_l} \subseteq \widetilde{C_i} \tag{3.25}$$

donde

$$\omega_{I_t(\widetilde{T'})} = \begin{cases} \omega_{I_t(\widetilde{T_1})} & si\ I_t(\widetilde{S}) \in (\widetilde{T_1} - \widetilde{T_2}), \\ \omega_{I_t(\widetilde{T_2})} & si\ I_t(\widetilde{S}) \in (\widetilde{T_2} - \widetilde{T_1}), \\ \omega_{I_t(\widetilde{T_1})} + \omega_{I_t(\widetilde{T_2})} - \omega_{I_t(\widetilde{T_1} \cap \widetilde{T_2})} & si\ I_t(\widetilde{S}) \in \widetilde{T_1}, \widetilde{T_2} \end{cases} \tag{3.26}$$

Los pesos en la unión de estructuras WAP se calculan de forma similar a los pesos en la superestructura WAP inducida (Definición 3.1.8).

Ejemplo 3.1.7. Unión de Estructuras WAP
Sea $X = \{lluvia, viento, sol, tormenta, nube, calor, brisa, nieve\}$
Sea $\widetilde{T_1} = \widetilde{g}(Y), Y = (\{viento, sol\}, \{lluvia\})$
Sea $\widetilde{T_2} = \widetilde{g}(Y'), Y' = (\{calor, sol\}, \{viento\})$
$\Rightarrow \widetilde{T_1} \cup \widetilde{T_2} = \widetilde{g}(\{viento, sol, calor\}, \{lluvia, calor, sol\}, \{lluvia, viento\})$

Los pesos se calcularían igual que en el Ejemplo3.1.5.

3.2 Estructura APO

La **"Estructura-AP Ordenada (APO)"** o estructura multitérmino se diferencia de la estructura-AP en que discrimina entre términos según el orden que éstos presenten en el texto y para construirla se ha de realizar una limpieza previa de los datos y aplicar un proceso posterior de minería de textos, igual que se hace para la estructura-AP.

La estructura APO también representa el dominio activo del atributo para el que se construye, teniendo en cuenta que este dominio está afectado de orden, es decir, está representado por secuencias en lugar de por conjuntos.

Para introducir la estructura APO, empezaremos definiendo los conjuntos-AP ordenados o secuencias-AP (**"AP-Seqs"**), a las que también conocemos como conjuntos multitérmino.

Además, puesto que la *tag cloud* monotérmino es la que más se ve en la Web, nos parece importante asignarle un modelo matemático que conoceremos como **"Estructura Monotérmino"** y para el que será necesario definir previamente los conjuntos monotérmino. De esta forma, podrán verse posteriormente las similitudes y diferencias de esta estructura con las que se proponen en este capítulo.

Cabe indicar que la estructura monotérmino no es más que una estructura APO o multitérmino donde todas las componentes tienen cardinal igual a uno.

3.2.1 Componente *K*-Término, Monotérmino y Multitérmino

Definición 3.2.1. Componente *k*-término
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 si consta de k elementos diferentes del conjunto X, con $1 \leq k \leq n$:

$$y = \{y_1, y_2, ..., y_k\}; \;\; y_i \in X \tag{3.27}$$

Es decir, y es una componente k-*término si:*

$$card(y) = k \;\; \forall \;\; k \in [1, n] \tag{3.28}$$

El cardinal de una componente es igual a la cantidad de elementos que posee del conjunto X.

Decimos que una componente *k*-término es **monotérmino** si $card(y) = 1$ y **multitérmino** si $card(y) \geq 2$.

3.2.2 Estructura Monotérmino

Para entender esta estructura empezaremos definiendo los conjuntos monotérmino.

Definición 3.2.2. Conjunto Monotérmino
Sea $X = \{x_1, x_2, ..., x_n\}$ un conjunto referencial de items. Se dice que $R \subseteq X$ conjunto de itemsets frecuentes es un conjunto monotérmino si y sólo si:

$$card(y) = 1 \; \forall \; y \in R \tag{3.29}$$

Un conjunto monotérmino es aquel que está formado exclusivamente por componentes monotérmino.

Una estructura monotérmino es una estructura obtenida a partir de conjuntos monotérmino, es decir, será un conjunto de conjuntos monotérmino, eliminando los elementos repetidos.

Definición 3.2.3. Estructura Monotérmino
Sea $X = \{x_1, x_2, ..., x_n\}$ un conjunto referencial de items y $S = \{A, B, ...\} \subseteq X$ un conjunto de conjuntos monotérmino, tal que:

$$\forall A, B \in S; A \nsubseteq B, B \nsubseteq A \tag{3.30}$$

llamaremos estructura monotérmino generada por S, $M = g(A, B, ...)$ al conjunto de conjuntos monotérmino cuyos conjuntos generadores son A, $B, ...$

Ejemplo 3.2.1. Estructura Monotérmino
Sea $X = \{lluvia, viento, sol, tormenta, nube, calor, brisa, nieve\}$
Sea $A = \{\{nieve\}, \{sol\}\}$
Sea $B = \{sol\}$
Sea $C = \{\{calor\}, \{sol\}, \{brisa\}\}$
Sea $D = \{brisa\}$
$\Rightarrow M = g(A, B, C, D) = (\{nieve\}, \{sol\}, \{calor\}, \{brisa\})$

3.2.3 Definición de AP-Seq y de Estructura APO

Las AP-Seqs o conjuntos multitérmino estarán formadas tanto por componentes multitérmino como monotérmino.

Definición 3.2.4. AP-Seq o 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 una AP-Seq o un conjunto multitérmino si y sólo si:

1. $\forall\ Z = \{z_i, ..., z_j\}$ *secuencia de* $R \Rightarrow$

$$\begin{cases} \{z_i, ..., z_{j-1}\} \in R \\ \{z_{i+1}, ..., z_j\} \in R \end{cases} \quad \textit{con } 1 \leq i \leq n-1,\ i+1 \leq j \leq n \tag{3.31}$$

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

$$\begin{aligned} &a)\ \textit{card}(Y) = \textit{max}_{\ Z \in R}\ (\textit{card}(Z)) \quad y \quad \nexists\, Y' \in R \mid \textit{card}(Y') = \textit{card}(Y) \\ &b)\ \forall\ Z \in R; Z \subseteq Y \end{aligned} \tag{3.32}$$

La secuencia Y de máxima cardinalidad caracteriza a la AP-Seq y se denomina secuencia generadora de R. Denotamos $R = g(Y)$, donde $g(Y)$ es la AP-Seq con secuencia generadora Y.

Llamamos nivel de $g(Y)$ al cardinal de Y, es decir, a la cantidad de elementos de la secuencia generadora Y. Las AP-Seqs de nivel 1 se corresponden con los elementos de X y la AP-Seq de nivel 0 es el conjunto vacío $\emptyset$.

Ejemplo 3.2.2. AP-Seq
Sea $X = \{lluvia, viento, sol, tormenta, nube, calor, brisa, nieve\}$.
Sea $R = (\{nube, lluvia, viento\}, \{nube, lluvia\}, \{lluvia, viento\}, \{nube\}, \{lluvia\}, \{viento\})$.
$\Rightarrow$ R es una AP-Seq de X, siendo la secuencia generadora Y la secuencia de mayor cardinalidad: $Y = \{nube, lluvia, viento\}$, estando todas las demás secuencias incluidas en ésta y cumpliendo la condición 3.31 de adyacencia estricta de los términos.

Para comprender mejor su definición, en la Figura 3.3 se representa el retículo de esta AP-Seq.Vemos que la secuencia Y de mayor cardinalidad es la raíz del árbol y en las hojas aparecen los elementos de los que se compone Y. En los niveles intermedios del retículo se encuentran las secuencias con nivel intermedio entre el 1 y el de la secuencia generadora, de forma escalonada.

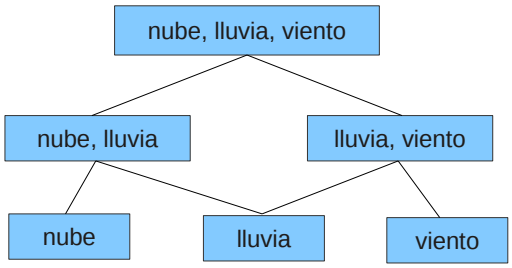


Figura 3.3: AP-Seq

Para distinguir con el retículo de un conjunto-AP, ver la Figura 2.2.

El concepto de AP-Seq establecido se usa para definir la estructura de información que se construye en base a ésta: la estructura APO. Para ello se calculan las secuencias frecuentes de elementos, a las que llamaremos a partir de ahora *"item-seqs" (item-sequences)* frecuentes (ver Definición 3.2.6).

La estructura APO también se obtiene de forma constructiva, generando inicialmente las *item-seqs* frecuentes (según un soporte mínimo establecido) con cardinal igual a uno (que son los monotérminos); luego éstas se combinan para obtener las de cardinal dos, y así sucesivamente hasta obtener las *item-seqs* frecuentes con cardinal maximal.

Por tanto, la estructura final es la de un conjunto de AP-Seqs frecuentes. Para generarla se usa una ligera modificación del algoritmo Apriori que veremos en la Sección 4.3.

Antes de introducir las operaciones entre AP-Seqs, definiremos la estructura APO, ya que algunas de estas operaciones darán como resultado una estructura. Posteriormente, en la Sección 3.3, se hablará de la estructura monotérmino ponderada y estructura APO ponderada o WAPO.

La estructura APO se define de forma similar a la estructura-AP, como un conjunto de AP-Seqs.

Definición 3.2.5. Estructura APO
Sea $X = \{x_1, x_2, ..., x_n\}$ *un conjunto referencial de items y* $S = \{A, B, ...\} \subseteq P(X)$ *un conjunto de item-seqs frecuentes de nivel mayor o igual que uno, siendo P(X) las partes de X y* $A, B, ...$ *AP-Seqs tales que:*

$$\forall\, A, B \in S;\ A \nsubseteq B, B \nsubseteq A \text{ y } B \neq A \tag{3.33}$$

Llamaremos estructura APO generada por S, $E = g(A, B, ...)$, *al conjunto de AP-Seqs cuyas secuencias generadoras son* $A, B, ...$

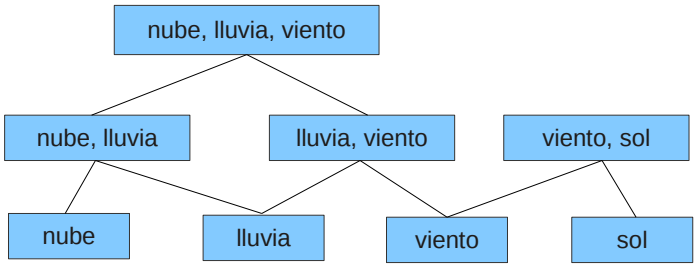


Figura 3.4: Estructura APO

Ejemplo 3.2.3. Estructura APO
En el retículo de la Figura 3.4 podemos ver la estructura APO, E, formada por las AP-Seqs A y B donde:
$A = g(\{nube, lluvia, viento\})$ y
$B = g(\{viento, sol\})$
$\Rightarrow E = g(\{nube, lluvia, viento\}, \{viento, sol\})$

Como vemos, ninguna de estas dos secuencias A y B está completamente incluida en la otra, aunque compartan elementos comunes.

3.2.4 Propiedades y Operaciones de las AP-Seqs y la Estructura APO

Para entender algunas operaciones entre AP-Seqs, como el acoplamiento, es necesario definir el concepto de *item-seq*.

Definición 3.2.6. Secuencia de Elementos o *item-seq* de una AP-Seq
Sea R = g(Y) una AP-Seq con conjunto referencial de items X, donde $Y = (y_1, ..., y_i, ..., y_j, ...y_n)$. Diremos que $\alpha_t \subseteq Y$ es una secuencia de elementos del conjunto Y si y sólo si:

$$\alpha_t = \{y_i, .., y_j\} \tag{3.34}$$

con $1 \leq i \leq n-1$; $i+1 \leq j \leq n$; $1 \leq t \leq \sum_{k=1}^{n} k$, donde $\sum_{k=1}^{n} k$ es el número máximo de subsecuencias que se pueden formar con los elementos de la secuencia Y.

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

Ejemplo 3.2.4. Secuencia de Elementos o *item-seq* de una AP-Seq
Sea $X = \{lluvia, viento, sol, tormenta, nube, calor, brisa, nieve\}$
Sea $R = g(Y) = \{tormenta, sol\}$
Sea $S = g(Y) = \{sol, nube, viento\}$

$$\begin{aligned} \Rightarrow \quad & \alpha_1(R) = \{tormenta\}, \alpha_2(R) = \{sol\}, \alpha_3(R) = \{tormenta, sol\} \\ \Rightarrow \quad & \alpha_1(S) = \{sol\}, \alpha_2(S) = \{nube\}, \alpha_3(S) = \{viento\}, \\ & \alpha_4(S) = \{sol, nube\}, \alpha_5(S) = \{nube, viento\} \\ & \alpha_6(S) = \{sol, nube, viento\} \end{aligned}$$

En este ejemplo vemos todas las posibles *item-seqs* que conforman las AP-Seqs R y S.

Definición 3.2.7. Inclusión de AP-Seqs
Sea $R = g(Y)$ una AP-Seq con conjunto referencial de items X. Sea $S = g(Y')$

otra AP-Seq con el mismo conjunto referencial X. Se dice que R está incluida en S si y sólo si:

$$R \subseteq S \Leftrightarrow \begin{cases} Y = (y_1, ..., y_n) \\ Y' = (y'_1, .., y'_i, .., y'_j, ..., y'_m) \end{cases} \Rightarrow (y_1, .., y_n) = (y'_i, .., y'_j) \quad (3.35)$$
$$\textit{para } 1 \leq i \leq m-1,\ i+1 \leq j \leq m$$

Es decir, una AP-Seq R estará incluida en otra AP-Seq S, cuando exista una secuencia de elementos exactamente igual en la secuencia generadora de S a la secuencia generadora de R.

Ejemplo 3.2.5. Inclusión de AP-Seqs
Sea $X = \{lluvia, viento, sol, tormenta, nube, calor, brisa, nieve\}$
Sea $R = g(Y)$, $Y = \{tormenta, sol, lluvia, nube\}$
Sea $S = g(Y')$, $Y' = \{sol, lluvia, nube\}$
Sea $T = g(Y'')$, $Y'' = \{tormenta, lluvia, nube\}$

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

Veamos:

$R = (\{tormenta, sol, lluvia, nube\}, \{tormenta, sol, lluvia\}, \{sol, lluvia, nube\}, \{tormenta, sol\}, \{sol, lluvia\}, \{lluvia, nube\}, \{tormenta\}, \{sol\}, \{lluvia\}, \{nube\})$

$S = (\{sol, lluvia, nube\}, \{sol, lluvia\}, \{lluvia, nube\}, \{sol\}, \{lluvia\}, \{nube\})$

$T = (\{tormenta, lluvia, nube\}, \{tormenta, lluvia\}, \{lluvia, nube\}, \{tormenta\}, \{lluvia\}, \{nube\})$

Todas las *item-seqs* de S están en R, pero no todas las *item-seqs* de T lo están, como es el caso de {*tormenta, lluvia*}, que está en T, pero no está en R, a pesar de que todos los elementos de la secuencia generadora de T, Y'', están contenidos en la secuencia generadora de R, Y. Sin embargo, los términos "tormenta" y "lluvia"

son estrictamente adyacentes en Y'', pero no en Y, por lo que T no estará incluida en R, mientras que S sí lo estaría.

Definición 3.2.8. Acoplamiento de AP-Seqs
Sean $R = g(Y)$ y $S = g(Y')$ AP-Seqs con conjunto referencial X. Se define el acoplamiento entre R y S ($R \frown S$) como la estructura APO $M = g(\beta_a, \beta_b, ...)$, tal que:

$$1.\forall\ \beta_a \in \{\beta_1, ..., \beta_n\} \Rightarrow \exists\ \alpha_t, \alpha'_s \in Y, Y'\ respectivamente, \quad tal\ que \quad \beta_a = \alpha_t = \alpha'_s \tag{3.36}$$

$$2.\forall\ \alpha_t = \alpha'_s \in Y, Y'\ respectivamente \Rightarrow \exists\ \beta_a \in \{\beta_1, .., \beta_n\} \quad tal\ que\ \alpha_t \subseteq \beta_a\ \alpha'_s \subseteq \beta_a \tag{3.37}$$

donde α_t es una item-seq de Y y α'_s es item-seq de Y'.

Es decir, el acoplamiento es igual a la estructura generada por las *item-seqs* o subsecuencias coincidentes en Y e Y', eliminando aquellas que no sean maximales (ver Definición 3.2.6).

Ejemplo 3.2.6. Acoplamiento de AP-Seqs
Sea $X = \{$*lluvia, viento, sol, tormenta, nube, calor, brisa, nieve*$\}$
Sea $R = g(Y)$, $Y = \{$*viento, sol, tormenta, lluvia, nube, calor*$\}$
Sea $S = g(Y')$, $Y' = \{$*viento, sol, brisa, nube, calor*$\}$

Item-seqs coincidentes:
$\beta_a = (y_1, y_2) = (y'_1, y'_2) = \{viento, sol\}$
$\beta_b = (y_4, y_5) = (y'_4, y'_5) = \{nube, calor\}$
Acoplamiento:
$\Rightarrow M = R \frown S = g(\{viento, sol\}, \{nube, calor\}) = (\{viento, sol\},$
$\{nube, calor\}, \{viento\}, \{sol\}, \{nube\}, \{calor\})$

Definición 3.2.9. Unión de AP-Seqs
Sean $R = g(Y)$ y $S = g(Y')$ AP-Seqs con conjunto referencial X. Se define la

unión de R y S como la estructura APO, U, generada por Y e Y'

$$U = g(R \cup S) = g(Y, Y') \tag{3.38}$$

Ejemplo 3.2.7. Unión de AP-Seqs
Sea $X = \{lluvia, viento, sol, tormenta, nube, calor, brisa, nieve\}$
Sea $R = g(Y), Y = \{tormenta, sol, lluvia, nube\}$
Sea $S = g(Y'), Y' = \{viento, sol, lluvia, tormenta\}$
$\Rightarrow g(Y \cup Y') = g(\{tormenta, sol, lluvia, nube\}, \{viento, sol, lluvia, tormenta\})$
$= (\{tormenta, sol, lluvia, nube\}, \{viento, sol, lluvia, tormenta\}, \{tormenta,$
$sol, lluvia\}, \{sol, lluvia, nube\}, \{viento, sol, lluvia\}, \{sol, lluvia, tormenta\},$
$\{tormenta, sol\}, \{sol, lluvia\}, \{lluvia, nube\}, \{viento, sol\}, \{lluvia, tormenta\},$
$\{tormenta\}, \{sol\}, \{lluvia\}, \{nube\}, \{viento\})$

En las operaciones de la estructura-AP (Sección 2.2.1), podíamos distinguir entre "Subconjunto-AP Inducido" y "Subestructura-AP inducida". Para la estructura APO sin embargo, siempre se hablará de "Subestructura APO Inducida", si bien esta subestructura puede proceder de una AP-Seq o de una estructura APO. Esto es así porque el resultado del acoplamiento de una AP-Seq con una secuencia, no dará una subAP-Seq, sino un conjunto de subAP-Seqs, es decir, una estructura.

La operación subestructura APO de una AP-Seq inducida por una secuencia se encargará de obtener el conjunto APO particular que se genera al acoplar el retículo global de la AP-Seq con una secuencia dada.

Definición 3.2.10. Subestructura APO de una AP-Seq Inducida por una Secuencia
Sea $R = g(Y)$ una AP-Seq con conjunto referencial X y sea $Y' \subseteq X$, diremos que S es la subestructura APO de la AP-Seq R inducida por Y' si y sólo si:

$$S = g(Y \frown Y') \tag{3.39}$$

es decir, S es una subestructura APO 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 Definición 3.2.8). Como hemos dicho, el acoplamiento entre AP-Seqs da lugar a estructuras APO, por ello obtenemos una subestructura APO en lugar una subAP-Seq .

El Ejemplo 3.2.8 clarifica estas ideas.

Ejemplo 3.2.8. Subestructura APO de una AP-Seq Inducida por una Secuencia
Sea $X = \{lluvia, viento, sol, tormenta, nube, calor, brisa, nieve\}$
Sea $R = g(Y)$, $Y = \{viento, sol, tormenta, lluvia, nube, calor\}$
Sea $Y' = \{viento, brisa, lluvia, nube\}$

Item-seqs coincidentes:
$\beta_a = (y_1) = (y'_1) = \{viento\}$
$\beta_b = (y_4, y_5) = (y'_3, y'_4) = \{lluvia, nube\}$
Subestructura APO inducida:
$\Rightarrow\ S = g(Y \frown Y') = g(\{viento\}, \{lluvia, nube\}) = (\{lluvia, nube\}, \{lluvia\}, \{nube\}, \{viento\})$

Al tener más de una secuencia generadora, S no es una subAP-Seq, sino una subestructura.

Igual ocurrirá con la superestructura APO inducida: la unión de dos secuencias dará lugar a una estructura, por lo que nunca hablaremos de "SuperAP-Seq Inducida" como hablábamos de "Superconjunto-AP Inducido" cuando veíamos las operaciones de los conjuntos-AP (Sección 2.2.1).

Definición 3.2.11. Superestructura APO de una AP-Seq Inducida por una Secuencia
Sea $R = g(Y)$ una AP-Seq con conjunto referencial X y sea $Y' \subseteq X$ diremos que V es la superestructura APO de la AP-Seq R inducida por Y' si y sólo si:

$$V = g(Y \cup Y') \tag{3.40}$$

La superestructura V será la generada por la unión de Y e Y', siendo la unión la que se ha visto en la Definición 3.2.9.

Ejemplo 3.2.9. Superestructura APO de una AP-Seq Inducida por una Secuencia
Sea $X = \{lluvia, viento, sol, tormenta, nube, calor, brisa\}$
Sea $R = g(Y)$, $Y = \{viento, sol\}$
Sea $Y' = \{viento, brisa, lluvia\}$
$\Rightarrow\ V = g(\{viento, sol\}, \{viento, brisa, lluvia\})$

Vemos que V es una estructura al tener más de una secuencia generadora.

Omitiremos la definición de otras propiedades y operaciones de la estructura APO como "inclusión de estructuras APO", "subestructura y superestructura de una estructura APO inducidas por una secuencia" y "unión e intersección de estructuras APO", para no hacer demasiado repetitiva la lectura de esta memoria, ya que todas estas operaciones se definirán igual para la estructura WAPO (Sección 3.3), sólo que más completas en esta última por la inclusión de la ponderación.

3.3 Estructura WAPO

Previamente a la definición de estructura WAPO, estableceremos la de estructura monotérmino ponderada, con el fin de poder compararlas. Esta última se corresponde con una estructura WAPO donde todas las *item-seqs* son de nivel uno.

3.3.1 Estructura Monotérmino Ponderada

Las estructuras monotérmino ponderadas, son estructuras monotérmino (ver Sección 3.2.2) en las que cada conjunto estará formado por una secuencia ponderada de nivel uno. Estos conjuntos se denominan "Conjuntos Monotérmino Ponderados."

Definición 3.3.1. Estructura Monotérmino Ponderada
Sea $X = \{x_1, x_2, ..., x_n\}$ *un conjunto referencial de items y* $\widetilde{S} = \{\widetilde{A}, \widetilde{B}, ...\} \subseteq X$ *un conjunto de conjuntos monotérmino ponderados, tales que:*

$$\forall \widetilde{A}, \widetilde{B} \in \widetilde{S}; \widetilde{A} \nsubseteq \widetilde{B}, \widetilde{B} \nsubseteq \widetilde{A} \tag{3.41}$$

llamaremos estructura monotérmino generada por $\widetilde{S}$, $\widetilde{E} = g(\widetilde{A}, \widetilde{B}, ...)$, *al conjunto de conjuntos monotérmino cuyos generadores son* $\widetilde{A}$, $\widetilde{B}, ...$ *siendo* $\widetilde{A} = \{A_i, \omega_i; 1 \leq i \leq n\}$

Ejemplo 3.3.1. Estructura Monotérmino Ponderada
Sea $X = \{lluvia, viento, sol, tormenta, nube, calor, brisa\}$
Sea $\widetilde{A} = \{A, (\omega_A)\} = \{\{lluvia, (5)\}, \{sol, (10)\}\}$
Sea $\widetilde{B} = \{B, (\omega_B)\} = \{sol, (10)\}$
Sea $\widetilde{C} = \{C, (\omega_C)\} = \{\{calor, (12)\}, \{sol, (10)\}, \{brisa, (8)\}\}$
Sea $\widetilde{D} = \{D, (\omega_D)\} = \{brisa, (8)\}$
$\Rightarrow \widetilde{E} = g(\widetilde{A}, \widetilde{B}, \widetilde{C}, \widetilde{D}) = (\{lluvia, (5)\}, \{sol, (10)\}, \{calor, (12)\}, \{brisa, (8)\})$

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

En la Figura 3.5 podemos ver la *tag cloud* de la estructura monotérmino ponderada del Ejemplo 3.3.1.

Figura 3.5: Estructura monotérmino

Para definir la estructura WAPO, comenzaremos definiendo las **"AP-Seqs Ponderadas"** o **"Conjuntos WAPO"**.

3.3.2 Definición, Propiedades y Operaciones de las AP-Seqs Ponderadas

Las estructuras WAPO se componen por AP-Seqs ponderadas y éstas, a su vez, por *item-seqs* ponderadas.

Definición 3.3.2. *Item-seq* ponderada de una AP-Seq
Sea R = g(Y) una AP-Seq con conjunto referencial de items X. Diremos que $\widetilde{\alpha_t} \subseteq Y$ es una item-seq ponderada del conjunto Y si y sólo si:

$$\widetilde{\alpha_t} = [\alpha_t, \omega_t] \tag{3.42}$$

donde α_t es la item-seq definida en la Definición 3.2.6 y ω_t su peso.

Como las *item-seqs* se extraerán del texto, el peso se corresponderá con su frecuencia de aparición en éste ($\omega_t \in \mathbb{N}$).

Propiedad 3.3.2.

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.

$$Si\ \alpha_1 \subseteq \alpha_2 \Rightarrow \omega_1 \geq \omega_2 \tag{3.43}$$

Ejemplo 3.3.2. *Item-seq* ponderada de una AP-Seq
Sea $X = \{lluvia, viento, sol, tormenta, nube, calor, brisa, nieve\}$
Sea $R = g(Y) = \{viento, lluvia, brisa\}$

$$\begin{aligned} \Rightarrow \quad & \widetilde{\alpha_1}(R) = \{viento, (8)\}, \widetilde{\alpha_2}(R) = [\{lluvia\}, (5)], \widetilde{\alpha_3}(R) = [\{brisa\}, (5)] \\ & \widetilde{\alpha_4}(R) = [\{viento, lluvia\}, (3)], \widetilde{\alpha_5}(R) = [\{lluvia, brisa\}, (4)] \\ & \widetilde{\alpha_6}(R) = [\{viento, lluvia, brisa\}, (2)] \end{aligned}$$

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

Definición 3.3.3. AP-Seq Ponderada o Conjunto WAPO
Sea R un conjunto APO, diremos que $\widetilde{R}$ es un conjunto WAPO si y sólo si:

$$R = \{\alpha_1, \alpha_2, ...\} \Rightarrow \widetilde{R} = \{\widetilde{\alpha_1}, \widetilde{\alpha_2}, ...\} \tag{3.44}$$

siendo $\widetilde{\alpha}_i$ una *item-seq* ponderada.

Propiedades y Operaciones de las AP-Seqs Ponderadas

A continuación se introduce la "Inclusión de AP-Seqs Ponderadas" y la "Subestructura WAPO de una AP-Seq Ponderada Inducida por una Secuencia", por revestir ambas especial importancia. El resto de operaciones que se vieron para las AP-Seqs sin ponderar (Sección 3.2.4), se definirían del mismo modo para las AP-Seqs ponderadas, pero considerando que las *item-seqs* tienen un peso asociado.

Definición 3.3.4. Inclusión de AP-Seqs Ponderadas
Sean $\widetilde{R} = \widetilde{g}(Y)$ *y* $\widetilde{S} = \widetilde{g}(Y')$ *dos AP-Seqs ponderadas con el mismo conjunto referencial de items* X*:*

$$\widetilde{R} \subseteq \widetilde{S} \Leftrightarrow Y \subseteq Y' \tag{3.45}$$

con

$$\omega_{\alpha_t}(\widetilde{R}) = \omega_{\alpha_t}(\widetilde{S}) \ \ \forall\, t = 1, 2, ... \tag{3.46}$$

El peso de las *item-seqs* será el mismo en $\widetilde{R}$ y en $\widetilde{S}$, ya que ambas proceden del mismo texto.

Ejemplo 3.3.3. Inclusión de AP-Seqs Ponderadas
Sea $X = \{lluvia, viento, sol, tormenta, nube, calor, brisa, nieve\}$
Sea $\widetilde{R} = \widetilde{g}(\{viento, lluvia\}) = ([\{viento, lluvia, \}(3)], [\{viento\}, (4)], [\{lluvia\}, (7)])$
Sea $\widetilde{S} = \widetilde{g}(\{viento, sol, lluvia\}) = ([\{viento, lluvia, sol\}, (2)], [\{viento, sol\}, (4)], [\{sol, lluvia\}, (5)], [\{viento\}, (4)], [\{lluvia\}, (7)], [\{sol\}, (6)]$

$$\Rightarrow \widetilde{R} \not\subseteq \widetilde{S}$$

La AP-Seq ponderada $\widetilde{R}$ no está incluida en la AP-Seq ponderada $\widetilde{S}$, ya que su secuencia generadora, {*viento, lluvia*}, no está incluida en la secuencia generadora de $\widetilde{S}$, {*viento, sol, lluvia*}, por no existir el mismo orden estricto de adyacencia entre los *items viento* y *lluvia*.

Definición 3.3.5. Subestructura WAPO de una AP-Seq Ponderada Inducida por una Secuencia

Sea $\widetilde{R} = \widetilde{g}(Y)$ diremos que $\widetilde{S}$ es la subestructura WAPO de una AP-Seq ponderada inducida por Y'si y sólo si:

$$\widetilde{S} = g(\widetilde{R} \frown Y') \tag{3.47}$$

donde

$$\omega_{\alpha_t}(\widetilde{S}) = \omega_{\alpha_t}(\widetilde{R}) \tag{3.48}$$

La secuencia Y' no es ponderada, ya que representa las *item-seqs* de una tupla tras la limpieza o los términos introducidos en una consulta. En este último caso, el acoplamiento representa las *item-seqs* en $\widetilde{R}$ que se recuperan con Y', por lo que el peso de las *item-seqs* tras el acoplamiento, será el mismo que tuvieran en $\widetilde{R}$.

Podemos ver la operación de acoplamiento de AP-Seqs (denotada por "$\frown$") en la Definición 3.2.8.

3.3.3 Definición, Propiedades y Operaciones de la Estructura WAPO

Pasemos a ver la definición de la estructura WAPO, que de forma equivalente a las estructuras WAP y APO, se definirá como un conjunto de AP-Seqs ponderadas.

Definición 3.3.6. Estructura WAPO

Sea $X = \{x_1, x_2, ..., x_n\}$ un conjunto referencial de items y $\widetilde{S} = \{\widetilde{A}, \widetilde{B}, ...\} \subseteq P(X)$ un conjunto de item-seqs frecuentes ponderadas de nivel mayor o igual que uno, siendo P(X) las partes de X y $\widetilde{A}, \widetilde{B}, ...$ AP-Seqs ponderadas tales que:

$$\forall\ \widetilde{A}, \widetilde{B} \in \widetilde{S};\ \ \widetilde{A} \not\subseteq \widetilde{B},\ \ \widetilde{B} \not\subseteq \widetilde{A}\ \ y\ \ \widetilde{B} \neq \widetilde{A} \tag{3.49}$$

Llamaremos estructura WAPO generada por $\widetilde{S}$, $\widetilde{E} = g(\widetilde{A}, \widetilde{B}, ...)$, al conjunto de AP-Seqs ponderadas cuyas secuencias generadoras son $\widetilde{A}, \widetilde{B}, ...$

Nota.- La AP-Seq $\widetilde{A}$ puede expresarse como $\widetilde{g}(A)$

Ejemplo 3.3.4. Estructura WAPO

$$
\begin{aligned}
\text{Sea } \widetilde{A} &= \widetilde{g}(\{lluvia, tormenta, nube\}) \\
&= ([\{lluvia, tormenta, nube\}, (4)], [\{lluvia, tormenta\}, (6)], [\{tormenta, \\
&\quad nube\}, (5)], [\{lluvia\}, (7)], [\{tormenta\}, (10)], [\{nube\}, (8)]) \text{ y} \\
\text{Sea } \widetilde{B} &= \widetilde{g}(\{lluvia, sol\} \\
&= ([\{lluvia, sol\}, (7)], [\{lluvia\}, (7)], [\{sol\}, (12)]), \text{ entonces :} \\
\widetilde{E} &= \widetilde{g}(\{lluvia, tormenta, nube\}, \{lluvia, sol\}) \\
&= ([\{lluvia, tormenta, nube\}, (4)], [\{lluvia, tormenta\}, (6)], [\{tormenta, \\
&\quad nube\}, (5)], [\{lluvia, sol\}, (7)], [\{lluvia\}, (7)], [\{tormenta\}, (10)], \\
&\quad [\{nube\}, (8)], [\{sol\}, (12)])
\end{aligned}
$$

Visualización de la Estructura WAPO en forma de *Tag Cloud*.

En la Figura 3.6 podemos ver la *tag cloud* de la estructura WAPO del Ejemplo 3.3.4.

Figura 3.6: Estructura WAPO

Propiedades y Operaciones de la Estructura WAPO

Empezaremos definiendo la operación de inclusión de estructuras WAPO. Una estructura WAPO $\widetilde{E_1}$ estará contenida en otra estructura WAPO $\widetilde{E_2}$ si todas las secuencias generadoras de $\widetilde{E_1}$ aparecen incluidas en alguna de las secuencias generadoras de $\widetilde{E_2}$.

Definición 3.3.7. Inclusión de Estructuras WAPO
Sean $\widetilde{E_1}$ y $\widetilde{E_2}$ dos estructuras WAPO con el mismo conjunto referencial de items X. Diremos que $\widetilde{E_1}$ está incluida en $\widetilde{E_2}$ ($\widetilde{E_1} \subseteq \widetilde{E_2}$) si y sólo si:

$$\begin{aligned} &\forall\ \widetilde{R}\ \textit{AP-Seq ponderada de}\widetilde{E_1} \Rightarrow \exists\ \widetilde{S}\ \textit{AP-Seq ponderada de}\widetilde{E_2}, \\ &tal\ que\ \widetilde{R} \subseteq \widetilde{S} (\textit{análogamente}\ R \subseteq S) \end{aligned} \tag{3.50}$$

siendo

$$\omega_{\alpha_t}(\widetilde{E_1}) = \omega_{\alpha_t}(\widetilde{E_2})\ \forall\ t = 1, 2, ... \tag{3.51}$$

Para aplicar esta propiedad, nos servimos de la Definición 3.3.4 de inclusión de AP-Seqs ponderadas.

Para encontrar la representación del TDA correspondiente a una tupla dada de la base de datos, utilizaremos la operación subestructura APO inducida. La subestructura WAPO inducida se definirá igual que ésta, pero considerando el peso y representará los resultados obtenidos con una consulta, donde el peso indicaría el número de tuplas recuperadas.

Definición 3.3.8. Subestructura WAPO Inducida
Sea la estructura WAPO $\widetilde{E} = g(\widetilde{A_1}, \widetilde{A_2}, ..., \widetilde{A_n})$ con conjunto referencial de items X e $Y \subseteq X$. Definiremos la sub-estructura WAPO de E, $\widetilde{E'}$ inducida por Y como:

$$\widetilde{E'} = \widetilde{E} \wedge Y = g(\widetilde{B_1}, \widetilde{B_2}, ..., \widetilde{B_m}) \tag{3.52}$$

donde

$$\begin{aligned} \forall\ \widetilde{B_i} \in \{\widetilde{B_1}, ..., \widetilde{B_m}\} \Rightarrow &\exists\ \widetilde{A_j} \in \{\widetilde{A_1}, ..., \widetilde{A_n}\} \\ &tal\ que B_i = A_j \frown Y \end{aligned} \tag{3.53}$$

$$\begin{aligned} \forall\ \widetilde{A_j} \in \{\widetilde{A_1}, ..., \widetilde{A_n}\} \Rightarrow &\exists\ \widetilde{B_i} \in \{\widetilde{B_1}, ..., \widetilde{B_m}\} \\ &tal\ que A_j \frown Y \subseteq B_i \end{aligned} \tag{3.54}$$

con

$$\omega_{\alpha_t}(\widetilde{E'}) = \omega_{\alpha_t}(\widetilde{E}) \tag{3.55}$$

$\widetilde{E'}$ es la estructura WAPO generada por el acoplamiento de Y con las secuencias generadoras de $\widetilde{E}$. Para ello, se hace el acoplamiento de cada *item-seq* de Y con todas las *item-seqs* generadas por $(A_1, A_2, ..., A_n)$ de $\widetilde{E}$, tal como se indica en la Definición 3.2.8. En las AP-Seqs B_i de $\widetilde{E'}$ irán sólo las *item-seqs* que no estén completamente incluidas en otra AP-Seq de B_i, ya que si no serían redundantes y la estructura WAPO resultante no estaría constituida únicamente por secuencias maximales.

No tiene sentido que la secuencia Y sea ponderada, ya que normalmente representa los elementos introducidos en la consulta.

Ilustraremos lo anterior con el Ejemplo 3.3.5.

Ejemplo 3.3.5. Subestructura WAPO Inducida
Sea $X = \{lluvia, viento, sol, tormenta, nube, calor, brisa, nieve\}$

$$\begin{aligned}
Sea\ \widetilde{E} &= \widetilde{g}(\{viento, sol, nube\}, \{lluvia, nube\}) \\
&= ([\{viento, sol, nube\}, (4)], [\{viento, sol\}, (6)], [\{sol, nube\}, (4)], \\
&\quad [\{lluvia, nube\}, (2)], [\{viento\}, (7)], [\{sol\}, (8)], [\{nube\}, (7)], \\
&\quad [\{lluvia\}, (3)] \\
Sea\ \widetilde{Y} &= g(\{viento, lluvia, nube\}) \\
\Rightarrow \widetilde{E} \wedge Y &= \widetilde{g}(\{viento\}, \{lluvia, nube\}) \\
&= ([\{viento\}, (7)], [\{lluvia, nube\}, (2)], [\{lluvia\}, (3)], [\{nube\}, (7)])
\end{aligned}$$

Vemos que, aunque la *item-seq* $\{nube\}$ también se obtendría del acoplamiento de $\widetilde{Y}$ con la primera secuencia generadora de $\widetilde{E}$, $A_1 = \{viento, sol, nube\}$, ésta no se considera como secuencia generadora en E' por no ser maximal en esta estructura.

El peso de las *item-seqs* en la sub-estructura WAPO inducida es el que tenían en la estructura WAPO original. La subestructura WAPO inducida representaría el resultado de la consulta, la AP-Seq Y contendría los términos de ésta y la estructura WAPO simbolizaría la información contenida en el atributo textual consultado.

De forma análoga a la definición anterior, se define ahora la operación superestructura WAPO inducida. Esta operación permite obtener la estructura WAPO generada por la unión de un conjunto dado con una estructura WAPO determinada.

Definición 3.3.9. Superestructura WAPO Inducida
Sea la estructura WAPO $\widetilde{E} = g(\widetilde{A_1}, \widetilde{A_2}, ..., \widetilde{A_n})$ *con conjunto referencial de items* X *e* $Y \subseteq X$. *Definiremos la superestructura WAPO de* E, $\widetilde{E'}$ *inducida por* $\widetilde{Y}$ *como:*

$$\widetilde{E'} = \widetilde{E} \vee \widetilde{Y} = g(\widetilde{B_1}, \widetilde{B_2}, ..., \widetilde{B_m}) \tag{3.56}$$

donde

$$\begin{aligned} \forall\, \widetilde{B_i} \in \{\widetilde{B_1}, ..., \widetilde{B_m}\} \Rightarrow & \exists\, \widetilde{A_j} \in \{\widetilde{A_1}, ..., \widetilde{A_n}\} \\ & tal\ que\ B_i = A_j \cup Y \end{aligned} \tag{3.57}$$

$$\begin{aligned} \forall\, \widetilde{A_j} \in \{\widetilde{A_1}, ..., \widetilde{A_n}\} \Rightarrow & \exists\, \widetilde{B_i} \in \{\widetilde{B_1}, ..., \widetilde{B_m}\} \\ & tal\ que\ A_j \cup Y \subseteq B_i \end{aligned} \tag{3.58}$$

donde

$$\omega_{\alpha_t(\widetilde{E'})} = \begin{cases} \omega_{\alpha_t(\widetilde{E})} & si\ \alpha_t(\widetilde{E'}) \in (\widetilde{E} - \widetilde{Y}), \\ \omega_{\alpha_t(\widetilde{Y})} & si\ \alpha_t(\widetilde{E'}) \in (\widetilde{Y} - \widetilde{E}) \\ \omega_{\alpha_t(\widetilde{E})} + \omega_{\alpha_t(\widetilde{Y})} - \omega_{\alpha_t(\widetilde{E} \cap \widetilde{Y})} & si\ \alpha_t(\widetilde{E'}) \in \widetilde{E}, \widetilde{Y} \end{cases} \tag{3.59}$$

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

En este caso, las *item-seqs* que estaban presentes tanto en $\widetilde{E}$ como en $\widetilde{Y}$, aparecerán en $\widetilde{E'}$ con peso igual a la suma de los pesos que presentaran en $\widetilde{E}$ y en $\widetilde{Y}$ menos el peso de las *item-seqs* presentes en la intersección, que serán aquellas provenientes de entradas repetidas en $\widetilde{E}$ e $\widetilde{Y}$. Las *item-seqs* que estuvieran sólo en $\widetilde{E}$ o $\widetilde{Y}$ conservarán su peso.

Al igual que en la superestructura WAP inducida (ver Definición 3.1.8), aquí sí tiene sentido que la $\widetilde{Y}$ sea ponderada, puesto que estas operaciones representan la agregación de información a la base de datos.

Ejemplo 3.3.6. Superestructura WAPO Inducida
Sea $X = \{lluvia, viento, sol, tormenta, nube, calor, brisa, nieve\}$

$$\begin{aligned}
Sea\ \widetilde{E} &= \widetilde{g}(\{viento, sol, nube\}) \\
&= ([\{viento, sol, nube\}, (4)], [\{viento, sol\}, (6)], [\{sol, nube\}, (4)], \\
&\quad [\{viento\}, (7)], [\{sol\}, (8)], [\{nube\}, (7)]), \\
Sea\ \widetilde{Y} &= \widetilde{g}(\{viento, sol, lluvia\}) \\
&= ([\{viento, sol, lluvia\}, (2)], [\{viento, sol\}, (3)], [\{sol, lluvia\}, (3)], \\
&\quad [\{viento\}, (5)], [\{sol\}, (4)], [\{lluvia\}, (3)]) \\
\Rightarrow \widetilde{E} \vee \widetilde{Y} &= \widetilde{g}(\{viento, sol, nube\}, \{viento, sol, lluvia\}) \\
&= ([\{viento, sol, nube\}, (4)], [\{viento, sol, lluvia\}, (2)], \\
&\quad [\{viento, sol\}, (8)], [\{sol, nube\}, (4)], [\{sol, lluvia\}, (3)] \\
&\quad [\{viento\}, (10)], [\{sol\}, (11)], [\{nube\}, (7)], [\{lluvia\}, (3)])
\end{aligned}$$

Si observamos por ejemplo la *item-seq* $\{viento, sol\}$, vemos que aparece tanto en $\widetilde{E}$ como en $\widetilde{Y}$ con pesos 6 y 3, respectivamente. El peso en la superestructura WAPO inducida $\widetilde{E'}$ igual a 8, indica que existe una elemento en $\widetilde{E}$ repetido en $\widetilde{Y}$ y que no se ha considerado en el cálculo del peso de $\widetilde{E'}$, para no contabilizarlo doblemente.

El acoplamiento entre dos estructuras WAPO nos dará los elementos comunes en dos atributos textuales.

Definición 3.3.10. Acoplamiento de Estructuras WAPO
Sean $\widetilde{E_1} = g(\widetilde{A_1}, \widetilde{A_2}, ..., \widetilde{A_n})$ *y* $\widetilde{E_2} = g(\widetilde{B_1}, \widetilde{B_2}, ..., \widetilde{B_m})$ *dos estructuras WAPO, se define el acoplamiento de* $\widetilde{E_1}$ *y* $\widetilde{E_2}$ *como el conjunto generado por:*

$$S = \widetilde{E_1} \frown \widetilde{E_2} = g(\widetilde{C_1}, \widetilde{C_2}, ..., \widetilde{C_h}) \tag{3.60}$$

verificando:

$$\begin{aligned}
&\forall \widetilde{C_i} \in \{\widetilde{C_1}, ..., \widetilde{C_h}\};\ \exists\ \widetilde{A_p} \in \{\widetilde{A_1}, ..., \widetilde{A_n}\} \\
&y\ \widetilde{B_q} \in \{\widetilde{B_1}, ..., \widetilde{B_m}\}\ /\ \widetilde{C_i} = \widetilde{A_p} \frown \widetilde{A_q}
\end{aligned} \tag{3.61}$$

con

$$\omega_{\alpha_t(E)} \leq \omega_{\alpha_t(\widetilde{E_1})}, \omega_{\alpha_t(\widetilde{E_2})} \tag{3.62}$$

El peso o frecuencia de las *item-seqs* resultantes tras el acoplamiento, siempre será menor o igual al peso que tuvieran en sus estructuras de origen. Igualmente, el peso o frecuencia de las entradas comunes en dos atributos textuales, será menor o igual al peso que tengan en estos atributos.

Ejemplo 3.3.7. Acoplamiento de Estructuras WAPO
Sea $X = \{lluvia, viento, sol, tormenta, nube, calor, brisa, nieve\}$

$$\begin{aligned}
Sea\ \widetilde{E_1} &= \widetilde{g}(\{viento, sol, nube\}, \{lluvia, nube\}) \\
&= ([\{viento, sol, nube\}, (4)], [\{viento, sol\}, (6)], [\{sol, nube\}, (4)], \\
&\quad [\{lluvia, nube\}, (2)], [\{viento\}, (7)], [\{sol\}, (8)], [\{nube\}, (7)], \\
&\quad [\{lluvia\}, (3)] \\
Sea\ \widetilde{E_2} &= \widetilde{g}(\{viento, lluvia, nube\}) \\
&= ([\{viento, lluvia, nube\}, (3)], [\{viento, lluvia\}, (4)], [\{lluvia, nu- \\
&\quad be\}, (3)\}], [\{viento\}, (5)], [\{lluvia\}, (6)], [\{nube\}, (5)]) \\
\Rightarrow \widetilde{E} \frown \widetilde{E_2} &= \widetilde{g}(\{viento\}, \{lluvia, nube\}) \\
&= ([\{viento\}, (4)], [\{lluvia, nube\}, (1)], [\{lluvia\}, (2)], [\{nube\}, (2)])
\end{aligned}$$

La operación de unión de dos estructuras WAPO representa la unión de dos atributos textuales en una sola columna de las base de datos.

Definición 3.3.11. Unión de Estructuras WAPO
Sean $\widetilde{E_1} = g(\widetilde{A_1}, \widetilde{A_2}, ..., \widetilde{A_n})$ *y* $\widetilde{E_2} = g(\widetilde{B_1}, \widetilde{B_2}, ..., \widetilde{B_m})$ *dos estructuras WAPO, se define la unión como:*

$$S = \widetilde{E_1} \cup \widetilde{E_2} = g(\widetilde{C_1}, \widetilde{C_2}, ..., \widetilde{C_l}) \tag{3.63}$$

verificando:

$$i) \quad \forall \widetilde{C_i} \in \{\widetilde{C_1}, ..., \widetilde{C_l}\} \Rightarrow (\ \exists\ \widetilde{A_j}/\widetilde{C_i} = \widetilde{A_j})\ o\ (\exists\ \widetilde{B_l}/\widetilde{C_i} = \widetilde{B_l}) \tag{3.64}$$

$$ii) \quad \forall\ \widetilde{A_j} \in \{\widetilde{A_1}, ..., \widetilde{A_n}\} \Rightarrow \exists\ \widetilde{C_i}/\widetilde{A_j} \subseteq \widetilde{C_i} \tag{3.65}$$

$$iii) \quad \forall\ \widetilde{B_l} \in \{\widetilde{B_1}, ..., \widetilde{B_m}\} \Rightarrow \exists\ \widetilde{C_i}/\widetilde{B_l} \subseteq \widetilde{C_i} \tag{3.66}$$

donde

$$\omega_{I_t(\widetilde{E'})} = \begin{cases} \omega_{\alpha_t(\widetilde{E_1})} & si\ \alpha_t(\widetilde{S}) \in (\widetilde{E_1} - \widetilde{E_2}), \\ \omega_{\alpha_t(\widetilde{E_2})} & si\ \alpha_t(\widetilde{S}) \in (\widetilde{E_2} - \widetilde{E_1}), \\ \omega_{\alpha_t(\widetilde{E_1})} + \omega_{\alpha_t(\widetilde{E_2})} - \omega_{\alpha_t(\widetilde{E_1} \cap \widetilde{E_2})} & si\ \alpha_t(\widetilde{S}) \in \widetilde{E_1}, \widetilde{E_2} \end{cases} \quad (3.67)$$

Ejemplo 3.3.8. Unión de Estructuras WAPO
Sea $X = \{lluvia, viento, sol, tormenta, nube, calor, brisa, nieve\}$
Sea $\widetilde{E_1} = \widetilde{g}(Y), Y = \{\{viento, sol\}, \{lluvia\}\}$
Sea $\widetilde{E_2} = \widetilde{g}(Y'), Y' = \{\{calor, sol\}, \{viento\}\}$
$\Rightarrow \widetilde{E_1} \cup \widetilde{E_2} = \widetilde{g}(\{viento, sol\}, \{calor, sol\}, \{lluvia\})$

Como se puede ver, en la unión de $\widetilde{E_1}$ y $\widetilde{E_2}$ no se ha incluido la secuencia $\{viento\}$, debido a que ésta forma parte de otra secuencia maximal, la $\{viento, sol\}$.

3.4 Consultas

Teniendo en cuenta que las estructuras WAP y WAPO se obtienen a partir de los términos relevantes del atributo textual que se está procesando, se puede decir que contendrán la mayoría de los términos representativos que aparecen en dicho atributo. De aquí podemos afirmar entonces, que dichas estructuras constituyen el dominio activo del atributo del que se obtienen, cada una con sus peculiaridades específicas.

La idea es que el usuario expresa sus requerimientos como secuencias de términos, para ser consultados sobre atributos textuales en la base de datos. Dichos atributos estarán representados por sus estructuras WAP y WAPO particulares.

Tras realizar la intersección o el acoplamiento de la estructura particular del atributo textual con la secuencia de consulta, se obtiene la subestructura WAP o WAPO inducida, según corresponda, y esta subestructura representa la información que se recupera con la consulta.

Por otra parte, también se calculará el TDA de cada tupla, que nos ayudará para saber qué tuplas exactamente son las que se recuperan. Para obtener el TDA se realiza la intersección o el acoplamiento entre la estructura de conocimiento y el texto

corto modificado que se obtiene para la tupla a la salida del proceso de limpieza de datos, o lo que es lo mismo, se calcula la subestructura-AP o APO inducida por el conjunto o secuencia formada por los elementos de esa tupla, una vez que el texto está limpio. Como el cálculo de este TDA se realiza para cada tupla en concreto, no tiene sentido hacerlo con las estructuras ponderadas, ya que el TDA de cada tupla es independiente del de las demás y no conlleva peso alguno.

Así mismo, es posible que el usuario consulte directamente sobre el TDA de las tuplas en lugar de sobre la subestructura-AP inducida.

De esta forma, dos tipos diferentes de consultas pueden ser resueltas:

1. **Consultas sobre la estructura o dominio activo**

   El usuario puede dar una lista inicial de términos en su consulta, sin tener conocimiento alguno sobre el vocabulario de la estructura WAP o WAPO, mas que por los términos visualizados en la *tag cloud*. En este caso, puede preguntar inicialmente sobre el dominio activo del atributo, representado por su estructura particular, buscando los términos que se acoplan con su lista inicial. Si está interesado en que para el cálculo de este dominio se haya discriminado según el orden estricto de adyacencia de los términos, preguntaría por el dominio activo de la estructura WAPO, de no ser así, preguntaría por el dominio activo de la estructura WAP.

   Cuando se conozcan los términos sobre los que se va a consultar, se realiza un acoplamiento de los mismos con la estructura de conocimiento. El usuario tiene la opción de elegir, además de la estructura con la que se realiza el acoplamiento (WAP o WAPO), que el acoplamiento sea fuerte o débil (se verá la definición de acoplamiento fuerte y débil en la Sección 3.4.1). Si elige acoplamiento fuerte, se realizará un acoplamiento con los conjuntos o secuencias generadoras que incluyan completamente los términos de la consulta, descartando las demás. Si elige un acoplamiento débil, se realizará un acoplamiento con todos los conjuntos o secuencias generadoras de la estructura que incluyan algún término de la consulta. En este caso, el usuario puede captar nuevos términos relacionados con su búsqueda que se encuentran almacenados en la estructura de conocimiento, para en un paso posterior, preguntar por estos términos a través del acoplamiento fuerte.

2. **Consultas sobre cada tupla de la base de datos**

   En este tipo de consultas el usuario podría dar una lista inicial de términos para preguntar directamente sobre el TDA particular de cada tupla, con lo que los procedimientos de acoplamiento se invocarían directamente sobre dichas estructuras.

No olvidemos que la *tag cloud*, aparte de representar el contenido de la información, también actúa como asistente para la consulta, por lo que no siempre la forma de consultar será introduciendo los términos, sino que el usuario también puede directamente pulsar en una de las etiquetas de la *tag cloud* y los términos de dicha etiqueta actuarán como términos de consulta.

En el esquema de la Figura 3.7, se resumen las distintas opciones de consulta que tiene el usuario.

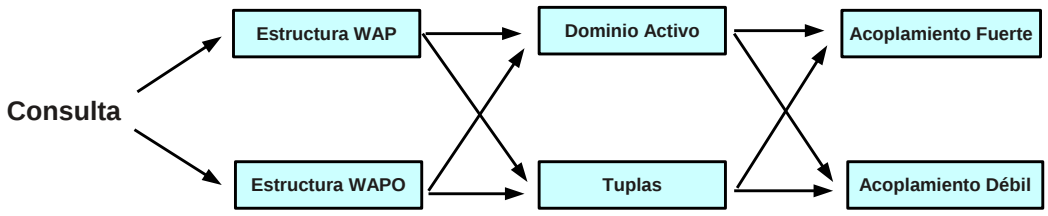


Figura 3.7: Esquema de las distintas opciones de consulta

3.4.1 Acoplamiento de Secuencias con Estructuras APO: Acoplamientos Fuerte y Débil

Consideremos la estructura APO como tipo de dato. Para ver el acoplamiento cuando el tipo de dato es la estructura-AP referir a la Sección 2.2.2.

Tomamos la estructura APO como tipo de dato en lugar de la estructura WAPO, porque para ver si existe acoplamiento entre una estructura y un conjunto o secuencia, sólo hace falta mirar si los términos del conjunto se encuentran en alguno de los generadores de la estructura, independientemente de su peso en ésta. Aún así,

cuando calculemos la subestructura que se genera tras el acoplamiento, si es conveniente tener el cuenta el peso en las *item-seqs*, que será el mismo que tuvieran en la estructura WAPO particular del atributo del que proceden. Este peso, nos aportaría información acerca del número de tuplas que se recuperan.

Dicho esto, cada atributo textual de una base de datos sobre la que consulta el usuario, tendrá su estructura APO particular. Y supongamos que el usuario expresa sus requerimientos a través de secuencias de términos, estas secuencias puede escribirlas el mismo usuario o pueden ser etiquetas representadas en la *tag cloud* particular del atributo. La información recuperada con este tipo de consulta nos vendrá dada a través del acoplamiento de la secuencia de términos consulta y la estructura APO, por lo que resulta necesario establecer algunos tipos de acoplamientos para satisfacer las consultas que se pueden realizar sobre dichas estructuras. Dos tipos distintos de acoplamiento que aparecen directamente, uno cuando se desea encontrar todos los términos de la consulta, el acoplamiento fuerte, y otro cuando se desea encontrar al menos uno de los términos que se están buscando, el acoplamiento débil.

Definición 3.4.1. Acoplamiento Fuerte
Sea la estructura APO $E = g(A_1, A_2, ..., A_n)$ con conjunto referencial de items X e $Y \subseteq X$. Se define el acoplamiento fuerte entre Y y E como la operación lógica:

$$Y \odot E = \begin{cases} verdadero & \textit{si } \exists\, A_i \in \{A_1, ..., A_n\} / Y \subseteq A_i \\ falso & \textit{en otro caso} \end{cases} \tag{3.68}$$

El acoplamiento fuerte será, por lo tanto, el que tendrá una secuencia Y con la estructura APO E cuando dicha secuencia aparezca completamente incluida en alguna de las secuencias generadoras de E, manteniendo la relación de orden entre los elementos.

Definición 3.4.2. Acoplamiento Débil
Sea la estructura APO $E = g(A_1, A_2, ..., A_n)$ con conjunto referencial de items X e $Y \subseteq X$. Se define el acoplamiento débil entre Y y E como la operación lógica:

$$Y \oplus E = \begin{cases} verdadero & \textit{si } \exists\, A_i \in \{A_1, ..., A_n\} / Y \frown A_i \neq \emptyset \\ falso & \textit{en otro caso} \end{cases} \tag{3.69}$$

El acoplamiento débil será, por lo tanto, el que tendrá una secuencia Y con la estructura APO E cuando dicha secuencia aparezca parcialmente incluida en alguna de las secuencias generadoras de E, es decir, el acoplamiento de Y con E sea distinto de cero.

Ejemplo 3.4.1. Acoplamientos Fuerte y Débil
$X = \{lluvia, viento, sol, tormenta, nube, calor, brisa, nieve\}$
Sea $E_1 = g(Y), Y = (\{viento, sol, brisa\}, \{lluvia, nube\}, \{calor, tormenta\})$
Sea $Y' = (\{sol, brisa\})$
Sea $Y'' = (\{brisa, viento, sol\})$

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

Vemos que Y' está completamente incluida en la primera secuencia generadora de E, por lo tanto tendrá un acoplamiento fuerte con E. Y'' en cambio, sólo está incluida parcialmente en esta primera secuencia, aunque aparezcan incluidos sus tres elementos. Esto ocurre porque la AP-Seq $\{$*viento, sol, brisa*$\}$ no es igual que la AP-Seq $\{$*brisa, viento, sol*$\}$.

A continuación ,en la Sección 3.4.2 presentamos un índice para cuantificar el grado de bondad de los acoplamientos, entendiendo que un acoplamiento será mejor cuanto más se ajuste la secuencia de consulta a la subestructura APO inducida del atributo consultado. Un mayor ajuste tendrá mayor respuesta y por lo tanto, su índice de acoplamiento será mayor.

3.4.2 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 todas las AP-Seqs que componen la estructura APO, sumando el grado de acoplamien-

to de cada AP-Seq, multiplicando por el número de AP-Seqs que acoplan y dividiendo entre el número total de AP-Seqs en la estructura APO.

2. Cálculo de índice por el máximo: Para su cálculo sólo se tiene en cuenta la AP-Seq perteneciente a la estructura APO con la que mejor se acopla el conjunto o secuencia de términos buscada, que será la secuencia donde la cardinalidad del acoplamiento sea mayor. Esta cardinalidad se calcula sumando la cardinalidad de todos los emparejamientos resultantes tras el acoplamiento de la AP-Seq con la estructura APO.

Sea la estructura APO $E = g(A_1, A_2, ..., A_m)$ con conjunto refencial de items X e $Y \subseteq X$. Cada secuencia A_i puede expresarse como un conjunto de subsecuencias o *item-seqs*: $A_i = (\alpha_1, \alpha_2, ...)$, donde $\alpha_1, \alpha_2, ...$ son todas las posibles subsecuencias, de cualquier longitud, que se pueden formar con los términos de la secuencia generadora A_i (ver Ejemplo 3.2.4).

LLamaremos δ_{ij} al valor del cardinal de la *item-seq* α_j resultante del acoplamiento del conjunto o la secuencia Y con A_i:

$$\delta_{ij} = card(\alpha_j | A_i \frown Y) \tag{3.70}$$

Y γ_i al valor del cardinal de A_i:

$$\gamma_i = card(A_i) \tag{3.71}$$

Veamos exactamente a qué nos referimos con esta notación en el Ejemplo 3.4.2.

Ejemplo 3.4.2. Cálculo de cardinales $\delta_{i,j}$ y γ_i
Sea $E = g(\{lluvia, tormenta, nieve\}, \{lluvia, nieve\})$
Sea $Y = \{lluvia, nieve\}$
Las item-seqs resultantes tras el acoplamiento de Y con la primera secuencia generadora de E, $A_1 = \{lluvia, tormenta, nieve\}$, son:
$\alpha_1 = \{lluvia\}, \alpha_2 = \{nieve\}$, entonces: $\delta_{1,1} = 1, \delta_{1,2} = 1$
Y las item-seqs resultantes tras el acoplamiento de Y con la segunda secuencia generadora de E, $A_2 = \{lluvia, nieve\}$, son:
$\alpha_3 = \{lluvia, nieve\}$, entonces: $\delta_{2,3} = 2$
siendo $\gamma_1 = 3$ y $\gamma_2 = 2$.

Aunque cuando hallamos la subestructura APO inducida, las *item-seqs* α_1 y α_2 se suprimen de la secuencias generadoras de la subestructura por estar contenidas en α_3, evitando así las redundancias, aquí las consideramos porque entendemos que para medir la bondad del acoplamiento de la secuencia Y con la estructura E hay que tener en cuenta todas las secuencias generadoras de E con las que se acopla la secuencia Y, teniendo mayor bondad de acoplamiento la secuencia Y que se acople con un mayor número de secuencias generadoras de E.

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

Aplicando la probabilidad de Laplace,

$$p(\delta_{ij}) = \frac{\text{casos favorables}}{\text{casos posibles}} \tag{3.72}$$

Como "casos posibles" consideraremos todas las posibles reordenaciones de los elementos de A_i independientemente del cardinal de las *item-seqs* del acoplamiento de Y con estos elementos. Con "casos favorables" consideraremos todas las posibles reordenaciones de los elementos de A_i que den como resultado que el cardinal de la *item-seq* α_j del acoplamiento de Y con A_i sea δ_{ij}.

Para hallar el número de casos posibles o posibles reordenaciones de los elementos de A_i, calculamos las permutaciones del cardinal de A_i:

$$\text{casos posibles} = \gamma_i! \tag{3.73}$$

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

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

Con lo que el número de casos favorables serán las permutaciones de los elementos de A_i menos los elementos de la *item-seq* α_j resultante del acoplamiento

de Y con A_i más uno (porque α_j contará como un sólo elemento):

$$\text{casos favorables} = [\gamma_i - (\delta_{ij} - 1)]! = (\gamma_i - \delta_{ij} + 1)! \tag{3.74}$$

Con lo que:

$$p(\delta_{ij}) = \frac{[\gamma_i - (\delta_{ij} - 1)]!}{\gamma_i!} = \frac{(\gamma_i - \delta_{ij} + 1)!}{\gamma_i!} \tag{3.75}$$

Y con esto tenemos la probabilidad de que el cardinal de la *item-seq* α_j del acoplamiento de Y con A_i sea δ_{ij}. Lo que pretendemos con esto es ponderar estas *item-seqs* resultantes 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, por ejemplo.

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 $p(\delta_{ij})$ para el cálculo del índice de bondad de acoplamiento, para dar de esta forma más peso a los acoplamientos de secuencias con mayor número de términos (mayor $\delta_{i,j}$).

Explicado lo anterior, ya podemos introducir el cálculo de los índices de acoplamiento fuerte y débil.

Definición 3.4.3. Cálculo del Índice de Acoplamiento Fuerte

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

$$\forall A_i \in \{A_1, ..., A_n\},\ \forall \alpha_j \in \{\alpha_1, ..., \alpha_m\}, \quad m_{ij}(Y) = \frac{\delta_{ij}\ \frac{\gamma_i!}{(\gamma_i - \delta_{ij}+1)!}}{\gamma_i} \tag{3.76}$$

$$S = \{i \in \{1, ..., n\}, j \in \{1, 2, ...\} |\ Y \subseteq A_i,\ \alpha_j \subseteq A_i\} \tag{3.77}$$

Con lo que se define:

1. *Índice fuerte por el promedio*

$$S(Y|E) = \frac{\sum_{i,j \in S} m_{ij}(Y)}{\sum_{i \in S} \gamma_i!} \cdot \frac{m}{n} \tag{3.78}$$

donde m es el número de AP-Seqs $\{A_1, ..., A_n\} \in E$ tales que $Y \subseteq A_i\ \forall i \in \{1, ..., n\}$ y n el número total de AP-Seqs en E.

2. *Índice fuerte por el máximo*

$$S(Y|E) = max_i \left(\frac{m_{ij}(Y)}{\gamma_i!} \right) ;\ i, j\ \in S \tag{3.79}$$

Definición 3.4.4. Cálculo del Índice de Acoplamiento Débil

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

$$\forall A_i \in \{A_1, ..., A_n\},\ \forall \alpha_j \in \{\alpha_1, ..., \alpha_m\}, \quad m_{ij}(Y) = \frac{\delta_{ij}\ \frac{\gamma_i!}{(\gamma_i - \delta_{ij}+1)!}}{\gamma_i} \tag{3.80}$$

$$W = \{i \in \{1, ..., n\}, j \in \{1, ..., m\} |\ Y \frown \widetilde{A_i} \neq \emptyset,\ \alpha_j \subseteq A_i\} \tag{3.81}$$

Con lo que se define:

1. *Índice débil por el promedio*

$$W(Y|E) = \frac{\sum_{i,j \in W} m_{ij}(Y)}{\sum_{i \in W} \gamma_i!} \cdot \frac{m}{n} \tag{3.82}$$

donde m es el número de AP-Seqs $\{A_1, ..., A_n\} \in E$ tales que $Y \frown A_i \neq \emptyset\ \forall i \in \{1, ..., n\}$ y n el número total de AP-Seqs en E.

2. *Índice débil por el máximo*

$$W(Y|E) = max_i \left(\frac{\sum_j m_{ij}(Y)}{\gamma_i!} \right) ;\ i, j \in W \tag{3.83}$$

Propiedades de los Índices

1. Todos los índices están comprendidos entre 0 y 1.

   Para un mismo acoplamiento:

2. El índice basado en el máximo será siempre mayor o igual que el índice por el promedio.

3. El índice débil de acoplamiento basado en el máximo será siempre mayor o igual que el índice fuerte de acoplamiento basado en el máximo.

4. El índice débil de acoplamiento basado en el promedio será siempre mayor o igual que el índice fuerte de acoplamiento basado en el promedio.

Ejemplo 3.4.3. Cálculo del Índice de Acoplamiento Fuerte
Sea $E = (\{1,2,3,4\},\{2,3,1\},\{3,5,4\})$
Sea $Y = (\{2,3,1\})$
$\Rightarrow Y \subseteq A_2 \Rightarrow A_2 \frown Y = \{2,3,1\}$

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

$$S(Y|E) = \frac{\frac{3\left(\frac{3!}{(3-3+1)!}\right)}{3}}{3!} \cdot \frac{1}{3} = \frac{1}{3}$$

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

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

Ejemplo 3.4.4. Cálculo del Índice de Acoplamiento Fuerte
Sea $E = (\{1,2,3,4,5\},\{1,2,3,6\},\{3,5,4\})$
Sea $Y = (\{1,2,3\})$
$\Rightarrow Y \subseteq A_1, A_2 \Rightarrow A_1 \frown Y = \{1,2,3\}, A_2 \frown Y = \{1,2,3\}$

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

$$\begin{aligned} S(Y|E) &= \frac{\frac{3\left(\frac{5!}{(5-3+1)!}\right)}{5} + \frac{3\left(\frac{4!}{(4-3+1)!}\right)}{4}}{5! + 4!} \cdot \frac{2}{3} \\ &= \frac{12+9}{5!+4!} \cdot \frac{2}{3} = 0.1 = \frac{1}{3} \end{aligned}$$

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

$$S(Y|E) = max(\tfrac{12}{5!}, \tfrac{9}{4!}) = max(0.1, 0.375) = 0.375$$

Ejemplo 3.4.5. Cálculo del Índice de Acoplamiento Débil
Sea $E = (\{1,2,3,4\},\{2,3,1\},\{3,5,4\})$
Sea $Y = (\{2,3,1\})$
$\Rightarrow E \frown Y = \{\alpha_1, \alpha_2, \alpha_3, \alpha_4\} = \{\{2,3\},\{1\},\{2,3,1\},\{3\}\}$

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

$$
\begin{aligned}
W(Y|E) &= \frac{\frac{2\left(\frac{4!}{(4-2+1)!}\right)}{4} + \frac{1\left(\frac{4!}{(4-1+1)!}\right)}{4} + \frac{3\left(\frac{3!}{(3-3+1)!}\right)}{3} + \frac{1\left(\frac{3!}{(3-1+1)!}\right)}{3}}{4! + 3! + 3!} \cdot \frac{3}{3} \\
&= \frac{2 + {}^{1}/_{4} + 3! + {}^{1}/_{3}}{4! + 3! + 3!} \\
&= \frac{2 + 0.25 + 6 + 0.33}{36} \\
&= 0.24
\end{aligned}
$$

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

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

3.5 Ejemplo Práctico

3.5.1 Comparación de las Estructuras Monotérmino Ponderada, WAP y WAPO

Supongamos que se extraen los siguientes titulares de noticias relacionadas con el empleo de una página de Internet (ver Tabla 3.1)

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 3.2 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 3.3.

	TITULARES
1	Faltan funcionarios en las oficinas de empleo
2	Deterioro del empleo de los funcionarios
3	El empleo de oficina el mejor valorado
4	Los funcionarios se abarrotan frente a la oficina de empleo
5	Disminución de empleo
6	Los funcionarios van a la oficina de empleo
7	El empleo de los funcionarios peligra
8	Disminución de empleo y sueldo en 2013
9	Trabajadores critican el empleo en su oficina
10	Empleo en verano y disminución del paro

Tabla 3.1: Muestra de titulares relacionados con el empleo

n	1^{er} item	2^{o} item	3^{er} item	4^{o} item
1	faltan	funcionarios	oficina	empleo
2	deterioro	empleo	funcionarios	
3	empleo	oficina	mejor	valorado
4	funcionarios	abarrotan	oficina	empleo
5	disminución	empleo		
6	funcionarios	van	oficina	empleo
7	empleo	funcionarios	peligra	
8	disminución	empleo	sueldo	2013
9	trabajadores	critican	empleo	oficina
10	empleo	verano	disminución	paro

Tabla 3.2: Titulares tras la limpieza de datos

Generación y Visualización de la Estructura Monotérmino Ponderada

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 o relativa, de cada término (ver Tabla 3.4) y la estructura monotérmino estará compuesta por aquellos términos que se consideren frecuentes según un soporte mínimo establecido.

Aunque casi todos los sitios web emplean la frecuencia absoluta para determinar cuando una palabra aparecerá el la *tag cloud*, nosotros hemos calculado también

n	Itemsets
1	{*faltan, funcionarios, oficina, empleo*}
2	{*deterioro, empleo, funcionarios*}
3	{*empleo, oficina, mejor, valorado*}
4	{*funcionarios, abarrotan, oficina, empleo*}
5	{*disminución, empleo*}
6	{*funcionarios, van, oficina, empleo*}
7	{*empleo, funcionarios, peligra*}
8	{*disminución, empleo, sueldo, 2013*}
9	{*trabajadores, critican, empleo, oficina*}
10	{*empleo, verano, disminución, paro*}

Tabla 3.3: Conjunto de *itemsets* tras la limpieza

n	Palabra	n_i	$f_i(\%) = n_i/n$
1	faltan	1	2.86
2	funcionarios	5	14.29
3	oficina	5	14.29
4	empleo	10	28.57
5	deterioro	1	2.86
6	mejor	1	2.86
7	valorado	1	2.86
8	abarrotan	1	2.86
9	disminución	3	8.57
10	van	1	2.86
11	peligra	1	2.86
12	sueldo	1	2.86
13	2013	1	2.86
14	trabajadores	1	2.86
15	critican	1	2.86
16	verano	1	2.86
17	paro	1	2.86
		n=35	$\sum_i f_i(\%) = 100$

Tabla 3.4: Frecuencia absoluta y relativa de cada término en el texto

la frecuencia relativa f_i en tanto por ciento, siendo n el número total de palabras en el texto limpio, en nuestro caso $n = 35$.

Los generadores de *tag cloud* que encontramos en la web [1] 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 supera al menos al 5 % ($f_i(\,\%) > 5\,\%$), lo que equivale en nuestro caso a una frecuencia absoluta superior o igual a 1.75 ($n_i \geq 1.75$). Como la frecuencia absoluta es un número entero, tomaremos los términos cuya n_i sea igual o mayor que 2 ($n_i \geq 2$).

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

Palabra (x)	F_i
funcionarios	5
oficina	5
empleo	10
disminución	3

Tabla 3.5: Términos en la estructura monotérmino

El cardinal de esta estructura monotérmino es:

$$card(M) = 4$$

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

$$\begin{aligned} \Rightarrow \widetilde{M} &= g(\widetilde{A}, \widetilde{B}, \widetilde{C}, \widetilde{D}) \\ &= \{[\{funcionarios\}, (5)], [\{oficina\}, (5)], [\{empleo\}, (10)], [\{dismi - \\ &\quad nución\}, (3)]\} \end{aligned}$$

[1] TagCrowd, TagCloudGenerator, etc.

En la Figura 3.8 se puede ver esta visualización en forma de *tag cloud*. Los distintos tamaños de fuente indican la frecuencia de los términos.

Figura 3.8: *Tag cloud* de la estructura monotérmino ponderada

Generación y Visualización de la Estructura WAP

Empezaremos generando los conjuntos-AP que conformarán la estructura-AP haciendo uso del algoritmo Apriori [Agr94], que veremos en el Capítulo 4. Luego introduciremos la ponderación en los *itemsets* para obtener la estructura WAP.

En la Sección 4.3 podemos ver el esquema de este algoritmo que a continuación se desarrolla paso a paso.

Algoritmo Apriori paso a paso para la generación de los *itemsets* frecuentes de 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 la fase C_1 o *itemsets* candidatos de nivel 1. Estos *itemsets* pueden verse en la Tabla 3.6. En este caso, N se corresponde con el número de tuplas o número total de transacciones ($N = 10$).

  A continuación, de entre estos *itemsets* candidatos, se seleccionan los *itemsets* frecuentes en lo que se conoce como fase L_1. En nuestro ejemplo, consideraremos que un *itemset* es frecuente cuando su soporte sea igual o superior al 20 % ($Supp(x) \geq 20\,\%$). Ver Tabla 3.7.

C_1	***Itemset*(x)**	n_i	$Supp(x)(\%) = n_i/N$
1	faltan	1	10
2	funcionarios	5	50
3	oficina	5	50
4	empleo	10	100
5	deterioro	1	10
6	mejor	1	10
7	valorado	1	10
8	abarrotan	1	10
9	disminución	3	30
10	van	1	10
11	peligra	1	10
12	sueldo	1	10
13	2013	1	10
14	trabajadores	1	10
15	critican	1	10
16	verano	1	10
17	paro	1	10

Tabla 3.6: Algoritmo Apriori en fase C_1. *Itemsets* candidatos

L_1	**Itemset(x)**	n_i	$Supp(x)(\%) = n_i/N$
1	{*funcionarios*}	5	50
2	{*oficina*}	5	50
3	{*empleo*}	10	100
4	{*disminución*}	3	30

Tabla 3.7: Algoritmo Apriori en fase L_1. *Itemsets* frecuentes.

- 2º **paso.-** Se generan los *itemsets* candidatos en C_2 o *itemsets* candidatos de nivel 2. Para ello, combinamos entre sí todos los *itemsets* frecuentes en L_1 y eliminamos aquellos en que aparezcan los mismos elementos en distinto orden. Ver Tabla 3.8.

  Es fácil darse cuenta de que en el cuadro de *itemsets* candidatos en la fase C_2 (Tabla 3.8), 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, ya que como en la estructura-AP

C_2	Itemset(x)	n_i	$Supp(x)(\%) = n_i/N$
1	{*funcionarios, oficina*}	3	30
2	{*funcionarios, empleo*}	5	50
3	{*funcionarios, disminución*}	0	0
4	{*oficina, empleo*}	5	50
5	{*oficina, disminución*}	0	0
6	{*empleo, disminución*}	3	30

Tabla 3.8: Algoritmo Apriori en fase C_2. *Itemsets* candidatos.

no importa el orden, estos *itemsets* se considera redundantes.

A continuación, se seleccionan de entre estos *itemsets* candidatos los *itemsets* frecuentes en lo que se denomina la fase L_2. En nuestro caso seleccionamos los *itemsets* candidatos cuyo soporte es superior o igual al 20 %. Ver Tabla 3.9.

L_2	Itemset(x)	n_i	$Supp(x)(\%) = n_i/N$
1	{*funcionarios, oficina*}	3	30
2	{*funcionarios, empleo*}	5	50
3	{*oficina, empleo*}	5	50
4	{*empleo, disminución*}	3	30

Tabla 3.9: Algoritmo Apriori en fase L_2. *Itemsets* frecuentes.

- 3^{er} **paso.-** Se generan los *itemsets* candidatos en C_3 o *itemsets* candidatos de nivel 3. Para ello, se combinan entre sí todos los *itemsets* frecuentes en L_2 que tengan algún *item* en común (en C_k se combinarían los *itemsets* de L_{k-1} que tuvieran $k-2$ *items* en común) y se eliminan los que resulten con los mismos elementos y orden diferente. Podemos ver los *itemsets* candidatos en la Tabla 3.10.

  De los *itemsets* candidatos en C_3 se seleccionan los *itemsets* frecuentes en L_3. Esto podemos verlo en la Tabla 3.11.

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

C_3	**Itemset(x)**	n_i	$Supp(x)(\%) = n_i/N$
1	{*funcionarios, oficina, empleo*}	3	30
2	{*funcionarios, empleo, disminución*}	0	0
3	{*oficina, empleo, disminución*}	0	0

Tabla 3.10: Algoritmo Apriori en fase C_3. *Itemsets* candidatos.

L_3	**Itemset(x)**	n_i	$Supp(x)(\%) = n_i/N$
1	{*funcionarios, oficina, empleo*}	3	30

Tabla 3.11: Algoritmo Apriori en fase L_3. *Itemsets* frecuentes.

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

$$A = \{funcionarios, oficina, empleo\}$$
$$B = \{empleo, disminución\}$$

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

$$T = g(A, B)$$

$$T = \{\{funcionarios, oficina, empleo\}, \{funcionarios, oficina\}, \{funcio - narios, empleo\}, \{oficina, empleo\}, \{funcionarios, disminución\}, \{funcionarios\}, \{oficina\}, \{empleo\}, \{disminución\}\}$$

Y el cardinal de la estructura-AP es $card(T) = 9$

Ponderamos la estructura-AP para obtener la estructura WAP:

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

$$\begin{aligned}\widetilde{T} \;=\; & \{[\{funcionarios, oficina, empleo\}, (3)], [\{funcionarios, oficina\}, (3)], \\ & [\{funcionarios, empleo\}, (5)], [\{oficina, empleo\}, (5)], [\{funcionarios, \\ & disminución\}, (3)], [\{funcionarios\}, (5)], [\{oficina\}, (4)], [\{empleo\}, \\ & (10)], [\{disminución\}, (3)]\}\end{aligned}$$

En la Figura 3.9 podemos ver cómo quedaría esta estructura WAP visualizada en forma de *Tag Cloud*.

Figura 3.9: *Tag cloud* de la estructura WAP

Generación y Visualización de la Estructura WAPO

Modificación del algoritmo Apriori para generar las *item-seqs* frecuentes de las AP-Seqs.

Para la generación de las *item-seqs* frecuentes que compondrán las AP-Seqs, se aplica una modificación del algoritmo Apriori empleado para la generación de los *itemsets* frecuentes que componían los conjuntos-AP. Podemos ver esta modificación en la Sección 4.3. Una vez que tengamos las AP-Seqs que constituyen la estructura APO, añadiremos la ponderación para obtener la estructura WAPO.

A continuación se detalla dicha modificación del algoritmo Apriori paso por paso:

- 1^{er} **paso.-** Se generan las *item-seqs* candidatas en C_1 o *item-seqs* candidatas de nivel 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 los Tablas C_1 y L_1 generadas para los conjuntos-AP. Ver Tablas 3.6 y 3.7.

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

  El cuadro que tendríamos, sería el mismo cuadro que para los *itemsets* candidatos en fase C_2 (Tabla 3.8), pero con el doble de elementos, es decir, por cada *itemset* del Tabla 3.8, tendríamos una *item-seq* con los mismos elementos del *itemset* en el mismo orden y otra *item-seq* con los elementos en orden inverso.

  Lo que hacemos es tomar cada uno de los *items* en L_1 e ir añadiendo los *items* restantes a su derecha, así obtendríamos todas las ordenaciones posibles de *itemsets* en C_2. Ver Tabla 3.12.

  Con lo que las *item-seqs* candidatas en C_2 son las que vemos en la Tabla 3.13.

  En la Tabla 3.14 vemos las *item-seqs* frecuentes en L_2 según el soporte mínimo establecido del 20 %.

- 3^{er} **paso.-** Para construir las *item-seqs* candidatas en C_3, combinamos las *item-seqs* en L_2 de la siguiente forma: tomamos cada *item-seq* y la enlazamos con todas las demás que empiecen por la subsecuencia de nivel 1 con que ésta termina. Si de esta forma, resulta una *item-seq* con dos *items* repetidos, la *item-seq* se elimina. Por ejemplo, la primera *item-seq* "{oficina, empleo}",

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

Tabla 3.12: Adyacencias de las *item-seqs* de nivel 1

C_2	Item-seq(x)	n_i	$Supp(x)(\%) = n_i/N$
1	{*funcionarios, oficina*}	1	10
2	{*funcionarios, empleo*}	0	0
3	{*funcionarios, disminución*}	0	0
4	{*oficina, funcionarios*}	0	0
5	{*oficina, empleo*}	3	30
6	{*oficina, disminución*}	0	0
7	{*empleo, funcionarios*}	2	20
8	{*empleo, oficina*}	2	20
9	{*empleo, disminución*}	0	0
10	{*disminución, funcionarios*}	0	0
11	{*disminución, oficina*}	0	0
12	{*disminución, empleo*}	2	20

Tabla 3.13: Algoritmo Apriori modificado para en fase C_2. *Item-seqs* candidatas

al terminar por la subsecuencia “empleo”, puede combinarse con las *item-seqs* “{empleo, funcionarios}” y “{empleo, oficina}”, que son las dos que empiezan por “empleo”. De la primera combinación resultaría la *item-seq* “{oficina, empleo, funcionarios}” y de la segunda combinación la *item-seq* “{oficina, empleo, oficina}”, pero como en esta segunda *item-seq* de nivel tres hay un *item* repetido, la *item-seq* no se considera.

L_2	**Item-seq(x)**	n_i	$Supp(x)(\%) = n_i/N$
1	{*oficina, empleo*}	3	30
2	{*empleo, funcionarios*}	2	20
3	{*empleo, oficina*}	2	20
4	{*disminución, empleo*}	2	20

Tabla 3.14: Algoritmo Apriori modificado en fase L_2. *Item-seqs* frecuentes

De igual forma, para construir las *item-seqs* candidatas en C_k, tomaríamos las subsecuencias de nivel $k-2$ con que terminan las *item-seqs* en L_{k-1} y las enlazamos con el resto de *item-seqs* en L_{k-1} que empiecen por la subsecuencia tomada.

En C_3 tendríamos las *item-seqs* del Tabla 3.15.

C_3	**Item-seq(x)**	n_i	$Supp(x)(\%) = n_i/N$
1	{*oficina, empleo, funcionarios*}	0	0
2	{*disminución, empleo, funcionarios*}	0	0
3	{*disminución, empleo, oficina*}	0	0

Tabla 3.15: Algoritmo Apriori modificado en fase C_3. *Item-seqs* candidatas

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

Una vez generadas las *item-seqs* frecuentes, estudiaríamos cuáles son las AP-Seqs que contienen esas *item-seqs*. Resultando las siguientes AP-Seqs:

$$A' = \{oficina, empleo\}$$
$$B' = \{empleo, funcionarios\}$$
$$C' = \{empleo, oficina\}$$
$$D' = \{disminución, empleo\}$$

Con lo que la estructura APO sería la generada por esas AP-Seqs:

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

$$\begin{aligned} E = \{&\{oficina, empleo\}, \{empleo, funcionarios\}, \{empleo, oficina\} \\ &\{disminución, empleo\}, \{oficina\}, \{empleo\}, \{funcionarios\}, \\ &\{disminución\}\} \end{aligned}$$

Siendo el cardinal de la estructura APO es $card(E) = 8$

La estructura WAPO sería la generada por las AP-Seqs ponderadas:

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

$$\begin{aligned} \widetilde{E} = \{&[\{oficina, empleo\}(3)], [\{empleo, funcionarios\}, (2)], [\{empleo, \\ &oficina\}, (2)], [\{disminución, empleo\}, (2)], [\{oficina\}, (5)], [\{em - \\ &pleo\}, (10)], [\{funcionarios\}, (5)], [\{disminución\}, (3)]\} \end{aligned}$$

En la Figura 3.10 podemos ver visualizada en forma de *tag cloud* esta estructura WAPO.

Figura 3.10: *Tag cloud* de la estructura WAPO

Comparación de las tres estructuras

	MONOTÉRMINO	WAP	WAPO
Notación	$\widetilde{M}$	$\widetilde{T} = g(\widetilde{A}, \widetilde{B})$	$\widetilde{E} = g(\widetilde{A'}, \widetilde{B'}, \widetilde{C'}, \widetilde{D'})$
Generadores	No tiene	$A = \{funcionarios, oficina, empleo\}$ $B = \{empleo, disminución\}$	$A' = \{oficina, empleo\}$ $B' = \{empleo, funcionarios\}$ $C' = \{empleo, oficina\}$ $D' = \{disminución, empleo\}$
Cardinal	4	9	8
Itemsets o ponderados/as	$[\{empleo\}, (10)]$ $[\{funcionarios\}, (5)]$ $[\{oficina\}, (5)]$ $[\{disminució\}, (3)]$	$[\{funcionarios, oficina, empleo\}, (3)]$ $[\{funcionarios, empleo\}, (5)]$ $[\{oficina, empleo\}, (5)]$ $[\{funcionarios, oficina\}, (3)]$ *Item-seqs* $[empleo, disminución\}, (3)]$ $[\{empleo\}, (10)]$ $[\{funcionarios\}, (5)]$ $[\{oficina\}, (5)]$ $[\{disminución\}, (3)]$	$[\{oficina, empleo\}, (3)]$ $[\{empleo, funcionarios\}, (2)]$ $[\{empleo, oficina\}, (2)]$ $[\{disminución, empleo\}, (2)]$ $[\{empleo\}, (10)]$ $[\{funcionarios\}, (5)]$ $[\{oficina\}, (5)]$ $[\{disminución\}, (3)]$

Tabla 3.16: Comparación de la estructura monotérmino ponderada, la estructura WAP y la estructura WAPO

Diferencias entre la Estructura WAP y la Estructura WAPO

A partir del Tabla (3.16) podemos deducir las siguientes diferencias entre la estructura WAP y la estructura WAPO:

- **El nivel de las secuencias generadoras de la estructura WAP es mayor o igual que el nivel de las secuencias generadoras de la estructura WAPO.** En el ejemplo vemos que, aunque la estructura WAP tiene un menor número de secuencias generadoras que la WAPO, el nivel de estas secuencias es mayor. De hecho, la secuencia generadora de máximo nivel, es de nivel tres para la estructura WAP y dos para la WAPO. Esto ocurre siempre así, ya que en la formación de los *itemsets* frecuentes de la estructura WAP no se respeta el orden estricto de adyacencia, al contrario que en la estructura WAPO, por lo

que esta formación es mucho menos restrictiva para la estructura WAP y sus *itemsets* alcanzarán frecuencias más altas dando lugar a combinaciones de mayor número de *items*. En caso de que se quisieran tener secuencias generadoras de mayor nivel en la estructura WAPO, bastaría con bajar el soporte.

- **El cardinal de la estructura WAP suele ser mayor que el de la estructura WAPO.** Esto ocurre como consecuencia de lo anterior y es el caso del ejemplo. Pero no siempre tiene por qué ser así, ya que existe un mayor número de combinaciones posibles de *item-seqs* en la estructura WAPO, al afectar el orden de los términos, y si un mayor número de estas combinaciones resulta frecuente, tendríamos un cardinal mayor en la estructura WAPO. Como normalmente el cardinal de la estructura WAP será mayor, tendremos un mayor número de *itemsets* en la visualización de la *tag cloud*.

- **El soporte y la frecuencia de las *item-seqs* de la estructura WAPO siempre es menor o igual que el soporte de los *itemsets* de la estructura WAP.** También esto es consecuencia de lo anterior. Debido a ello, un mismo conjunto de términos podrá verse con mayor tamaño en la visualización de la estructura WAP que en la visualización de la estructura WAPO.

- **El orden de los términos en los *itemsets* puede no ser el mismo en las estructuras WAP y WAPO.** Vemos, por ejemplo, que el *itemset* $\{empleo, dis-minución\}$ aparece en este orden de elementos en la estructura WAP, pero en orden inverso de elementos en la *item-seq* $\{disminución, empleo\}$ de la estructura WAPO. Esto es otra causa del orden estricto de adyacencia, tenido en cuenta en la estructura WAPO, ya que el orden más frecuente con que aparecen estos términos en el texto, es primero "disminución" y luego "empleo" y no al contrario.

  Fijémonos también en que tenemos dos *item-seqs* en la estructura WAPO con los mismos términos cambiados de orden, son las *item-seqs* $\{oficina, empleo\}$ y $\{empleo, oficina\}$. Esto se debe a que ambos términos se encuentran en el texto con los dos órdenes y ambas adyacencias resultan frecuentes según el soporte establecido. Esta especificación del orden nos ayuda a precisar y a refinar la búsqueda a través de la *tag cloud*.

"La consulta a través de la tag cloud *de la estructura WAP produce más resultados que a través de la* tag cloud *de la estructura WAPO, pero la mayoría de las veces estos resultados serán erróneos o poco precisos."*

- **La visualización de la estructura WAP suele ser más densa que la visualización de la estructura WAPO (Figura 3.11)**. Esto a veces dificulta la localización de términos y se produce el solapamiento semántico de los *itemsets*.

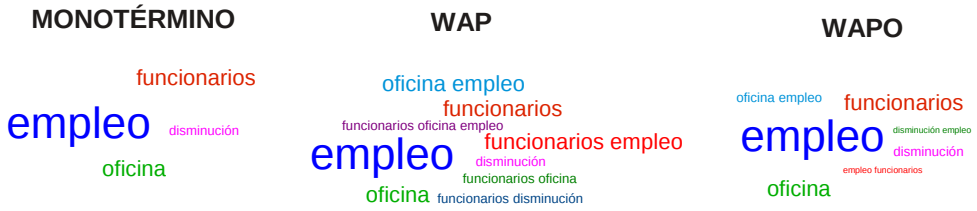


Figura 3.11: Comparación de las *tag cloud* de las diferentes estructuras

- **¿Cuándo será mejor utilizar la estructura WAP y cuándo la estructura WAPO?**

  - Utilizaremos la estructura WAPO 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 WAPO 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 WAPO la búsqueda es más restrictiva, más precisa y visualmente, los términos en la nube aparecen menos aglomerados al ser menores en número, por lo que es más fácil identificarlos.

  - Utilizaremos la estructura WAP 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 WAP en forma de nube presenta mayor número de términos que la estructura WAPO, 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.

¿Cómo Mejoran la Estructura WAP y la Estructura WAPO a la Estructura Monotérmino Ponderada?

Tanto la estructura WAP como la estructura WAPO ofrecen incuestinables mejoras sobre la estructura monotérmino ponderada. Citaremos las más evidentes:

- Recuperan información más precisa.
- Ofrecen mayor cantidad de sugerencias de búsqueda y exploración.
- Permiten identificar relaciones entre conceptos y sugieren términos relacionados con el término o términos de la consulta.
- Mediante los componentes multitérmino, facilitan la discriminación entre conceptos.
- Es posible 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.

Además, la matemática subyacente de las estructuras WAP y WAPO, permite definir todas las operaciones realizadas sobre la base de datos. Y ambas se obtienen a través de un método estándar perfectamente establecido.

3.5.2 Cálculo de la Subestructura Inducida

Para satisfacer las consultas que van a realizarse sobre la base de datos, es necesario el cálculo de previo de la subestructura inducida para cada tupla, así el usuario puede consultar directamente el TDA de cada tupla sin necesidad de consultar sobre el dominio activo completo del atributo textual.

También a través de este TDA es posible saber cuáles son las tuplas que se recuperan cuando el usuario consulta sobre el dominio del atributo.

Las subestructuras inducidas WAP y WAPO por una secuencia Y, no pueden calcularse previamente, ya que es necesario conocer la secuencia Y, que será la secuencia que contenga los términos de la consulta.

A continuación se calculan ambos tipos de subestructuras inducidas para los datos de nuestro ejemplo.

1. **Cálculo de la subestructura-AP y APO inducidas para cada tupla**

   A través de este proceso se obtiene la subestructura inducida de cada tupla de la base de datos y se escribe su representación como un TDA en una nueva columna de la tabla de la que provienen los datos originales, como un nuevo atributo. Para obtener el TDA, se realiza la intersección o el acoplamiento entre la estructura-AP o APO y el texto corto modificado de la tupla que se obtiene a la salida del proceso de limpieza de datos.

   Como la subestructura inducida que vamos a calcular es para representar las tuplas como TDA, no tiene sentido emplear la ponderación, ya que el peso o frecuencia de cada tupla es igual a uno, por eso hacemos el acoplamiento con la estructura de conocimiento sin ponderar.

   Es posible que se pierdan algunos términos en el valor de alguna tupla tras el acoplamiento, dado que, la subestructura inducida se calcula a partir de la estructura-AP o APO y ésta, ha podido perder los términos del lenguaje inicial del atributo que no cumplan con el soporte mínimo establecido, por lo que esos términos no aparecerán en el TDA.

   En la Tabla 3.17 se representan las subestructuras AP y APO inducidas para cada tupla del Tabla 3.3. Las estructuras AP y APO obtenidas a partir de dicha tabla, pueden verse en la Tabla 3.18.

n	Itemsets	Subestructura-AP Inducida	Subestructura APO Inducida
1	{*faltan, funcionarios, oficina, empleo*}	{*funcionarios, oficina, empleo*}	{*oficina, empleo*}{*funcionarios*}
2	{*deterioro, empleo, funcionarios*}	{*funcionarios, empleo*}	{*empleo, funcionarios*}
3	{*empleo, oficina, mejor, valorado*}	{*empleo, oficina*}	{*empleo, oficina*}
4	{*funcionarios, abarrotan, oficina, empleo*}	{*funcionarios, oficina, empleo*}	{*funcionarios*}{*oficina, empleo*}
5	{*disminución, empleo*}	{*disminución, empleo*}	{*disminución, empleo*}
6	{*funcionarios, van, oficina, empleo*}	{*funcionarios, oficina, empleo*}	{*funcionarios*}{*oficina, empleo*}
7	{*empleo, funcionarios, peligra*}	{*empleo, funcionarios*}	{*empleo, funcionarios*}
8	{*disminución, empleo, sueldo, 2013*}	{*disminución, empleo*}	{*disminución, empleo*}
9	{*trabajadores, critican, empleo, oficina*}	{*empleo, oficina*}	{*empleo, oficina*}
10	{*empleo, verano, disminución, paro*}	{*empleo, disminución*}	{*empleo*}{*disminución*}

Tabla 3.17: TDA. Subestructura inducida por tuplas

Estructura-AP	Estructura APO
$T = g(A, B)$ $A = \{funcionarios, oficina, empleo\}$ $B = \{empleo, disminución\}$	$E = g(A', B', C', D')$ $A' = \{oficina, empleo\}$ $B' = \{empleo, funcionarios\}$ $C' = \{empleo, oficina\}$ $D' = \{disminución, empleo\}$

Tabla 3.18: Estructuras AP y APO

Vemos que son bastantes las palabras que se pierden de los *itemsets* originales en las subestructuras inducidas, en concreto todas las palabras que tienen una sola aparición y por lo tanto un soporte bajo. Es el caso de la palabra "faltan" en la primera tupla o "deterioro" en la segunda. En este ejemplo, las subestructuras AP y APO inducidas se comportan igual con respecto a la pérdida de palabras.

El hecho de que la forma de representación obtenida elimina los términos que no cumplen con el soporte mínimo, constituye una limitación con respecto a la forma tradicional en que se procesan los textos cortos en las bases de datos.

2. **Cálculo de las subestructuras inducidas WAP y WAPO**

   Supongamos ahora que un usuario realiza una consulta sobre un atributo textual de la base de datos. Este requerimiento se expresa a través de un conjunto

de términos y las estructuras WAP y WAPO representan el dominio activo de dicho atributo, cada una con sus características particulares.

Ahora sí incluimos la ponderación en las estructuras de dominio, ya que queremos que la subestructura inducida sea ponderada, porque el peso aportará información importante, como el número de tuplas que se recuperan con la consulta. Veamos esto detalladamente.

Sea X conjunto referencial de items: X = {funcionarios, empleo, oficina, sueldo, disminución, faltan, deterioro, mejor, valorado, abarrotan, van, peligra, 2013, trabajadores, critican, verano, paro}.

Sea la estructura WAPO $\widetilde{E} = g(\widetilde{A_1}, \widetilde{A_2}, .., \widetilde{A_n})$ con conjunto referencial de items X. Sea $\widetilde{T} = g(\widetilde{A_1}, \widetilde{A_2}, .., \widetilde{A_m})$ una estructura WAP con conjunto referencial de items X e $Y \subseteq X$ un conjunto-AP o una AP-Seq según proceda. Este conjunto Y no es ponderado ya que representa los términos de la consulta.

En nuestro ejemplo (ver Tabla 3.16):

$$\begin{aligned}\widetilde{T} &= g([\{funcionarios, oficina, empleo\}, (3)], [\{empleo, disminución\}, \\ &\quad (3)]) \\ \widetilde{E} &= g([\{oficina, empleo\}, (3)], [\{empleo, funcionarios\}, (2)], [\{empleo, \\ &\quad oficina\}, (2)], [\{disminución, empleo\}, (2)])\end{aligned}$$

Y supongamos que tenemos los siguientes conjuntos de consulta Y:

Y_1={*disminución, empleo*}
Y_2={*funcionarios, empleo*}
Y_3={*oficina, empleo*}
Y_4={*empleo*}
Y_5={*funcionarios, sueldo*}

En la Tabla 3.19 podemos ver la subestructura WAP inducida para cada uno de estos conjuntos Y. Y en la Tabla 3.20 la subestructura WAPO.

Consultas	Subestructura WAP Inducida $\widetilde{T'}$
Y_1	*g({disminución, empleo})=[{disminución, empleo},(3)], [{disminución},(3)], [{empleo},(10)]*
Y_2	*g({funcionarios, empleo})=[{funcionarios, empleo},(5)], [{funcionarios},(5)], [{empleo},(10)]*
Y_3	*g({oficina, empleo})=[{oficina, empleo},(5)],[{oficina},(5)],[{empleo},(10)]*
Y_4	*g({empleo})=[{empleo},(10)]*
Y_5	*g({funcionarios})=[{funcionarios},(5)]*

Tabla 3.19: Subestructura WAP inducidas

Consultas	Subestructura WAPO Inducida $\widetilde{E'}$
Y_1	*g({disminución, empleo})=[{disminución, empleo},(2)], [{disminución},(3)], [{empleo},(10)]*
Y_2	*g({funcionarios}, {empleo})=[{funcionarios}, (5)], [{empleo},(10)]*
Y_3	*g({oficina, empleo})=[{oficina, empleo},(3)],[{oficina},(5)],[{empleo},(10)]*
Y_4	*g({empleo})=[{empleo},(10)]*
Y_5	*g({funcionarios})=[{funcionarios},(5)]*

Tabla 3.20: Subestructura WAPO inducidas

Supongamos ahora que un usuario ha formulado una consulta sobre los términos "disminución empleo" a través de la estructura WAP y que ha pedido que esta consulta se realice a través de un acoplamiento fuerte. Según el Tabla 3.19, la subestructura WAP inducida por esta consulta contiene los *itemsets:* $\{disminución, empleo\}$, $\{disminución\}$ y $\{empleo\}$. La ponderación de estos *itemsets* indica el número de tuplas que contienen en su TDA al *itemset* en cuestión. Al seleccionar el acoplamiento fuerte, el usuario solicita que todos los términos de su consulta estén incluidos en el TDA de la tuplas que se recuperan, es decir, no pregunta por un término u otro, si no por los dos términos juntos en una misma frase. Mirando los *itemsets* en la subestructura WAP inducida, vemos que el *itemset* ponderado que contiene los dos términos a la vez es *[*$\{disminución, empleo\}$*,(3)]* cuya ponderación nos indica que son tres las tuplas que contienen el *itemset* en su TDA, bien por sí mismo o bien incluido en un *itemset* mayor. Si vamos al Tabla 3.17, vemos que estas tuplas son la 5, la 8 y la 10, entonces esas son las tuplas que se devolverían con esa consulta.

Si en lugar de seleccionar la estructura WAP, el usuario hubiera seleccinado la WAPO, siguiendo el mismo procedimiento, vemos que se recuperarían sólo dos tuplas, que son la 5 y la 8. Si consultamos la tabla original (Tabla 3.1), vemos que el usuario posiblemente no quisiera recuperar la tupla 10, ya que la información que aporta esta tupla no se ajusta a los requerimientos de la consulta con total precisión.

Si en lugar de consultar a través del acoplamiento fuerte, lo hubiera hecho a través del acoplamiento débil, se recuperarían, tanto con una como con otra estructura, todas las tuplas que contengan en su TDA los *itemsets* $\{$*disminución, empleo*$\}$ o $\{$*disminución*$\}$ o $\{$*empleo*$\}$, puesto que ahora no es necesario que todas las palabras de la consulta estén incluidas en los *itemsets*. En el caso de este ejemplo, se recuperarían todas las tuplas, al estar la palabra "empleo" presente en todas ellas.

Veremos las tuplas que se recuperan con cada una de las consultas, estructuras y acoplamiento y realizaremos una discusión al respecto en la siguiente sección (Sección 3.5.3), concretamente, se visualizan estas tuplas en los Tablas 3.25 y 3.26, junto a los índices de acoplamiento fuerte y débil de las consultas con las estructuras AP y APO.

3.5.3 Cálculo de los Índices de Acoplamiento Fuerte y Débil de un Conjunto con las Estructuras AP y APO

Resulta interesante cuantificar el grado de acoplamiento del conjunto de términos introducidos en la consulta con ambas estructuras AP y APO, como también analizar si todos los términos están completamente incluidos en algún conjunto generador de la estructura de conocimiento (en cuyo caso habría al menos una tupla que satisfacería todos los términos), si sólo unos pocos (de ser así podrían sugerirse términos relacionados para refinar la búsqueda) o ninguno de ellos.

Esto es lo que hacemos cuando comprobamos si el acoplamiento es fuerte (todos los términos están incluidos) o débil (sólo unos pocos lo están) y calculamos los índices de bondad.

Ahora tomamos las estructuras sin ponderar como tipo de dato, porque para ver si existe acoplamiento entre una estructura y un conjunto o secuencia, sólo hace falta mirar si los términos del conjunto se encuentran en alguno de los generadores de la estructura, independientemente de su peso en ésta. Igualmente, el cálculo de los índices de acoplamiento es independiente del peso.

Cuando calculamos un índice de acoplamiento por el promedio, lo que calculamos es medida de similitud del atributo textual y los términos de la consulta, por eso se tienen en cuenta todos los conjuntos que componen la estructura para su cálculo. Cuando calculamos el índice de acoplamiento por el máximo, calculamos una medida de similitud de los términos de la consulta con el conjunto generador de la estructura con el que tienen un mayor grado de acoplamiento.

1. **Índice de acoplamiento fuerte**

   Cuando calculamos si existe acoplamiento fuerte, lo que hacemos es comprobar si todos los términos del conjunto consulta Y están completamente incluidos en algún conjunto generador de la estructura de conocimiento. Con el índice de acoplamiento fuerte damos una medida de la bondad de dicho acoplamiento, entendiendo que el acoplamiento será mejor cuanto más se asemeje el conjunto Y al conjunto o conjuntos de la estructura en que aparece incluido.

   a) Índice de acoplamiento fuerte de Y con la estructura-AP por el promedio y por el máximo (ver Definición 2.2.9).

   En la Tabla 3.21 podemos ver si existe acoplamiento fuerte de la estructura-AP de nuestro ejemplo (ver Tabla **??** con los ejemplos de consultas vistos en la sección anterior (Sección 3.5.2) y el índice de bondad de dicho acoplamiento. Existirá acoplamiento fuerte siempre que el índice de este acoplamiento sea distinto de cero.

   b) Índice de acoplamiento fuerte de Y con la estructura APO por el promedio y por el máximo (ver Definición 3.4.3).

   En la Tabla 3.22 podemos ver si existe acoplamiento fuerte de la estructura APO de nuestro ejemplo (ver Tabla **??**) con las consultas propuestas, así como el índice de bondad de dicho acoplamiento. Igual que

Consultas	Ind. de aco. fuerte por el promedio	Ind. de aco. fuerte por el máximo
{*disminución, empleo*}	$S(Y\|T) = \frac{2/2}{2} = 0.5$	$S(Y\|T) = max\left(\frac{2}{2}\right) = 1$
{*funcionarios, empleo*}	$S(Y\|T) = \frac{2/3}{2} = 0.33$	$S(Y\|T) = max\left(\frac{2}{3}\right) = 0.67$
{*oficina, empleo*}	$S(Y\|T) = \frac{2/3}{2} = 0.33$	$S(Y\|T) = max\left(\frac{2}{3}\right) = 0.67$
{*empleo*}	$S(Y\|T) = \frac{(1/3+1/2)}{2} = 0.42$	$S(Y\|T) = max\left(\frac{1}{3}, \frac{1}{2}\right) = 0.5$
{*funcionarios, sueldo*}	$S(Y\|T) = 0$	$S(Y\|T) = 0$

Tabla 3.21: Índice de acoplamiento fuerte con la estructura-AP

antes, diremos que existe acoplamiento fuerte cuando este índice sea distinto de cero.

c) Comparación del acoplamiento fuerte del conjunto Y con T y E.

El dominio de la estructura-AP es distinto del dominio de la estructura APO y además el índice fuerte se calcula de forma diferente para una y otra estructura, ya que para la estructura APO y con el fin de considerar el orden de los términos en la ponderación, se calcula teniendo en cuenta todas las combinaciones posibles de los elementos dentro de las secuencias generadoras, por lo que una comparación directa del índice fuerte para ambas estructuras no tiene mucho sentido.

En la Tabla 3.25 podemos ver representados los índices de acoplamiento fuerte por el promedio y por el máximo para los ejemplos de consulta propuestos con las dos estructuras. En las columnas seis y siete de dicho cuadro podemos ver las tuplas que se recuperan con cada estructura y cada una de las posibles consultas. El color azul en estas columnas indica que la tupla recuperada contiene la información requerida y el color rojo que la tupla recuperada no contiene la información requerida. En la

Consultas	Ind. de aco. fuerte por el promedio	Ind. de aco. fuerte por el máximo
{*disminución, empleo*}	$S(Y\|T) = \frac{2 \cdot \left(2!/_{1!}\right)}{\frac{2}{2!}} \cdot \frac{1}{4} = 0.25$	$S(Y\|T) = max\left(\frac{2 \cdot \left(2!/_{1!}\right)}{\frac{2}{2!}}\right) = 1$
{*funcionarios, empleo*}	$S(Y\|T) = 0$	$S(Y\|T) = 0$
{*oficina, empleo*}	$S(Y\|T) = \frac{2 \cdot \left(2!/_{1!}\right)}{\frac{2}{2!}} \cdot \frac{1}{4} = 0.25$	$S(Y\|T) = max\left(\frac{2 \cdot \left(2!/_{1!}\right)}{\frac{2}{2!}}\right) = 1$
{*empleo*}	$S(Y\|T) = \frac{4\frac{1 \cdot \left(2!/_{2!}\right)}{2}}{2!+2!+2!+2!} \cdot \frac{4}{4} = 0.25$	$S(Y\|T) = max\left(\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}\right) = 0.25$
{*funcionarios, sueldo*}	$S(Y\|T) = 0$	$S(Y\|T) = 0$

Tabla 3.22: Índice de acoplamiento fuerte con la estructura APO

última columna, podemos ver cuáles son todas las tuplas que se desean recuperar con cada una de las consultas.

Recordemos que, en este caso, estamos consultando con el acoplamiento fuerte, lo que quiere decir que queremos que el sistema nos devuelva aquellas tuplas que contienen íntegramente todos los términos de la consulta y, en el caso de la estructura APO, además deben mantener el orden con que los escribimos.

Si atendemos al primer ejemplo de consulta representada mediante el conjunto $Y_1 = \{disminución, empleo\}$ y haciendo uso del Tabla 3.17 donde se representa el dominio activo particular de cada tupla, dado por su subestructura inducida, vemos que las tuplas que contienen en su dominio activo el conjunto consulta Y_1, son las tuplas 5, 8 y 10 para la subestructura-AP inducida y las tuplas 5 y 8 para la subestructura APO inducida. Como vimos en la sección anterior, si buscamos estas tuplas en el conjunto de datos original (ver Tabla 3.1), vemos que sólo

las tuplas 5 y 8 contienen información sobre disminución de empleo, en cambio la tupla 10 contiene información imprecisa, distinta en parte a la información que se requiere. Si en este caso esta tupla no se recupera con la estructura APO, no es debido al orden de los términos en el texto, ya que en la información original aportada por esta tupla, el orden de los términos es el mismo que el orden establecido en la consulta, sino porque ambos términos no son adyacentes de forma estricta en el texto. Sin embargo, si hubiésemos consultado a través del acoplamiento débil, esta tupla sí se recuparía con la estructura APO.

Para tener conocimiento a priori del número de tuplas que se recuperan, hay que calcular la subestructura inducida ponderada (ver Tablas 3.19 y 3.20) y así sabríamos que se recuperan tres tuplas con la estructura-AP y dos tuplas con la estructura APO.

Independientemente de esto, vemos que el índice de acoplamiento fuerte por el máximo para Y_1 es 1 con ambas estructuras. Esto quiere decir que, tanto en la estructura-AP como en la estructura APO, hay un conjunto o secuencia generadora exactamente igual al conjunto de consulta y que, evidentemente, este es el conjunto o secuencia generadora con que más se asemeja Y_1. En cambio, el índice de acoplamiento fuerte por el promedio para Y_1 es distinto con ambas estructuras. Esto es normal si tenemos en cuenta que el número de conjuntos o secuencias generadoras y la longitud de éstas no coincide para T y E. Un índice fuerte por el promedio igual a 0.5 o 0.25 indica que el dominio activo del atributo textual representado mediante la estructura-AP o estructura APO según corresponda, se asemeja a los términos de la consulta en un 50 % o 25 % respectivamente, considerando para el cálculo de dicha similitud, que todos los términos de la consulta estén incluidos en dicho dominio.

Si atendemos ahora al segundo ejemplo de consulta, representada por el conjunto $Y_2 = \{funcionarios, empleo\}$, vemos que los índices de acoplamiento fuerte con la estructura APO por el promedio y por el máximo, son cero en ambos casos, lo que vendría a decirnos que no existe acoplamiento fuerte del conjunto Y_2 con E y por lo tanto no se

recuperaría ninguna información. Sin embargo, con la estructura-AP sí se recupera información y no toda es imprecisa. En la Tabla 3.19 vemos que exactamente se recuperan dos tuplas con esta consulta y en la Tabla 3.17 vemos que estas tuplas son la 2 y la 7 del atributo textual original (Tabla 3.1), que sí podrían tener alguna relación con lo que se consulta, entendiendo que a través de Y_2 se está consultando acerca del empleo de los funcionarios, en cuyo caso el orden más lógico de consulta hubiera sido el inverso al planteado.

Con el conjunto consulta $Y_3 = \{oficina, empleo\}$ se recuperan con la estructura-AP dos tuplas que versan sobre el empleo de oficina y no sobre la oficina de empleo.

$Y_4 = \{empleo\}$ recupera todas las tuplas del atributo textual con ambas estructuras, ya que en todas aparece contenido y al ser un monotérmino, no afecta el orden ni la adyacencia, por lo que las estructuras AP y APO se comportan igual respecto a la recuperación de información a través de monotérminos e igual que lo haría la estructura monotérmino.

El conjunto de consulta $Y_5 = \{funcionarios, sueldo\}$ no tiene acoplamiento fuerte con ninguna de las dos estructura, por eso el índice de acoplamiento fuerte es cero en todos los casos y no se recupera ninguna tupla.

Nótese que, en este ejemplo, tanto con la estructura-AP como con la estructura APO, se recuperan con las consultas ejemplo todas las tuplas que contienen la información que se requiere.

También se puede verificar que el índice de acoplamiento fuerte por el máximo es siempre mayor o igual que el índice de acoplamiento fuerte por el promedio y que todos los índices toman valores comprendidos entre 0 y 1.

2. **Índice de acoplamiento débil**

   Existe acoplamiento débil de un conjunto consulta Y cuando la intersección de Y con al menos un conjunto generador de la estructura de conocimiento es distinta de cero. Con el índice de acoplamiento débil damos una medida de la bondad de dicho acoplamiento, entendiendo que el acoplamiento es mejor cuanto mayor sea el número de conjuntos generadores con los que presenta intersección no vacía y cuanto mayor sea el número de elementos presentes en esas intersecciones.

   Se puede calcular el índice de acoplamiento débil para todos los conjuntos cuya intersección con la estructura de conocimiento sea distinta de cero, aunque estos conjuntos se encuentren incluidos completamente en uno de los generadores de la estructura y por lo tanto presenten un acoplamiento fuerte. Puede haber ocasiones en que estemos interesados en calcular el acoplamiento débil de un conjunto en lugar de el fuerte, aunque ese conjunto presente ambos, ya que mediante el acoplamiento débil nos hacemos idea de la semejanza de la consulta con todos los conjuntos generadores de la estructura y no sólo con los que contienen todos los términos de ésta.

   a) Índice de acoplamiento débil de Y con la Estructura-AP por el promedio y por el máximo (ver Definición 2.2.9).

   En la Tabla 3.23 podemos ver si existe acoplamiento débil de los ejemplos anteriores de posibles consultas y la estructura-AP y el índice de bondad de dicho acoplamiento. Existirá acoplamiento débil siempre que dicho índice sea distinto de cero.

   b) Índice de acoplamiento débil de Y con la Estructura APO por el promedio y por el máximo (ver Definición 3.4.4).

   En la Tabla 3.24 podemos ver los índices de acoplamiento débil de las consultas con la estructura APO. Igual que antes, sabemos que se da acoplamiento débil para un conjunto consulta siempre que el índice de dicho acoplamiento sea distinto de cero.

   c) Comparación del acoplamiento débil del conjunto Y con T y E.

Consultas	Ind. de aco. débil por el promedio	Ind. de aco. débil por el máximo
{*disminución, empleo*}	$W(Y\|T) = \frac{(1/3+1)}{2} = 0.67$	$W(Y\|T) = max\left(\frac{1}{3}, 1\right) = 1$
{*funcionarios, empleo*}	$W(Y\|T) = \frac{(2/3+1/2)}{2} = 0.58$	$W(Y\|T) = max\left(\frac{2}{3}, \frac{1}{2}\right) = 0.67$
{*oficina, empleo*}	$W(Y\|T) = \frac{(2/3+1/2)}{2} = 0.58$	$W(Y\|T) = max\left(\frac{2}{3}, \frac{1}{2}\right) = 0.67$
{*empleo*}	$W(Y\|T) = \frac{(1/3+1/2)}{2} = 0.42$	$W(Y\|T) = max\left(\frac{1}{3}, \frac{1}{2}\right) = 0.5$
{*funcionarios, sueldo*}	$W(Y\|T) = \frac{1/3}{2} = 0.17$	$W(Y\|T) = max\left(\frac{1}{3}\right) = 0.33$

Tabla 3.23: Índice de acoplamiento débil con la estructura-AP

Consultas	Índice de aco. débil por el promedio	Índice de aco. débil por el máximo
{*disminución, empleo*}	$W(Y\|T) = \frac{3\frac{1\cdot(2!/2!)}{2}+\frac{2\cdot(2!/1!)}{2}}{2!+2!+2!+2!} \cdot \frac{4}{4} = 0.44$	$W(Y\|T) = max\left(\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, 1\right) = 1$
{*funcionarios, empleo*}	$W(Y\|T) = \frac{5\frac{1\cdot(2!/2!)}{2}}{2!+2!+2!+2!} \cdot \frac{4}{4} = 0.31$	$W(Y\|T) = max\left(\frac{1}{4}, \frac{1}{2}, \frac{1}{4}, \frac{1}{4}\right) = 0.5$
{*oficina, empleo*}	$W(Y\|T) = \frac{3\frac{1\cdot(2!/2!)}{2}+\frac{2\cdot(2!/1!)}{2}}{2!+2!+2!+2!} \cdot \frac{4}{4} = 0.44$	$W(Y\|T) = max\left(1, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}\right) = 1$
{*empleo*}	$W(Y\|T) = \frac{4\frac{1\cdot(2!/2!)}{2}}{2!+2!+2!+2!} \cdot \frac{4}{4} = 0.25$	$W(Y\|T) = max\left(\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}\right) = 0.25$
{*funcionarios, sueldo*}	$W(Y\|T) = \frac{\frac{1\cdot(2!/2!)}{2}}{2!} \cdot \frac{1}{4} = 0.06$	$W(Y\|T) = max\left(\frac{1}{4}\right) = 0.25$

Tabla 3.24: Índice de acoplamiento débil con la estructura APO

Una comparación directa del índice débil para ambas estructuras tam-

	Índice de Acoplamiento Fuerte						
	Estructura-AP T		Estructura APO E		Tuplas Recuperadas		
Consulta	**Promedio**	**Máximo**	**Promedio**	**Máximo**	T	E	**Requeridas**
Y_1={*disminución, empleo*}	0.5	1	0.25	1	5 8 10	5 8	5 8
Y_2={*funcionarios, empleo*}	0.33	0.67	0	0	1 2 4 6 7	–	2 7
Y_3={*oficina, empleo*}	0.33	0.67	0.25	1	1 3 4 6 9	1 4 6	1 4 6
Y_4={*empleo*}	0.42	0.5	0.25	0.25	1...10	1...10	1...10
Y_5={*funcionarios, sueldo*}	0	0	0	0	–	–	–

Tabla 3.25: Comparación del acoplamiento fuerte

	Índice de Acoplamiento Débil						
	Estructura-AP T		Estructura APO E		Tuplas Recuperadas		
Consulta	**Promedio**	**Máximo**	**Promedio**	**Máximo**	T	E	**Requeridas**
Y_1={*disminución, empleo*}	0.67	1	0.44	1	1...10	1...10	1...10
Y_2={*funcionarios, empleo*}	0.58	0.67	0.31	0.5	1...10	1...10	1...10
Y_3={*oficina, empleo*}	0.58	0.67	0.44	1	1...10	1...10	1...10
Y_4={*empleo*}	0.42	0.5	0.25	0.25	1...10	1...10	1...10
Y_5={*funcionarios, sueldo*}	0.17	0.33	0.06	0.25	1 2 4 6 7	1 2 4 6 7	1 2 4 6 7 8

Tabla 3.26: Comparación del acoplamiento débil

poco tiene mucho sentido, por las mismas razones que no la tiene la comparación del índice fuerte: el dominio de la estructura-AP es distinto del dominio de la estructura APO y además los índices se calculan de forma diferente para una y otra estructura, ya que para la estructura APO 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 secuencias generadoras.

En la Tabla 3.26 se recogen los índices de acoplamiento débil por el promedio y por el máximo de los ejemlos de consulta propuestos con la estructura-AP y con la estructura APO, junto con las tuplas que se recuperan con cada una de las consultas y estructuras y las que se desean recuperar. Al estar consultando a través del acoplamiento débil, es de suponer que no se recupera ninguna tupla que no se requiera, ya que no se exige precisión, si no solamente recuperar las tuplas que contengan algún o algunos de los términos de la consulta, para lo que bastaría que alguno de los términos del conjunto Y aparezca en el dominio activo de cada tupla, representado por su subestructura inducida (ver Tabla 3.17).

Recordemos que el acoplamiento débil se usa cuando estamos más interesados en explorar el contenido del atributo textual o en buscar palabras relacionadas con un determinado término.

Vemos que para las cuatro primeras consultas, se recuperan mediante el acoplamiento débil todas las tuplas que tenemos en nuestro ejemplo, esto es porque todas las tuplas contienen en su TDA al menos una de las palabras del conjunto consulta Y. Sin embargo, para la última consulta representada mediante el conjunto $Y_5 = \{funcionarios, sueldo\}$ se recuperan, tanto con la estructura-AP como con la estructura APO, las tuplas 1,2,4,6 y 7, que son las que contienen el término "funcionarios" en la subestructura inducida correspondiente (Tabla 3.17), pero no se recupera la tupla 8 que recoge información relacionada con el otro término, el término "sueldo". Esto es porque en el dominio activo de la tupla 8, dado por su subestructura inducida, no aparece el término "sueldo" en ninguno de los conjuntos o secuencias generadoras al no ser un término frecuente que se elimina en la primera fase del algoritmo Apriori y del algoritmo Apriori modificado. Este ejemplo pone de manifiesto que no siempre se recuperará toda la información por la que se consulta a través de las estructuras AP y APO, siempre quedará una parte de información de difícil acceso y es la que se intenta minimizar. De todas formas, el usuario siempre tendrá la opción de consultar mediante los métodos tradicionales.

El índice de acoplamiento débil siempre es mayor o igual que el índice de acoplamiento fuerte, por lo que si un conjunto Y tiene un índice fuerte igual a 1, el índice débil también será 1. Es el caso de Y_1 en el acoplamiento por el máximo con las dos estructuras y de Y_3 en el acoplamiento por el máximo con la estructura APO.

Vemos que en todos los casos existe el acoplamiento débil, ya que no tenemos en nuestro ejemplo ningún índice débil que sea igual a cero, esto es porque no hay ninguna intersección vacía entre los conjuntos consulta Y y los conjuntos o secuencias generadoras de las estructuras-AP y APO.

Para Y_1, un índice débil por el promedio igual a 0.67 o 0.44 indica que el dominio activo del atributo textual consultado se asemeja a los términos de la consulta en un 67 % o 44 % respectivamente, considerando para el cálculo de dicha similitud las pecularidades de la estructura que representa en cada caso el dominio activo y las del tipo de acoplamiento.

Para Y_2, un índice débil por el máximo igual a 0.67 o 0.5 indica que el conjunto consulta se asemeja al conjunto o secuencia generadora de la estructura con que mejor se acopla en un 67 % o 50 % respectivamente.

Vemos como, normalmente, los índices de acoplamiento son mayores para la estructura-AP que para la estructura-APO, ya que se considera que los conjuntos de consulta Y acoplan mejor con la estructura-AP, ya que el orden no se tiene en cuenta para el cálculo de la bondad de estos acoplamientos ni para el acoplamiento en sí.

3. Comparación de los Índices Fuerte y Débil

   Por último, compararemos los índices fuerte y débil obtenidos para la estructura-AP y la estructura APO (Tablas 3.27 y 3.28) para corroborar las propiedades.

   Tanto para la estructura-AP como para la estructura APO se cumple:

   - 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 es mayor o igual para todos los casos que el índice de acoplamiento fuerte.

3.5.4 Cálculo de los Índices de Acoplamiento Fuerte y Débil de un Conjunto con el TDA para cada Tupla

Como se dijo con anterioridad, también es posible consultar directamente sobre el TDA de cada tupla en lugar de consultar sobre la estructura de conocimiento.

	Ind.Acoplamiento fuerte		Ind. Acoplamiento débil	
Estructura-AP T	**Promedio**	**Máximo**	**Promedio**	**Máximo**
***Y*={*disminución, empleo*}**	0.5	1	0.67	1
***Y*={*funcionarios, empleo*}**	0.33	0.67	0.58	0.67
***Y*={*oficina, empleo*}**	0.33	0.67	0.58	0.67
***Y*={*empleo*}**	0.42	0.5	0.42	0.5
***Y*={*funcionarios, sueldo*}**	0	0	0.17	0.33

Tabla 3.27: Comparación de los índices de acoplamiento fuerte y débil con la estructura-AP

	Ind.Acoplamiento fuerte		Ind. Acoplamiento débil	
Estructura APO E	**Promedio**	**Máximo**	**Promedio**	**Máximo**
***Y*={*disminución, empleo*}**	0.25	1	0.44	1
***Y*={*funcionarios, empleo*}**	0	0	0.31	0.5
***Y*={*oficina, empleo*}**	0.25	1	0.44	1
***Y*={*empleo*}**	0.25	0.25	0.25	0.25
***Y*={*funcionarios, sueldo*}**	0	0	0.06	0.25

Tabla 3.28: Comparación de los índices de acoplamiento fuerte y débil con la estructura APO

Supongamos que tenemos la consulta $Y_2 = \{funcionarios, empleo\}$ y tomemos las tres primeras tuplas del Tabla 3.17 con la subestructura inducida por tuplas ya calculada (Tabla 3.29).

n	Subestructura-AP Inducida	Subestructura APO Inducida
1	{*funcionarios, oficina, empleo*}	{*oficina, empleo*}{*funcionarios*}
2	{*funcionarios, empleo*}	{*empleo, funcionarios*}
3	{*empleo, oficina*}	{*empleo, oficina*}

Tabla 3.29: Subconjunto de datos seleccionado para el cálculo de los índices de acoplamiento para cada tupla

Resulta evidente que existirá acoplamiento fuerte de Y_2 con la subestructura-AP inducida de la segunda tupla, sin embargo, no existirá acoplamiento fuerte alguno con ninguna de las tuplas en la subestructura APO inducida.

De igual modo, también es fácil ver que el conjunto Y_2 acopla de forma débil con todas las tuplas de las subestructuras AP y APO inducidas.

Para darnos cuenta de esto, lo que hacemos es un acoplamiento (fuerte o débil) de las subestructuras inducidas con el conjunto Y_2. Dicho acoplamiento llevará igualmente asociados unos índices de bondad.

En la Tabla 3.30 se muestran los índices de acoplamiento del conjunto $Y_2 = \{funcionarios, empleo\}$ con las tres tuplas de la subestructura-AP inducida y en la Tabla 3.31 los índices de acoplamiento de Y_2, entendido como secuencia, con la subestructura APO inducida.

n	Subestructura-AP Inducida	Indice Fuerte Promedio/Máximo	Indice Débil Promedio/Máximo
1	{*funcionarios, oficina, empleo*}	0.67/0.67	0.67/0.67
2	{*funcionarios, empleo*}	1/1	1/1
3	{*empleo, oficina*}	0/0	0.5/0.5

Tabla 3.30: Indices de acoplamiento para la subestructura-AP inducida de cada tupla

n	Subestructura APO Inducida	Indice Fuerte Promedio/Máximo	Indice Débil Promedio/Máximo
1	{*oficina, empleo*}{*funcionarios*}	0/0	0.75/1
2	{*empleo, funcionarios*}	0/0	0.5/0.5
3	{*empleo, oficina*}	0/0	0.25/0.25

Tabla 3.31: Indices de acoplamiento para la subestructura APO inducida de cada tupla

La interpretación es análoga a la que hacíamos para los índices calculados sobre la estructura-AP o APO. Si el usuario consulta a través del acoplamiento fuerte con los términos "*funcionarios, empleo*", con la subestructura-AP inducida se recuperarán las tuplas 1 y 2, mientras que con la subestructura-APO inducida no se recuperará ninguna de las tres tuplas, ya que el valor cero en el índice de acoplamiento fuerte indica que no se da dicho acoplamiento. Un valor uno en el índice de acoplamiento fuerte, indica que el conjunto de términos de la consulta es exactamente igual que la subestructura inducida de esa tupla, en cambio un valor de

0.67 indica que se da el acoplamiento fuerte pero que la semejanza del conjunto de términos de la consulta con la subestructura inducida es de un 67 %.

Los índices de acoplamiento débil representan también la semejanza de la consulta con la subestructura inducida de la tupla, la única diferencia es que el cálculo de esta semejanza se realiza aunque no todos los términos de consulta estén incluidos en la subestructura, basta que con esté incluido alguno de ellos.

3.5.5 Cálculo del Índice F1

Por último, vamos a realizar un ejemplo simple de cálculo de alguna medida de *precisión* y *exhaustividad*. Hemos pensado en la F_1 *Score* por ser muy conocida en el ámbito de recuperación de información.

La F_1 *Score* es la media armónica de la *precisión* y la *exhaustividad* y se calcula como:

$$F_1 = 2 \cdot \frac{precision \cdot exhaustividad}{precision + exhaustividad} \tag{3.84}$$

donde la *precisión* es la fracción de instancias recuperadas que son relevantes y la *exhaustividad* es la fracción de instancias relevantes que son recuperadas. Calcularemos por tanto la *precisión* como el número de resultados correctos dividido entre el número de todos los resultados devueltos por la consulta y la *exhaustividad* como el número de resultados correctos dividido por el número de resultados que se debían haber devuelto.

Realizaremos este ejemplo con las tres primeras consultas propuestas $Y_1 = \{disminución, empleo\}, Y_2 = \{funcionarios, empleo\}$ y en el último caso con $Y_3 = \{oficina, empleo\}$, suponiendo que el usuario las selecciona entre los términos de la *tag cloud*. Si por ejemplo selecciona Y_3, es de suponer, que la información que quiere recuperar es la relacionada con la oficina de empleo, luego realizaremos todas las búsquedas a través del acoplamiento fuerte.

Si el usuario consulta mediante la *tag cloud* monotérmino, podrá hacerlo haciendo pulsando en ”oficina“ o bien en ”empleo“, pero no en las dos etiquetas simultáneamente. Si consulta con la estructura monotérmino se realizara introduciendo él mismo los términos de dicha consulta, el tipo de acoplamiento con que debe llevar a cabo su requirimiento, será con el acoplamiento débil, ya que no es

posible el acoplamiento fuerte de dos términos de consulta en la estructura monotérmino. Mediante este acoplamiento débil, se recuperarían todas las tuplas que tengan que ver tanto con ”oficina“ como con ”empleo“.

Podemos ver las tuplas que se recuperan con cada consulta y las que se desea recuperar en la Tabla 3.25. En la Tabla 3.32 podemos ver la F_1 *Score* calculada para Y_1 con la estructura-AP, la estructura APO y la estructura monotérmino, considerando dos opciones para esta última, que el usuario pulse en el primer término que desea encontrar o lo haga en el segundo. En la Tabla 3.33 vemos la calculada para Y_2 y en la Tabla 3.34 la calculada para Y_3.

	Estructura-AP T {***disminución, empleo***}	**Estructura APO** E {***disminución, empleo***}	**Estructura Monotérmino** M {***disminución***}	{***empleo***}
precision	0.67	1	0.67	0.3
exhaustividad	1	1	1	1
F_1 **Score**	0.8	1	0.75	0.46

Tabla 3.32: Cálculo de la F_1 *Score* para Y_1

	Estructura-AP T {***funcionarios, empleo***}	**Estructura APO** E {***funcionarios, empleo***}	**Estructura Monotérmino** M {***funcionarios***}	{***empleo***}
precision	0.4	–	0.4	0.3
exhaustividad	1	0	1	1
F_1 **Score**	0.57	–	0.57	0.46

Tabla 3.33: Cálculo de la F_1 *Score* para Y_2

	Estructura-AP T {***oficina, empleo***}	**Estructura APO** E {***oficina, empleo***}	**Estructura Monotérmino** M {***oficina***}	{***empleo***}
precision	0.67	1	0.6	0.3
exhaustividad	1	1	1	1
F_1 **Score**	0.8	1	0.75	0.46

Tabla 3.34: Cálculo de la F_1 *Score* para Y_3

Para la consulta Y_2 no se puede calcular la F_1 *Score* con la estructura APO, ya que no existe la *precisión* para esa consulta al no recuperarse ninguna instancia. Esto ocurre porque los términos de esta etiqueta están en orden inverso al lógico de consulta, ya que se han considerado como resultados correctos aquellos que contienen información sobre el empleo de los funcionarios.

En los demás casos, la F_1 *Score* calculada para la estructura APO, mejora la F_1 *Score* calculada para la estructura-AP y ésta última, mejora o iguala la calculada para la estructura monotérmino con uno u otro elemento de consulta.

En el Capítulo 6 se realiza un estudio más completo sobre la *precisión* y la *exhaustividad* con las distintas estructuras.

3.6 Resumen y Conclusiones

En este capítulo se han presentado tres estructuras alternativas a la estructura-AP [TP13b]:

- *Estructura WAP*: es el resultado de añadir ponderación a la estructura-AP.
- *Estructura APO*: introduce el orden en los *itemsets*.
- *Estructura WAPO*: establece ponderación y orden en los *itemsets*.

Estas estructuras se han definido matemáticamente junto con sus operaciones y algunas propiedades y se han expuesto diversos ejemplos para facilitar la comprensión de estas definiciones.

Una de las operaciones más importante es el acoplamiento de un conjunto de términos con una estructura. Su importancia reside en que es la operación utilizada en la consulta. Se ha visto como el acoplamiento puede ser fuerte o débil dependiendo de si queremos que todos los términos de la consulta estén presentes en las tuplas devueltas por el sistema o nos baste con que estén sólo unos pocos. También se han presentado unos índices para medir la bondad de estos acoplamientos.

En la Sección 3.5 se ha expuesto un ejemplo práctico donde se han comparado las estructuras monotérmino ponderada, WAP y WAPO. El motivo de incluir la estructura monotérmino en este ejemplo, es que es la estructura que normalmente vemos representada en Internet con forma de *tag cloud*.

Las principales conclusiones obtenidas de esta comparación son:

- Las estructuras WAP y WAPO recuperan información más precisa que la estructura monotérmino.
- Las estructuras WAP y WAPO ofrecen mayor cantidad de sugerencias de búsqueda y exploración que la estructura monotérmino.
- Las estructuras WAP y WAPO permiten identificar relaciones entre conceptos y sugieren términos relacionados para la consulta, no así la estructura monotérmino.
- Las estructuras WAP y WAPO facilitan la identificación del contenido debido a los componentes multitérmino y discriminan mejor entre conceptos.
- Mediante las estructuras WAP y WAPO es posible definir todas las operaciones que se realizan sobre la base de datos.
- La estructura WAPO es más apta que la estructura WAP en consultas donde el orden de los términos es importante. Mediante la estructura WAPO la búsqueda es más restrictiva, más precisa y visualmente, los terminos aparecen menos aglomerados, por lo que es más fácil identificarlos.
- La estructura WAP es más apta que la estructura WAPO cuando estamos más interesados en explorar el entorno que en realizar una búsqueda específica, ya que ofrece un mayor número de sugerencias o posibles términos relacionados.

En el ejemplo presentado, además de esta comparación, se han calculado los índices de acoplamiento de diversos conjuntos consulta y la F_1 *Score* como medida estándar de *precisión* y *exhaustividad*, obteniendo que, cuando una consulta está bien expresada, la *precisión* es mayor con la estructura APO que con la estructura-AP y ambas mejoran la *precisión* obtenida con la estructura monotérmino, mientras que la *exhaustividad* es la misma para las tres estructuras. Esto se traduce en que la estructura APO es la que mayor índice F1 presenta para estas consultas.

A partir de ahora nos centraremos en las estructuras WAP y WAPO, que son las que permiten su visualización a través de la *tag cloud*.

CAPÍTULO

4

Distintos Algoritmos para la Generación de las Estructuras

En este capítulo empezaremos repasando diversas técnicas y algoritmos que pueden ser empleados (con ligeras modificaciones en algunos casos) para la obtención de los *itemsets* o las *item-seqs* frecuentes.

En la Sección 4.1 veremos los algoritmos más conocidos para la obtención de los *itemsets* frecuentes. En la sección 4.2 se hará una revisión de algoritmos para la extracción de secuencias frecuentes.

En la Sección 4.3 describiremos los algoritmos que hemos empleado nosotros para la generación de los *itemsets* y las *item-seqs*, que son el algoritmo Apriori y el algoritmo Apriori modificado, respectivamente.

Muchos de estos algoritmos trabajan de forma similar a como lo haría un índice invertido o lista invertida. En la Sección 4.4 se hará una breve revisión de estos índices, formulando una definición adaptada de los conceptos de lista invertida e índice invertido completo.

Estos conceptos nos servirán de base para entender el método de generación de las estructuras WAP y WAPO, alternativo al algoritmo Apriori, que se propone en la Sección 4.5, junto con un ejemplo práctico que ayuda a su comprensión.

Terminaremos en la Sección 4.6 con un resumen y algunas conclusiones acerca de cuándo es mejor aplicar este método y cuándo será mejor el algoritmo Apriori o su modificación.

4.1 Algoritmos para la Obtención de *Itemsets* Frecuentes

El propósito de estos algoritmos es encontrar todos los conjuntos de *items* o "*itemsets*", cuyo soporte sobrepase el mínimo establecido, llamados *itemsets* frecuentes. Recordemos que el soporte de un *itemset* se calcula como la proporción de transacciones que contienen dicho *itemset*.

Los algoritmos más conocidos son el algoritmo "Apriori" y sus variantes "AprioriTid" y "AprioriHybrid" [Agr94].

Algoritmo Apriori

El algoritmo Apriori construye de forma iterativa el conjunto de términos frecuentes, utilizando los encontrados en un paso para construir los del paso siguiente. Para ello realiza múltiples lecturas de los datos.

En el primer paso se calcula el soporte de los *itemsets* elementales o de nivel uno y se determina cuáles se consideran frecuentes según el soporte mínimo. En cada paso posterior, se empieza con un conjunto "semilla" formado por los *itemsets* frecuentes encontrados en el paso anterior, los cuales se combinan entre sí para generar los *itemsets* candidatos. Para comprobar si estos *itemsets* candidatos son o no frecuentes, se requiere en cada paso realizar una nueva lectura de los datos.

En la Tabla 4.1 encontramos la notación necesaria para entender el algoritmo presentado a continuación.

k-itemsets	**Un *itemset* con *k items***
L_k	Conjunto de *k-itemsets* frecuentes Cada componente de este conjunto tiene dos campos: i) El *itemset* y ii) El soporte del *itemset*.
C_k	Conjunto de *k-itemsets* candidatos Cada componente de este conjunto tiene dos campos: i) El *itemset* y ii) El soporte del *itemset*.
$\overline{C_k}$	Conjunto de *k-itemsets* candidatos donde los identificadores (*TIDs*) de las transacciones se almacenan junto a los candidatos

Tabla 4.1: Notación para los Algoritmos Apriori y AprioriTid

Algoritmo Apriori:
L_1 = *itemsets* frecuentes de nivel 1;
for $(k = 2;\ L_{k-1} \neq \emptyset;\ k++)$ **do begin**
  C_k = apriori-gen(L_{k-1}); // Nuevos candidatos
  forall transacciones $t \in D$ **do begin**
    C_t = subconjunto (C_k, t); // Candidatos en t
    forall candidatos $c \in C_t$ **do**
      c.count++;
  end
  $L_k = \{c \in C_k | c.count \geq minsup\}$
end
Resultado = $\bigcup_k L_k$;

La función **apriori-gen** es la encargada de generar nuevos *itemsets* candidatos. Su funcionamiento se detalla en la Tabla 4.2.

Podemos ver un ejemplo práctico de la aplicación del algoritmo Apriori en la Sección 3.5.

Algoritmo AprioriTid

El algoritmo AprioriTid tiene la propiedad de que no es necesario recorrer toda la base de datos para calcular el soporte de los *itemsets* después del primer paso.

Para ello, se realiza una codificación de los encontrados en el paso previo, antes de decidir si son frecuentes en el paso posterior. En pasos sucesivos, el tamaño de esta codificación se va haciendo menor que el de la base de datos, ahorrando mucho esfuerzo de lectura.

Esta codificación trabaja de forma similar a la de un índice invertido.

Después de cada paso, no se emplean las transacciones en las que se encuentran los *itemsets* frecuentes, sino una codificación de éstos y si la transacción no contiene ningún *itemset* frecuente en el paso actual, no se considera en los pasos siguientes, mientras que en el algoritmo Apriori considera todas las transacciones en cada paso.

Este algoritmo también utiliza la función "apriori-gen" para determinar los *itemsets* candidatos.

Algoritmo AprioriTid:
L_1 = *itemsets* frecuentes de nivel 1;
$\overline{C}_1$ = base de datos D;
for ($k = 2;\ L_{k-1} \neq \emptyset;\ k++$) **do begin**
  C_k = apriori-gen(L_{k-1}); // Nuevos candidatos
  $\overline{C}_k = \emptyset$;
  forall entradas $t \in \overline{C}_{k-1}$ **do begin**
    // Determinar los *itemsets* candidatos en C_k contenidos en la transacción con identificador $t.TID$
    $C_t = \{c \in C_k | (c - c[k]) \in t.(conjunto\ de\ items) \wedge (c - c[k-1]) \in t.(conjunto\ de\ items)\}$;
    forall candidatos $c \in C_t$ **do**
      c.count++;
    if ($C_t \neq \emptyset$) **then** $\overline{C}_k += < t.TID, C_t >$;
  end
  $L_k = \{c \in C_k | c.count \geq minsup\}$
end
Resultado = $\bigcup_k L_k$;

En lugar de usar la base de datos D para el cálculo del soporte, se utiliza el conjunto C_k. Cada miembro de este conjunto es de la forma $< TID, \{X_k\} >$, donde cada X_k es un *itemset* potencialmente frecuente en la transacción con identificador TID:

Los experimentos realizados por Agrawal y Srikant [Agr94] muestran que el rendimiento del algoritmo Apriori es superior al rendimiento del algoritmo AprioriTid, debido a que en los primeros pasos el Apriori funciona mejor que el AprioriTid.

Nota.- En el caso de emplear estos algoritmos para la obtención de secuencias frecuentes (*item-seqs*) es necesario utilizar un índice invertido completo o lista invertida en la que se almacena también la posición de las palabras en el texto.

Algoritmo AprioriHybrid

El algoritmo AprioriHybrid es una combinación del Apriori y del AprioriTid para mejorar la escalabilidad y el rendimiento. Consiste en usar el algoritmo Apriori para los primeros pasos y cambiar al AprioriTid en los siguientes. Pero este cambio entre algoritmos también conlleva un coste.

El algoritmo AprioriHybrid funcionará mejor que el Apriori dependiendo de la base de datos en que se aplique.

Otros Algoritmos

Si las listas invertidas están disponibles, pueden usarse algoritmos como el algoritmo de "Partición" o el "Eclat".

El **algoritmo de Partición** [Sav95] necesita recorrer la base de datos como mucho dos veces para generar todas las reglas de asociación importantes.

Se ejecuta en dos etapas: en la primera se divide la base de datos en particiones sin intersecciones comunes, cuyo tamaño se calcula de forma que cada partición pueda acomodarse bien en la memoria principal y se generan los *itemsets* frecuentes de estas particiones. Al final de esta etapa, estos *itemsets* se combinan para generar el conjunto de *itemsets* candidatos a ser frecuentes. En la segunda etapa, se calcula el soporte de estos *itemsets* para identificar los frecuentes, con lo que las particiones se leen sólo una vez en cada etapa.

El algoritmo de Partición funciona mejor que el Apriori para soportes pequeños. Su principal problema es que, conforme se incrementa el número de particiones, el número de *itemsets* que son local pero no globalmente frecuentes, también se incrementa. Si se aleatorizan las particiones, estas tendrán un gran número de *itemsets* frecuentes en común, consumiendo gran cantidad de tiempo en identificar las redundancias.

En Zaki et al. [Zak97] se presentan seis algoritmos que recorren la base de datos solamente una vez. Se caracterizan por el empleo de técnicas de *clustering* para agrupar los *itemsets* relacionados y se diferencian según la técnica empleada. El mejor de estos algoritmos es **Eclat**, que usa únicamente intersecciones simples para generar los *itemsets* globalmente frecuentes, evitando la identificación de redundancias y no trabaja con varios *clusters* a la vez.

Otro algoritmo muy conocido es **FP-Growth** [Han00]. A diferencia de los anteriores, no adopta la forma del algoritmo Apriori para la generación de los *itemsets* candidatos, sino que se sirve de tres técnicas:

- La compresión de la base de datos en otra condensada, mucho más pequeña, llamada "árbol de patrones frecuentes (*FP-tree*)", lo que evita el coste de recorrer repetidas veces la base de datos original.

- La creación de una estructura de árbol que adopta un método mediante el cual se examinan únicamente los patrones contenidos en una "sub-base de datos" o base de datos condicional, que contiene el conjunto de *items* frecuentes que co-ocurren, evitando el coste de generar un gran número de *itemsets* candidatos.

- El uso de un método, "divide y vencerás", que descompone la tarea de generar los *itemsets* en pequeñas tareas de establecer patrones en las bases de datos condicionales, lo que reduce el espacio de búsqueda.

En Hipp et al. [Hip00] se realiza una comparación de todos estos algoritmos en cuanto a rendimiento, comprobándose que todos tienen un comportamiento similar con respecto al tiempo de ejecución. Mientras algunos algoritmos como Eclat y el algoritmo de Partición emplean la mayor parte del tiempo en determinar el soporte

de los *itemsets* candidatos de nivel inferior a cuatro, el algoritmo Apriori encuentra la mayor dificultad en el cálculo del soporte de los *itemsets* de nivel cuatro o superior.

Como el nivel de los *itemsets* frecuentes encontrados en nuestros experimentos suele ser inferior a cuatro y, en cualquier caso, no muy superior, nosotros empleamos el algoritmo Apriori, que además es el más conocido y resulta fácil de usar y implementar.

4.2 Algoritmos para la Obtención de Secuencias Frecuentes

La diferencia entre un *itemset* y una secuencia es que la secuencia es un conjunto ordenado de elementos o *items* (*item-seq*), mientras que en un *itemset* el orden de los elementos no importa.

En el campo de la obtención de secuencias frecuentes destacan trabajos como el de Agrawal y Srikant [Agr95] para el descubrimiento de patrones secuenciales en las bases de datos transaccionales. Estos autores proponen tres métodos basados en el algoritmo Apriori: el "AprioriAll", el "AprioriSome" y el "DinamicSome". Los elementos en los patrones secuenciales obtenidos por estos algoritmos no tienen por qué ser contiguos en el texto, lo que los diferencia de los patrones secuenciales que obtenemos nosotros para la formación de las *item-seqs* frecuentes que componen la estructura APO.

El algoritmo **AprioriAll** considera todas las secuencias frecuentes, incluidas las no maximales, mientras que el algortimo **AprioriSome** y el algoritmo **DinamicSome** sólo consideran las maximales, evitando las incluidas en éstas, comenzando así la búsqueda de secuencias por aquellas de mayor longitud. Sin embargo, el tiempo que se ahorran estos dos últimos al no considerar las secuencias contenidas en otras maximales, puede ser menor que el tiempo consumido considerando secuencias que tienen soporte inferior al mínimo, que nunca se hubieran tenido en cuenta, ya que sus subsecuencias no son frecuentes como consecuencia del bajo valor de su soporte. Por ello el algoritmo AprioriAll se considera el más eficiente de los tres. Éste fue el primer algoritmo en minería de patrones secuenciales.

El principal inconveniente que posee es que, igual que con el algoritmo Apriori, deben realizarse múltiples lecturas sobre la base de datos y se generan muchos candidatos; aún así es la base de un gran número de algoritmos eficientes que se han desarrollado con posterioridad.

El algoritmo AprioriAll consiste primero en generar aquellas secuencias que podrían ser frecuentes, denominadas ”secuencias candidatas“. Posteriormente, se escanea la base de datos para determinar el soporte de estas secuencias y ver si son consideradas frecuentes. El proceso de generación de las secuencias candidatas es similar al ”AprioriGen“ que hemos visto para la generación de *itemsets* frecuentes [Agr94] (ver Sección 4.1). La propiedad Apriori también se usa para eliminar aquellas secuencias candidatas cuyas subsecuencias no son frecuentes. La diferencia con el algoritmo Apriori es que al generar el candidato mediante la unión de los patrones frecuentes en el paso previo, un orden diferente en la combinación de los elementos daría lugar a diferentes candidatos.

Otro algoritmo muy conocido es el algoritmo **GSP** (Patrones Secuenciales Generalizados) [Sri96]. También está basado en el Apriori, pero además integra restricciones de tiempo y relaja la definición de transacción, a la vez que considera el conocimiento adquirido a través de taxonomías. Con este algoritmo resulta difícil el cálculo del soporte de las secuencias.

El algoritmo **PrefixSpan** [Pei01] es más eficiente que el algoritmo GSP, ya que puede tratar con bases de datos muy extensas y no requiere la generación de secuencias candidatas, sin embargo, necesita una mayor espacio de almacenamiento y el coste temporal de ejecución es mayor que con el anterior.

El algoritmo **SPADE** [Zak01] utiliza técnicas de búsqueda en retículos y uniones simples. Todas las secuencias se descubren con sólo tres recorridos sobre la base de datos. Descompone el principal problema en pequeños sub-problemas, que pueden ser más fácilmente almacenados en la memoria principal. Con este enfoque, la base de datos secuencial se transforma en un lista vertical de identificadores asociados a los *items* correspondientes. Trabaja del mismo modo que lo haría una lista invertida y también emplea la propiedad Apriori.

Un algoritmo que sólo requiere una lectura sobre la base de datos o como mucho dos en bases de datos extensas, es el algoritmo **MEMISP** (Indexación de Memoría para Minería de Patrones Secuenciales) [Lin02]. Para ello utiliza una búsqueda

recursiva y una estrategia de indexación, por lo que también funcionaría como un índice invertido.

Podemos encontrar una revisión de estos algoritmos en [Zha03].

El algoritmo que más se ajusta a nuestras necesidades y modo de trabajo para la generación de las *item-seqs* frecuentes, es el algoritmo AprioriAll [Agr95], pero como nosotros pretendemos que los elementos que componen las *item-seqs* aparezcan contiguos en el texto, realizamos una modificación distinta del Apriori. En la Sección 4.3 vemos en qué consiste esta modificación.

4.3 Algoritmos Apriori y Apriori Modificado para la Generación de *Item-seqs* Frecuentes

Aunque ya hemos visto el algoritmo Apriori dado por Agrawal et al. [Agr94] en la Sección 4.1 para la generación de *itemsets* frecuentes, volvemos a enunciarlo de forma diferente en la Tabla 4.2 junto con el Apriori modificado para la generación de *item-seqs* frecuentes.

La diferencia básica entre ambos está en la etapa de generación de nuevos candidatos. Mientras que para la generación de los *itemsets* candidatos en un paso se combinan los *itemsets* frecuentes del paso anterior que tengan todos los elementos comunes menos uno, sin importar el orden de estos elementos, para la generación de las *item-seqs* candidatas en un paso, se combinan las *item-seqs* frecuentes del paso anterior que tienen todos los elementos comunes menos uno y además, estos elementos conservan la misma relación de orden, por lo que una *item-seq* frecuente en L_{k-1} podrá combinarse con otra, sólo en el caso de que ésta última empiece por la subsecuencia de nivel $k-2$ con que termina la primera.

La notación para entender la Tabla 4.2 es la misma que teníamos en la Tabla 4.1.

Vemos que la eliminación de los *itemsets* candidatos que contengan algún subconjunto no frecuente en la etapa anterior L_{k-1} es un paso extra que no está en la generación de las *item-seqs* candidatas, ya que por la forma en que se construyen,

<table>
<tr><th colspan="2">Algoritmos Apriori y Apriori Modificado</th></tr>
<tr><td colspan="2">Paso 1
1. Generar los itemsets (item-seqs) candidatos en C_1
2. Almacenar los itemsets (item-seqs) frecuentes en L_1</td></tr>
<tr><td colspan="2">Paso k
1.Generar los itemsets (item-seqs) candidatos en C_k a partir de los itemsets (item-seqs) frecuentes en L_{k-1}:</td></tr>
<tr><td>Generación de itemsets
a) Unir $L_{k-1}p$ con $L_{k-1}q$ como sigue:
insert into C_k
select
$p.item_1, p.item_2, ..., p.item_{k-2}, p.item_m, q.item_n$
from $L_{k-1}p$, $L_{k-1}q$
where
$\forall\, p.item_i\ \exists\, q.item_j\ \ i,j \in [1,k-2]\ t.q$
$p.item_i = q.item_j\ y \nexists\ i,j \in [1,k-1]$
$t.q.\ p.item_m = q.item_j;\ q.item_n = p.item_i$
b) Eliminar los itemsets candidatos en C_k
donde algún subconjunto no sea un itemset de L_{k-1}
c) Eliminar redundancias</td><td>Generación de item-seqs
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) Eliminar redundancias</td></tr>
<tr><td colspan="2">2. Leer las transacciones de la base de datos para determinar el soporte de cada itemset (item-seq) en C_k
3. Almacenar los itemsets (item-seqs) frecuentes en L_k</td></tr>
</table>

Tabla 4.2: Algoritmos Apriori y Apriori modificado

éstas sólo pueden dividirse en dos subsecuencias para un nivel inferior, que son las dos subsecuencias en L_{k-1} cuya unión da la *item-seq* en C_k, luego serán frecuentes.

Por otro lado, el paso que consiste en la eliminación de redundancias, no se realiza de igual forma en ambos casos. En el caso de los *itemsets*, se eliminarían aquellos que tienen los mismos elementos, independientemente del orden en que aparezcan y en el caso de las *item-seqs*, se eliminarían aquellas que tengan los mismos elementos en el mismo orden y aquellas en que aparezca algún elemento repetido en una misma *item-seq*, algo que en los *itemsets* no es posible que ocurra.

Por otra parte, cuando se escanean las transacciones de la base de datos para determinar el soporte de cada conjunto o secuencia en C_k, las frecuencias para el cálculo de este soporte no se contabilizan igual si estamos buscando conjuntos o secuencias, ya que en los conjuntos los elementos no tienen por qué estar ordenados y en las secuencias sí.

El siguiente ejemplo ilustra las diferencias entre el proceso de generación de los *itemsets* y de generación de las *item-seqs*.

Ejemplo 4.3.1. Generación de *itemsets* e *item-seqs*

1. Supongamos que $L_2 = \{a, b\}, \{a, c\}, \{m, a\}$
   Generación de *itemsets* $\Rightarrow C_3 = \{a, b, c\}, \{a, b, m\}, \{a, c, m\}$
   Generación de *item-seqs* $\Rightarrow C_3 = \{m, a, b\}, \{m, a, c\}$

2. Supongamos que $L_3 = \{x, y, z\}, \{v, y, x\}$
   Generación de *itemsets* $\Rightarrow C_4 = \{x, y, z, v\}$
   Generación de *item-seqs* $\Rightarrow C_4 = \emptyset$

3. Supongamos que $L_1 = \{3\}, \{5\}, \{7\}$
   Generación de *itemsets* $\Rightarrow C_2 = \{3, 5\}, \{3, 7\}, \{5, 7\}$
   Generación de *item-seqs* $\Rightarrow C_2 = \{3, 5\}, \{5, 3\}, \{3, 7\}, \{7, 3\}, \{5, 7\}, \{7, 5\}$

En la Sección 3.5 se explican estos procesos de forma maś detallada.

4.4 Índices Invertidos

Ya hemos visto como muchos de los algoritmos mencionados utilizan índices invertidos o listas invertidas con la intención de recortar el tiempo de ejecución, como son AprioriTid, AprioriHybrid o SPADE. La mayoría emplean la propiedad Apriori.

Según Tomasic et al.[Tom94], la lista invertida se construye para una determinada palabra o cadena y el indíce invertido es el conjunto de todas las listas invertidas para todas las palabras que aparecen en un documento, junto al número de veces que aparecen.

El inconveniente de trabajar con índices invertidos es el del almacenamiento en bases de datos muy extensas. En la lista referencial de índice invertido hay un valor que se corresponde con cada *item* en un *itemset*, con lo que el espacio de almacenamiento del índice invertido es casi el mismo que el del conjunto de *itemsets* maximales si no se aplica ninguna medida de reducción. Si se almacena tanto el

índice invertido como el conjunto de *itemsets* maximales, el espacio de almacenamiento será mayor.

Araújo et al. [Ara97] presentan un índice invertido completo para el acoplamiento de cadenas en textos largos que se compone de una tabla con todas las palabras del texto y una lista con la posición que corresponde a cada palabra. El algoritmo que presentan estos autores permite la búsqueda de frases, conjuntos de caracteres, etc. de forma exacta, así como la búsqueda aproximada con coste no uniforme y expresiones regulares arbitrarias.

Este algoritmo emplea un tiempo de procesamiento de consulta de orden raíz de n, "$(O(n1/2))$", alrededor de 7 segundos para una consulta de 3 términos.

Para crear el índice, Araújo et al. [Ara97] consideran el texto como una secuencia de palabras, separadas por delimitadores y se almacena cada ocurrencia de cada palabra en una lista que mantiene el orden que ocupan las palabras en el texto. Cuando se realiza una consulta de una sola palabra, simplemente se busca en la tabla y en la lista de ocurrencias. Cuando la consulta es de más de una palabra, se busca en la tabla cada una de ellas y se recupera la correspondiente lista de ocurrencias, luego se obtiene la intersección de la lista mirando los punteros que tienen la misma posición relativa en la consulta y en el texto.

El tiempo que se tarda en construir el índice es directamente proporcional al tamaño del texto, cerca de 1 minuto por cada 4 Mb.

Qiao and Zhang [Qia12] han propuesto soluciones a las desventajas del índice invertido, construyendo su mapa de bits y usándolo para realizar las tareas de acoplamiento, con lo que han ahorrado mucho tiempo y espacio. Sin embargo, esta solución sólo ofrece mejoras en bases de datos extensas.

En el siguiente apartado estableceremos los conceptos de lista invertida e índice invertido completo. Estos conceptos nos serán útiles para la Sección 4.5, en la que propondremos una forma alternativa de calcular las estructuras WAP y WAPO con la ayuda de un índice invertido y de algunas de las definiciones que se dan a continuación.

4.4.1 Lista Invertida

Definición 4.4.1. (Patil et al. [Pat11]) Lista invertida para un término en una colección de documentos
Sea $D = \{d_1, d_2, ..., d_{\|D\|}\}$ *una colección de* n *documentos de un alfabeto* Σ*. La lista invertida asociada a un término* u *consiste en pares de la forma* $(d_j, puntuacion(u, d_j))$ *para* $j = 1, 2, 3, ..., \|D\|$*, donde la puntuación de un documento* d_j *depende del número de ocurrencias de* u *en* d_j*.*

En esta definición no se especifica la posición del término dentro del documento.

Khancome y Boonjing [Kha07] dan la definición de lista invertida para una cadena. Nosotros vamos a adaptarla para los *items* de un *itemset* de un atributo textual.

Definición 4.4.2. Lista Invertida de los *items* de un *Itemset* I_i
Sea I_i *un itemset de un atributo textual de una base de datos, con* items $t_1, t_2, ..., t_m$ *de un conjunto referencial* X*. Sea* $\Theta = \theta_{a_{i,1}}, \theta_{b_{i,2}}, \theta_{c_{i,3}}, ..., \theta_{...i,m}$*, donde* $\theta_{n_{i,j}}$ *se corresponde con los* t_j *para* $j = 1, ...m$*, con* i *indicando la posición del itemset. Dado* $\Theta = \theta_{a_{i,1}}, \theta_{b_{i,2}}, \theta_{c_{i,3}}, ..., \theta_{...i,m}$ *de* I_i*, entonces la lista invertida de* Θ*, denotada como* L_Θ *es el conjunto definido como:*

$$L_\Theta = \{\theta_a :< i, 1 >, \theta_b :< i, 2 >, \theta_c :< i, 3 >, ..., \theta_{...} :< i, m >\} \tag{4.1}$$

Ejemplo 4.4.1. Lista Invertida de los *items* del *Itemset* I_i
Supongamos, que tras el preprocesamiento del texto, el *itemset* que ocupa la tercera posición en el atributo textual se compone de los siguientes *items*:

$I_3 = \{sol, nube, sol, brisa, nieve\}$ entonces:
$\theta_{a_{3,1}} = sol,$
$\theta_{b_{3,2}} = nube,$
$\theta_{c_{3,3}} = sol,$
$\theta_{d_{3,4}} = brisa,$
$\theta_{e_{3,5}} = nieve$
$\Rightarrow \Theta = sol_{3,1}\ \ nube_{3,2}\ \ sol_{3,3}\ \ brisa_{3,4}\ \ nieve_{3,5}$

La lista invertida para los *items* de $I_3 = \{sol, nube, sol, brisa, nieve\}$ sería:
$L_\Theta = \{sol :< 3,1 >, nube :< 3,2 >, sol :< 3,3 >, brisa :< 3,4 >, nieve : < 3,5 >\}$

A continuación adaptamos la definición de tabla de listas invertidas de Khancome y Boonjing [Kha07].

Definición 4.4.3. Tabla de Listas Invertidas
Una tabla de listas invertidas τ es un conjunto de pares ordenados $(\theta_\lambda, P_\lambda)$, donde $\theta_\lambda = \theta_{n_{i,j}}$ representa un item del itemset I_i y P_λ de θ_λ es un conjunto que contiene los elementos $< i, j >$ de θ_λ.

Ejemplo 4.4.2. Tabla de Listas Invertidas
Podemos ver la tabla de listas invertidas del *itemset* $I_3 = \{sol, nube, sol, brisa, nieve\}$ del Ejemplo 4.4.1 en la Tabla 4.3.

θ_λ	P_λ
sol	$< 3,1 >, < 3,3 >$
nube	$< 3,2 >$
brisa	$< 3,4 >$
nieve	$< 3,5 >$

Tabla 4.3: Tabla de listas invertidas

4.4.2 Índice Invertido Completo

El índice invertido consistirá en el conjunto de las listas invertidas de todos los *items* de los *itemsets* del atributo textual.

Empezaremos estableciendo la notación para la definición de índice invertido completo (4.4) e introduciendo unos conceptos que se usarán en ésta.

Definición 4.4.4. Prefijo y Sufijo de un *itemset* Θ
Si $\Theta = \rho\theta\kappa$ para los itemsets $\rho, \kappa, \theta \in Sub(S)$, $\Theta \in S \Rightarrow "\rho"$ es un prefijo de Θ y "κ" es un sufijo de Θ.

Notación	Concepto
Σ	Alfabeto finito no vacío
Σ^*	Conjunto de todos los *itemsets* sobre $\sum$
ϕ	*Itemset* vacío
Σ^+	$\Sigma^* - \{\phi\}$
S	Conjunto finito de $\Theta \ \ \forall\, \Theta \ \in \ \sum^*$
$Sub(S)$	Conjunto de todos los sub-*itemsets* de Θ, $\forall\, \Theta \in S$

Tabla 4.4: Notación usada en la definición de índice invertido completo

La definición formal de índice invertido completo que incluimos a continuación, es una adaptación de la proporcionada por Blumer et al.[Blu87].

Definición 4.4.5. Índice Invertido Completo
Dado un alfabeto finito Σ, un conjunto de items $k \subseteq \Sigma^+$ y un conjunto de itemsets $S \subseteq \Sigma^+$, un índice invertido completo para (Σ, k, S) es un tipo de dato abstracto que implementa las siguientes funciones:

1. **find**: $\Sigma^+ \ \rightarrow \ k \ \cup \ \{\phi\}$, *donde find(θ) es el mayor itemset* $\Theta = \rho\theta\kappa$ *que contiene* θ *con* $\rho, \kappa \ \in \ k \cup \{\phi\}$ *y* Θ *ocurre en* S.

2. **freq**: $k \rightarrow \mathbb{N}$, *donde freq(θ) es el número de veces que* θ *ocurre en* S.

3. **loc**: $k \rightarrow 2^{N \times N}$, *donde loc(θ) es el número de pares ordenados indicando el número del itemset en que ocurre* θ *y su posición dentro de éste.*

4.4.3 Obtención de las Estructuras WAP y WAPO a través de Índices Invertidos

Es posible obtener las estructuras WAP y WAPO a partir de los índices invertidos principalmente de dos formas:

1. Haciendo uso del algoritmo Apriori, de forma similar a como lo hacen los algoritmos AprioriTid y AprioriHybrid [Agr94], realizando una codificación que permita recorrer los índices invertidos en lugar de toda la base de datos para la aplicación del algoritmo Apriori y desechando las posiciones de los *items* que no se corresponden con un *itemset* frecuente en el paso anterior.

2. Descubriendo primero los *itemsets* maximales y descomponiendo aquellos que no sean frecuentes en sub-*itemsets* que sí lo sean, hasta que nos hayamos quedado con un conjunto de *itemsets* frecuentes del mayor nivel posible, los cuales se corresponderán con los conjuntos generadores de la estructura que pretendamos obtener.

   En la Sección 4.5 se explica cómo se aplicaría este método paso a paso, para lo que se empieza definiendo algunos conceptos necesarios y se acaba con un ejemplo para su comprensión.

4.5 Método Alternativo al Algoritmo Apriori para la Generación de las Estructuras

Comenzamos esta sección introduciendo unos conceptos que serán necesarios posteriormente.

4.5.1 Regla, Regla Primaria e Implicación

Blumer et al. [Blu87] definen los conceptos de regla, regla primaria e implicación, nosotros los adaptaremos presentando seis definiciones, distinguiendo para cada uno de estos conceptos dos casos diferentes:

1. Cuando los *items* de un *itemset* mantengan una relación de orden (que sería el caso de las *item-seqs*),

2. Cuando no importe el orden que presenten los *items* dentro de los *itemsets*.

Posteriormente, daremos las definiciones de regla primaria frecuente e implicación frecuente para los dos casos arriba indicados.

Definición 4.5.1. Regla de S
Una regla de S es una aplicación:

$$\theta \rightarrow \Gamma \quad \textit{donde} \quad \theta \in Sub(S), \Gamma \in S \tag{4.2}$$

que ocurre cada vez que los elementos de θ están contenidos en Γ ($\theta \subseteq \Gamma$).

Definición 4.5.2. Regla ordenada de S

Una regla ordenada de S es una aplicación:

$$\theta \rightarrow \rho\theta\kappa \quad \textit{donde} \quad \theta, \gamma, \beta \ \in \ Sub(S) \tag{4.3}$$

que ocurre cada vez que θ está precedido por ρ y seguido por κ en S.

Definición 4.5.3. Regla Primaria de S

$\theta \rightarrow \Gamma$ es una regla primaria de S si es una regla de S y $\Gamma \in S$ es un itemset maximal, esto es:

$$\nexists\, \Gamma' \ \in S \textit{ tal que } \ \Gamma \subset \Gamma' \tag{4.4}$$

o lo que es lo mismo

$$\nexists\, \Gamma' \ \in S \textit{ tal que } \ \theta \rightarrow \Gamma' \textit{ sea un regla de } S, \textit{ con } \Gamma \subset \Gamma' \tag{4.5}$$

Definición 4.5.4. Regla Ordenada Primaria de S

$\theta \rightarrow \rho\theta\kappa$ es una regla ordenada primaria de S si es una regla ordenada de S y ρ y $\kappa \ \in \ Sub(S)$ son itemsets del mayor orden posible, esto es:

$$\nexists\, \delta, \epsilon \ \in \ Sub(S) \ \textit{ con } \ \delta, \epsilon \neq \phi \ \textit{ tales que } \ \theta \rightarrow \delta\rho\theta\kappa\epsilon \ \textit{ sea una regla ordenada de } S. \tag{4.6}$$

Definición 4.5.5. Implicación de θ en S

Si $\theta \rightarrow \Gamma$ es una regla primaria de S, entonces Γ se llama implicación de θ en S y se denota $imp_S(\theta)$, siendo $P(S)$ el conjunto de todas las implicaciones en S:

$$P(S) = \{imp_S(\theta) : \theta \ \in Sub(S)\} \tag{4.7}$$

Definición 4.5.6. Implicación Ordenada de θ en S

Si $\theta \rightarrow \rho\theta\kappa$ es una regla ordenada primaria de S, entonces $\rho\theta\kappa$ se llama implicación ordenada de θ en S y se denota $impo_S(\theta)$, siendo $Po(S)$ el conjunto de todas las implicaciones ordenadas en S:

$$Po(S) = \{impo_S(\theta) : \theta \ \in Sub(S)\} \tag{4.8}$$

Definición 4.5.7. Regla Frecuente Primaria de S

Sea $Subf(S)$ el conjunto de todos los sub-itemsets frecuentes de Θ, $\forall\Theta \in S$. Diremos que $\theta \rightarrow \Gamma$ es una regla frecuente primaria de S, si es una regla de S y Γ es un itemset maximal en $Subf(S)$, esto es:

$$\nexists\, \Gamma' \text{ tal que } \Gamma \subset \Gamma' \text{ con } \Gamma, \Gamma' \in Subf(S) \tag{4.9}$$

o lo que es lo mismo

$$\nexists\, \Gamma' \text{ tal que } \theta \rightarrow \Gamma' \text{ sea un regla de } S, \text{ con } \Gamma \subset \Gamma',\ \Gamma, \Gamma' \in Subf(S) \tag{4.10}$$

Definición 4.5.8. Regla Ordenada Frecuente Primaria de S

Sea $Subof(S)$ el conjunto de todas las subsecuencias frecuentes de Θ, $\forall\Theta \in S$. Diremos que $\theta \rightarrow \rho\theta\kappa$ es una regla ordenada frecuente primaria de S, si es una regla ordenada de S, pertenece al conjunto $Subof(S)$ y ρ y $\kappa \in Subof(S)$ son item-seqs del mayor orden posible, esto es:

$$\nexists\, \delta, \epsilon \in Subof(S) \text{ con } \delta, \epsilon \neq \phi \text{ tales que } \theta \rightarrow \delta\rho\theta\kappa\epsilon \text{ sea una regla ordenada de } S \text{ y } \delta\rho\theta\kappa\epsilon \text{ pertenezca al conjunto } Subof(S) \tag{4.11}$$

Definición 4.5.9. Implicación Frecuente de θ en S

Si $\theta \rightarrow \Gamma$ es una regla frecuente primaria de S, entonces Γ se llama implicación frecuente de θ en S y se denota $imp_S^f(\theta)$, siendo $P^f(S)$ el conjunto de todas las implicaciones frecuentes en S:

$$P^f(S) = \{imp_S^f(\theta) : \theta \in Subf(S)\} \tag{4.12}$$

Definición 4.5.10. Implicación Ordenada Frecuente de θ en S

Si $\theta \rightarrow \rho\theta\kappa$ es una regla ordenada frecuente primaria de S, entonces $\rho\theta\kappa$ se llama implicación ordenada frecuente de θ en S y se denota $impo_S^f(\theta)$, siendo $Po^f(S)$ el conjunto de todas las implicaciones ordenadas frecuentes en S:

$$Po^f(S) = \{impo_S^f(\theta) : \theta \in Subof(S)\} \tag{4.13}$$

4.5.2 Proceso de obtención de estructuras a partir de implicaciones frecuentes

Podemos obtener las estructuras WAP y WAPO identificando las implicaciones de $\theta = \{I_j\}$ con las secuencias generadoras de la Estructura APO, siendo I_j cada uno de los *itemsets* frecuentes de nivel uno en S conjunto finito de *itemsets*. El problema es que tendríamos que identificar aquellas implicaciones que, a su vez, fueran frecuentes.

El proceso completo consistiría en obtener previamente todas las reglas de θ y seleccionar únicamente las frecuentes. Aquellas no frecuentes, se dividen en subreglas hasta que lo sean. De entre estas reglas frecuentes, estudiaremos cuáles son reglas primarias y una vez que se tengan las reglas frecuentes primarias de θ, éstas serán las implicaciones frecuentes de θ en S. El último paso, consiste en identificar cada una de estas implicaciones frecuentes con un conjunto o secuencia generadora de la estructura buscada.

Correspondencia entre las Implicaciones Frecuentes y los Conjuntos Generadores de la Estructura WAP

Sea $P^f(S) = \{imp_S^f(\theta) : \theta \in Subf(S)\}$ el conjunto de todas las implicaciones frecuentes en S. Sea $\widetilde{T} = \widetilde{g}(A, B, ...)$ la estructura WAP generada por los conjuntos ponderados $\widetilde{A}, \widetilde{B}, ...$, entonces:

$$\begin{aligned} &A = imp_S^f(\theta_1), \;\; B = imp_S^f(\theta_2), ... \text{ eliminando redundancias,} \\ &\text{con } \theta_j = t_j, \text{ siendo } t_j \text{ itemset frecuente de nivel uno } \forall\, j = 1, 2, ... \end{aligned} \tag{4.14}$$

Correspondencia entre las Implicaciones Ordenadas Frecuentes y las Secuencias Generadoras de la Estructura WAPO

Sea $Po^f(S) = \{impo_S^f(\theta) : \theta \in Subf(S)\}$ el conjunto de todas las implicaciones ordenadas frecuentes en S. Sea $\widetilde{E} = \widetilde{g}(A, B, ...)$ la estructura WAPO generada por las secuencias ponderadas $\widetilde{A}, \widetilde{B}, ...$, entonces:

$$\begin{aligned} &A = impo_S^f(\theta_1), \;\; B = impo_S^f(\theta_2), ... \text{ eliminando redundancias,} \\ &\text{con } \theta_j = t_j, \text{ siendo } t_j \text{ item-seq frecuente de nivel uno } \forall\, j = 1, 2, ... \end{aligned} \tag{4.15}$$

En la Tabla 4.5 se recoge la nueva notación que se ha empleado para definir el proceso de obtención de las estructuras WAP y WAPO a través de implicaciones frecuentes.

Notación	Concepto
CMF	Conjunto de Monotérminos Frecuentes
CR	Conjunto de Reglas
CRO	Conjunto de Reglas Ordenadas
CRF	Conjunto de Reglas Frecuentes
$CROF$	Conjunto de Reglas Ordenadas Frecuentes
S'	Subconjunto de S obtenido tras eliminar en S los *itemsets* frecuentes
S''	Subconjunto de S obtenido tras eliminar en S las *itemseqs* frecuentes
$P^f(S)$	Conjunto de Implicaciones Frecuentes de S
$Po^f(S)$	Conjunto de Implicaciones Ordenadas Frecuentes de S
r_k	k-ésima regla
ro_k	k-ésima regla ordenada
r_k^f	k-ésima regla frecuente
ro_k^f	k-ésima regla ordenada frecuente
rp_k^f	k-ésima regla frecuente primaria
rop_k^f	k-ésima regla ordenada frecuente primaria

Tabla 4.5: Notación usada en la definición del proceso de obtención de las estructuras a partir de implicaciones frecuentes

Esquema Algorítmico del Proceso de Obtención de las Estructuras a través de Implicaciones Frecuentes

Σ = Alfabeto finito, k = conjunto de items, S = conjunto de itemsets, $k \subseteq S \Rightarrow$ (Σ, k, S) Índice invertido completo con funciones $freq(\theta), loc(\theta)$ y $find(\theta)$.

Partimos de S conjunto de itemsets:

1. *Identificar los monotérminos frecuentes según soporte:*
   Si $freq(t_j) > soporte,\ t_j \in k \ \Rightarrow\ t_j$ es un término frecuente $\Rightarrow t_j = \theta_j$

2. *Almacenar monotérminos frecuentes y su frecuencia en el Conjunto de Monotérminos Frecuentes:*

Sea CMF *el conjunto de monotérminos frecuentes* $\Rightarrow$ *Insertar* $(\theta_j, freq(\theta_j))$ *en* CMF

3. *Eliminar los monotérminos no frecuentes en* S *y almacenar resultado en* $S'(S'')$*:*

   a) **Estructura WAP**: *Si* $t_j \in k$ *y* $t_j \notin CMF \Rightarrow$ *Eliminar* t_j *y almacenar resultado en* S'

   b) **Estructura WAPO**:*Si* $t_j \in k$ *y* $t_j \notin CMF \Rightarrow$ *Eliminar* t_j *conservando el orden en las* item-seqs *resultantes y almacenar resultado en* S''

4. *Identificar reglas (reglas ordenadas) maximales de* $\theta_j \in CMF$ *en* $S'(S'')$*:*

   a) **Estructura WAP**: *Si* $\theta_j \subseteq I_i$; $\theta_j \in CMF$, $I_i \in S' \Rightarrow I_i$ *es una regla* (r_k) *de* S'

   b) **Estructura WAPO**: *Si* $\theta_j \subseteq I_i$; $\theta_j \in CMF$, $I_i \in S'' \Rightarrow I_i$ *es una regla ordenada* (ro_k) *de* S''

5. *Almacenar reglas (reglas ordenadas) junto a su frecuencia en el Conjunto de Reglas* (CR) *(Conjunto de Reglas Ordenadas* (CRO)*):*

   a) **Estructura WAP**: *Si* r_k *es una regla de* $S' \Rightarrow$ *Insertar* $(r_k, freq(r_k))$ *en* CR

   b) **Estructura WAPO**: *Si* ro_k *es una regla ordenada en* $S'' \Rightarrow$ *Insertar* $(ro_k, freq(ro_k))$ *en* CRO

6. *Eliminar redundancias en* $CR(CRO)$*:*

   a) **Estructura WAP**: *Si* $r_k = r_{k'}$ *con* $r_k, r_{k'} \in CR$ *y* $k \neq k' \Rightarrow$ *Eliminar* $(r_{k'}, freq(r_{k'}))$

   b) **Estructura WAPO**: *Si* $ro_k = ro_{k'}$ *con* $ro_k, ro_{k'} \in CRO$ *y* $k \neq k' \Rightarrow$ *Eliminar* $(ro_{k'}, freq(ro_{k'}))$

7. *Identificar reglas frecuentes en* $CR(CRO)$ *según soporte:*

   a) **Estructura WAP**: *Si* $freq(r_k) > soporte$, $r_k \in CR \Rightarrow r_k$ *es una regla frecuente* (r_k^f)

   b) **Estructura WAPO**: *Si* $freq(ro_k) > soporte$, $ro_k \in CRO \Rightarrow ro_k$ *es una regla ordenada frecuente* (ro_k^f)

8. *Almacenar las reglas frecuentes (reglas ordenadas frecuentes) junto a su frecuencia en el Conjunto de Reglas Frecuentes (CRF) (Conjunto de Reglas Ordenadas Frecuentes ($CROF$)):*

   a) **Estructura WAP**: *Si r_k^f es una regla frecuente de $S \Rightarrow$ Insertar $(r_k^f, freq(r_k^f))$ en CRF*

   b) **Estructura WAPO**: *Si ro_k^f es una regla ordenada frecuente de $S \Rightarrow$ Insertar $(ro_k^f, freq(ro_k^f))$ en $CROF$*

9. *Descomponer las reglas (reglas ordenadas) no frecuentes en CR/CRO en sub-reglas (sub-reglas ordenadas):*

   a) **Estructura WAP**: *Si $r_k \in CR$, $r_k \notin CRF$ y $r_k = I_i$ $itemset$ de nivel n $\Rightarrow$ Descomponer r_k en n sub-reglas (sub-*itemsets*) de nivel $n-1$: $r_{k1}, ..., r_{kn}$*

   b) **Estructura WAPO**: *Si $ro_k \in CRO$, $ro_k \notin CROF$ y $ro_k = I_i$* item-seq *de nivel n $\Rightarrow$ Descomponer ro_k en 2 sub-reglas ordenadas (sub-*item-seqs*) de nivel $n-1$: r_{k1}, r_{k2}*

10. *Volver al paso 5:*

    a) **Estructura WAP**: *Si $freq(r_k) < soporte$ para algún $k \Rightarrow$ Ir al paso 5.*

    b) **Estructura WAPO**: *Si $freq(ro_k < soporte$ para algún $k \Rightarrow$ Ir al paso 5.*

11. *Eliminar redundancias en $CRF(CROF)$:*

    a) **Estructura WAP**: *Si $r_k^f = r_{k'}^f$ con $r_k^f, r_{k'}^f \in CRF$ y $k \neq k' \Rightarrow$ Eliminar $(r_{k'}^f, freq(r_{k'}^f))$*

    b) **Estructura WAPO**: *Si $ro_k^f = ro_{k'}^f$ con $ro_k^f, ro_{k'}^f \in CROF$ y $k \neq k' \Rightarrow$ Eliminar $(ro_{k'}^f, freq(ro_{k'}^f))$*

12. *Identificar reglas frecuentes primarias (reglas ordenadas frecuentes primarias):*

    a) **Estructura WAP**: *Si $r_k^f : \theta_j \rightarrow S'$; $r_k^f \in CRF, S' \subseteq S$, cumple con la Definición 4.5.7 $\Rightarrow$ r_k^f es una regla frecuente primaria (rp_k^f) de S.*

    b) **Estructura WAPO**: *Si $ro_k^f : \theta_j \rightarrow S''$; $ro_k^f \in CROF, S'' \subseteq S$, cumple con la Definición 4.5.8 $\Rightarrow$ r_k^f es una regla ordenada frecuente primaria (rop_k^f) de S.*

13. *Almacenar reglas frecuentes primarias (reglas ordenadas frecuentes primarias) junto con su frecuencia en el Conjunto de Implicaciones Frecuentes de S ($P^f(S)$) (Conjunto de Implicaciones Ordenadas Frecuentes de S ($Po^f(S)$)):*

    a) **Estructura WAP**: *Si rp_k^f es una regla frecuente primaria $\Rightarrow$ Insertar (rp_k^f, $freq(rp_k^f)$) en $P^f(S)$*

    b) **Estructura WAPO**: *Si rop_k^f es una regla ordenada frecuente primaria $\Rightarrow$ Insertar (rop_k^f, $freq(rop_k^f)$) en $Po_k^f(S)$*

14. *Identificar cada una de las implicaciones frecuentes en $P^f(S)$ (implicaciones ordenadas frecuentes en $Po^f(S)$) con un conjunto generador de la estructura WAP (secuencia generadora de la estructura WAPO).*

    a) **Estructura WAP**: *Establecer correspondencias indicadas en la Definición 4.5.9*

    b) **Estructura WAPO**: *Establecer correspondencias indicadas en la Definición 4.5.10*

15. *Construir la estructura WAP (WAPO) a partir de sus conjuntos generadores (secuencias generadoras), siendo $freq(I_i)$ el peso de cada I_i* itemset (item-seq) *en la estructura WAP (WAPO).*

4.5.3 Ejemplo

En la Sección 3.5 vimos un ejemplo de obtención de las estructuras a partir del algoritmo Apriori y su modificación para la generación de las *item-seqs* en la estructura APO. Volvemos a usar en este ejemplo la muestra de titulares relacionados con el empleo recogidos en la Tabla 3.1. Los conjuntos de *itemsets* tras la limpieza de datos, se mostraban en la Tabla 3.3, que transcribimos aquí para una mejor comprensión (Tabla 4.6).

Lo que tratamos de hacer es volver a generar las estructuras para esta muestra de tuplas, pero en lugar de utilizar el algoritmo Apriori, lo haremos a partir de las implicaciones de los términos frecuentes de nivel uno, para lo que es necesario contar con el índice invertido completo.

Primero construiremos la imagen de la función **loc(θ)** para los monotérminos (ver Tabla 4.7). Y posteriormente, la imagen de la función **freq(θ)** (ver Tabla 4.8).

n	Itemsets
1	{*faltan, funcionarios, oficina, empleo*}
2	{*deterioro, empleo, funcionarios*}
3	{*empleo, oficina, mejor, valorado*}
4	{*funcionarios, abarrotan, oficina, empleo*}
5	{*disminución, empleo*}
6	{*funcionarios, van, oficina, empleo*}
7	{*empleo, funcionarios, peligra*}
8	{*disminución, empleo, sueldo, 2013*}
9	{*trabajadores, critican, empleo, oficina*}
10	{*empleo, verano, disminución, paro*}

Tabla 4.6: Conjunto de *itemsets* tras la limpieza

Con la función freq(θ) podemos saber cuándo los términos son frecuentes dependiendo del soporte mínimo establecido. En el ejemplo de la Sección 3.5 tomamos un soporte del 20 % que, en este caso, se corresponde con una frecuencia absoluta igual a dos, por lo que los términos frecuentes son los que sobrepasan esa frecuencia y coinciden con los que teníamos en la Tabla 3.7. Son los términos: {*funcionarios*}, {*oficina*}, {*empleo*} y {*disminución*}.

La obtención de los monotérminos frecuentes y su almacenamiento en el conjunto de monotérminos frecuentes, se correspondería con los pasos 1 y 2 del esquema algorítmico presentado en la Sección 4.5.

A partir del tercer paso en dicho esquema, distinguimos en el proceso según la estructura que queramos obtener. Empezaremos generando la estructura WAP y posteriormente, generaremos la estructura WAPO.

***(i)* Obtención de la Estructura WAP**

El paso 3 en la obtención de la estructura WAP consiste en eliminar los términos no frecuentes del atributo textual y almacenar el resultado en una nueva columna de la base de datos. En la Tabla 4.9 podemos ver cómo quedarían los *itemsets* en la nueva columna tras la eliminación de los términos no frecuentes en la columna original.

El objetivo ahora es calcular las implicaciones frecuentes de los *itemsets* frecuentes de nivel uno, para lo que comenzamos calculando las reglas y viendo cuáles

t_j	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9	I_{10}
faltan	$< 1,1 >$									
funcionarios	$< 1,2 >$	$< 2,3 >$		$< 4,1 >$		$< 6,1 >$	$< 7,2 >$			
oficina	$< 1,3 >$		$< 3,2 >$	$< 4,3 >$		$< 6,3 >$			$< 9,4 >$	
empleo	$< 1,4 >$	$< 2,2 >$	$< 3,1 >$	$< 4,4 >$	$< 5,2 >$	$< 6,4 >$	$< 7,1 >$	$< 8,2 >$	$< 9,3 >$	$< 10,1 >$
deterioro		$< 2,1 >$								
mejor			$< 3,3 >$							
valorado			$< 3,4 >$							
abarrotan				$< 4,2 >$						
disminución					$< 5,1 >$			$< 8,1 >$		$< 10,3 >$
van						$< 6,2 >$				
peligra							$< 7,3 >$			
sueldo								$< 8,3 >$		
2013								$< 8,4 >$		
trabajadores									$< 9,1 >$	
critican									$< 9,2 >$	
verano										$< 10,2 >$
paro										$< 10,4 >$

Tabla 4.7: Función $loc(\theta)$

θ_j	**loc(θ_j)**	$freq(\theta_j)$
faltan	$\{<1,1>\}$	1
funcionarios	$\{<1,2>,<2,3>,<4,1>,<6,1>,<7,2>\}$	5
oficina	$\{<1,3>,<3,2>,<4,3>,<6,3>,<9,4>\}$	5
empleo	$\{<1,4>,<2,2>,<3,1>,<4,4>,<5,2>,<6,4>,<7,1>,<8,2>,<9,3>,<10,1>\}$	10
deterioro	$\{<2,1>\}$	1
mejor	$\{<3,3>\}$	1
valorado	$\{<3,4>\}$	1
abarrotan	$\{<4,2>\}$	1
disminución	$\{<5,1>,<8,1>,<10,3>\}$	3
van	$\{<6,2>\}$	1
peligra	$\{<7,3>\}$	1
sueldo	$\{<8,3>\}$	1
2013	$\{<8,4>\}$	1
trabajadores	$\{<9,1>\}$	1
critican	$\{<9,2>\}$	1
verano	$\{<10,2>\}$	1
paro	$\{<10,4>\}$	1

Tabla 4.8: Función *freq(θ)*

n	*Itemsets* Originales S	*Itemsets* Resultantes S'
1	{*faltan, funcionarios, oficina, empleo*}	{*funcionarios, oficina, empleo*}
2	{*deterioro, empleo, funcionarios*}	{*empleo, funcionarios*}
3	{*empleo, oficina, mejor, valorado*}	{*empleo, oficina*}
4	{*funcionarios, abarrotan, oficina, empleo*}	{*funcionarios, oficina, empleo*}
5	{*disminución, empleo*}	{*disminución, empleo*}
6	{*funcionarios, van, oficina, empleo*}	{*funcionarios, oficina, empleo*}
7	{*empleo, funcionarios, peligra*}	{*empleo, funcionarios*}
8	{*disminución, empleo, sueldo, 2013*}	{*disminución, empleo*}
9	{*trabajadores, critican, empleo, oficina*}	{*empleo, oficina*}
10	{*empleo, verano, disminución, paro*}	{*empleo, disminución*}

Tabla 4.9: *Itemsets* resultantes tras la eliminación de términos no frecuentes

son primarias frecuentes.

Las tareas 4 y 5 que se indican en el esquema algorítmico consisten en calcular las reglas de los monotérminos frecuentes y almacenarlas en el conjunto de reglas CR, junto con su frecuencia. En la Tabla 4.10 podemos ver lo que sería el resultado de ejecutar estas tareas. La primera columna se añade para entender de dónde procede cada una de las reglas.

θ_j	k	$r_k(\theta_j)$	$freq(r_k(\theta_j))$
{*funcionarios*}	1	{*funcionarios, oficina, empleo*}	3
	2	{*empleo, funcionarios*}	5
{*oficina*}	3	{*funcionarios, oficina, empleo*}	3
	4	{*empleo, oficina*}	5
{*empleo*}	5	{*funcionarios, oficina, empleo*}	3
	6	{*empleo, funcionarios*}	5
	7	{*empleo, oficina*}	5
	8	{*disminución, empleo*}	3
{*disminución*}	9	{*disminución, empleo*}	3

Tabla 4.10: Conjunto de reglas (CR) de los *itemsets* frecuentes de nivel uno

Eliminando las reglas repetidas (paso 6), el conjunto de reglas quedaría como se muestra en la Tabla 4.11.

Como todas las reglas son frecuentes según el soporte establecido, no se tiene que descomponer ninguna de ellas y todas son almacenadas en el conjunto de reglas

k	r_k	$freq(r_k)$
1	{*funcionarios, oficina, empleo*}	3
2	{*empleo, funcionarios*}	5
3	{*empleo, oficina*}	5
4	{*disminución, empleo*}	3

Tabla 4.11: Conjunto de reglas (CR) tras la eliminación de redundancias

frecuentes (CRF), por lo que pasaríamos directamente al paso 12, que consiste en identificar que reglas son primarias, es decir, cuáles no están contenidas en otra regla.

Como vemos, r_2 y r_3 están contenidas en r_1, por lo que r_1 será regla primaria y r_2 y r_3 no lo serán. En la Tabla 4.12 podemos ver lo que sería el conjunto de reglas frecuentes primarias, o lo que es lo mismo, el conjunto de implicaciones frecuentes $(P^f(S))$ (paso 13).

k	rp_k^f	$freq(rp_k^f)$	imp_k
1	{*funcionarios, oficina, empleo*}	3	imp_1
2	{*disminución, empleo*}	3	imp_2

Tabla 4.12: Conjunto de reglas frecuentes primarias o implicaciones frecuentes $P^f(S)$

Identificar las implicaciones frecuentes a partir de las reglas primarias frecuentes:

- $imp_S(x_1) = imp(\{funcionarios\}) = \{funcionarios, oficina, empleo\}$
- $imp_S(x_2) = imp(\{oficina\}) = \{funcionarios, oficina, empleo\}$
- $imp_S(x_3) = imp(\{empleo\}) = \{funcionarios, oficina, empleo\}$ y $\{disminución, empleo\}$
- $imp_S(x_4) = imp(\{disminución\}) = \{disminución, empleo\}$

Las implicaciones frecuentes serán los conjuntos generadores de la estructura WAP, es decir, cada conjunto generador se corresponderá con una implicación del conjunto de implicaciones frecuentes (paso 14).

Sea $\widetilde{T}$ una estructura WAP, con conjuntos generadores $\widetilde{A}$ y $\widetilde{B}$:

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

Dichos conjuntos generadores se corresponden con las implicaciones encontradas en la Tabla 4.12:

$A = imp_1$ y $B = imp_2 \quad \Rightarrow$

$T = g(\{funcionarios, oficina, empleo\}, \{disminución, empleo\})$

Con lo que la estructura-AP resultante tendrá cardinal 9:

$T = (\{funcionarios, oficina, empleo\}, \{disminución, empleo\},$
$\{funcionarios, oficina\}, \{oficina, empleo\}, \{funcionarios\}, \{oficina\},$
$\{empleo\}$ y $\{disminución\}$

A partir de T obtenemos la estructura WAP:

$\widetilde{T} = \widetilde{g}([\{funcionarios, oficina, empleo\}, (3)], [\{empleo, disminución\}, (3)])$

El último paso del proceso consiste en hallar la estructura WAP completa a partir de sus conjuntos generadores. La única dificultad se encuentra en conocer el peso de todos los *itemsets* que la componen. Hallar el peso de cada *itemset* del retículo es tan fácil como hacer uso de las funciones loc(θ_j) y freq(θ_j), con θ_j = $I_j \in T$.

La función **find**(θ) del índice invertido completo nos será útil para recuperar la información a través de los índices invertidos mediante la consulta.

***(ii)* Obtención de la Estructura WAPO**

Una vez que se han llevado a cabo los pasos 1 y 2 del esquema del proceso de obtención de las estructuras a través de implicaciones y ya se tienen identificados y almacenados los monotérminos frecuentes: {*funcionarios*}, {*oficina*}, {*empleo*}

y {*disminución*}, el siguiente paso para la obtención de la estructura WAPO, sería eliminar el resto de términos y almacenar el resultado en una nueva columna de la base de datos, como hacíamos para la estructura WAP. La diferencia es que ahora importa el orden y la adyacencia de los términos en las *item-seqs*, por lo que la supresión de un término en el interior de una *item-seq*, daría lugar a dos *item-seqs* diferentes, ya que los términos en los extremos de cada una de ellas, no tendrían por qué ser adyacentes en el texto.

En la Tabla 4.13 podemos ver la columna original de *item-seqs* y la nueva columna tras la eliminación de los términos no frecuentes.

n	***Item-seqs* Originales S**	***Item-seqs* Resultantes S''**
1	{*faltan, funcionarios, oficina, empleo*}	{*funcionarios, oficina, empleo*}
2	{*deterioro, empleo, funcionarios*}	{*empleo, funcionarios*}
3	{*empleo, oficina, mejor, valorado*}	{*empleo, oficina*}
4	{*funcionarios, abarrotan, oficina, empleo*}	{*funcionarios*}, {*oficina, empleo*}
5	{*disminución, empleo*}	{*disminución, empleo*}
6	{*funcionarios, van, oficina, empleo*}	{*funcionarios*}, {*oficina, empleo*}
7	{*empleo, funcionarios, peligra*}	{*empleo, funcionarios*}
8	{*disminución, empleo, sueldo, 2013*}	{*disminución, empleo*}
9	{*trabajadores, critican, empleo, oficina*}	{*empleo, oficina*}
10	{*empleo, verano, disminución, paro*}	{*empleo*}, {*disminución*}

Tabla 4.13: *Item-seqs* resultantes tras la eliminación de los términos no frecuentes

A continuación se identifican las reglas ordenadas de los términos frecuentes y se almacenan en el conjunto de reglas ordenadas (CRO) junto a su frecuencia (pasos 4 y 5). Ver Tabla 4.14.

El conjunto de reglas ordenadas tras eliminar las repetidas (paso 6), puede verse en la Tabla 4.15.

Los pasos 7 y 8 consisten en identificar las reglas ordenadas frecuentes y almacenarlas en el conjunto de reglas ordenadas frecuentes $(CROF)$.

Como no todas las reglas ordenadas son frecuentes, en el paso 9 se descomponen aquellas que no lo son en sub-reglas ordenadas y se estudia la frecuencia de las sub-reglas. Hacemos esto solamente con la regla r_1, que es la única que resulta no frecuente según el soporte establecido (Tabla 4.16).

θ_j	k	$ro_k(\theta_j)$	$freq(ro_k(\theta_j))$
{*funcionarios*}	1	{*funcionarios, oficina, empleo*}	1
	2	{*empleo, funcionarios*}	2
	3	{*funcionarios*}	5
{*oficina*}	4	{*funcionarios, oficina, empleo*}	1
	5	{*empleo, oficina*}	2
	6	{*oficina, empleo*}	3
{*empleo*}	7	{*funcionarios, oficina, empleo*}	1
	8	{*empleo, funcionarios*}	2
	9	{*empleo, oficina*}	2
	10	{*oficina, empleo*}	3
	11	{*disminución, empleo*}	2
	12	{*empleo*}	10
{*disminución*}	13	{*disminución, empleo*}	2
	14	{*disminución*}	3

Tabla 4.14: Conjunto de reglas ordenadas (CRO) de las *item-seqs* frecuentes de nivel uno

k	ro_k	$freq(ro_k)$
1	{*funcionarios, oficina, empleo*}	1
2	{*empleo, funcionarios*}	2
3	{*funcionarios*}	5
4	{*empleo, oficina*}	2
5	{*oficina, empleo*}	3
6	{*disminución, empleo*}	2
7	{*empleo*}	10
8	{*disminución*}	3

Tabla 4.15: Conjunto de reglas ordenadas (COR) tras la eliminación de redundancias

i	ro_{1i}	$freq(ro_{1i})$
1	{*funcionarios, oficina*}	1
2	{*oficina, empleo*}	3

Tabla 4.16: Sub-reglas de ro_1

Después del paso 9, volvemos al 5 (según lo indicado en el 10) para almacenar las nuevas reglas ordenadas obtenidas de la descomposición de r_1. Ahora volverían

a eliminarse las redundancias en el caso de haberse producido y si alguna de las sub-reglas es frecuente, se almacenaría en $CROF$. Las sub-reglas no frecuentes se descomponen de nuevo. Es el caso de la regla r_{11} (Tabla 4.17).

j	ro_{11j}	$freq(ro_{11j})$
1	{*funcionarios*}	5
2	{*oficina*}	5

Tabla 4.17: Sub-reglas de ro_{11}

El proceso comprendido entre los pasos 5 y 10 se repite hasta que todas las reglas sean frecuentes y ya no quede ninguna por descomponer.

El paso 11 consiste en eliminar las redundancias que se hubieran producido en $CROF$, que quedaría como se muestra en la Tabla 4.18.

k	ro_k^f	$freq(ro_k^f)$
1	{*oficina*}	5
2	{*empleo, funcionarios*}	2
3	{*funcionarios*}	5
4	{*empleo, oficina*}	2
5	{*oficina, empleo*}	3
6	{*disminución, empleo*}	2
7	{*empleo*}	10
8	{*disminución*}	3

Tabla 4.18: Conjunto de reglas ordenadas frecuentes $(CROF)$ tras la eliminación de redundancias

Partiendo de las reglas ordenadas frecuentes obtenidas, las siguientes tareas (12 y 13) consisten en establecer cuáles son primarias y crear el conjunto de las reglas ordenadas frecuentes primarias o conjunto de implicaciones ordenadas frecuentes $(Po^f(S))$.

De todas las reglas en la Tabla 4.18, podemos ver las primarias en la Tabla 4.19.

Ya podemos identificar las implicaciones de las *item-seqs* frecuentes de nivel uno:

- $imp_S(x_1) = imp(\{funcionarios\}) = \{empleo, funcionarios\}$

k	rop_k^f	$freq(rop_k^f)$	imp_k
1	{*empleo, funcionarios*}	2	imp_1
2	{*empleo, oficina*}	2	imp_2
3	{*oficina, empleo*}	3	imp_3
4	{*disminución, empleo*}	2	imp_4

Tabla 4.19: Conjunto de reglas ordenadas frecuentes primarias o implicaciones ordenadas frecuentes $Po^f(S)$

- $imp_S(x_2) = imp(\{oficina\}) = \{empleo, oficina\}$ y $\{oficina, empleo\}$
- $imp_S(x_3) = imp(\{empleo\}) = \{empleo, funcionarios\}, \{empleo, oficina\}$, $\{oficina, empleo\}$ y $\{disminución, empleo\}$
- $imp_S(x_4) = imp(\{disminución\}) = \{disminución, empleo\}$

Las implicaciones ordenadas frecuentes serán las secuencias generadoras de la estructura WAPO. Podemos buscar la correspondencia directamente en el conjunto de reglas ordenadas frecuentes primarias del Tabla 4.19. Como vemos, la estructura WAPO tendrá cuatro secuencias generadoras correspondientes a cada una de las reglas frecuentes primarias (paso 14).

Con lo que $\widetilde{E} = \widetilde{g}(A, B, C, D)$

Dichas secuencias generadoras se corresponden con las implicaciones ordenadas encontradas en la Tabla 4.19:

$A = imp_1$, $B = imp_2$, $C = imp_3$ y $D = imp_4 \quad \Rightarrow$

$E = g(\{empleo, funcionarios\}, \{empleo, oficina\}, \{oficina, empleo\}$ y $\{disminución, empleo\})$

La estructura APO resultante tendrá cardinal 8:

$E = (\{empleo, funcionarios\}, \{empleo, oficina\}, \{oficina, empleo\}, \{disminución, empleo\}, \{funcionarios\}, \{oficina\}, \{empleo\}$ y $\{disminu - ción\}$

A partir de E se obtiene la estructura WAPO:

$\widetilde{E} = g([\{empleo, funcionarios\}, (2)], [\{empleo, oficina\}, (2)], [\{oficina, empleo\}, (2)]y[\{disminución, empleo\}, (3)])$

En el último paso, para hallar el peso del resto de *item-seqs* de la estructura WAPO, haríamos uso de las funciones loc(θ_j) y freq(θ_j), con $\theta_j = I_j \in E$.

Comprobamos, que tanto la estructura WAP como la WAPO, halladas a través de este método, coinciden con las obtenidas en la Sección 3.5 (ver Tabla 3.16).

4.6 Resumen y Conclusiones

En las Secciones 4.1 y 4.2 se recoge una revisión sobre los algoritmos más usados para la generación de *itemsets* o secuencias frecuentes, algunos de los cuales utilizan índices invertidos o listas invertidas con la intención de recortar el tiempo de procesamiento, como son AprioriTid, AprioriHybrid [Agr94] o SPADE [Zak01] y la mayoría de ellos emplean la propiedad Apriori.

En Hipp et al. [Hip00] se realiza una comparación de varios de los algoritmos en cuanto a rendimiento para la obtención de *itemsets* frecuentes, comprobándose que todos tienen un comportamiento similar con respecto al tiempo de ejecución, ya que unos emplean más tiempo en unas tareas y otros en otras. Por ejemplo, Eclat [Zak97] y el algoritmo de Partición [Sav95], emplean la mayor parte del tiempo en determinar el soporte de los *itemsets* candidatos de nivel inferior a cuatro, mientras que el algoritmo Apriori encuentra la mayor dificultad en el cálculo del soporte de los *itemsets* de nivel cuatro o superior.

Como nuestro objetivo es obtener la estructura de dominio de los atributos textuales de las bases de datos y éstos están compuestos por entradas de texto corto, muy raramente vamos a encontrar secuencias de nivel cuatro o superior, por lo que

nos parece la mejor opción aplicar el algoritmo Apriori, que es el más conocido por su simpleza y facilidad de implementación.

Aparte del Apriori, en este capítulo se ha propuesto un método alternativo para la generación de las estructuras, que consiste en identificar los conjuntos o secuencias generadoras con las implicaciones de los *itemsets* frecuentes de nivel uno, pero deberá ocurrir que estas implicaciones también sean frecuentes, si no habrá que descomponerlas hasta encontrar el nivel en que lo sean.

El mayor inconveniente de este método es que cuando se encuentran muchos *itemsets* o *item-seqs* maximales no frecuentes, se pierde mucho tiempo en la descomposición de éstos hasta encontrar el nivel en que son frecuentes, pudiendo hallar así las implicaciones frecuentes, lo que lo convierte en un método más apropiado en un texto en el que se encuentran muchas secuencias de gran tamaño repetidas, garantizando en cierto modo que la mayor parte de los conjuntos de términos maximales encontrados van a ser frecuentes.

Por esta razón, este método funcionará mejor que el Apriori sólo en algunos casos, dependiendo de las características del texto en que se aplique.

CAPÍTULO

5

Del Atributo Textual a la Tag Cloud

En este capítulo presentamos una metodología para el procesamiento de los textos cortos no estructurados que aparecen en los campos textuales de las bases de datos, con el fin de obtener cierta semántica de éstos y ser capaces de representar su contenido.

Principalmente, se pretende obtener el dominio de un atributo cualquiera de la base de datos, extrayendo la información relevante del texto con la ayuda de técnicas de minería de datos y texto y representarla a través de alguna de las formas intermedias, provistas de estructura, vistas en el Capítulo 3.

Posteriormente, visualizaremos la forma intermedia generada, para permitir que se identifique el contenido del atributo de forma visual y posibilitar que ésta pueda usarse como asistente que ayude en la consulta y la exploración.

En nuestro caso particular, emplearemos una *tag cloud* multitérmino como forma de visualización, por todas las ventajas que se expusieron sobre ésta en el Capítulo 2.

En la Sección 5.1 se plantea una metodología general, estructurada en etapas secuenciales, para la representación semántica de textos no estructurados. En la

Sección 5.2 se desarrollan las etapas de preprocesamiento, indicando las herramientas que pueden usarse en cada una de ellas. En la Sección 5.3 se detalla la de generación de formas intermedias y la de postprocesamiento. En la Sección 5.4 se expone la de visualización. En la Sección 5.5 se concretan las herramientas utilizadas por nosotros en la aplicación de la metodología.

Se termina con un resumen y la exposición de algunas conclusiones en la Sección 5.6.

5.1 Metodología para la Representación Semántica de Textos no Estructurados

Vamos a proponer una metodología de carácter general que requiere la aplicación de una amplia variedad de herramientas, las cuales son bien conocidas y nos permitirán obtener información relevante sobre los campos textuales.

Esta metodología consta de las siguientes etapas:

- **Preprocesamiento Sintáctico**: Lo primero que hacemos es una limpieza basada en la **tokenización** [1] de los términos textuales, con la consiguiente eliminación de signos gramaticales y de puntuación y, en caso de considerarse necesario, la **lematización** [2]. Es en esta etapa donde se realiza la eliminación de palabras vacías.

- **Preprocesamiento Semántico**: Después de realizar el preprocesamiento sintáctico, se lleva a cabo la detección de términos sinónimos y se sustituyen todos ellos por un único término. El proceso consiste en agruparlos en conjuntos y seleccionar un término de cada uno de estos conjuntos que será considerado el representante canónico de todos los demás, sustituyéndolos en el texto a procesar.

  La detección de sinónimos se lleva a cabo en los siguientes cuatro pasos:

[1] La tokenización es el proceso de descomposición de una cadena de texto en palabras, frases, símbolos u otros elementos significativos llamados "tokens"

[2] La lematización es el proceso lingüístico que consiste en, dada una forma flexionada (plural, femenino, conjugación, etc.), hallar el lema correspondiente. El lema es la forma que por convenio se acepta como representante de todas las formas flexionadas de una misma palabra

1. **Etiquetado de categoría gramatical (*Part Of Speech Tagging*)**: Primero se etiqueta el término con la categoría gramatical que desempeña en la sentencia de la que se ha extraído.
2. **Desambiguación (*Word Sense Disambiguation*)**: Una vez que se conoce la categoría gramatical, se determina el significado del término en cuestión, lo que es imprescindible para poder determinar sus sinónimos.
3. **Generación de conjuntos de sinónimos**: Posteriormente se agrupan los términos en base a su significado en los denominados conjuntos de sinónimos. Para esta tarea se puede hacer uso de diccionarios electrónicos.
4. **Selección del representante canónico de los conjuntos de sinónimos**: Por último, se determina un término del conjunto de sinónimos que ejerza como representante canónico de éste y sustituya a todos los demás en el texto y que, por tanto, los represente.

- **Generación de la Forma de Representación Intermedia**: Una vez concluido el preprocesamiento del texto, los datos ya están limpios para generar, a partir de ellos, alguna de las formas intermedias de representación vistas en el Capítulo 3, las cuales continúan aportando semántica al permitir que los términos relacionados puedan permanecer unidos. Al finalizar esta etapa, se habrá conseguido dotar de estructura al texto no estructurado y será mucho más fácil trabajar con los atributos textuales, recuperar la información que contienen y realizar búsquedas semánticas satisfactorias.

- **Visualización**: Finalmente, se visualiza la forma o formas intermedias obtenidas, que actúan como dominio del texto procesado, lo que permite identificar el contenido de la información. Además de esto, la visualización funciona como una interfaz gráfica que facilita la consulta a través de cada uno de los términos representados.

En la Figura 5.1 se resume este proceso de forma gráfica. Podemos ver las distintas etapas de la metodología desde que se tiene el texto no estructurado hasta que se genera la *tag cloud* de salida. Las herramientas externas utilizadas son diversas y se comentan para cada caso particular.

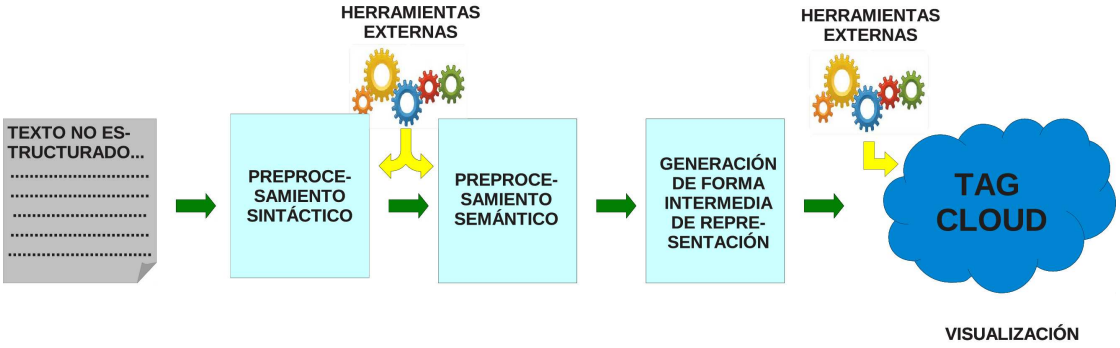


Figura 5.1: Proceso general para la representación semántica de textos no estructurados

5.2 Preprocesamiento de los Datos

En este apartado se describen con mayor detalle las etapas de preprocesamiento sintáctico y semántico. También se referencian posibles herramientas externas a utilizar. En la Sección 5.5 se comentan las usadas por nosotros, que han sido desarrolladas de forma colaborativa en nuestro grupo de investigación.

5.2.1 Preprocesamiento Sintáctico

Mediante este proceso, se eliminan elementos sintácticos que no aportan información, así como signos de puntuación que complican el procesamiento automático del texto. La herramienta que se utiliza es una que aplica filtros para la tokenización y la eliminación de palabras vacías (*stop-words*).

En nuestro caso, es mejor no aplicar ningún filtro de lematización o emplear solamente "*S-stemmer*", que lo único que hace es convertir las formas plurales a singulares , ya que el diccionario externo empleado, podría no reconocer las formas lematizadas, con lo que se complicaría enormemente la etapa siguiente de preprocesamiento semántico. Además, el empleo de estas formas dificultaría la identificación del contenido cuando se representen en la *tag cloud*. Para ver algunos filtros de lematización, consultar [Por80].

.

Son muchas las herramientas existentes para realizar preprocesamiento sintáctico de datos textuales. La usada por nosotros se describe en la Sección 5.5.

Tras realizar el preprocesamiento sintáctico, los datos ya estarían preparados para ser procesados. Sin embargo, en nuestro caso, el siguiente paso consistiría en la generación de las formas intermedias de representación y para su generación las técnicas utilizadas se basan en la detección de *itemsets* o *item-seqs* frecuentes, las cuales se componen de conjuntos de términos. Como sabemos, un mismo concepto puede estar expresado por varios términos distintos, lo que podría dar lugar a que, un término que es en realidad relevante, no fuera frecuente debido a la amplia gama de expresiones sintácticas en las que viene dado, por lo que se quedaría fuera de la estructura de representación. También podría darse el caso contrario, en el varios significados vienen representados por un término genérico, que sería frecuente en el texto y por lo tanto aparecería en la forma intermedia de representación, pero que en realidad no sería relevante.

Para evitar que se produzcan estas situaciones, se realiza un preprocesamiento semántico a continuación del sintáctico.

5.2.2 Preprocesamiento Semántico

Lo que se pretende en esta fase es homogeneizar la representación sintáctica de los conceptos presentes en el texto, para ello se sustituyen todas las palabras sinónimas por una única forma que se conoce como "representante canónico del conjunto de sinónimos".

Para esta tarea, nos servimos de diccionarios externos y bases de conocimiento existentes. Podría pensarse en crear un diccionario propio de términos sinónimos, pero debido a la extensión de los dominios tratados y a la generalidad de los mismos, esta opción no es abordable.

A continuación se explica, de forma general, el proceso mediante el cual se preparan los datos de forma semántica. Empezaremos definiendo los conjuntos de sinónimos y el concepto de representante canónico de dichos conjuntos.

Definición 5.2.1. Conjunto de Sinónimos
Sea $\mathcal{T} = \{T_1, \ldots, T_n\}$ *un conjunto de textos. Un texto* $T_i \in \mathcal{T}$ *estará compuesto*

por uno o más términos $t_j, j = 1, \ldots, m$. Cada término t_j tiene asociado uno o más significados s_k, $k = 1, \ldots, p$, dependiendo de los contextos en que se pueda emplear el término, entonces, un término t_j cuyo contexto está determinado, sólo podrá tener un único significado desambiguado m_k. A su vez, un significado desambiguado m_k puede ser el significado de uno o más términos t_j.

$\Rightarrow$ Llamamos conjunto de sinónimos S_k a todos aquellos términos t_j cuyo significado desambiguado sea m_k.

Definición 5.2.2. Representante Canónico de un Conjunto de Sinónimos
Para todo conjunto de sinónimos S_k existe un término $t_j \in S_k$ que recibe el nombre de representante canónico de S_k, que es único y se denota como r_k.

Todo término t_j perteneciente al conjunto de sinónimos S_k, puede ser sustituido por el representante canónico del conjunto r_k, sin que cambie la semántica del texto original.

Dicho de otra forma, r_k será el representante canónico de un conjunto de sinónimos S_k si:

$$\forall t_j \in S_k, \exists r_k | t_j = r_k \tag{5.1}$$

Para obtener los términos sinónimos de uno dado, debe conocerse el significado de todos los implicados, lo que es una tarea compleja que, a su vez, conlleva la realización de otras tareas también complejas. Para determinar este significado, primero es necesario conocer la categoría gramatical de la palabra y el contexto en el que se usa, ya que términos con la misma forma sintáctica tienen significados diferentes dependiendo de esto.

Sólo una vez conocido el significado concreto de una palabra, podrán determinarse sus sinónimos.

A continuación vemos en qué consiste el proceso de determinar la categoría gramatical.

Etiquetado de Categoría Gramatical

Lo que se hace mediante este proceso es añadir a cada palabra del texto una etiqueta, la cual expresa su categoría gramatical dentro de la frase en la que está incluida. Ejemplos: sustantivo, adverbio, adjetivo, verbo, pronombre, etc.

La categoría gramatical de un término puede determinarse en función de su definición (ya que hay palabras que sólo pueden funcionar en una categoría) o en función del contexto en el que se encuentra. En esta última situación, el proceso de etiquetado es bastante complejo.

Tomemos por ejemplo la palabra "pienso" que, dependiendo del contexto actuará como nombre o como verbo: "los animales comen pienso (nombre)" o "pienso (verbo) en los animales".

Encontramos numerosas aproximaciones en la literatura para tratar con este problema, las cuales pueden dividirse en lingüísticas y basadas en corpus, aunque también existen híbridas que optimizan los resultados.

- **Aproximaciones Lingüísticas**: Los expertos lingüistas intervienen en éstas, creando conjuntos de reglas que ayudan a determinar la categoría gramatical de cada término según unos modelos establecidos.

- **Aproximaciones Basadas en Corpus**: Éstas pueden realizarse de forma automática, con supervisión humana o sin ella. Están basadas en técnicas de aprendizaje sobre los datos. Cuando existe supervisión, a partir de un corpus ya etiquetado, se construye un clasificador que asigna las etiquetas a las nuevas palabras que se introducen. Cuando no existe supervisión, se etiqueta el texto en base a estadísticas y se aprende a partir de un corpus no etiquetado previamente.

Los etiquetadores más utilizados hasta el momento son los de tipo estadístico, que pertenecen al grupo de aproximaciones basadas en corpus sin supervisión. Algunos de los que ofrecen mejores resultados son:

- **TnT** [Bra00]: Su uso está muy extendido. El nombre procede de la forma breve de "*Trigrams "n" Tags*". Fue desarrollado en la Universidad de Saarlandes, Alemania. Se basa en modelos ocultos de Markov y es muy eficiente, es capaz de trabajar con palabras desconocidas y ofrece la posibilidad de entrenarlo para diferentes lenguajes y para cualquier conjunto de etiquetas mediante un corpus anotado.

- **TreeTagger** [Sch99] : Se desarrolló en el Instituto de Lingüística Computacional de la Universidad de Stuttgart, Alemania. Al igual que TnT, se basa se basa en modelos ocultos de Markov. TreeTagger es capaz de etiquetar textos en seis idiomas (español, inglés, francés, alemán, italiano y griego), aunque puede adaptarse a otros lenguajes si se dispone de un *lexicon* y un corpus anotado. Etiqueta los textos con información de la categoría gramatical y del lema del término correspondiente.

- **Stanford Tagger** [Tou03]: Actualmente, es el que mayor precisión obtiene para el idioma inglés. Se desarrolló en la Universidad de Stanford, California.

- **SVMTool** [Gim04]: Es de código abierto que emplea *"Support Vector Machines (SVM)"* para clasificación. Se desarrolló en la Universidad Politécnica de Cataluña, España.

Una vez determinada la categoría gramatical de un término se procede a su desambiguación, lo que permite encontrar el sentido concreto de éste tras haber eliminado todos los pertenecientes a otras categorías gramaticales.

Desambiguación

Este proceso es fundamental dentro del preprocesamiento semántico. De él depende que se haga una buena identificación del sentido de las palabras y que se elija el conjunto de sinónimos adecuado para cada una de ellas.

El algoritmo más simple de desambiguación es el del sentido más frecuente. Consiste en tomar como sentido de una palabra el que posee con mayor frecuencia una vez conocida su categoría gramatical. Para hacer esto existen diccionarios electrónicos sobre el uso de las palabras según su sentido, los cuales normalmente muestran los sentidos ordenados según la frecuencia con que se usan, con lo que se tomaría el primero presentado.

Este algoritmo es simple y no es lo bastante bueno para alcanzar resultados ambiciosos, por lo que suele usarse en evaluaciones de algoritmos de desambiguación, para compararlo con otros.

A continuación, se describen algunos algoritmos de desambiguación más acordes a nuestras necesidades:

- **Algoritmo de Lesk [Les86]**. Es un algoritmo básico en desambiguación basada en conocimiento y uno de los primeros desarrollados para desambiguación semántica de todas las palabras en un texto sin restricciones. Se sirve de un conjunto de entradas de diccionario (una por cada posible sentido de la palabra) y del conocimiento sobre el contexto inmediato en el que se realiza la desambiguación. La idea principal es desambiguar las palabras encontrando el solapamiento entre las definiciones de sus distintos sentidos.

  Los algoritmos basados en corpus actuales nacen a partir de éste, ya que todos trabajan, de una u otra forma, con el solapamiento medido entre el contexto de un término ambiguo y los contextos específicos correspondientes a varios significados de esa palabra, aprendidos de anotaciones anteriores.

  Este algoritmo no es apto para desambiguar más de dos palabras, ya que esto ocasionaría múltiples combinaciones.

  Muchas variantes han buscado soluciones a este problema, como la de "enfriamiento simulado", que intenta reducir el número de combinaciones con métodos de optimización, o el "algoritmo simplificado de Lesk".

- **Algoritmo Simplificado de Lesk [Vas04]**. Es una variación simplificada del algoritmo de Lesk. En esta versión se realiza un proceso de desambiguación distinto por cada palabra del texto de entrada. El significado correcto se determina de forma individual, encontrando el sentido que produce un mayor solapamiento entre su definición en el diccionario y el contexto actual. Es decir, la idea de este algoritmo es buscar el sentido que contenga más palabras del contexto en el que se encuentra aquella a desambiguar, en lugar de intentar encontrar el sentido a todas las palabras del texto de forma simultánea.

- **Algoritmo Adaptado de Lesk [Ban02]**. Es otra variante del algoritmo de Lesk. En ésta se emplean, además de las definiciones de las propias palabras, otras relacionadas para determinar el contexto de éstas y en función del mismo, su sentido más probable. Esto es lo que le diferencia del algoritmo de Lesk original, que únicamente tiene en cuenta la definición de las palabras, pero no la de palabras relacionadas.

Por ejemplo, para el caso concreto de WordNet, el algoritmo tiene en cuenta su jerarquía para construir un contexto más amplio para un significado de una palabra. Esta jerarquía define hiperónimos, hipónimos, holonimos, merónimos, tropónimos, relaciones de atributos y sus definiciones asociadas. Así, se extiende el contexto del diccionario con las definiciones de conceptos semánticamente relacionados.

Tras haber determinado la categoría gramatical de los términos del texto y realizado su desambiguación, se asocian las palabras a los conjuntos de sinónimos S_k que correspondan, según el sentido que se les ha asignado. Y una vez que se tienen todas las palabras asociadas a un conjunto de sinónimos, se elije el representante canónico de este conjunto r_k, el cual sustituirá en el texto a todas las palabras contenidas en S_k

Selección del Representante Canónico

Tras obtener la serie de conjuntos de sinónimos $S_k,\ k = 1, \ldots, n$, cada palabra o término t_j estará asociado a un único S_k. Semánticamente, todos los términos asociados a un mismo conjunto de sinónimos son equivalentes, por lo que se podía tomar cualquiera de ellos aleatoriamente como representante canónico. Sin embargo, nos parece más acertado realizar la sustitución por el que aparece con mayor frecuencia dentro del texto original, ya que de esta forma se conserva la legibilidad y cohesión contextual.

El procedimiento es el siguiente:

1. Se calcula el número de veces que los términos asociados a cada conjunto S_k aparecen en el texto de origen, considerando que una misma forma sintáctica con diferente categoría gramatical pertenecerá a otro conjunto S_k.

2. Una vez conocida la frecuencia de cada término t_j, se selecciona aquel que aparece en el texto mayor número de veces y se designa como representante canónico r_k.

3. Por último, se sustituyen todos los términos sinónimos ($t_j \in S_k$) por el representante canónico r_k del conjunto S_k.

5.3 Generación de las Formas Intermedias

Una vez finalizado todo el preprocesamiento de los textos, la siguiente etapa consiste en generar la forma intermedia de representación. Ésta debe contener información acerca de los términos más relevantes del texto y de sus relaciones. Nosotros utilizamos como formas intermedias las estructuras WAP y WAPO planteadas en el Capítulo 3.

Los principales pasos realizados en esta tarea son:

1. Obtener un diccionario de datos consistente en una lista de términos de la columna textual a representar.

2. Transformar los datos en una base de datos transaccional, donde los atributos se componen por los diferentes términos del diccionario de datos. Cada tupla se corresponde con una entrada en la tabla original.

3. Obtener los *itemsets* o *item-seqs* frecuentes a partir el algoritmo Apriori [Agr94] y su modificación (Sección 4.3), aunque puede hacerse a partir de otros métodos como vimos en el Capítulo 4. Para esta tarea será necesario establecer un soporte mínimo, cuyas variaciones afectarán al número de *itemsets* o *item-seqs* en la estructura final. La única forma de determinar este soporte es empíricamente, según el número de términos frecuentes que se quiera tener representados en la *tag cloud*.

4. Construir dos estructuras bien diferenciadas:

   - Una de dominio que describe el atributo almacenado en la columna de la base de datos que contiene los textos procesados. (En la Sección 3.5 se vio un ejemplo de cómo se obtenía esta estructura, la WAP y la WAPO, concretamente pueden verse las estructuras que se hallaron en la Tabla 3.16)
   - Otra por cada tupla de la relación original. Ésta es la subestructura inducida por los términos textuales en la estructura de dominio y representa al texto corto de la tupla en cuestión. (Como ejemplo puede verse la

Tabla 3.17 que presenta la estructura de cada tupla del ejemplo de la Sección 3.5).

El principal inconveniente en la generación de las formas intermedias es que hay palabras vacías de contenido que se usan frecuentemente como nexos en las frases y se identifican como términos relevantes cuando en realidad son todo lo contrario. Este inconveniente es menor en la estructura WAPO, donde todos los términos presentes en las *item-seqs* son estrictamente adyacentes en el texto, lo que evita en parte la aparición en la representación de nexos indeseados.

5.4 Postprocesamiento y Visualización

Determinada la forma de representación intermedia a utilizar y expresado el conjunto de textos en dicha forma, se procede a la visualización de la estructura. Nosotros hemos elegido para ésta una *tag cloud* multitérmino, ya que presenta numerosas ventajas como se vio en la Sección 2.1.

Los términos más apropiados para representar en la *tag cloud* son los nombres. Los verbos y adverbios no son útiles en la visualización, ya que aportan poca descripción del contenido. Los adjetivos aislados también se consideran palabras superfluas, pero aportan mucha información cuando van acompañando al nombre. Teniendo esto en cuenta y conociendo la categoría gramatical de cada término gracias al etiquetado, optamos por eliminar los verbos y adverbios y establecer unas reglas para determinar cuándo un conjunto de términos es apto para aparecer en la *tag cloud* atendiendo a su categoría gramatical.

Un *itemset* o *item-seq* será aceptado para su visualización dentro de la *tag cloud* si cumple con alguna de las siguientes condiciones (para el idioma inglés):

- Los términos individuales se consideran candidatos a ser una etiqueta en la *tag cloud* si son nombres [N].
- Los conjuntos de dos términos se consideran candidatos a etiqueta en la *tag cloud* si son nombres precedidos de un adjetivo [AN] o son nombres compuestos [NN]

- Serán conjuntos de términos candidatos a etiquetas en la *tag cloud* aquellos compuestos por una combinación válida en los niveles previos precedida por un adjetivo o seguida por un nombre. Como ejemplo, combinaciones válidas de nivel tres serán [AAN], [ANN] y [NNN].

Todos los conjuntos de términos frecuentes que se obtengan que no cumplan ninguna de estas reglas, serán desechados. El resto se usará para generar la *tag cloud*.

De esta forma se evita, entre otras cosas, que en la visualización aparezcan palabras frecuentes poco relevantes (como comentábamos en la Sección 5.3) que normalmente hacen de conexión entre frases, palabras tales como verbos conjugados o adverbios.

Para el español las condiciones serán las mismas, pero considerando que el adjetivo puede ir tanto delante del nombre, como detrás. Éste y otros aspectos del postprocesamiento se ven mejor en los experimentos del Capítulo 6.

En el caso de que en la *tag cloud* aparezcan demasiados términos, lo que afectaría a su identificación, se puede optar por representar únicamente los conjuntos maximales (como hacían Kaptein y Marx [Kap10]), que serán en definitiva los que resumen el contenido de la estructura, evitando también así algunas redundancias y solapamiento semántico. De esta forma, es posible disminuir el soporte para que aparezcan nuevos términos relevantes, sin que el diseño aparezca demasiado aglomerado.

También los aspectos concernientes a la visualización se ven mejor en los experimentos del Capítulo 6.

La *tag cloud* construida se usa, además de para representar el contenido de un atributo textual almacenado en una columna de una tabla de la base de datos, para generar consultas sobre él.

En la Figura 5.2 podemos ver la arquitectura general del sistema y los distintos módulos que la componen y que hemos ido explicando en este capítulo.

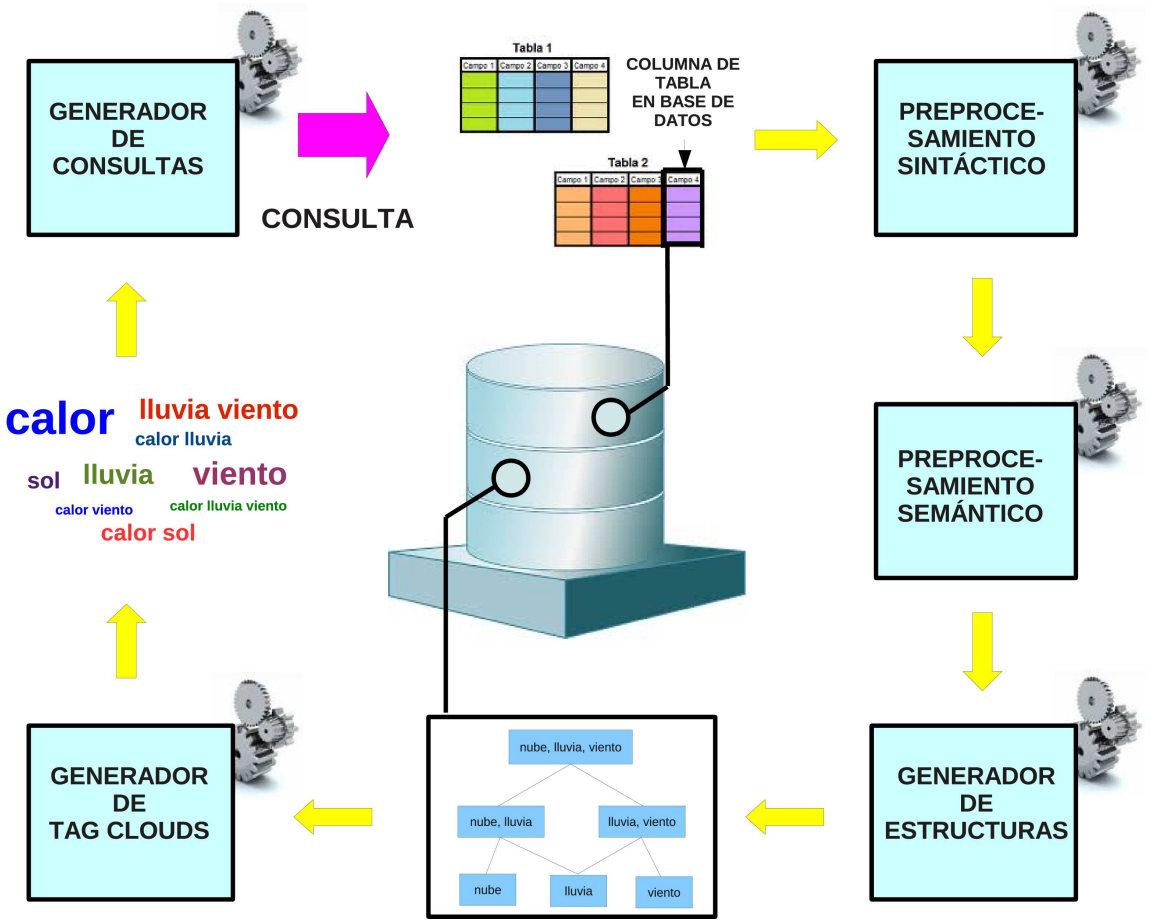


Figura 5.2: Arquitectura del sistema

5.5 Herramientas Utilizadas

En esta sección se detallan las herramientas utilizadas por nosotros, las cuales se han desarrollado dentro de los proyectos P07-TIC-02786 [1] y P10-TIC-6109 [2].

5.5.1 Herramientas para el Preprocesamiento Sintáctico

Para realizar un proceso de limpieza selectiva sobre un conjunto de datos utilizaremos la siguiente herramienta genérica de filtrado.

Herramienta de Filtrado de Datos

"*DB2DS*" es el nombre que se le dio a esta herramienta, la cual está realizada en Java usando JDBC (*Java DataBase Conectivity*), para poder adaptarla a la mayor cantidad posible de SGBDs (Sistemas Gestores de Bases de Datos). Además combina rutinas implementadas en la base de datos en lenguaje PL/SQL (Procedural Language/Structured Query Language).

Un usuario cualquiera, sin conocimiento previo sobre el funcionamiento de esta herramienta, podrá construir conjuntos de datos a partir de un esquema de base de datos con la ayuda de esta herramienta, que cuenta para ello con una interfaz sencilla e intuitiva.

Además, ofrece la posibilidad de realizar un procesamiento completo o de llamar a otros métodos nativos que ejecuten el procesamiento de forma interna en la base de datos.

Se comienza estableciendo una conexión con la base de datos, para ello se introducen los datos de conexión, que pueden memorizarse de tal forma que la conexión sea persistente y nos ahorremos tener que volver a introducirlos en ocasiones sucesivas. Así, la conexión a través de unos datos ya establecidos con anterioridad, se realizará de forma automática y sólo nos preguntará la contraseña si no la hemos guardado previamente.

[1] Un sistema para la movilización del conocimiento contenido en una base de datos poco estructurada. Aplicación a los textos de una base de historias clínicas.

[2] Nuevas representaciones estructuradas de textos para data warehousing y minería de datos: representaciones visuales de tipo tag clouds usando folksonomías. Aplicación a la información médica.

Podrán guardarse tantas configuraciones de conexiones distintas como se desee, pero solamente una de ellas podrá mantenerse activa. Las conexiones persistentes podrán modificarse o borrarse.

La datos de configuración de las estas conexiones se almacenan en XML (*Extensible Markup Language*), para poder importarlos y exportarlos con facilidad.

Cuando la conexión con la base de datos se ha establecido, ya se puede comenzar a seleccionar datos. Existen dos tipos de mecanismos de selección:

- **Horizontales.** Permiten seleccionar tuplas a lo largo de todo el esquema bajo una condición impuesta por el usuario sobre uno o más campos.
- **Verticales.** Permiten realizar proyecciones sobre el conjunto de datos y seleccionar aquellas tablas y campos que contengan los datos buscados. Pueden introducirse textualmente condiciones complejas.

Seleccionados los datos, se identifican las operaciones de filtrado a realizar. Para esta tarea es necesario indicar los procesos de filtrado que desean efectuarse sobre los campos que se vayan a procesar y el orden en que deben ejecutarse. Podemos indicar estos procesos de forma independiente para cada uno de los campos o podemos definir un mismo tipo de filtrado sobre un grupo de campos.

Finalmente, una vez realizado el filtrado del texto, se obtiene un conjunto de Metadatos, los que permitirán la generación de conjuntos de datos, relacionales o transaccionales, conocidos como "*datasets*".

Para poder conocer el procesamiento que ha dado lugar a los datos actuales, todas las operaciones realizadas sobre éstos en cada etapa, se recogen en un fichero de configuración XML, que a su vez puede ser utilizado para repetir el proceso sobre otro conjunto de datos distinto que comparta el mismo esquema de base de datos.

Para realizar todo el proceso de forma guiada, la herramienta dispone de un asistente de generación de conjuntos de datos que ejecuta cada uno de los pasos en el siguiente orden:

1. Conexión con la base de datos
2. Selección de tablas

3. Aplicación de condiciones en selección horizontal de atributos
4. Aplicación de condiciones en selección vertical de atributos
5. Selección del tipo de filtrado a realizar sobre cada uno de los campos. En caso de seleccionar más de un filtrado, se indicará el orden en que deben realizarse. Tipos de filtrado:
   - Tokenización.
   - Aplicación de Diccionarios.
   - Eliminación de *Stop Words*.
   - Lematización (*Stemming*).
   - Ordenación alfabética.
   - Otros.
6. Inclusión de filtrado a nivel de registros para eliminar tuplas repetidas
7. Discretización de campos numéricos y selección de las representaciones intermedias a generar
8. Generación de Metadatos
9. Creación de tablas relacionales
10. Creación de tablas transaccionales

    El paso 3 se realiza antes que el paso 4 por si se quieren establecer condiciones sobre atributos que no se van a incluir en el conjunto de datos resultante.

Veamos cada una de las etapas de forma detallada:

Gráficamente, partiendo de un esquema de bases de datos, se define una consulta que realiza una selección horizontal y vertical, mediante la secuencia de menús "*Tablas* $\rightarrow$ *Vistas* $\rightarrow$ *Creación de una consulta*".

Para la selección horizontal, existen mecanismos que realizan reunión entre tablas y aplican condiciones de selección según un atributo.

Para la selección vertical, se cuenta con una serie de *checkboxes* que nos ayudan a seleccionar la columna que queremos incluir en la base de datos.

Cuando la selección se ha completado, se define una nueva vista por la generación de la consulta SQL que se ha llevado a cabo.

A continuación se aplican las técnicas de filtrado, para ello distinguiremos entre campos de texto y campos numéricos.

- **Campos de Texto**

  Tras haber obtenido la vista, se puede aplicar un filtrado para uno o más campos textuales. Este filtrado se aplica mediante la secuencia de menús "*Vistas* $\rightarrow$ *Metadatos* $\rightarrow$ *Aplicación de filtros y discretización"*.

  El orden de aplicación de filtros afecta a los resultados, por lo que éstos serán los siguientes y se aplicarán en el siguiente orden:

  1. Tokenización
  2. *S-Stemmer* en inglés
  3. *Porter Stemming* en inglés
  4. *S-Stemmer* en español
  5. *Porter Stemming* en español
  6. Eliminación de *Stop Words* en inglés
  7. Eliminación de *Stop Words* en español

  Como vemos, estos filtros combinan los idiomas inglés y español.

  La lematización empleada (*S-Stemmer*) es muy sencilla, consiste únicamente en convertir las formas plurales en singulares.

  Los Metadatos son los textos filtrados que se almacenan junto al resto de datos en la estructura del catálogo.

  Como sabemos, un término puede componerse por una o más palabras (por ejemplo "*Inteligencia Artificial*" o "*Red Social*"), sin embargo a la hora de procesar los campos textuales, por defecto se considera que cada palabra

representa un término. Para procesar el conjunto de términos de más de una palabra, se usa un diccionario.

Los diccionarios recogen los distintos términos complejos de más de una palabra y permiten su preservación en el procesamiento de datos. Los usuarios pueden añadir, modificar o borrar términos del diccionario. Para facilitar el proceso de generación de éstos, es posible crearlos a partir de otros ya definidos o incluir nuevos términos provenientes de diferentes diccionarios.

El diccionario utilizado puede ser general para toda la base de datos, específico para cada usuario o incluso un diccionario concreto por columna.

- **Campos numéricos**

  Existen dos formas principales de discretizar los campos numéricos:

  - Semiautomáticamente: Se crean intervalos semiautomáticos, equidistantes o equiprobables. El parámetro que se pide es el número de intervalos a crear
  - Manualmente: Puede definirse la discretización a aplicar mediante la interfaz gráfica

  Los valores numéricos discretizados también se almacenan en forma de Metadatos, de modo que a partir de ellos pueden generarse los conjuntos de datos necesarios.

Una vez generados los metadatos en los procesos de filtrado de texto y discretización de datos numéricos, éstos se almacenan en unas tablas del sistema, con el objetivo de construir los *datasets*, que podrán ser tanto relacionales como transaccionales ("*Metadatos* $\rightarrow$ *Dataset Relacional/Transaccional*"). Éstos podrán exportarse a un fichero de texto o ser visualizados en forma de tabla en la base de datos.

Las características de los *datasets* que se generen, variarán en función del uso que se vaya a hacer de ellos. En el caso de los campos textuales, se puede optar por reagrupar los textos o mantenerlos como términos individuales.

Los *datasets* generados serán utilizados por otras herramientas para realizar procesos de minería de datos y texto sobre ellos.

Para ver los aspectos generales del diseño, la implementación de la herramienta y más información sobre las transformaciones de datos y el filtrado de textos, consultar [Cn11].

5.5.2 Herramientas para el Preprocesamiento Semántico

La herramienta utilizada en esta etapa del procesamiento se denomina "*SemanticPreprocessor*". Se trata de un único programa Java que realiza todas las etapas del preprocesamiento semántico. Aún así, para alguna de éstas, es preciso el uso de herramientas externas.

Se parte de una columna de la base de datos, se lee el texto, se procesa y una vez procesado, se almacena en una nueva columna de la tabla.

El programa trabaja a partir de un fichero de configuración donde se indica la conexión con la base de datos, la columna que contiene el texto a procesar, la tabla en la que se encuentra la columna y el nombre de la nueva columna en la que se escribirán los datos procesados. Este fichero también puede contener datos para indicar las herramientas externas que se usarán. En este caso, usaremos la herramienta "WordNet", en su versión 3.0 para sistemas UNIX (ver Apéndice A).

El acceso a WordNet a través de *SemanticPreprocessor* se realiza a través de la API de la librería JWNL, en su versión 1.4-rc2 de 10 de Julio de 2008.

Las etapas del preprocesamiento semántico se vieron en la Sección 5.2.2. A continuación, se resumen indicando lo realizado por nuestra herramienta en cada caso y las herramientas externas utilizadas.

- **Etiquetado de categoría gramatical (*Part Of Speech Tagging*)**.

  En esta etapa se toma el contenido textual de una fila y se etiquetan los términos con la categoría gramatical que desempeñan en la frase de la que se extraen.

  El etiquetado se realiza utilizando la API para etiquetado lingüístico de "*The Stanford Natural Language Processing Group*" [Tou00, Tou03]. Se usa la

librería "`stanford-postagger.jar`" (versión 3.0 de 26 de Mayo de 2010).

Tras pasar el texto al etiquetador, éste devuelve una lista de palabras con su etiqueta correspondiente. Procesamos esta lista de palabras y adaptamos las etiquetas "*Penn Treebank* [1]" a WordNet. En el Apéndice A.2, podemos ver todo el proceso de etiquetado y también se muestra la tabla de correspondencias entre ambos conjuntos de etiquetas.

Una vez que esta etapa ha concluido pasamos a la siguiente, que consiste en desambiguar el texto.

- **Desambiguación (*Word Sense Disambiguation*)**.

  Se determina el significado del término en cuestión, lo que será necesario para conocer sus sinónimos.

  Se toma la lista de palabras etiquetadas en la etapa anterior y se le aplica el algoritmo de desambiguación.

  Los algoritmos de desambiguación implementados en la herramienta son los siguientes:

  - Sin desambiguación: No se aplica desambiguación alguna. Se incluye esta posibilidad para comparar los resultados de las aproximaciones que usan desambiguación con un caso base y ver si, efectivamente, se mejoran los resultados aplicando desambiguación.
  - El sentido más frecuente: Como vimos, este algoritmo lo que hace es tomar el sentido más frecuente de un término. Dado que WordNet almacena los sentidos por orden de frecuencia, es tan sencillo como devolver el primero que ofrezca, por lo que se utiliza WordNet en este algoritmo.
  - Algoritmo simplificado de Lesk: Se basa en el solapamiento del contexto de una palabra con las definiciones de los sentidos en WordNet. La implementación se realizó siguiendo las especificaciones del algoritmo presentadas en [Vas04].

[1] "Penn Treebank English POS Tag Set" [Mar93]

- Algoritmo adaptado de Lesk: Recordemos que este algoritmo es muy similar al anterior, con la salvedad de que en la desambiguación no sólo se emplea la definición de cada uno de los sentidos en WordNet, sino también las definiciones de palabras relacionadas, hipónimos, tropónimos, etc. La implementación se realizó según la descripción de [Ban02].

- **Generación de conjuntos de sinónimos**.

  En esta etapa, una vez desambiguadas las palabras, se agrupan aquellas que comparten un mismo sentido, es decir, las que son sinónimas, formando agrupaciones de términos sinónimos.

- **Selección del representante canónico de los conjuntos de sinónimos**.

  Por último, se selecciona el término más frecuente de cada uno de los conjuntos de sinónimos como representante del conjunto. Dicho término sustituirá a todos los demás de su conjunto de sinónimos en el texto.

Con esto terminaría preprocesamiento semántico. La salida se guarda en la nueva columna que se indicó en el fichero de configuración.

En un futuro, pretende integrarse este programa dentro de la herramienta de preprocesamiento sintáctico.

5.5.3 Herramientas para la Generación de Formas Intermedias

La herramienta utilizada para generar las formas intermedias consiste en una extensión del programa Java "***TextAnalyzerTest***". Este programa se creó originalmente en nuestro grupo de investigación por los doctores Serrano Chica de la Universidad de Jaén y Martínez Folgoso de la Universidad de Camagüey (Cuba) y posteriormente, se amplió por el Dr. Campaña Gómez de la Universidad de Granada.

TextAnalyzerTest partía en un principio de un atributo textual en una base de datos PostgreSQL, analizaba el texto, generaba una representación basada en Conjuntos AP y la almacenaba en la base de datos junto al atributo original. Una descripción detallada de su funcionamiento se puede encontrar en [MF08].

Como PostgreSQL presentaba algunas limitaciones referentes a la representación de objetos complejos, se decidió abordar la implementación de los Conjuntos-AP en un sistema objeto-relacional más potente, como es Oracle™. Así, se llevó a cabo la modificación del programa original para soportar Oracle™, por el Dr. Campaña Gómez en colaboración con el Dr. Martínez Folgoso.

Esta modificación mantenía los métodos de interfaz y el paquete SQL que creaba la definición de la estructura de los conjuntos-AP, sus propiedades, métodos y algunas funciones auxiliares en PostgreSQL, para poder usarlos desde programas externos.

Más adelante, se incluyeron nuevas modificaciones con el fin de poder procesar términos que tuviesen asociado su identificador de *synset* en WordNet (ver Apéndice A) y por tanto, su sentido. Los términos etiquetados con su sentido se generan en la etapa de preprocesamiento semántico.

Y por último, se han incluido en el programa las estructuras APO, WAP y WAPO.

El funcionamiento es bastante sencillo: tras definir los parámetros en un fichero de configuración que es leído por el programa a través de la conexión a la base de datos, se calculan los conjuntos de términos (*itemsets* o *item-seqs*) maximales, según el soporte que se indique en el fichero de configuración y se genera la salida. Ésta se realiza a un fichero de texto y también se almacena en la base de datos como un tipo de objeto.

Además de la estructura de dominio, se calcula la subestructura correspondiente para cada una de las tuplas procesadas. Como hemos visto, la primera representa el contenido del atributo textual original, que se incluye en el catálogo de la base de datos y en una nueva columna de la tabla original y la segunda, representa la porción de la estructura general que se corresponde con cada tupla particular. Esto ayudará en la ejecución de las tareas de consulta.

Una vez que la estructura se genera y almacena, se puede usar para interactuar con los textos de la base de datos a través de ontologías o visualizaciones como la *tag cloud*.

5.5.4 Herramientas para la Generación de la *Tag Cloud*

Para la generación de la *tag cloud* hemos usado un programa Java que simula el algoritmo de distribución usado por Wordle. Éste asigna a cada etiqueta un tamaño de fuente proporcional a su frecuencia en el texto y la sitúa en una posición aleatoria alrededor del centro de la *tag cloud*.

Si un término se coloca en un lugar ocupado por otro, este último se mueve un paso hacia adelante dentro de una espiral creciente en torno al centro, hasta que su posición sea correcta.

El algoritmo también asigna color a las etiquetas, en principio de forma aleatoria.

La visualización generada resume y representa el contenido de la información del atributo textual correspondiente y además se usa para consultarlo.

La *tag cloud* se presenta al usuario como una interfaz donde, gracias a un módulo de consulta, cada etiqueta actúa como un conjunto de términos consulta, así cuando un usuario pulsa en alguna, el sistema devuelve las tuplas de la columna de texto que contienen los términos de dicha etiqueta.

5.6 Resumen y Conclusiones

En este capítulo hemos presentado una metodología para el procesamiento de los textos cortos no estructurados que aparecen en los campos textuales de las bases de datos.

Partimos de un atributo textual y a través de técnicas de minería de datos y texto obtenemos su dominio, representado por alguna de las formas intermedias provistas de estructura vistas en el Capítulo 3 y lo visualizamos a través de una *tag cloud*, permitiendo la identificación del contenido del atributo de forma visual y asistiendo a la consulta y la exploración.

La metodología propuesta se lleva a cabo en cuatro etapas:

1. Preprocesamiento sintáctico.

2. Preprocesamiento semántico.

3. Generación de la forma de representación intermedia.

4. Visualización.

En la etapa de preprocesamiento sintáctico utilizamos la herramienta de filtrado de datos “DB2DS” y aplicamos los siguientes filtros en este orden:

1. Tokenización

2. *S-Stemmer* en inglés

3. *Porter Stemming* en inglés

4. *S-Stemmer* en español

5. *Porter Stemming* en español

6. Eliminación de *Stop Words* en inglés

7. Eliminación de *Stop Words* en español

En la etapa de preprocesamiento semántico la herramienta utilizada se denomina “SemanticPreprocessor” y realiza las siguientes tareas en este orden:

1. Etiquetado de la categoría gramatical

2. Desambiguación

3. Generación de conjuntos de sinónimos

4. Selección del representante canónico de los conjuntos de sinónimos.

Este preprocesamiento semántico se realiza con ayuda de la herramienta externa “WordNet”, por lo que se ha presentado una descripción de esta herramienta especificando sus términos y conceptos básicos en el Apéndice A.

En la etapa de generación de formas intermedias usamos el programa “TextAnalyzerTest” y en la de visualización se representa la forma intermedia a través de un programa Java que simula el algoritmo de distribución de etiquetas usado por

Wordle, que asigna a cada etiqueta un tamaño de fuente proporcional a su frecuencia en el texto y la sitúa en una posición aleatoria.

Sin embargo, no todos los *itemsets* frecuentes serán aceptados para su visualización dentro de la *tag cloud*, sino solamente los que cumplan un conjunto de reglas atendiendo a su categoría gramatical.

Si el número de términos representados es muy alto, se puede optar por representar únicamente los conjuntos maximales, facilitando la identificación y evitando al mismo tiempo algunas redundancias. De esta forma, es posible disminuir el soporte para que aparezcan nuevos términos relevantes.

CAPÍTULO

6

Evaluación Experimental

Este capítulo se dedica a evaluar el método propuesto en base a su aplicación sobre varios conjuntos de datos reales.

En la Sección 6.1 se describe un primer experimento realizado sobre una base de artículos científicos de la revista "Security and Communication Networks" de la editorial "Wiley". Consiste en generar de forma automática una *tag cloud* a través de nuestra propuesta y compararla con la que proporciona la propia revista en su sitio web.

En el experimento de la Sección 6.2 se emplea una base de historias clínicas. Este segundo conjunto de datos es más extenso que el primero y presenta mayores necesidades de limpieza. Se prueba que la *tag cloud* construida cumple las expectativas de recuperación de información y de representación del contenido, a partir también del cálculo de métricas y de una encuesta de satisfacción.

Vemos con menor nivel de detalle otros experimentos realizados sobre diferentes bases de datos en la Sección 6.3 y se termina con un resumen y algunas conclusiones en la Sección 6.4.

6.1 Evaluación Experimental sobre una Base de Artículos Científicos

6.1.1 Descripción del Conjunto de Datos

Entre los sitios web que ofrecen *tag clouds* con propósitos de recuperación y visualización de información, hemos escogido el de la editorial "Wiley": **Wiley Online Library** [1]. Algunas de las revistas presentes en este sitio, ofrecen la posibilidad de explorar su contenido a través de *tag clouds* construidas a partir de las palabras clave (*keywords*) de los artículos de investigación.

Aunque no existe información sobre el método usado para generarlas, han sido posiblemente creadas por expertos. Algunos de los términos que aparecen en ellas no son relevantes en los datos originales y parecen haber sido finamente ajustados de forma manual. Este hecho carece de importancia ya que, independientemente del método de generación usado, el principal factor a considerar es que estas *tag clouds* son las proporcionadas por la propia editorial, por lo que es de suponer, que se consideran una herramienta efectiva para la exploración y la búsqueda de los contenidos de las revistas y son vistas por los expertos del sitio como herramientas optimizadas para estos propósitos.

De acuerdo con esto, hemos recogido información procedente de diferentes revistas, con la intención de seleccionar aquellas que puedan ser usadas para probar en ellas nuestro método. Estas revistas, deben de ofrecer una *tag cloud* para explorar su contenido (ya que no todas lo hacen) y también deben contener todos sus artículos en formato electrónico. Esto último garantiza que la *tag cloud* que muestran sea reproducible a partir de los datos.

La revista **"Security and Communication Networks"** [2], que está clasificada en la categoría de *Electrical and Electronics Engineering*, cumple estas dos premisas.

Hemos extraído de esta revista toda la información sobre los artículos disponibles, en total 346, aunque algunos de ellos se han desechado por no contener las palabras clave. Toda la información recogida se ha almacenado en una base de

[1] http://onlinelibrary.wiley.com/

[2] http://onlinelibrary.wiley.com/journal/10.1002/(ISSN)1939-0122

datos relacional usando un atributo para cada uno de los campos (título, resumen, autores,...) y añadiendo un identificador a cada tupla [TP13b].

6.1.2 Descripción de la Metodología

Nuestro objetivo es la generación automática de una *tag cloud* que rivalice en calidad con la elegida de la revista de Wiley.

En esta sección se describe la metodología para su obtención a partir de los datos en sucio. Habrá que decidir qué tipo de preprocesamiento aplicar (si únicamente sintáctico o sintáctico más semántico), qué tipo de estructura extraer (WAP o WAPO), qué atributo escoger de la base de datos, etc. A continuación se describe cada uno de estos procesos y posteriormente se comentan los aspectos que nos han hecho escoger los método aplicados finalmente.

Preprocesamiento

El preprocesamiento sintáctico lo hemos llevado a cabo con la herramienta de filtrado descrita en la Sección 5.5.1. Los filtros que se aplican en los datos almacencados en la base de datos son los siguientes y se emplean en el mismo orden en que se exponen:

1. Tokenización
2. Eliminación de *Stop Words*
3. Lematización simple para eliminar formas de género y plural.

Gracias a este proceso es posible eliminar signos de puntuación y palabras que no aportan significado.

Usamos una lematización simple porque en la segunda fase de preprocesamiento, la semántica de los términos se determina con herramientas externas que no reconocen las formas lematizadas y tampoco sería fácil identificar el significado de las mismas si aparecen en una *tag cloud*.

Los datos preprocesados sintácticamente se almacenan en nuevas columnas de la base de datos.

Los algoritmos Apriori y Apriori modificado funcionan a partir de los términos frecuentes encontrados en el texto. El principal inconveniente a la hora de calcular la frecuencia de un término determinado, es que ciertos conceptos pueden expresarse a través de varias formas léxicas, por lo que un término puede aparecer menos veces de lo que en realidad está presente, debido al hecho de que el concepto que representa está expresado en el texto con otras formas distintas.

Otro inconveniente de la existencia de esta variedad lingüística con la que se representan los conceptos, es que si todos los términos equivalentes en significado se visualizan en la *tag cloud*, además de existir solapamiento semántico, la identificación del contenido puede verse distorsionada.

Para abordar este problema, es necesario aplicar un preprocesamiento semántico. El objetivo del mismo es identificar todas las formas léxicas referidas a un único concepto y agruparlas en un conjunto que llamaremos "conjunto de sinónimos", del cual se elige una única forma denominada "representante canónico del conjunto de sinónimos" para reemplazar a todas las demás en el texto. De esta manera, cada concepto queda representado por un único término cuya frecuencia puede calcularse con precisión.

Este proceso se ha llevado a cabo con la herramienta de preprocesamiento semántico descrita en la Sección 5.5.2 que se desarrolla en los siguientes pasos [TP13a]:

1. Etiquetado de la categoría gramatical (*Part of speech tagging*), con la ayuda de "Stanford Tagger[1]". Se han etiquetado los términos con la categoría gramatical que desempeñan en la sentencia de la que se extraen.

2. Desambiguación *Word sense disambiguation*, con el algoritmo adaptado de Lesk [Ban02] y el diccionario léxico WordNet. Se ha determinado el significado del término en cuestión, ya que sólo conociendo su significado podrán determinarse los sinónimos.

3. Generación de conjuntos de sinónimos. Los acrónimos y sinónimos son reconocidos con la ayuda de WordNet.

[1] http://nlp.stanford.edu/software/tagger.shtml

4. Selección de los representantes canónicos de los conjuntos de sinónimos. Hemos seleccionado el término más frecuente de cada conjunto de sinónimos como representante del conjunto, el cual sustituye en el texto todos los demás términos que pertenecen al mismo conjunto de sinónimos.

Una vez que los datos han sido preprocesados semánticamente, el resultado se almacena en nuevas columnas de la base de datos.

Generación de la Forma Intermedia de Representación y Visualización

Las formas de representación intermedias (estructuras WAP y WAPO) se extraen con la herramienta descrita en la Sección 5.5.3.

Como atributos textuales de interés se han seleccionado los siguientes campos:

1. Palabras clave (*keywords*)
2. Títulos
3. *Keywords* + Títulos
4. Resúmenes
5. *Keywords* + Títulos + Resúmenes

El tercero de estos campos consiste en la unión en un sólo atributo de las palabras clave y los títulos y el último en la unión de palabras clave, títulos y resúmenes.

Para generar las estructuras hay que especificar el soporte mínimo considerado, teniendo en cuenta que pequeñas variaciones en este soporte, afectarán al número de elementos en la estructura final, el cual debe ser lo suficientemente grande como para permitir que la visualización de la estructura sea representativa, pero no tan grande que dificulte la correcta identificación de los términos.

Nuestro objetivo es obtener un número de elementos aproximado al que encontramos en la *tag cloud* que ofrece la propia revista en su sitio web y que tomamos como referencia.

Para calcular el valor del soporte que nos de este número de elementos, hemos realizado un análisis de ensayo y error, en el que un grupo de expertos pertenecientes a nuestro grupo de investigación han evaluado visualmente las *tag clouds*

obtenidas para distintos valores. Finalmente se ha acordando establecer como definitivo un soporte del 2 %, por ser el que ofrece un número de elementos más próximo al contemplado en la visualización de referencia.

Como ya sabemos, las estructuras son retículos y cada retículo viene determinado por los *itemsets* o *item-seqs* maximales, que son aquellas que no están contenidas en ninguna otra. La siguiente cuestión a considerar es si será mejor representar todos los *itemsets* frecuentes en la *tag cloud* o sólo los maximales, como hacían Kaptein y Marx [Kap10],

Postprocesamiento

Una vez generadas las estructuras WAP y WAPO y obtenida toda la información de los *itemsets* frecuentes y el peso de cada uno de ellos, pasamos a la fase de postprocesamiento.

Recordemos que en la fase de preprocesamiento semántico, se habían etiquetado todos los términos con su categoría gramatical con la ayuda de WordNet. Las categorías gramaticales contempladas son: nombre, verbo, adjetivo y adverbio. Los nombres son buenos candidatos para aparecer representados en la *tag cloud*, pero los verbos y los adverbios no son palabras especialmente útiles para representar el contenido de la información. Es por esta razón, que en la etapa de postprocesamiento, hemos eliminado todos los *itemsets* que contenían verbos o adverbios. Los adjetivos por sí sólos no son informativos, pero sí lo son si van acompañando a un nombre, dado que lo modifican añadiéndole semántica.

Considerando estas observaciones, se definió el conjunto de reglas que vimos en la Sección 5.4 para determinar cuando un *itemset* cumplía los requisitos para su visualización en la *tag cloud* atendiendo a su categoría gramatical. Estas reglas son las siguientes (recordemos que están pensadas para el idioma inglés) (Consultar [TP13b]):

- ***Itemsets* de nivel uno:** Se consideran buenos candidatos si son nombres [N].
- ***Itemsets* de nivel dos:** Se consideran buenos candidatos si son nombres precedidos de adjetivos [AN] o nombres compuestos [NN].

- ***Itemsets* de nivel n:** Se consideran buenos candidatos si están compuestos por una combinación válida de términos en los niveles previos, precedida por un adjetivo o seguida por un nombre. Por ejemplo: [AAN], [ANN] y [NNN].

Además de los *itemsets* frecuentes que contenían verbos o adverbios, hemos eliminado todos los que no cumplían estas reglas. Los restantes serán los que aparezcan representados en la *tag cloud* correspondiente.

Posibles Formas de Generar la *Tag Cloud*

Atendiendo al preprocesamiento aplicado, al atributo procesado, al tipo de estructura generada y a los *itemsets* visualizados, tendríamos un total de 40 posibles formas distintas de construir la *tag cloud*. Estas formas se resumen en la Tabla 6.1 y algunas de las visualizaciones pueden verse a continuación. La razón de que no se muestren todas es que éstas se han ido descartando conforme las hemos ido analizando. Así, primeramente se han descartado las construidas a partir de textos largos y luego las obtenidas tras el preprocesamiento exclusivamente sintáctico.

A continuación discutimos el método empleado finalmente.

Selección del Método a Emplear

Atendiendo al atributo empleado para la generación de las estructuras WAP y WAPO, se observa que las *tag clouds* generadas a partir de textos largos como resúmenes, parecen menos apropiadas para este tipo de procesamiento que las obtenidas a partir de textos cortos como títulos o palabras clave, por lo que se han descartado los dos últimos atributos considerados en un principio.

En las *tag clouds* de estas estructuras los términos aparecen visualmente muy aglomerados y esta mayor densidad en la representación dificulta la identificación del contenido. Además, el frecuente uso de formalismos y conectores en los textos largos, hace que estas palabras posean frecuencias muy altas, por lo que se tiene un gran número de palabras vacías de contenido que se extraen como relevantes cuando en realidad sólo sirven para articular el discurso.

En el caso de querer obtener las formas intermedias de textos largos, la estructura WAPO resulta más apropiada que la estructura WAP, ya que la falta de orden

Preprocesamiento	Atributo Seleccionado	Estructura	*Itemsets* Representados
Sintáctico	*Keywords*	WAP	Todos (**Figura 6.1**)
			Maximales
		WAPO	Todos (**Figura 6.2**)
			Maximales
	Títulos	WAP	Todos (**Figura 6.3**)
			Maximales
		WAPO	Todos (**Figura 6.4**)
			Maximales
	Keywords + Títulos	WAP	Todos
			Maximales
		WAPO	Todos
			Maximales
	Resúmenes	WAP	Todos
			Maximales
		WAPO	Todos
			Maximales
	Keywords + Títulos + Resúmenes	WAP	Todos
			Maximales
		WAPO	Todos
			Maximales
Sintáctico + Semántico	*Keywords*	WAP	Todos (**Figura 6.5**)
			Maximales (**Figura 6.6**)
		WAPO	Todos (**Figura 6.7**)
			Maximales (**Figura 6.8**)
	Títulos	WAP	Todos (**Figura 6.9**)
			Maximales (**Figura 6.10**)
		WAPO	Todos (**Figura 6.11**)
			Maximales (**Figura 6.12**)
	Keywords + Títulos	WAP	Todos (**Figura 6.13**)
			Maximales (**Figura 6.14**)
		WAPO	Todos (**Figura 6.15**)
			Maximales (**Figura 6.16**)
	Resúmenes	WAP	Todos
			Maximales
		WAPO	Todos
			Maximales
	Keywords + Títulos + Resúmenes	WAP	Todos
			Maximales
		WAPO	Todos
			Maximales

Tabla 6.1: Número de representaciones obtenidas según tipo de preprocesamiento, atributo escogido, estructura obtenida e *itemsets* representados

en esta última acarrea mayores co-ocurrencias en los términos, lo que incrementa significativamente el número de conjuntos-AP. De hecho, cuanto más largo es un texto, será mayor el número de términos que co-ocurran y menos informativa la estructura.

Nos centramos únicamente en los campos "*Keywords*" y "Títulos" y "*Keywords* + Títulos".

Vemos como en todas las *tag clouds* construidas a partir estructura WAPO, el número de términos es menor que en las construidas a partir de la estructura WAP, resultando un diseño más espaciado, lo que facilita la identificación de las etiquetas.

Las *tag clouds* en las que se representan todos los *itemsets* aparecen sobrecargadas además de contener bastantes redundancias. En cambio, las *tag clouds* en las que se representan únicamente las secuencias maximales, parecen simples y compactas, por lo que resultan preferibles.

Sin embargo, vemos que al visualizar sólo las *item-seqs* maximales, algunas etiquetas cobran una importancia que no les corresponde, como es el caso del término "*using*" para el atributo "Títulos". Esto se soluciona añadiendo estos términos a una lista de palabras dependientes de contexto y considerándolos como *stop-words* [Kap10].

Tras analizar todos estos modelos para determinar cuál es el mejor, hemos decidido optar por el modelo alcanzado con las siguientes combinaciones:

1. **Preprocesamiento sintáctico y semántico**: Con el preprocesamiento semántico se identifican los diferentes sentidos de las palabras polisémicas, evitando que dominen la *tag cloud* con sus altas frecuencias cuando estos sentidos se representan con una sola palabra.

2. **Estructura WAPO como forma intermedia de representación**: La estructura WAPO es más precisa al tener en cuenta el orden de los términos en el texto y visualmente es más compacta y tiene menor densidad semántica que la estructura WAP. Además, la identificación del contenido no es tan clara con esta última.

Figura 6.1: *Tag cloud* para "*Keywords*" con estructura WAP y preprocesamiento sintáctico

Figura 6.2: *Tag cloud* para "*Keywords*" con estructura WAPO y preprocesamiento sintáctico

Figura 6.3: *Tag cloud* para "Títulos" con estructura WAP y preprocesamiento sintáctico

Figura 6.4: *Tag cloud* para "Títulos" con estructura WAPO y preprocesamiento sintáctico

Figura 6.5: *Tag cloud* para "*Keywords*" con estructura WAP y preprocesamientos sintáctico y semántico

Figura 6.6: *Tag cloud* de *itemsets* maximales para "*Keywords*" con estructura WAP y preprocesamientos sintáctico y semántico

Figura 6.7: *Tag cloud* para "*Keywords*" con estructura WAPO y preprocesamientos sintáctico y semántico

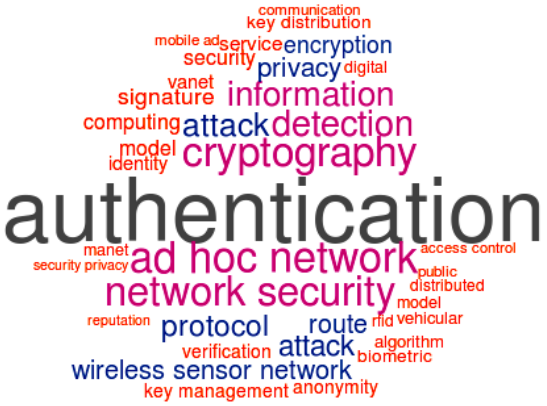


Figura 6.8: *Tag cloud* de *itemsets* maximales para "*Keywords*" con estructura WAPO y preprocesamientos sintáctico y semántico

Figura 6.9: *Tag cloud* para "Títulos" con estructura WAP y preprocesamientos sintáctico y semántico

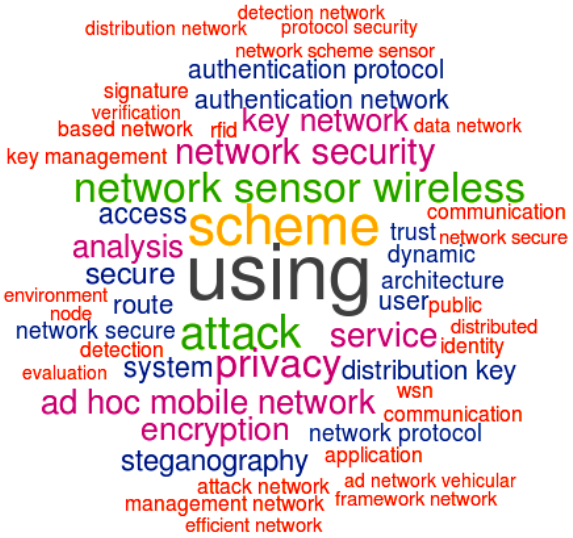


Figura 6.10: *Tag cloud* de *itemsets* maximales para "Títulos" con estructura WAP y preprocesamientos sintáctico y semántico

Figura 6.11: *Tag cloud* para "Títulos" con estructura WAPO y preprocesamientos sintáctico y semántico

Figura 6.12: *Tag cloud* de *itemsets* maximales para "Títulos" con estructura WAPO y preprocesamientos sintáctico y semántico

Figura 6.13: *Tag cloud* para "*Keywords* + Títulos" con estructura WAP y preprocesamientos sintáctico y semántico

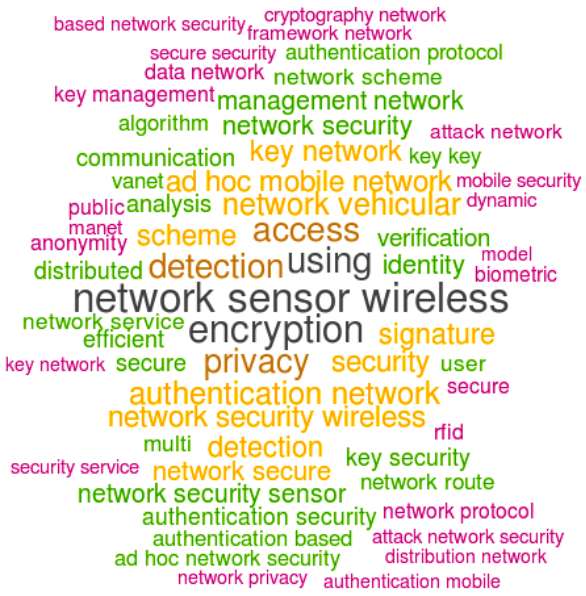


Figura 6.14: *Tag cloud* de *itemsets* maximales para "*Keywords* + Títulos" con estructura WAP y preprocesamientos sintáctico y semántico

Figura 6.15: *Tag cloud* para "*Keywords* + Títulos" con estructura WAPO y preprocesamientos sintáctico y semántico

Figura 6.16: *Tag cloud* de *itemsets* maximales para "*Keywords* + Títulos" con estructura WAPO y preprocesamientos sintáctico y semántico

3. **Representación sólo de *item-seqs* maximales**: Cuando se representan todas las *item-seqs* encontramos algunas redundancias y con la representación únicamente de las maximales, el diseño es más simple y compacto y permite destacar los componentes multitérmino, que de otra forma aparecen muy reducidos debido al gran soporte de los monotérminos.

Falta por decidir el atributo a procesar para establecer el modelo definitivo. En la Figura 6.17 podemos ver la *tag cloud* generada con esta configuración para los atributos textuales que habíamos seleccionado: "*Keywords*", "Títulos" y "*Keywords* + Títulos".

(a) Keywords (b) Títulos (c) Keywords+Títulos

Figura 6.17: *Tag clouds* generadas para los distintos atributos con el método seleccionado

Para la generación de estas *tag clouds* se ha usado el programa descrito en la Sección 5.5.4, que simula el algoritmo de distribución de etiquetas usado por Wordle, que lo que hace es asignar a cada palabra un tamaño de fuente atendiendo a su peso. El orden para colocar las palabras en el diseño es también según el peso, así se situarán primero las que mayor peso tengan, asignándose el resto de forma aleatoria. Cuando una palabra se solape con otra en el diseño, ésta se moverá un paso hacia adelante en espiral. Esta vez hemos usado los mismos colores que Wiley para facilitar la comparación visual con ésta última.

La *tag cloud* de referencia de la revista de Wiley está construida a partir de las *keywords* de los artículos y la consulta a través de las etiquetas también se realiza sobre este atributo. Nosotros hemos obtenido tres *tag clouds* similares a

la de referencia, las tres igualmente representativas y útiles para los propósitos perseguidos. Finalmente, elegiremos entre estas tres la construida sobre el atributo "*Keywords*" para hacer una comparación equivalente con la de Wiley.

6.1.3 Resultados

Comparación de *Tag Clouds*

Después de establecer el procedimiento de generación de la *tag cloud*, se ha seleccionado la primera (Figura 6.17(a)) entre las tres obtenidas en la Figura 6.17 para compararla con la de referencia, por estar ambas generadas a partir de las palabras clave.

La Figura 6.18 muestra la *tag cloud* de referencia, la que hemos generado automáticamente y las diferencias entre ellas. La Figura 6.18(b) muestra todos los términos que están en la de referencia y que no están en la nuestra. De forma análoga, la Figura 6.18(d) muestra los términos que están en la nuestra y que no están en la de Wiley.

En la *tag cloud* de referencia sólo se hace uso del color con propósitos de visualización, ya que no está relacionado con la frecuencia de las etiquetas ni con ningún otro aspecto, así que nosotros hemos usado los mismos colores con el mismo fin.

En las Figuras 6.18(b) y 6.18(d), que muestran las diferencias entre ambas representaciones, el color gris indica que el término está incluido parcialmente en la otra *tag cloud*, pero no totalmente. Por ejemplo, *Intrusion Detection*, está totalmente incluido en la de referencia, pero sólo parcialmente en la nuestra: *Detection*.

Evaluación a través de Métricas

Para medir de forma cuantitativa las diferencias desde la perspectiva de la recuperación de información entre la *tag cloud* de referencia y la que hemos generado de forma automática, hemos calculado las medidas de *precisión*, *exhaustividad* y F_1 *Score* para las etiquetas presentes en ambas visualizaciones, entendiendo que cada una de estas etiquetas funcionaría como término o términos de consulta cuando el usuario la elija.

La métrica F_1 *Score* (vista en la Sección 3.5.5) se calcula en función de los valores de *precisión* y *exhaustividad* y la usamos como medida estándar por ser

(a) *Tag cloud* de referencia

(b) Términos no presentes en (c)

(c) *Tag cloud* generada automáticamente

(d) Términos no presentes en (a)

Figura 6.18: Comparación de la *tag cloud* de referencia (a), y la *tag cloud* generada con nuestro método (c) a partir del atributo "*Keywords*"

muy conocida y ampliamente utilizada. Es la que encontramos en trabajos como Balachandran et al. [Bal12] o Masadan et al. [Mas12].

En realidad, como lo que queremos es comparar la *precisión* y la *exhaustividad* de la *tag cloud* de referencia y la generada automáticamente, sólo es necesario calcular éstas para las etiquetas que están presentes en una de las dos visualizaciones y ausentes en la otra, ya que el resto de etiquetas, al estar presentes en ambas, comparten el mismo valor y no aportan nada en términos comparativos.

Para este cálculo, previamente es necesario anotar los artículos de la base de datos con las etiquetas más apropiadas de la *tag cloud*, para así conocer cuáles de-

berían recuperarse con cada una de estas etiquetas. Esta tarea ha sido realizada con la ayuda de un grupo de expertos pertenecientes a nuestro grupo de investigación, a los que se les han proporcionado los datos originales, incluyendo las palabras clave, títulos y resúmenes.

En la Tabla 6.2 se muestran las medias de *precisión*, *exhaustividad* y F_1 *Score* calculadas. Como vemos, con la *tag cloud* de referencia se obtiene algo más de *precisión* que con la que hemos generado de forma automática y un poco menos *exhaustividad*. Esto significa que los resultados obtenidos con las consultas a través de la de referencia son un poco más relevantes que los obtenidos con la nuestra, pero la nuestra recupera resultados importantes que no se recuperan con la otra.

En general las medidas difieren muy poco unas de otras y la F_1 *score* es casi la misma para ambas visualizaciones (ver [TP13b]).

Tag Clouds	*Precisión*	*Exhaustividad*	F_1 *Score*
Referencia	**0.82609**	0.72743	**0.76208**
Generada Automáticamente	0.76809	**0.76476**	0.75173

Tabla 6.2: *Precisión media, exhaustividad* y F_1 *Score* para las *tag clouds* (a) y (c) de la Figura 6.18

En la Sección 2.1.4 se vieron algunas métricas para la evaluación de la efectividad de la *tag cloud*. Las principales son las proporcionadas por Venetis et al. [Ven11]. Sin embargo, el uso que le dan a la *tag cloud* estos autores y otros como Durao et al. [Dur12] difiere del nuestro, ya que ellos la construyen para resumir los resultados obtenidos tras una consulta previa y nosotros pretendemos principalmente representar el contenido global de los atributos textuales de la base de datos.

Por esta razón, no tiene sentido calcular las métricas de *cohesividad*, *relevancia*, *popularidad* e *independencia*, las cuales requieren distinguir entre el contenido del atributo y el conjunto de resultados que se recuperan con la consulta. Calcularemos únicamente las de *cobertura*, *solapamiento* y *balance* con las fórmulas proporcionadas por Venetis et al. [Ven11]. Para ello supondremos que con una consulta hipotética se recuperan todas las tuplas del atributo textual.

Las fórmulas de *relevancia*, *solapamiento* y *balance* corresponden a las Ecuaciones 2.3, 2.4 y 2.5, respectivamente. En nuestro caso particular, q representaría

los términos de la consulta hipotética inicial, S la *tag cloud*, t cada una de las etiquetas en S, C_q el contenido del atributo (que es lo que devolvería la consulta) y $A_q(t)$ el conjunto de objetos recuperados con t.

La *cobertura* nos da la fracción del texto original representada por los términos de la *tag cloud*. Esta métrica toma valores en el intervalo $[0, 1]$. Un valor cercano a 1 indica que los términos en la *tag cloud* representan la mayor parte del texto original.

El *solapamiento* determina el grado en que diferentes términos en la *tag cloud* representan la misma información en el texto original. Esta métrica toma valores también en el intervalo $[0, 1]$. Un valor cercano a 0 indica que el grado de solapamiento es bajo y por lo tanto, los términos representan información distinta.

El *balance* es una métrica relacionada con la cantidad de resultados recuperados por las etiquetas. Se dice que una *tag cloud* es equilibrada (*balanced*) si sus etiquetas recuperan un número similar de resultados. Esta métrica también toma valores en el intervalo $[0, 1]$. Un valor cercano a 1 indica que la *tag cloud* es equilibrada.

En la Tabla 6.3 podemos ver los valores obtenidos para estas métricas.

Tag Clouds	*Cobertura*	*Solapamiento*	*Balance*
Referencia	0.85838	**0.00455**	0
Generada Automáticamente	**0.92196**	0.00550	**0.06289**

Tabla 6.3: Comparación de la *cobertura*, *solapamiento* y *balance* de las *tag clouds* (a) y (c) de la Figura 6.18

Como podemos observar, la *tag cloud* que hemos generado automáticamente tiene mejor *cobertura* que la de referencia, lo que indica que representa mejor el texto original. Sin embargo, el *solapamiento* es un poco mejor en la de referencia, aunque ambos son practicamente 0, por lo que las etiquetas representarían distinta información del texto original.

Es importante indicar que el valor del *balance* para la *tag cloud* de referencia es 0 debido a que hay al menos una etiqueta (*data compression*) que no recupera ninguna entrada. Esta etiqueta es relevante en el ámbito de la revista y de acuerdo a los criterios de los expertos que la han incluido en la visualización, sin embargo no es útil desde el punto de vista de la recuperación de información.

Nuestra *tag cloud* tampoco es equilibrada. Podríamos intentar obtener mejor *balance* aumentado el soporte mínimo, lo que implicaría que las etiquetas menos relevantes saldrían fuera de la representación, pero de esta forma se perdería información importante y realmente no es necesario que sea equilibrada, es más, la heterogeneidad en la frecuencia de las etiquetas (aspecto relacionado con la cantidad de resultados que se recuperan) es lo que da a este tipo de visualización la capacidad de destacar los temas más relevantes.

En la Tabla 6.3 teníamos los valores de *cobertura*, *solapamiento* y *balance* calculados sobre el atributo "*Keywords*". Pero la *tag cloud* que hemos generado podría usarse, no sólo para consultar sobre este atributo, si no también para los demás. Veamos en qué proporción representaría a los atributos "Títulos" y "*Keywords* + Títulos" en comparación con la de referencia (ver Tabla 6.4).

Tag Clouds	***Cobertura* para "Títulos"**	***Cobertura* para "*Keywords* + Títulos"**
Referencia	0.69942	0.90751
Generada Automáticamente	**0.76300**	**0.95375**

Tabla 6.4: Comparación de la *cobertura* sobre los atributos "Títulos" y "*Keywords* + Títulos"

Aunque al final hemos seleccionado la *tag cloud* construida sobre el atributo "*Keywords*", ésta también ofrece buenos resultados de *cobertura* para los otros dos atributos: "Títulos" y "*Keywords* + Títulos", siendo incluso mayor el valor de *cobertura* para este último, en donde supera el 95 %.

En todos los casos hemos mejorado la *cobertura* de la *tag cloud* de referencia.

En general, los resultados obtenidos indican que hemos generado de forma automática una herramienta similar a la creada por expertos tomada de referencia y que presenta frente a ésta las siguientes ventajas (ver [TP13b]):

- Posee mayor *cobertura*
- Todos los términos recuperan información relevante
- No ha sido precisa la intervención humana para su generación

Encuesta de Usabilidad y Análisis Estadístico

Además de la comparar de las *tag clouds* mediante el cálculo de métricas, hemos llevado a cabo la realización de una encuesta para conocer la opinión de los usuarios.

Todos los resultados estadísticos presentados en este capítulo se han obtenido con el software *StatGraphics Plus 5.1* [1].

Participantes

Para estimar el tamaño de muestra necesario, se ha fijado un error máximo absoluto de 0.5 unidades. Considerando una desviación típica igual a 1 y una confianza del 95 %, el tamaño mínimo necesario es de 18 participantes:

```
Determinación del Tamaño de la Muestra
--------------------------------------
Parámetro a estimar: media normal
Tolerancia deseada: +- 0,5
Nivel de confianza: 95,0%
Sigma asumida: 1,0

El tamaño de la muestra requerido es de n=18 observaciones.
```

En total se ha conseguido reunir a 34 participantes anónimos con formación en Ciencias de la Computación y en Redes de Computadores. Como este tamaño de muestra es mucho mayor que el mínimo requerido, se tienen garantías de que el error no va a ser superior a 0.5 unidades, incluso considerando los posibles valores perdidos que se tengan en estas 34 observaciones.

Procedimiento

Se ha contactado vía e-mail con especialistas de las áreas antes mencionadas, solicitándoles colaboración. A estas personas se les han proporcionado las dos *tag clouds*, la de referencia y la generada por nosotros, sin especificarles cuál es una ni cuál es la otra. Para hacer el experimento lo más aleatorio posible y debido al hecho de que leemos de izquierda a derecha, a la mitad de las personas contactadas se les ha dado nuestra *tag cloud* a la izquierda y la de referencia a la derecha y a la otra

[1] http://www.statgraphics.com/statgraphics_plus.htm

mitad al contrario, para que el orden en que éstas se presentan a los participantes no influya en los resultados.

Junto con las *tag clouds* se les han dado unas instrucciones, pidiéndoles que experimenten con ambas herramientas para posteriormente realizar una serie de valoraciones en la comparación de éstas. En el Apéndice B.1 se incluye el documento con las instrucciones y unas cuestiones referentes a los distintos aspectos a valorar por los participantes.

En las tres primeras cuestiones planteadas se les pide que emitan una valoración en la comparación de las *tag clouds* con respecto a:

1. Facilidad de uso
2. Cantidad global de información recuperada
3. Magnitud y precisión del contenido representado

En las cuatro últimas cuestiones se les solicita que identifiquen 4 conceptos diferentes. Para ello se les dan 4 definiciones, correspondientes a cada uno de los conceptos y se les pide que, tras localizar el término apropiado para cada definición en cada una de las visualizaciones, comparen ambas con respecto a la facilidad para identificar dicho concepto.

En la Tabla 6.5 podemos ver las definiciones de los conceptos y el término que se espera que los participantes encuentren para cada una de ellas.

	Definición	**Término**
1	Vulneración del sistema	Ataque (*Attack*)
2	Verificación de identidad	Autentificación (*Authentication*)
3	Técnicas de cifrado y/o codificación	Criptografía (*Cryptography*)
4	Prevención de accesos no autorizados en una red	Seguridad de red (*Network Security*)

Tabla 6.5: Definición de conceptos y objetos esperados a identificar en las *tag clouds* (a) y (c) de la Figura 6.18

Las valoraciones emitidas por los participantes para cada una de las cuestiones evaluadas llevan asociada una codificación numérica en un rango del 1 al 5, con el siguiente detalle:

1. $TC1$ es mucho mejor que $TC2$
2. $TC1$ es un poco mejor que $TC2$
3. $TC1$ es igual que $TC2$
4. $TC1$ es un poco peor que $TC2$
5. $TC1$ es mucho peor que $TC2$

Para preservar el anonimato de las visualizaciones, las hemos llamado $TC1$ y $TC2$, siendo:

$TC1$: *Tag cloud* de referencia
$TC2$: *Tag cloud* generada automáticamente

Análisis Estadístico y Resultados

Empezaremos probando si las valoraciones en la facilidad de identificación de conceptos son independientes del concepto proporcionado. De ser así, podemos unir las valoraciones para los 4 conceptos en una única variable que represente la valoración global en la identificación de conceptos.

Lo primero que hacemos es un resumen estadístico donde se muestra la media, mediana y moda de las valoraciones en la facilidad de identificación de cada uno de los conceptos proporcionados:

Resumen Estadístico

	Frecuencia	Media	Mediana	Moda
concepto1	34	3,05882	3,0	4,0
concepto2	33	2,93939	3,0	2,0; 3,0; 4,0
concepto3	31	2,74194	2,0	2,0
concepto4	34	3,17647	3,0	4,0
Total	132	2,98485	3,0	4,0

Como la media de las 4 valoraciones está próxima a 3, valor que indica que $TC1$ y $TC2$ se comportan de la misma forma, a simple vista parece que ambas *tag clouds* se consideran iguales con respecto a su facilidad para identificar los conceptos.

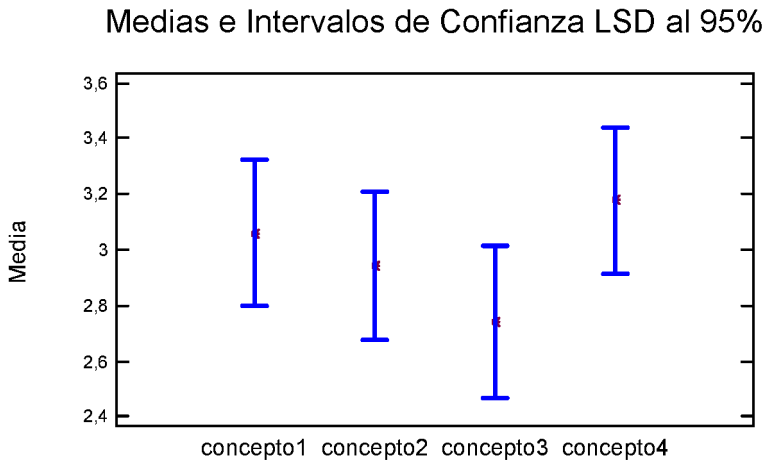


Figura 6.19: Gráfico de medias con intervalos de confianza al 95 % para la identificación de conceptos

En el gráfico de la Figura 6.19 se representan los intervalos al 95 % de confianza para estas medias. Vemos que existe solapamiento en todos estos intervalos, por lo que las medias podrían considerarse iguales a ese nivel de confianza. Así mismo, se observa que todos los intervalos incluyen el valor 3. Esto mismo puede verse numéricamente en la tabla de medias:

Tabla de Medias
con 95,0 intervalos LSD

	Frec.	Media	Error Estándar (s agrupada)	Límite inf.	Límite sup.
concepto1	34	3,05882	0,187278	2,7968	3,32085
concepto2	33	2,93939	0,190094	2,67343	3,20536
concepto3	31	2,74194	0,196131	2,46752	3,01635
concepto4	34	3,17647	0,187278	2,91444	3,4385
Total	132	2,98485			

En el contraste múltiple de rangos vemos que todos los grupos son homogéneos y que no existen diferencias significativas entre las medias de ninguno de los conceptos:

Contraste Múltiple de Rango

Método: 95,0 porcentaje LSD

	Frec.	Media	Grupos homogéneos
concepto3	31	2,74194	X
concepto2	33	2,93939	X
concepto1	34	3,05882	X
concepto4	34	3,17647	X

Contraste	Diferencias	+/- Límites
concepto1 - concepto2	0,11943	0,528009
concepto1 - concepto3	0,316888	0,536583
concepto1 - concepto4	-0,117647	0,524054
concepto2 - concepto3	0,197458	0,540446
concepto2 - concepto4	-0,237077	0,528009
concepto3 - concepto4	-0,434535	0,536583

* indica una diferencia significativa.

Para reafirmar este resultado, realizamos un análisis de la varianza para ver si podemos considerar iguales las medias de las valoraciones:

Tabla ANOVA

Análisis de la Varianza

Fuente	Sumas de cuad.	Gl	Cuadrado Medio	Cociente-F	P-Valor
Entre grupos	3,3319	3	1,11063	0,93	0,4277
Intra grupos	152,638	128	1,19248		
Total (Corr.)	155,97	131			

El estadístico experimental en el contraste sobre la igualdad de medias, F, tiene un valor igual a 0.93 y un p-valor de 0.4277, por lo que aceptamos la hipótesis nula, es decir, la igualdad de las medias.

Como última verificación, realizamos un test de independencia para comprobar si la valoración media en la facilidad de identificación de los distintos conceptos es independiente del concepto proporcionado. Esto implicaría que las valoraciones son consistentes para las *tag clouds* con independencia de los conceptos evaluados.

Este test de independencia está basado en el estadístico Chi-cuadrado [1]:

```
Contraste de Chi-cuadrado
---------------------------------------
Chi-cuadrado        GL         P-Valor
---------------------------------------
       10,06        12          0,6109
---------------------------------------
```

Este estadístico se ha calculado para un total de 132 observaciones en los 4 grupos, obteniendo un valor igual a 10.6 y un p-valor de 0.6109, lo que nos lleva a aceptar la hipótesis nula de independencia del concepto proporcionado en las valoraciones de identificación de conceptos.

También el coeficiente de contingencia tiene un valor igual a 0.2689, que probaría esto mismo al estar más cerca de 0 que de 1:

```
                              Valor
---------------------------------
Coef. Contingencia         0,2689
```

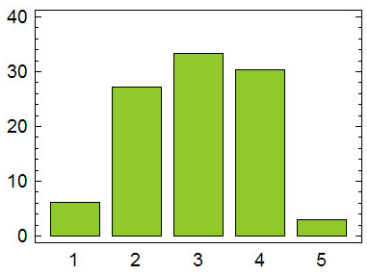
Una vez que hemos probado la independencia, podemos unificar los valores obtenidos para los 4 conceptos en una sola variable de valoración global en la facilidad de identificación de éstos, que junto con las otras 3, nos da un total de 4 variables:

1. *Facilidad_uso*: facilidad de uso
2. *Recuperación*: cantidad global de información recuperada
3. *Representación*: magnitud y precisión del contenido representado

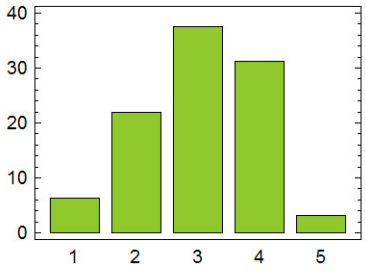
[1] http://www.ub.edu/aplica_infor/spss/cap52.htm

4. *Identificación*: facilidad en la identificación de conceptos

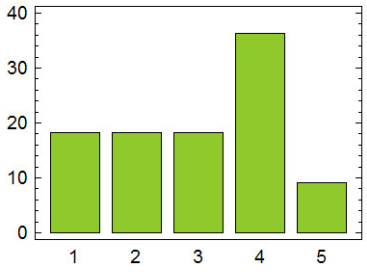
   En la Figura 6.20 podemos ver representados los diagramas de barras de estas variables, donde las barras representan porcentajes.



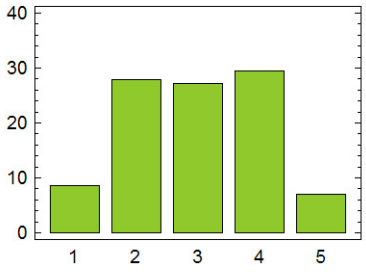
Facilidad de Uso



Información Recuperada



Representación del Contenido



Identificación de Conceptos

Figura 6.20: Diagramas de barras de los distintos aspectos

Y a continuación presentamos un resumen estadístico en el que se muestra la media, mediana y moda de las 4 variables.

Resumen Estadístico

	Frecuencia	Media	Mediana	Moda
Facilidad_uso	33	2,9697	3,0	3,0
Identificación	129	2,9845	3,0	4,0
Recuperación	32	3,03125	3,0	3,0
Representación	33	3,0	3,0	4,0
Total	227	2,99119	3,0	4,0

Vemos que la mediana es 3 para todos los casos, lo que indicaría que el 50 % de los participantes opinaría que $TC1$ es mejor que $TC2$ en los aspectos evaluados y el otro 50 % opinaría que es mejor $TC2$. Todas las medias están próximas a 3, por lo podría pensarse que, en media, los participantes opinan que $TC1$ y $TC2$ funcionan igual en todos los aspectos. Más adelante probaremos mediante un contraste de hipótesis que, efectivamente las medias pueden considerarse igual a 3.

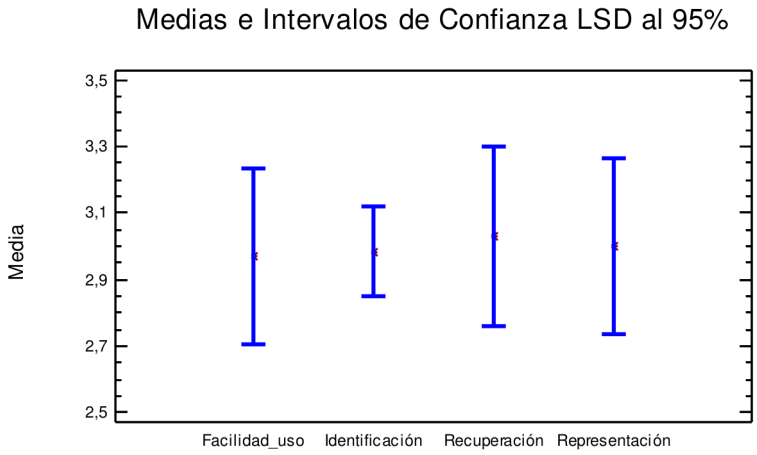


Figura 6.21: Gráfico de medias con intervalos de confianza al 95 % para los distintos aspectos

En la Figura 6.21 vemos el gráfico de los intervalos al 95 % de confianza de las valoraciones medias, resultado que también puede verse de forma numérica en la tabla de medias:

Tabla de Medias
con 95,0 intervalos LSD

	Frec.	Media	Error Estándar (s agrupada)	Límite inf.	Límite sup.
Facilidad_uso	33	2,9697	0,190802	2,70382	3,23557
Identificación	129	2,9845	0,096504	2,85002	3,11897
Recuperación	32	3,03125	0,19376	2,76125	3,30125
Representación	33	3,0	0,190802	2,73412	3,26588
Total	227	2,99119			

Vemos como todos los intervalos de confianza incluyen el valor 3. El correspondiente a las valoraciones en la identificación de conceptos es más pequeño que los demás, debido a esta variable posee mayor tamaño de muestra, ya que es el resultado de unificar en una sola las valoraciones de la facilidad de identificación de los 4 conceptos proporcionados.

A continuación realizamos contrastes de hipótesis para comprobar que, efectivamente, podemos considerar que la media de las valoraciones en los cuatro aspectos evaluados puede considerarse igual a 3.

En los cuatro casos se plantea un test bilateral para la media una muestra, basado en el estadístico *t de Student* [1]. En la hipótesis nula tenemos que comportamiento de $TC1$ y $TC2$ se considere el mismo en el aspecto evaluado, es decir, que la media de la valoraciones correspondientes a dicho aspecto sea igual a 3. Como hipótesis alternativa tenemos que este comportamiento se considere diferente, o lo que es lo mismo, que la media de las valoraciones sea distinta de 3.

En el caso de que la hipótesis nula se rechace en favor de la alternativa, plantearemos un nuevo contraste unilateral, para ver si la valoración media es inferior o superior a 3, lo que indicaría la preferencia de los participantes hacia una u otra *tag cloud*.

1. **Comparación de la facilidad de uso**

   Por lo tanto, el primer planteamiento de hipótesis que realizamos con respec-

[1] http://psico.fcep.urv.es/spss/inferencia/2medias.html

to a la comparación en la facilidad de uso es el siguiente:

$$\begin{cases} H_0: \ TC1 \text{ es similar a } TC2 \ \ (\text{La media de } \textit{Facilidad\_uso} = 3) \\ H_1: \ TC1 \text{ no es similar a } TC2 \ \ (\text{La media de } \textit{Facilidad\_uso} \neq 3) \end{cases}$$

Los resultados obtenidos para dicho contraste son:

```
contraste t
-----------
Hipótesis nula: media = 3,0
Alternativa: no igual

Estadístico t = -0,17695
P-valor = 0,860663

No se rechaza la hipótesis nula para alpha = 0,05.
```

El estadístico de contraste t tiene un valor igual a -0.17695 y el p-valor es de 0.860663, lo que indica que aceptamos la hipótesis nula para un nivel de significación del 5 %, es decir, no existen evidencias para considerar que $TC1$ y $TC2$ se comporten de forma diferente con respecto a su facilidad de uso, según la opinión de los encuestados.

Como hemos aceptado la hipótesis nula, no es necesario plantear un contraste unilateral.

2. **Comparación de la identificación de conceptos**
   El contraste de hipótesis planteado es el siguiente:

$$\begin{cases} H_0: \ TC1 \text{ es similar a } TC2 \ \ (\text{La media de } \textit{Identificación} = 3) \\ H_1: \ TC1 \text{ no es similar a } TC2 \ \ (\text{La media de } \textit{Identificación} \neq 3) \end{cases}$$

Los resultados obtenidos para dicho contraste son:

```
contraste t
-----------
Hipótesis nula: media = 3,0
```

```
Alternativa: no igual

Estadístico t = -0,160555
P-valor = 0,872697

No se rechaza la hipótesis nula para alpha = 0,05.
```

El estadístico de contraste t tiene un valor de -0.160555 y el p-valor es 0.872697, lo que indica que aceptamos la hipótesis nula para un nivel de significación del 5 %, es decir, no existen evidencias significativas para considerar que $TC1$ y $TC2$ se comporten de forma diferente con respecto a la facilidad en la identificación de conceptos.

3. **Comparación de la cantidad de información recuperada**
   Planteamos el siguiente contraste:

$$\begin{cases} H_0: \ TC1 \text{ es similar a } TC2 \ \ (\text{La media de } \textit{Recuperación} = 3) \\ H_1: \ TC1 \text{ no es similar a } TC2 \ \ (\text{La media de } \textit{Recuperación} \neq 3) \end{cases}$$

   Los resultados obtenidos para dicho contraste son:

```
contraste t
-----------
Hipótesis nula: media = 3,0
Alternativa: no igual

Estadístico t = 0,182869
P-valor = 0,856091

No se rechaza la hipótesis nula para alpha = 0,05.
```

El estadístico de contraste t tiene un valor igual a 0.182869 y el p-valor es de 0.856091, lo que indica que aceptamos la hipótesis nula para un nivel de significación del 5 %, es decir, no existen evidencias para considerar que

$TC1$ y $TC2$ se comporten de forma diferente con respecto a la cantidad de información recuperada.

4. **Comparación de la representación del contenido**
   Por último, planteamos un contraste de hipótesis para comprobar si los participantes consideran que ambas *tag clouds* se comportan igual con respecto a la precisión del contenido representado:

   $$\begin{cases} H_0: \ TC1 \text{ es similar a } TC2 \ \ (\text{La media de } \textit{Representación} = 3) \\ H_1: \ TC1 \text{ no es similar a } TC2 \ \ (\text{La media de } \textit{Representación} \neq 3) \end{cases}$$

   ```
   contraste t
   -----------
   Hipótesis nula: media = 3,0
   Alternativa: no igual

   Estadístico t = 0,0
   P-valor = 1,0

   No se rechaza la hipótesis nula para alpha = 0,05.
   ```

   El estadístico experimental para este contraste tiene un valor igual a 0, con un p-valor correspondiente igual a 1, por lo que de nuevo aceptamos la hipótesis nula para un nivel de significación del 5 % y concluimos que no existen evidencias significativas para considerar que $TC1$ y $TC2$ se comporten de forma diferente con respecto a la representación del contenido de la información.

En la Tabla 6.6 podemos ver un resumen de los resultados.

6.2 Evaluación Experimental sobre una Base de Historias Clínicas

En el ámbito médico, cada vez es más habitual contar con herramientas informáticas que ayuden en la automatización de las tareas diarias y en la gestión de

Resumen de los resultados
Comprobación de que las valoraciones dadas en la identificación de conceptos son independientes del concepto relacionado Test de independencia Chi-cuadrado $\begin{cases} H_0 : \text{Existe independencia} \\ H_1 : \text{No existe independencia} \end{cases}$ Para un 5 % de significación: $\chi^2_{exp} = 10.06$, p-valor=0.6109, $\Rightarrow$ Aceptamos independencia (H_0)
Comprobación de que la media de las valoraciones de los distintos aspectos evaluados puede considerarse 3, esto es, $TC1$ y $TC2$ se comportan igual Contraste de hipótesis para la media de una población $\begin{cases} H_0 : \ TC1 \text{ similar a } TC2 \ \text{ (Media} = 3) \\ H_1 : \ TC1 \text{ no similar a } TC2 \ \text{ (Media} \neq 3 \end{cases}$ Para un 5 % de significación: • Facilidad de uso: $t_{exp} = -0.17695$, p-valor=0.860663 $\Rightarrow$ Aceptamos H_0 ($TC1$ similar a $TC2$) • Identificación de conceptos: $t_{exp} = -0.160555$, p-valor=0.872697 $\Rightarrow$ Aceptamos H_0 ($TC1$ similar a $TC2$) • Información recuperada: $t_{exp} = -0.182869$, p-valor=0.856091 $\Rightarrow$ Aceptamos H_0 ($TC1$ similar a $TC2$) • Representación del contenido: $t_{exp} = 0$, p-valor=1 $\Rightarrow$ Aceptamos H_0 ($TC1$ similar a $TC2$)
Conclusión: La *tag cloud* que hemos generado de forma automática se considera similar que la creada por expertos en todos los aspectos evaluados, con la ventaja de que no ha sido precisa la intervención humana para su generación.

Tabla 6.6: Resumen de los resultados obtenidos en el experimento de la Sección 6.1

las historias clínicas. Si bien estas herramientas son imprescindibles para facilitar la actividad del personal sanitario y modernizar sus sistemas de gestión, contribuyen a la acumulación de grandes volúmenes de información y al incremento de la dificultad de procesarlos adecuadamente para que puedan llegar a transformarse en información útil para los usuarios.

En gestión de urgencias, en intervenciones quirúrgicas, en recursos humanos y en todos los demás ámbitos hospitalarios, cada día se recoge gran cantidad de información de la que es muy conveniente, incluso necesario, extraer conocimiento que ayude en la gestión general del hospital, en la atención a los pacientes y en la toma de decisiones.

El personal hospitalario no solicita simples resultados cuando realiza una consulta sobre sus datos almacenados, sino conocimientos basados en agrupaciones lógicas que permitan analizar la información bajo un determinado punto de vista [MF08]. Sin embargo, los sistemas que gestionan la información en los hospitales son, en este aspecto, claramente deficitarios.

En el caso de la información textual el problema se agrava, ya que esta información está sujeta a un gran número de variaciones sintácticas, uso frecuente de acronismos e incluso errores o recortes en la escritura, ocasionados por las prisas y la escasez de tiempo. Y además, suele ser introducida por diferentes personas que utilizan diversos patrones de escritura.

Es preciso contar con sistemas inteligentes que procesen los datos de forma semántica, además de sintáctica, que añadan estructura, que faciliten la formulación de consultas semánticas sobre los atributos textuales y que ofrezcan una interfaz sencilla para la formulación de estas consultas, que no requieran conocimientos previos para su uso y ahorren tiempo en la búsqueda de la información necesitada.

El uso de una *tag cloud* semántica, como la propuesta en esta memoria, sería ideal en este contexto. Es por ello, que también nos hemos planteado realizar una evaluación de nuestra metodología sobre este tipo de datos, los cuales difieren de los usados en el experimento de la Sección 6.1 en que están más desestructurados, son más extensos y requieren una mayor limpieza.

6.2.1 Descripción del Conjunto de Datos

Para la realización de este experimento contamos con datos anónimos de los registros de Historias Clínicas Electrónicas del Hospital Clínico "San Cecilio" de Granada, España. Estos datos han sido proporcionados por el mismo hospital y se han almacenado en una base de datos relacional que cuenta con varias tablas. Escogemos como punto de partida la tabla que recoge la información referente a las intervenciones quirúrgicas y que comprende 24481 registros. Los atributos principales de esta tabla son "Diagnóstico" e "Intervención Propuesta", cuyo contenido es texto corto compuesto por una o más frases.

6.2.2 Descripción de la Metodología

Como en el experimento anterior (Sección 6.1) se probó que la mejor configuración para la generación de la *tag cloud* se alcanzaba con la estructura WAPO, tras la aplicación de los preprocesamientos sintáctico y semántico y representando únicamente los *itemsets* maximales, usaremos en este experimento esa misma configuración.

Preprocesamiento

La mayor parte del preprocesamiento se llevó a cabo como parte de la tesis del Dr. Sandro Martínez Folgoso [MF08], que utilizó estos mismos datos para un experimento donde se contrastaba la estructura-AP. Nosotros los hemos tomado ya limpios para la generación de la estructura WAPO y su visualización.

Resumiremos brevemente en qué consistió este preprocesamiento, para más detalles se puede consultar la tesis arriba citada.

Igual que para el experimento de la Sección 6.1, se comenzó empleando un preprocesamiento sintáctico consistente en la aplicación de los filtros de tokenización, eliminación de *Stop Words* y lematización simple para eliminar formas de género y plural a los datos brutos.

Posteriormente, se aplicó un preprocesamiento semántico, para el cual se crearon ficheros auxiliares con la ayuda de expertos, conteniendo los sinónimos y acrónimos propios del lenguaje específico del entorno hospitalario.

Por ejemplo, el término "IZQUIERDO", aparecía en el texto original con todas las siguientes formas (consideradas sinónimos en el fichero auxiliar): IZQ, IZDA, IZDO, IZQD, IZQDO, IZQDA, IZ, IZQIERDO, IZQU, IZQUI, IZQUERDO, HIZQ, IZQ1UIERDA, IZQUIE, IZQUIEDO, IZQUIER, IZQUIERA, IZQUIERD, IZQUIERDOP, IZQUIERTDO, IZQUIRDO, IZQUUIERDO, IZQUIERDA. Y todo eso sin considerar las distintas variantes al mezclar mayúsculas y minúsculas.

El fichero de acrónimos utilizado, cumplía la función principal de obtener los acrónimos sin los puntos intermedios que separan las letras que los componen. Esta labor es necesaria dado que el punto se considera como separador de dos frases en una misma tupla.

En la Tabla 6.7 podemos ver una muestra de los acrónimos contenidos en dicho fichero.

Acrónimo	Término
EECC	Extracción extracapsular del cristalino
LIO	Lente intraocular
OI	Ojo izquierdo
EBA	Exploración bajo anestesia
CP	Cámara posterior
PA	Peritonitis aguda

Tabla 6.7: Ejemplos de acrónimos encontrados en el texto original

El preprocesamiento semántico consistió, básicamente, en eliminar los puntos de separación de las letras en los acrónimos y en sustituir los sinónimos encontrados en el texto original por un representante canónico (en el ejemplo visto para "IZQUIERDO", todas esas distintas formas de escribir el término, se sustituyeron por una sola: "IZQUIERDO").

En la Tabla 6.8 tenemos un ejemplo de cómo queda el texto limpio tras aplicar los preprocesamientos sintáctico y semántico descritos.

Generación de la Forma Intermedia de Representación y Visualización

El atributo textual a procesar escogido es "Intervención Propuesta" de la tabla de intervenciones quirúrgicas.

Texto corto	Texto corto modificado
AMPUTACION 4 DEDO PIE IZDO	AMPUTACION 4 DEDO PIE IZQUIERDO
ENDORREDUCCION DE F.A.V.	ENDORREDUCCION DE FAV
legrado	LEGRADO
Mastectomia mas D.A.	MASTECTOMIA MAS DA

Tabla 6.8: Ejemplo de limpieza del texto original

Hemos extraído la estructura WAPO con la herramienta descrita en la Sección 5.5.3 para distintos valores de soporte y para la visualización se ha usado el programa descrito en la Sección 5.5.4.

Para decidir qué valor de soporte utilizamos, hemos realizado otra vez un análisis de ensayo y error, con la ayuda del mismo grupo de expertos que evaluaron el soporte de las *tag clouds* del experimento anterior.

Usando valores pequeños para el soporte (Figuras 6.22 y 6.23), el número de términos en la visualización se ha considerado demasiado grande, lo que impide distinguirlos claramente. Para valores mayores (Figuras 6.26 y 6.27), se pierden términos importantes en la visualización, por lo que los soportes más adecuados son los que se han empleado en las Figuras 6.24 y 6.25, que conservan un número aceptable de términos, pero no tantos como para dificultar su identificación.

Finalmente, seleccionamos la estructura correspondiente a un 0.3 % de soporte (Figura 6.24), que contiene más información que la correspondiente al 0.4 % (Figura 6.25), pero mantiene un número aceptable de términos en la visualización. Como vemos, se observan palabras como “Izquierdo” o “Derecho”, que adquieren una relevancia desmesurada y sin embargo, no aportan información. Para eliminar estas palabras aplicamos el postprocesamiento.

Postprocesamiento

Las reglas que se vieron en la Sección 5.4 para determinar cuando un *itemset* debería aparecer en la *tag cloud* atendiendo a su categoría gramatical, estaban pensadas para el idioma inglés. Para el español son básicamente las mismas, se empieza eliminando los verbos y adverbios, así como los adjetivos que aparecen por sí solos, sin acompañar a ningún nombre.

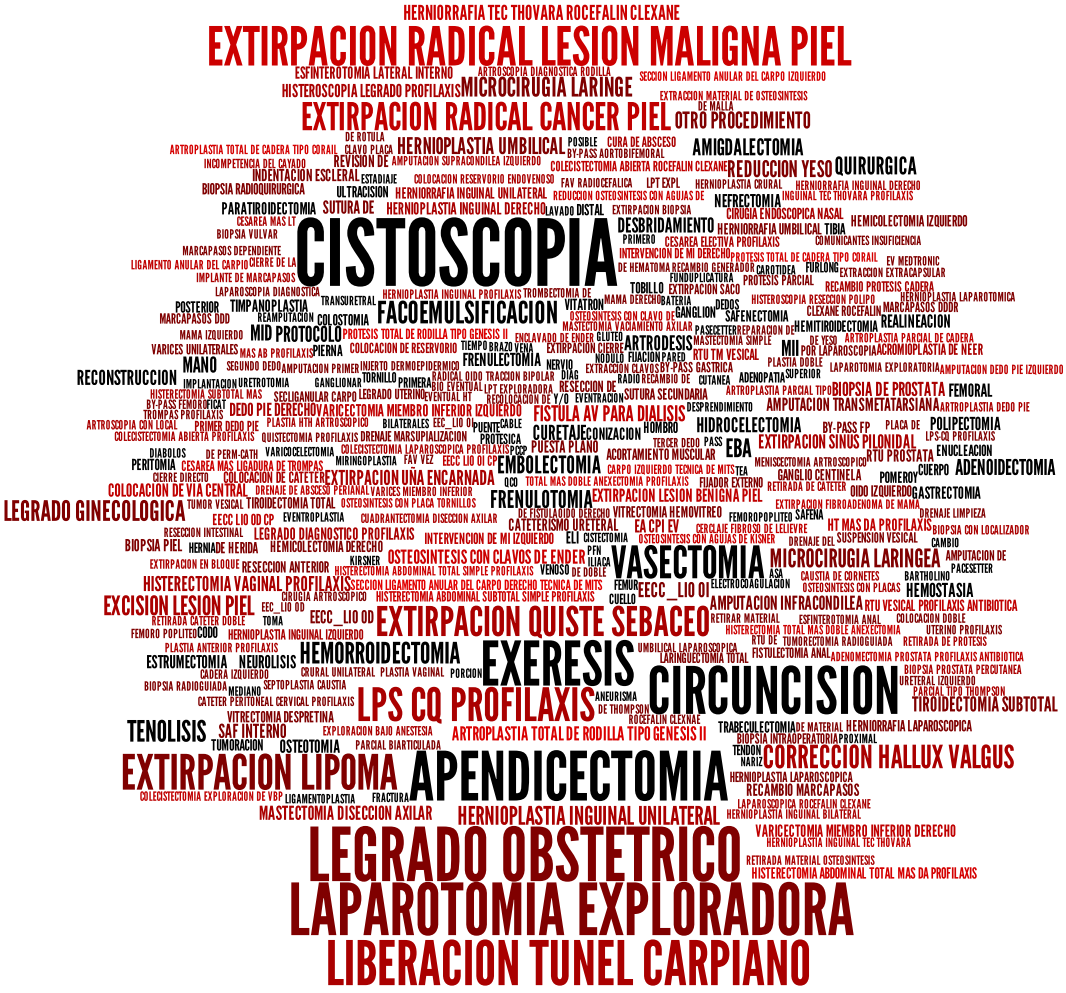


Figura 6.22: *Tag Cloud* generada para un 0.05 % de soporte

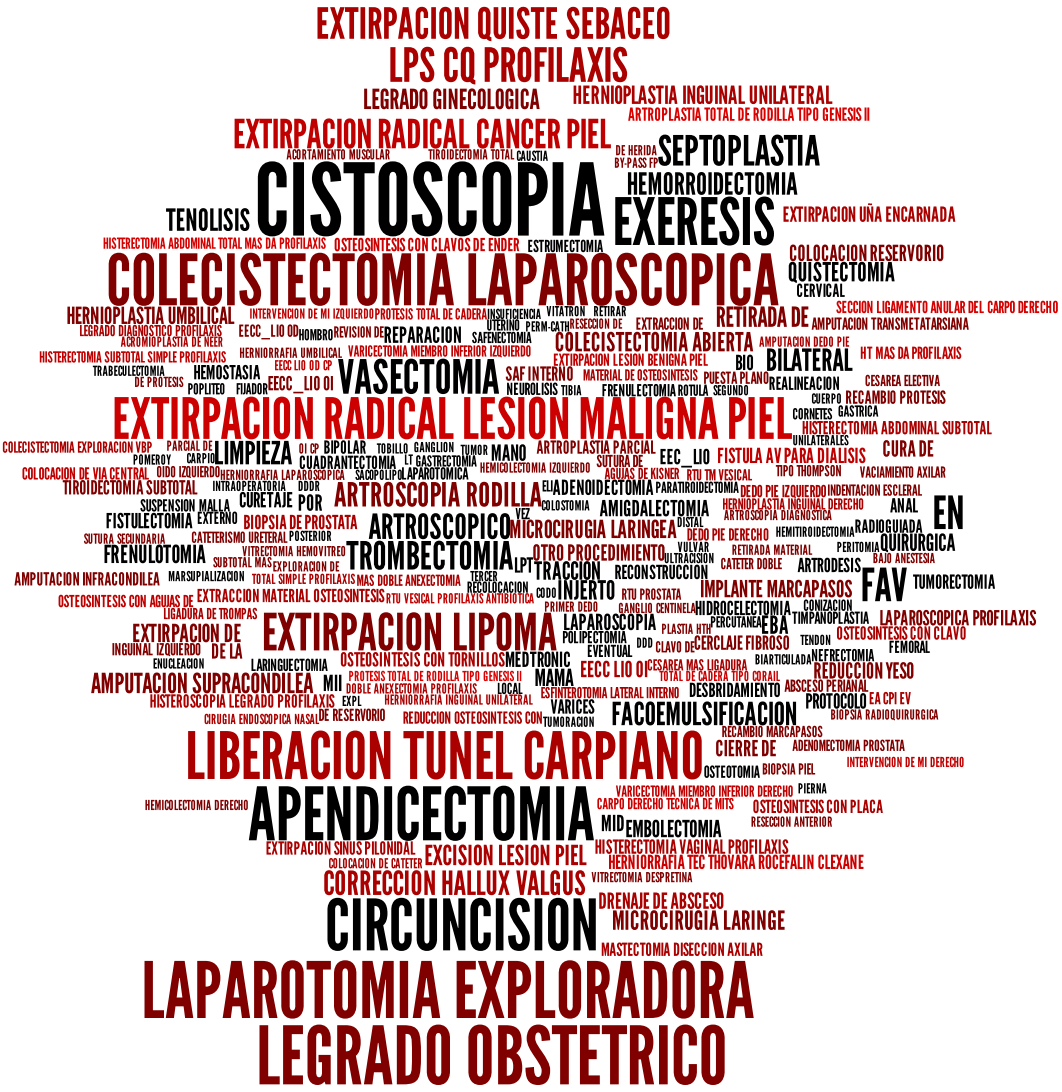


Figura 6.23: *Tag Cloud* generada para un 0.1 % de soporte

Figura 6.24: *Tag Cloud* generada para un 0.3 % de soporte

Figura 6.25: *Tag Cloud* generada para un 0.4 % de soporte

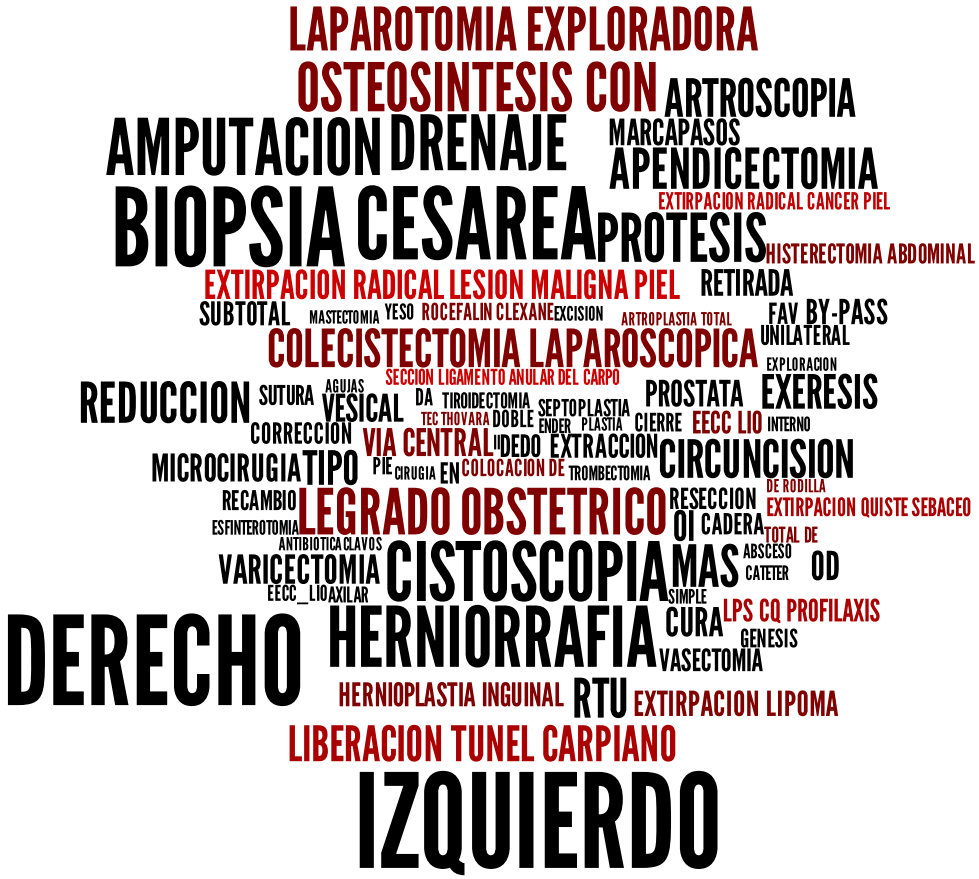


Figura 6.26: *Tag Cloud* generada para un 0.5 % de soporte

Figura 6.27: *Tag Cloud* generada para un 1 % de soporte

La única diferencia es que en español, el adjetivo puede ir tanto delante como detrás del nombre. El conjunto de reglas queda entonces de la siguiente forma:

- ***Itemsets* de nivel uno**: Se consideran buenos candidatos si son nombres [N].
- ***Itemsets* de nivel dos**: Se consideran buenos candidatos si se componen de dos nombres [NN] o de un adjetivo y un nombre: [AN] y [NA].
- ***Itemsets* de ninvel n**: Se consideran buenos candidatos si están compuestos por una combinación válida de términos en los niveles previos, más un nombre o un adjetivo. Para el nivel tres tendríamos: [NNN], [ANN], [NNA], [NAN], [ANN], [AAN] y [ANA].

Este postprocesamiento no se ha automatizado por no contar con ningún etiquetador en español para poder identificar la categoría gramatical de las palabras. Se ha realizado de forma manual, a la espera de conseguir hacerlo de manera automática.

Terminado el postprocesamiento, hemos dado color a las etiquetas de forma que sea más fácil distinguirlas, hemos optado por el color negro para los monotérminos y el rojo para los multitérminos en distintas tonalidades dependiendo de la cardinalidad de los mismos. De esta forma queda una *tag cloud* estética en la que los colores cumplen una funcionalidad (ver Figura 6.28).

6.2.3 Resultados

Evaluación a través de Métricas

Para este experimento se han calculado las mismas métricas que en el de la Sección 6.1.

Para el cálculo de la *precisión* y la *exhaustividad*, previamente es necesario anotar los registros del atributo textual con las etiquetas más apropiadas de la *tag cloud* para así conocer cuáles deberían recuperarse con cada una de estas etiquetas.

Como las restricciones de tiempo no nos permiten anotar los 24481 registros, hemos seleccionado una muestra de 500, tamaño que corresponde con un error relativo del 4.35 % para una confianza del 95 %.

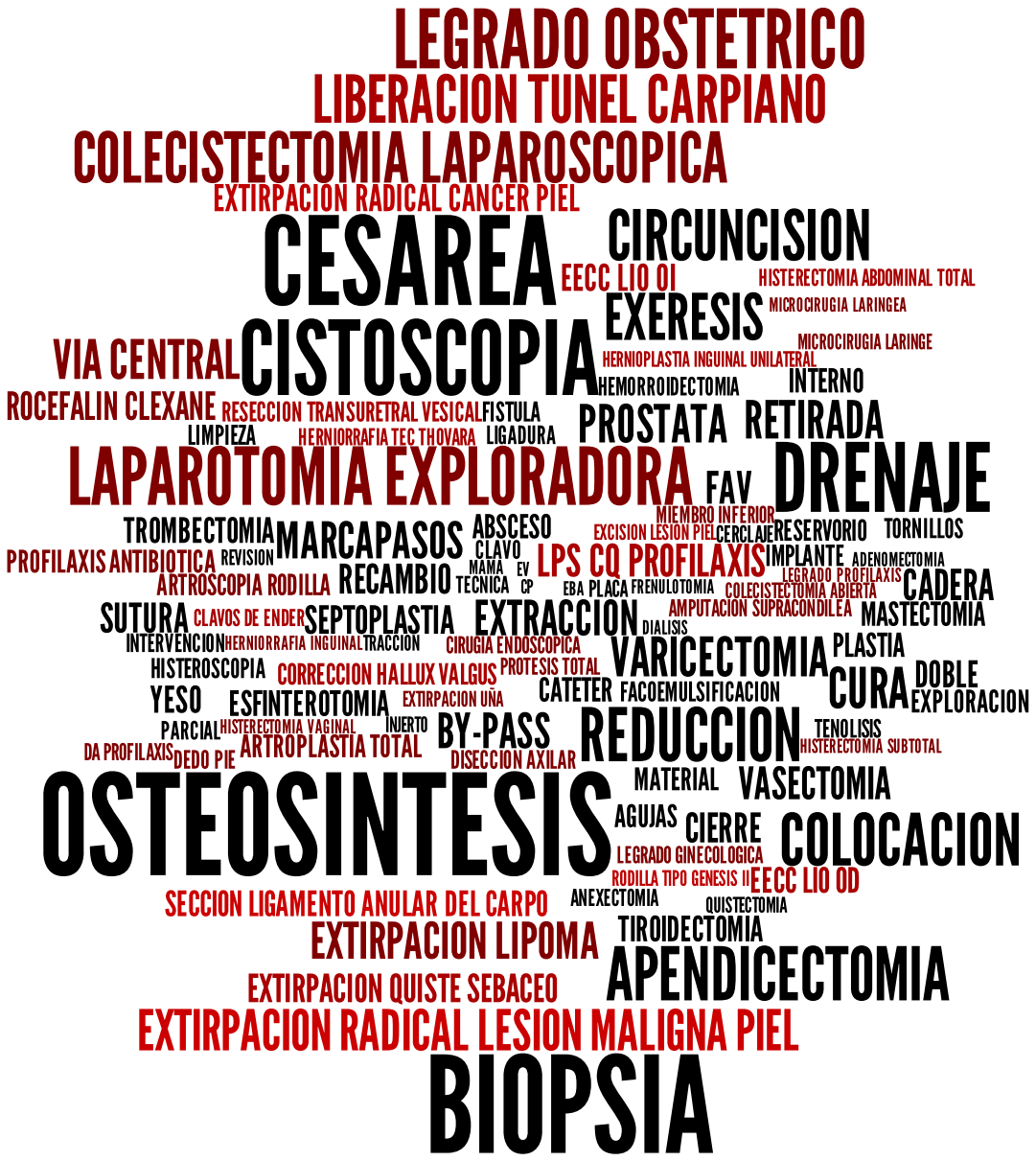


Figura 6.28: *Tag cloud* con soporte 0.3 % tras postprocesamiento y color

Las anotaciones se han llevado a cabo por un grupo de expertos en el campo de la salud, a los cuales se les han proporcionado los datos de la muestra y una *tag cloud* generada a partir de éstos que contiene las etiquetas con las que anotar los registros.

En la Tabla 6.9 se muestran las medias de *precisión*, *exhaustividad* y F_1 *Score*

calculadas. Este cálculo se ha realizado considerando dos tipos distintos de consulta:

- **Tipo I**.- Recupera todas las entradas en que los términos aparecen en el orden estricto de adyacencia que tienen en las etiquetas.
- **Tipo II**.- Recupera todas las entradas en que los términos aparecen en el mismo orden que tienen en las etiquetas, sin tener por qué respectar la adyacencia estricta

Tipos de Consulta	*Precisión*	*Exhaustividad*	F_1 *Score*
Tipo I	0.940	0.780	0.823
Tipo II	0.990	0.870	0.904

Tabla 6.9: *Precisión media, exhaustividad* y F_1 *Score* para la *tag cloud* de la Figura 6.28 para distintos tipos de consulta

Vemos que los valores de *precisión*, *exhaustividad* y F_1 *Score* son muy buenos en los dos casos, siendo un poco más altos para la consulta de tipo II.

Se obtiene mejor valor para la *precisión* que para la *exhaustividad*, lo que quiere decir que casi todas las entradas que se recuperan son relevantes, aunque queda una pequeña proporción de entradas relevantes que no se recuperan.

A continuación calculamos las métricas de *cobertura*, *solapamiento* y *balance* con las fórmulas proporcionadas por Venetis et al. [Ven11] y el mismo planteamiento realizado en el experimento de la Sección 6.1.

Además de la *cobertura* total de la *tag cloud*, se ha calculado la *cobertura* media por etiqueta o término, entendida ésta como la fracción de texto original representada, en media, por cada etiqueta en la *tag cloud*.

En la Tabla 6.10 podemos ver los valores obtenidos.

Cobertura	*Cobertura por Etiqueta*	*Solapamiento*	*Balance*
0.58793	0.00706	0.00003	0.05022

Tabla 6.10: *Cobertura*, *solapamiento* y *balance* de la *tag cloud* de la Figura 6.28

En este caso la *cobertura* no es tan buena como en el experimento de la Sección 6.1 debido a que la base de datos es mucho más extensa y heterogénea, por lo que

muchos de los registros no alcanzan representación en la *tag cloud* a causa de sus bajas frecuencias en comparación con el resto. Para ser un conjunto de datos con estas características, una cobertura cercana al 60 % es un valor bastante bueno.

El *solapamiento* es prácticamente 0, por lo que las distintas etiquetas representan diferente información.

Por último, el valor obtenido en la métrica de *balance* indica que la *tag cloud* es desequilibrada, lo que como ya vimos, es un característica intrínseca de este tipo de representación.

Encuesta de Satisfacción y Análisis Estadístico

Para evaluar el nivel de satisfacción de los usuarios con la *tag cloud*, hemos llevado a cabo la realización de una encuesta.

Participantes

El tamaño mínimo de muestra requerido para obtener un error absoluto inferior a 0.5 unidades, considerando una desviación típica igual a 1 y una confianza del 95 %, es de 18 participantes, como se vio en el experimento de la Sección 6.1.

En este caso hemos conseguido un total de 23 participantes anónimos con formación y experiencia en distintas áreas de la medicina.

Procedimiento

Se ha contactado a través del correo electrónico con diversos especialistas solicitándoles colaboración, a los cuales se les ha proporcionado un enlace a una página web en la que hemos habilitado la herramienta para su experimentación y posterior evaluación. En esta misma página hemos incrustado un formulario con unas breves instrucciones y las preguntas a responder.

En el Apéndice B.2 se muestra el contenido completo de la página.

Las cuestiones planteadas son similares a las de la encuesta del experimento anterior, con la diferencia de que ahora el propósito es evaluar la *tag cloud* sin compararla con ninguna otra. Éstas se formulan como afirmaciones con las que el encuestado debe expresar su grado de acuerdo.

Las cuatro primeras afirmaciones son las siguientes:

1. La *tag cloud* presentada me parece intuitiva y de fácil uso
2. La *tag cloud* presentada aporta información sobre el contenido de la base de datos
3. La información recuperada con las etiquetas de la *tag cloud* es coherente con dichas etiquetas.
4. Una *tag cloud* como la presentada me ayudaría a realizar búsquedas en una base de datos médica

Esta vez no hemos podido establecer ninguna cuestión sobre la cantidad global de información recuperada ya que el encuestado no conoce el contenido de la base de datos. En lugar de eso se ha preguntado sobre la coherencia de esta información.

Las cuatro últimas están relacionadas con la facilidad para identificar conceptos. Se formulan de la siguiente forma:

5-8. Me resulta fácil identificar un concepto en la *tag cloud* relacionado con... (*definiciones en Tabla 6.11*).

	Definiciones proporcionadas en las afirmaciones 5-8	**Término**
1	Intervención quirúrgica para el nacimiento de un bebé	Cesárea
2	Aborto quirúrgico o tratamiento tras aborto	Legrado obstétrico
3	Operación quirúrgica que tiene por objeto la reconstrucción completa de una articulación obstruida o anquilosada	Artroplastia total
4	Técnica muy utilizada en la operación de cataratas	Facoemulsificación

Tabla 6.11: Definición de conceptos y objetos esperados a identificar en la *tag cloud* de la Figura 6.28

Se han elegido estos 4 conceptos por estar situados en distintas partes de la visualización y con diferentes tamaños.

El grado de acuerdo con estas afirmaciones se expresa a través de una calificación numérica del 1 al 5, donde el 1 indica estar “completamente en desacuerdo” y el 5 “completamente de acuerdo”. Atendiendo a esta clasificación, las correspondencias del grado de acuerdo y las calificaciones emitidas serían las siguientes:

- 1: Completamente en desacuerdo
- 2: En desacuerdo
- 3: Indiferente
- 4: De acuerdo
- 5: Completamente de acuerdo

Por último, se pide a los participantes que emitan algunas sugerencias sobre los aspectos a mejorar.

Análisis Estadístico y Resultados

Igual que en el experimento de la Sección 6.1, empezaremos probando si las valoraciones sobre la facilidad de identificación de conceptos son independientes del concepto proporcionado.

Comenzamos mostrando un resumen estadístico con la media, mediana y moda de las cuatro variables que recogen estas valoraciones.

Resumen Estadístico

	Frecuencia	Media	Mediana	Moda
concepto1	23	4,0	4,0	4,0
concepto2	23	4,08696	4,0	4,0
concepto3	23	3,82609	4,0	4,0
concepto4	23	3,95652	4,0	4,0
Total	92	3,96739	4,0	4,0

Vemos que la media de los cuatro conceptos está próxima a 4, lo que indicaría que los participantes, en media, están "de acuerdo" en que les resulta fácil identificar en la *tag cloud* cada uno de ellos en función de sus definiciones.

Al ser todas las medianas iguales a 4, esta media es muy representativa y también 4 ha sido el grado de acuerdo expresado por la mayoría de los participantes con la afirmación de que resulta fácil identificar los conceptos, como indica el valor de la moda.

Podemos ver este resultado de forma más precisa a través de intervalos de confianza para las medias (ver gráficamente en Figura 6.29):

```
 Tabla de Medias
con 95,0 intervalos LSD
------------------------------------------------------------------
                            Error Estándar
            Frec.     Media  (s agrupada)  Límite inf.  Límite sup.
------------------------------------------------------------------
concepto1     23       4,0     0,224585       3,68441      4,31559
concepto2     23   4,08696     0,224585       3,77136      4,40255
concepto3     23   3,82609     0,224585       3,51049      4,14168
concepto4     23   3,95652     0,224585       3,64093      4,27211
------------------------------------------------------------------
Total        132   3,96739
```

Vemos como todos los intervalos contienen el valor 4 y en cualquier caso, toman valores mayores a 3. El solapamiento de los intervalos indicaría que las cuatro medias se pueden considerar iguales a ese nivel de confianza.

En el gráfico de cajas y bigotes (Figura 6.30) podemos ver que el primer cuartil es igual o superior a 3 para las cuatro variables.

El test de independencia Chi-cuadrado indica que podemos aceptar la hipótesis de que las valoraciones en la facilidad en la identificación de conceptos son independientes del concepto proporcionado (estadístico de contraste igual a 9.16 y p-valor de 0.6888):

```
Contraste de Chi-cuadrado
-----------------------------------------
    Chi-cuadrado        GL         P-Valor
-----------------------------------------
            9,16        12          0,6888
-----------------------------------------
```

El valor del coeficiente de contingencia es 0.3010, que al estar más cerca de 0 que de 1, señalaría esta misma independencia:

```
                          Valor
---------------------------------
Coef. Contingencia        0,3010
```

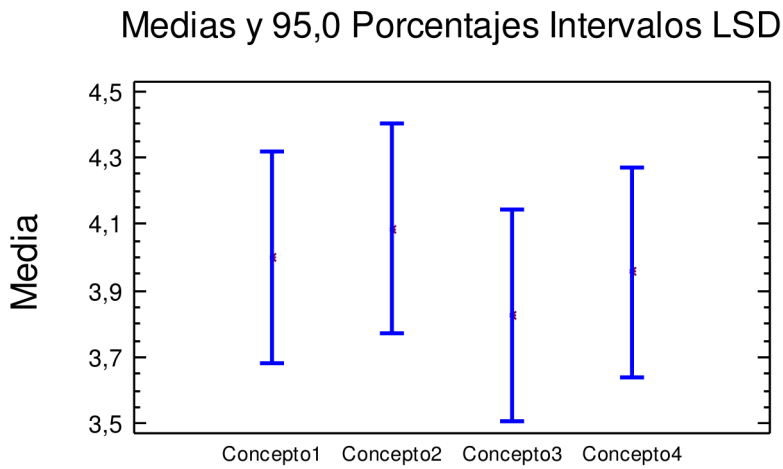


Figura 6.29: Gráfico de medias con intervalos de confianza al 95 % para la identificación de conceptos

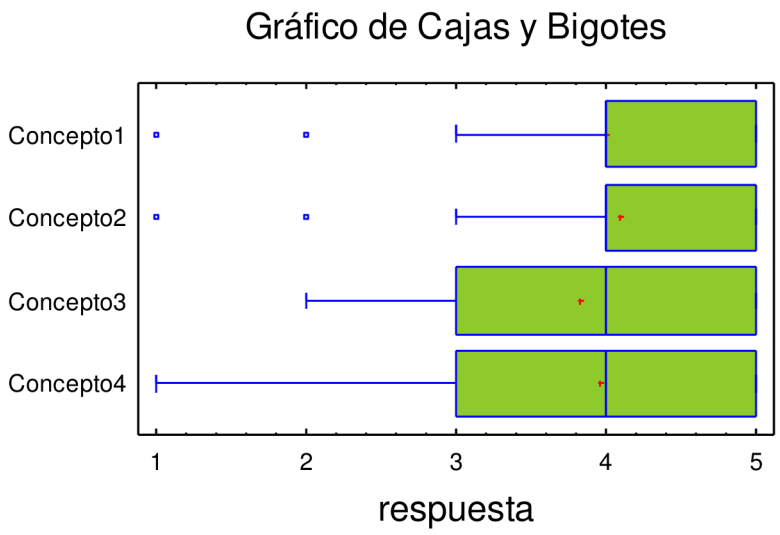


Figura 6.30: Gráfico de cajas y bigotes para la identificación de conceptos

A continuación unificamos los valores obtenidos en las cuatro variables sobre la identificación de conceptos en una sola, al haber probado la independencia del concepto.

Considerando las demás, tendríamos en total cinco variables que analizar:

1. *Facilidad_uso*: mide si la *tag cloud* es intuitiva y fácil de usar
2. *Identificación*: evalúa si es sencilla la identificación de conceptos de forma global
3. *Recuperación*: determina si la información recuperada es coherente con las etiquetas
4. *Representación*: constata si la *tag cloud* representa el contenido adecuadamente
5. *Utilidad_búsqueda*: comprueba si los usuarios emplearían la herramienta para realizar búsquedas sobre bases de datos médicas

En las Figuras 6.31, 6.32, 6.33, 6.34 y 6.35 podemos ver los diagramas de barras y los gráficos de sectores de cada de estas variables.

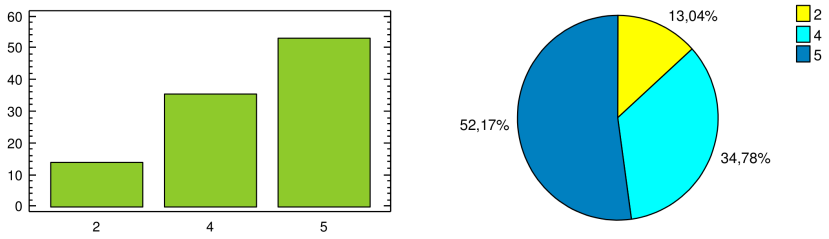


Figura 6.31: Diagrama de barras y gráfico de sectores para *Facilidad_uso*

A la vista de las figuras, es fácil darse cuenta que los porcentajes más altos corresponden a los valores 4 y 5 en las cuatro primeras variables. Para la variable *Utilidad_búsqueda* el porcentaje más alto es para el valor 3, pero éste no supera la suma de los porcentajes correspondientes a los valores 4 y 5. En todos los casos, podemos decir que más del 60 % de los encuestados emite valoraciones positivas (4

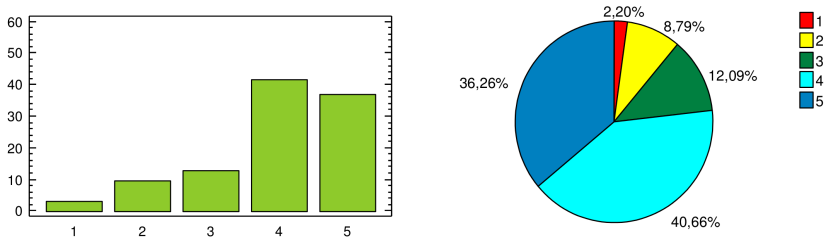


Figura 6.32: Diagrama de barras y gráfico de sectores para *Identificación*

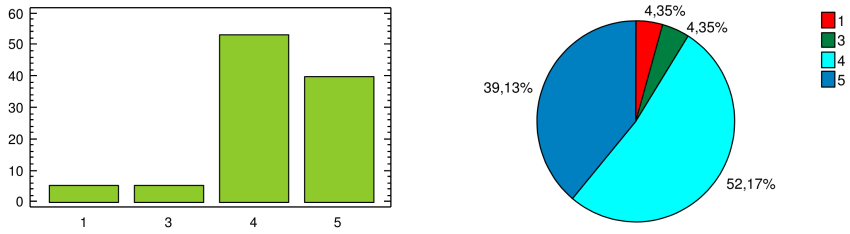


Figura 6.33: Diagrama de barras y gráfico de sectores para *Recuperación*

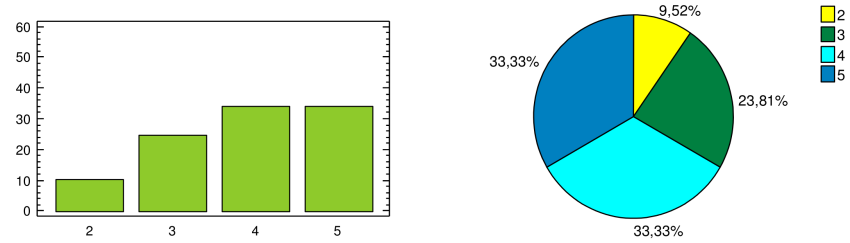


Figura 6.34: Diagrama de barras y gráfico de sectores para *Representación*

o 5) y este porcentaje es mucho mayor para las tres primeras variables, alcanzando casi un 90 % para *Facilidad_uso* y superándolo con *Recuperación*.

En el gráfico de cajas y bigotes de la Figura 6.36 vemos que el valor del primer cuartil es 4 para las tres primeras variables y 3 para las dos últimas. En todas ellas, el valor 1 se considera atípico, en los tres primeros casos también se considera atípico el 2 y para *Facilidad_uso* también el 3.

En el siguiente resumen estadístico se muestra la media, mediana y moda de las

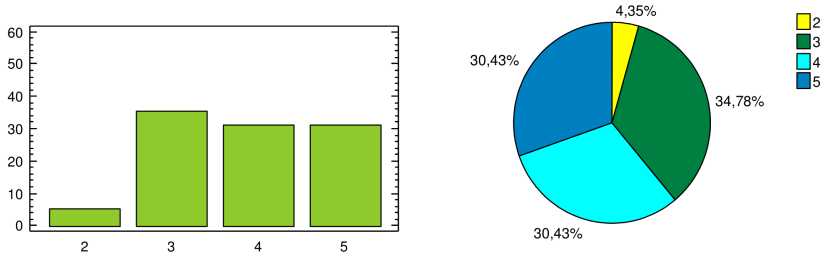


Figura 6.35: Diagrama de barras y gráfico de sectores para *Utilidad_búsqueda*

variables:

Resumen Estadístico

	Frecuencia	Media	Mediana	Moda
Facilidad_uso	23	4,26087	5,0	4,0
Identificación	92	3,96739	4,0	4,0
Recuperación	23	4,21739	4,0	4,0
Representación	21	3,90476	4,0	4,0
Utilidad_búsqueda	23	3,86957	4,0	3,0
Total	182	4,03315	4,0	4,0

Vemos que la media está cercana a 4 en todos los casos. El mejor valor obtenido es para la variable *Facilidad_uso* que tiene una media de 4.26 y una mediana igual a 5. Esto significa que de todas las afirmaciones realizadas, aquella con la que están más de acuerdo los participantes es con la que manifiesta que la *tag cloud* es intuitiva y fácil de usar. El grado de acuerdo parece ser muy bueno con todas ellas. Corroboremos este resultado mediante los intervalos de confianza para la media (ver gráficamente en Figura 6.37):

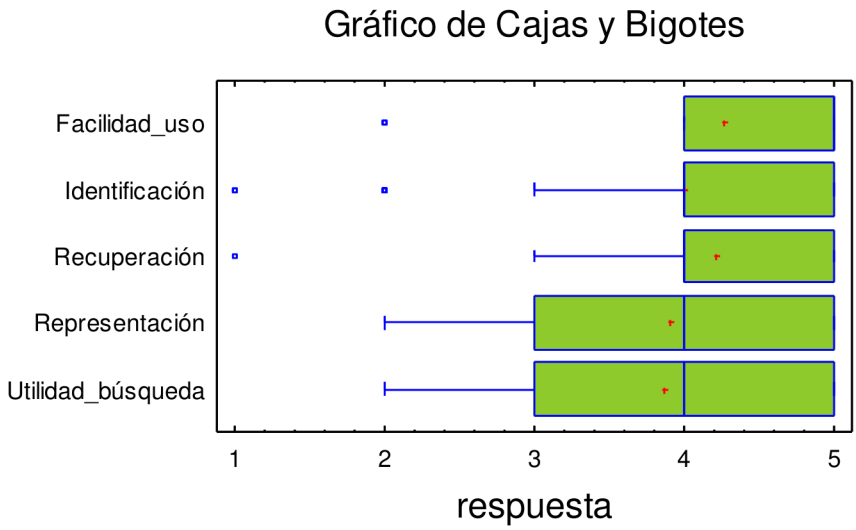


Figura 6.36: Gráficos de cajas y bigotes de las variables

Tabla de Medias
con 95,0 intervalos LSD

	Frec.	Media	Error Estándar (s agrupada)	Límite inf.	Límite sup.
Facilidad_uso	23	4,26087	0,20659	3,97257	4,54917
Identificación	92	3,96739	0,103861	3,85506	4,14494
Recuperación	23	4,21739	0,20659	3,92909	4,50569
Representación	21	3,90476	0,216204	3,60305	4,20647
Utilidad_búsqueda	23	3,86957	0,20659	3,58127	4,20647
Total	182	4,03315			

El intervalo de confianza que contiene valores más altos es el correspondiente a *Facilidad_uso*. Vemos que todos los intervalos toman valores mayores a 3, por lo que en ningún caso la media podría considerarse igual o inferior a 3 para un 95 % de confianza, lo que indicaría que, en media, los participantes estarían de acuerdo con todas las afirmaciones realizadas. Corroboraremos este resultado mediante contrastes de hipótesis.

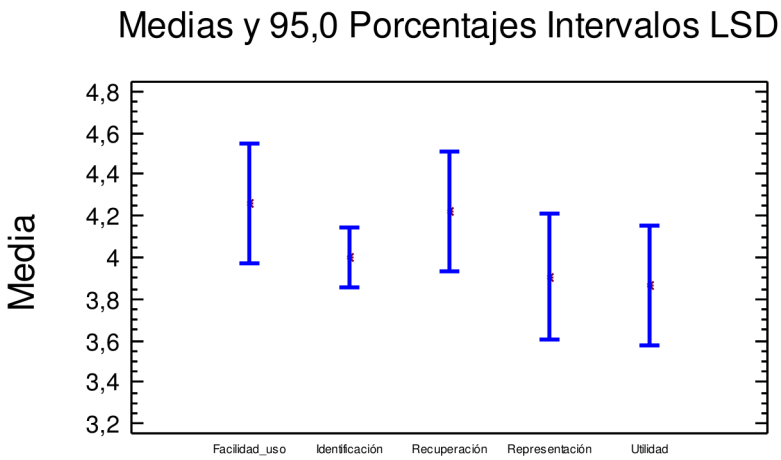


Figura 6.37: Gráfico de medias con intervalos de confianza al 95 % para los distintos aspectos

El contraste que se plantea es un test unilateral para la media de una muestra, que será el mismo para cada una de las cinco variables. En la hipótesis nula tendremos que la media sea igual a 3 y en la alternativa que sea superior. En el caso de rechazar la hipótesis nula en favor de la alternativa, habremos demostrado que, en media, los participantes están de acuerdo con las declaraciones formuladas sobre las capacidades de la *tag cloud*.

1. **La *tag cloud* presentada es intuitiva y fácil de usar**

$$\begin{cases} H_0 : \text{La media de } \textit{Facilidad\_uso} = 3 \\ H_1 : \text{La media de } \textit{Facilidad\_uso} > 3 \end{cases}$$

Los resultados obtenidos para dicho contraste son:

```
contraste t
-----------
Hipótesis nula: media = 3,0
Alternativa: mayor que

Estadístico t = 5,98804
P-valor = 0,00000250557
```

```
Se rechaza la hipótesis nula para alpha = 0,05.
```

El estadístico de contraste t tiene un valor igual a 5.98804 y el p-valor es de $2.5 \cdot 10^{-6}$, que como es menor de 0.05, rechazamos la hipótesis nula y concluimos que hay evidencias significativas para aceptar que la media de *Facilidad_uso* es superior a 3, con un 5 % de significación.

2. **Me resulta fácil identificar en la *tag cloud* los conceptos proporcionados**

$$\begin{cases} H_0 : \text{La media de } \textit{Identificación} = 3 \\ H_1 : \text{La media de } \textit{Identificación} > 3 \end{cases}$$

Los resultados obtenidos para dicho contraste son:

```
contraste t
-----------
Hipótesis nula: media = 3,0
Alternativa: mayor que

Estadístico t = 9,33422
P-valor = 0,0

Se rechaza la hipótesis nula para alpha = 0,05.
```

El estadístico de contraste t tiene un valor de 9.33422 y el p-valor es 0.0, por lo que de nuevo rechazamos la hipótesis nula en favor de la alternativa. Con un 5 % de significación, tenemos evidencias para concluir que la media de *Identificación* es superior a 3.

3. **La información recuperada con las etiquetas de la *tag cloud* es coherente con dichas etiquetas**

$$\begin{cases} H_0 : \text{La media de } \textit{Recuperación} = 3 \\ H_1 : \text{La media de } \textit{Recuperación} > 3 \end{cases}$$

Los resultados obtenidos para dicho contraste son:

```
contraste t
-----------
Hipótesis nula: media = 3,0
Alternativa: mayor que

Estadístico t = 6,47025
P-valor = 8,24563E-7

Se rechaza la hipótesis nula para alpha = 0,05.
```

El estadístico de contraste t tiene un valor igual a 6.47025 y el p-valor es de $8.25 \cdot 10^{-7}$, por lo que se rechaza la hipótesis nula para un 5 % de significación.

4. **La *tag cloud* presentada aporta información sobre el contenido de la base de datos**

$$\begin{cases} H_0 : \text{La media de } \textit{Representación} = 3 \\ H_1 : \text{La media de } \textit{Representación} > 3 \end{cases}$$

```
contraste t
-----------
Hipótesis nula: media = 3,0
Alternativa: mayor que

Estadístico t = 4,16603
P-valor = 0,000238578

Se rechaza la hipótesis nula para alpha = 0,05.
```

El estadístico experimental para este contraste tiene un valor igual a 4.16603, con un p-valor correspondiente igual a $2.3 \cdot 10^{-4}$. Rechazamos la hipótesis nula concluyendo que existen evidencias para considerar la media de *Representación* mayor que 3, con un 5 % de significación.

5. **Una *tag cloud* como la presentada me ayudaría a realizar búsquedas en una base de datos médica**

$$\begin{cases} H_0 : \text{La media de } \textit{Utilidad\_búsqueda} = 3 \\ H_1 : \text{La media de } \textit{Utilidad\_búsqueda} > 3 \end{cases}$$

```
contraste t
-----------
Hipótesis nula: media = 3,0
Alternativa: mayor que

Estadístico t = 4,5344
P-valor = 0,0000817716

Se rechaza la hipótesis nula para alpha = 0,05.
```

El estadístico experimental tiene un valor igual a 4.5344 y el p-valor del contraste es $8.1 \cdot 10^{-5}$, por lo que se rechaza la hipótesis nula y se concluye que para este nivel de significación podemos afirmar que la media de la variable *Utilidad_búsqueda* es superior a 3.

Los p-valores tan pequeños en todos los contrastes indican que si hubiéramos disminuido la significación hasta un 1 % o inferior, el resultado hubiera seguido siendo el mismo.

Con esto hemos demostrado que la media de todas las variables es mayor que 3, lo que equivale a decir que los participantes están de acuerdo con las afirmaciones realizadas sobre las capacidades de la *tag cloud*:

1. Resulta intuitiva y fácil de usar
2. Representa de forma adecuada el contenido
3. Recupera información coherente

4. Ayuda en la realización de búsquedas

5. Es fácil la identificación de conceptos

En la Tabla 6.12 podemos ver un resumen de los resultados.

6.3 Otras Evaluaciones Experimentales

En esta sección presentamos otros ejemplos de *tag clouds* obtenidas con nuestra propuesta sobre textos cortos de distinta temática. La calidad de éstas se pone de manifiesto mediante el cálculo de algunas métricas [TP13a].

Los ejemplos que mostramos corresponden con los conjuntos de datos:

1. **CCIA.** En esta tabla se almacenan 500 títulos de artículos científicos relacionados con la temática de ciencias de la computación e inteligencia artificial.

2. **MEDLINE.** Este conjunto de datos contiene 500 títulos de artículos científicos relacionados con la temática biomédica.

3. **SPORT.** En esta tabla se recogen 500 títulos de noticias sobre deportes procedentes de la agencia de noticias "Reuters", con sede en Reino Unido.

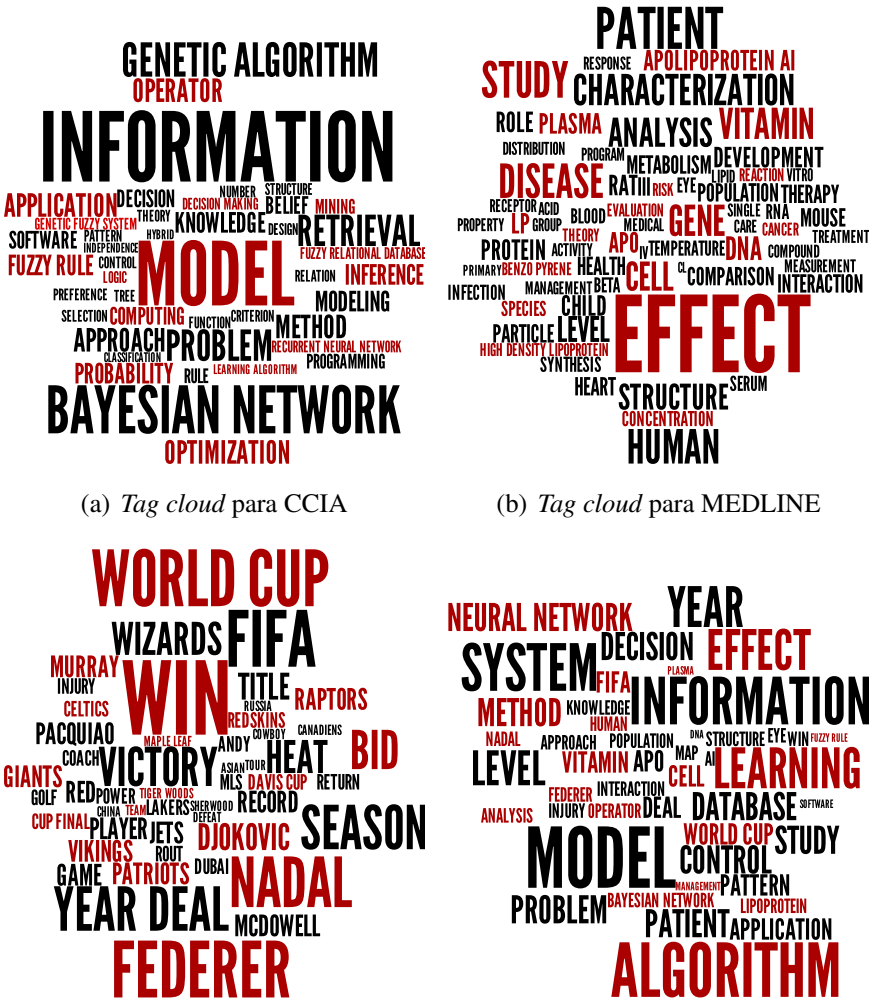
4. **MISC.** En este conjunto se incluyen 500 títulos seleccionados aleatoriamente de los tres conjuntos anteriores.

A cada uno de estos conjuntos de datos se le ha aplicado un preprocesamiento sintáctico y semántico con las herramientas descritas en las Secciones 5.5.1 y 5.5.2 y se ha obtenido la estructura WAPO como forma intermedia de representación. Tras un análisis visual, el soporte mínimo establecido para la selección de las *item-seqs* frecuentes es de 0.015 para el conjunto CCIA y 0.01 para los demás. La visualización de las *tag clouds* se muestra en la Figura 6.38.

De nuevo hemos usado las métricas adaptadas de Venetis et al. [Ven11] para el caso hipotético en que una consulta recupera todo el contenido de una columna de datos, ya que estos autores formulan las métricas para la *tag cloud* que se construye a partir de una consulta previa y nosotros la construimos a partir de la columna

Resumen de los resultados
Comprobación de que las valoraciones dadas en la identificación de conceptos son independientes del concepto relacionado Test de independencia Chi-cuadrado $\begin{cases} H_0 : \text{Existe independencia} \\ H_1 : \text{No existe independencia} \end{cases}$ Para un 5 % de significación: $\chi^2_{exp} = 9.16$, p-valor=0.6888, $\Rightarrow$ Aceptamos independencia (H_0)
Comprobación de que la media del grado de acuerdo con las afirmaciones sobre las capacidades de la *tag cloud* puede considerarse superior a 3, esto es, los participantes estan de acuerdo con dichas afirmaciones Contraste de hipótesis para la media de una población $\begin{cases} H_0 : \text{ Media} = 3 \\ H_1 : \text{ Media} > 3 \end{cases}$ Para un 5 % de significación: • *La tag cloud presentada es intuitiva y fácil de usar.* $t_{exp} = 5,98804$, p-valor=$2.5 \cdot 10^{-6} \Rightarrow$ Rechazamos $H_0 \Rightarrow$ **Deacuerdo con afirmación**. **Media muestral = 4,26087** • *Me resulta fácil identificar en la tag cloud los conceptos proporcionados.* $t_{exp} = 9,33422$, p-valor=0.0 $\Rightarrow$ Rechazamos $H_0 \Rightarrow$ **Deacuerdo con afirmación**. **Media muestral = 3,96739** • *La información recuperada con las etiquetas de la tag cloud es coherente con dichas etiquetas* $t_{exp} = 6,47025$, p-valor=$8.25 \cdot 10^{-7} \Rightarrow$ Rechazamos $H_0 \Rightarrow$ **Deacuerdo con afirmación**. **Media muestral = 4,21739** • *La tag cloud presentada aporta información sobre el contenido de la base de datos*. $t_{exp} = 4,16603$, p-valor=$2.3 \cdot 10^{-4} \Rightarrow$ Rechazamos $H_0 \Rightarrow$ **Deacuerdo con afirmación**. **Media muestral = 3,90476** • *Una tag cloud como la presentada me ayudaría a realizar búsquedas en una base de datos médica*. $t_{exp} = 4,5344$, p-valor=$8.1 \cdot 10^{-5} \Rightarrow$ Rechazamos $H_0 \Rightarrow$ **Deacuerdo con afirmación**. **Media muestral = 3,86957**

Tabla 6.12: Resumen de los resultados obtenidos en el experimento de la Sección 6.2



(a) *Tag cloud* para CCIA

(b) *Tag cloud* para MEDLINE

(c) *Tag cloud* para SPORT

(d) *Tag cloud* para MISC

Figura 6.38: *Tag clouds* obtenidas para diferentes conjuntos de datos

completa (ver Secciones 2.1.4 y 6.1.3). Las métricas calculadas son la de *cobertura*, *solapamiento* y *balance*.

Igual que en el experimento de la Sección 6.2 hemos calculado la *cobertura* total de la *tag cloud* y la *cobertura* media por etiqueta.

En la Tabla 6.13 podemos ver los valores de estas métricas para los cuatro conjuntos diferentes de datos.

Como vemos, los valores de *cobertura* son mayores cuando los datos son más homogéneos; es el caso de los conjuntos CCIA y MEDLINE. Para datos hete-

	Cobertura	*Cobertura por Etiqueta*	*Solapamiento*	*Balance*
CCIA	0.812	0.03586	0.00120	0.07792
MEDLINE	0.860	0.03510	0.00157	0.01960
SPORT	0.668	0.02283	0.00048	0.09803
MISC	0.628	0.02226	0.00052	0.11538

Tabla 6.13: Métricas de *cobertura*, *solapamiento* y *balance* de las *tag clouds* de la Figura 6.38

rogéneos como los de SPORT y MISC, que contienen información de temas más variados, los valores de *cobertura* son menores. El *solapamiento* es bueno en todos los casos, cercano a 0. Esto se debe en parte al uso de las componentes multitérmino, que ayudan en la contextualización de la información. Los conjuntos con mayor *solapamiento* son los más homogéneos mientras los más heterogéneos tienen un *solapamiento* menor.

Finalmente, observamos que todas las *tag clouds* son no equilibradas, hecho habitual cuando no se fuerza ningún tipo de criterio y no se adapta el soporte para hacer que las etiquetas representen un número similar de objetos. Nuestra intención no es mejorar el *balance*, ya que ello implicaría tamaños de fuente homogéneos en la visualización, impidiendo la posibilidad de destacar los temas más relevantes y de representar los términos frecuentes menos comunes.

6.4 Resumen y Conclusiones

En este capítulo se han realizado varias evaluaciones experimentales para probar nuestra metodología.

El primer experimento se ha desarrollado sobre un conjunto de datos formado por artículos de la revista “Security and Communication Networks”, de Wiley. Se ha elegido este conjunto debido a que Wiley proporciona una *tag cloud* en su sitio web para navegar por los contenidos de esta revista, así tenemos una *tag cloud* de referencia, creada por expertos, con la que comparar la obtenida con nuestra metodología.

Para esta comparación, hemos generado distintas *tag clouds* variando aspectos como el tipo de preprocesamiento empleado, la estructura representada o el

atributo procesado. La elegida como candidata para rivalizar con la de Wiley, ha sido la obtenida con la estructura WAPO, después de aplicar los preprocesamientos sintáctico y semántico. El atributo seleccionado ha sido el atributo que contenía las *keywords* de los artículos, ya que Wiley construye sus *tag clouds* también a partir de *keywords*.

Para confrontarlas desde el punto de vista de la recuperación de información, se ha calculado la *precisión*, *exhaustividad* y F_1 *Score*, obteniendo una F_1 similar para ambas visualizaciones.

También se han calculado las métricas de *cobertura*, *solapamiento* y *balance*, consiguiendo mayor *cobertura* con nuestra *tag cloud* y prácticamente el mismo *solapamiento* y *balance*.

Para finalizar este experimento, se ha llevado a cabo un estudio de usuario, en el que se ha seleccionado una muestra de participantes a los que se les ha pedido hacer unas determinadas tareas con ambas *tag clouds* con el objeto de compararlas. Los resultados de este estudio muestran que, desde el punto de vista de los participantes, ambas *tag clouds* pueden considerarse iguales para los objetivos propuestos, conclusión que es altamente satisfactoria, ya que hemos conseguido generar una visualización similar a la creada por expertos que además tiene las siguientes ventajas:

- Se genera automáticamente sin ningún tipo de intervención humana
- Posee mayor cobertura
- Todos los términos recuperan información relevante, lo cual no ocurre en la de referencia

Este experimento puede verse en [TP13b].

Una vez que hemos probado la eficiencia de la metodología, la hemos aplicado sobre una base de historias clínicas de un hospital universitario. Estos datos se diferencian de los anteriores en que están más desorganizados, son más extensos y acarrean mayores necesidades de limpieza.

La *tag cloud* generada ofrece buenos valores para todas las métricas calculadas. También en este caso se ha realizado una encuesta de satisfacción con usuarios

expertos mediante la que se ha efectuado una estimación del grado de acuerdo con las siguientes afirmaciones:

1. La *tag cloud* presentada me parece intuitiva y de fácil uso
2. La *tag cloud* presentada aporta información sobre el contenido de la base de datos
3. La información recuperada con las etiquetas de la *tag cloud* es coherente con dichas etiquetas.
4. Una *tag cloud* como la presentada me ayudaría a realizar búsquedas en una base de datos médica
5. Me resulta fácil identificar un concepto en la *tag cloud* relacionado con... (*cuatro definiciones distintas...*).

Tras analizar los datos recogidos con esta encuesta, el resultado es que, en media, los usuarios están "de acuerdo" con todas las afirmaciones, siendo las medias cercanas a 4 en todos los casos, medido el grado de acuerdo con calificaciones del 1 al 5, donde 1 significa estar "completamente en desacuerdo" y el 5 "completamente de acuerdo". Con esto hemos probado que la herramienta cumpliría todas las funciones para las que ha sido diseñada.

Por último, hemos visto otros experimentos que demuestran que todas las *tag clouds* generadas con nuestra metodología ofrecen en general buenos valores para las métricas de cobertura y solapamiento. Estos experimentos se encuentran publicados en [TP13a].

CAPÍTULO

7

Conclusiones y Trabajos Futuros

En esta tesis se ha presentado una metodología para obtener de forma automática una representación visual de información textual con propósitos de resumen, consulta y recuperación. Esta metodología implica las siguientes tareas:

1. Preprocesamiento sintáctico y semántico
2. Generación de una forma intermedia de representación
3. Postprocesamiento
4. Visualización a través de una *tag cloud*

Hay que destacar que se emplean componentes multitérmino, que aportan semántica y que la metodología posee una fundamentación teórica y está claramente definida, con lo que la visualización obtenida se distingue de otras, principalmente de la *tag cloud* monotérmino, que es la que encontramos en Internet con mayor frecuencia.

Así pues, podemos asegurar que se han alcanzado los objetivos específicos que se fijaron al principio de la memoria (Sección 1.2). Detallando un poco más, debemos resaltar las siguientes aportaciones:

1. Se han estudiado los antecedentes del problema planteado y de su solución. Se ha hecho un profundo repaso de los antecedentes de la *tag cloud* como interfaz visual, de los sistemas basados en *tagging* que fueron los que popularizaron esta herramienta y de los mecanismos existentes para dar semántica a las etiquetas en la *tag cloud*. También se han visto los antecedentes de la estructura-AP como forma matemática intermedia, comentando sus ventajas y sus principales limitaciones: ponderación y orden.

2. Se han establecido las definiciones formales de las extensiones esta estructura: estructuras WAP, APO y WAPO. Estas extensiones surgen de la inclusión de la ponderación (WAP), del orden estricto de adyacencia (APO) o de ambos (WAPO).

   También se han definido los índices de acoplamiento fuerte y débil de un conjunto con la estructura APO, como medidas de bondad del acoplamiento, operación que reviste de especial importancia en la consulta.

   Se ha desarrollado un amplio ejemplo en el que se ha comparado las estructura monotérmino ponderada con la WAP y WAPO, resaltando todos los aspectos en que estas últimas mejoran a la primera y comparándolas entre sí, especificando cuando resulta más útil usar cada una de ellas.

   También se ha ilustrado el proceso de consulta, calculando las subestructuras inducidas para cada tupla (lo que nos da su TDA) y las subestructuras inducidas a partir de las estructuras globales para algunos términos de consulta, subestructuras que representan la información recuperada.

3. Hemos estudiado la obtención de la forma intermedia a través de las implicaciones frecuentes de los términos frecuentes de nivel uno como método alternativo al algoritmo Apriori, concluyéndose que la conveniencia de usar uno u otro método dependerá de las características del texto en que se aplique.

4. Se ha especificado la metodología y la arquitectura del sistema para obtener el TDA del atributo textual. Se han descrito todas las herramientas usadas en las distintas etapas de preprocesamiento, generación de la forma intermedia y visualización y se han estudiado los procesos aplicados en cada etapa.

5. Se han realizado varios experimentos donde se ha validado todo el procedimiento planteado.

   El primero ha consistido en generar una *tag cloud* sobre una base de artículos científicos de la revista "Security and Commnunication Network" de Wiley y compararla con la que ofrece la propia revista en su página web. Se ha demostrado, por medio de una encuesta de usuario, que la *tag cloud* que generamos automáticamente con nuestra metodología es tan buena como aquella creada por expertos, con la ventaja de que no es necesaria la intervención humana para su generación. Además, hemos conseguido mejores resultados en las métricas de cobertura y exhaustividad (consultar [TP13b]).

   En el segundo experimento se ha trabajado con un conjunto de datos más extenso y desorganizado, una base de datos de historias clínicas. Para la *tag cloud* generada se han calculado las métricas de cobertura, solapamiento, balance, precisión y exhaustividad, obteniendo buenos valores para todas ellas. También se realizado una encuesta de satisfacción a especialistas médicos, consiguiendo resultados óptimos.

   Otras evaluaciones realizadas garantizan igualmente la efectividad de nuestra metodología (consultar [TP13a]).

Como trabajos futuros destacamos:

1. Calcular la precisión y exhaustividad contemplando otras posibilidades:

   - Calcular la precisión y la exhaustividad de la consulta cuando ésta se realiza a través de un acoplamiento débil. Por ejemplo, como en la *tag cloud* finalmente sólo se han representado las *item-seqs* maximales,

podría ocurrir que el usuario quisiera consultar los términos que están contenidos en una de ellas, sin que necesariamente tenga que hacerlo por la *item-seq* completa. Este tipo de consulta se realizaría a través de un acoplamiento débil, por lo que sería necesario calcular los índices de precisión y exhaustividad para cuando no se desea encontrar todos los términos, sino sólo algunos.

- Para el cálculo de la precisión y la exhaustividad de las etiquetas en la *tag cloud* hemos anotado las tuplas que deben recuperarse con cada etiqueta. Estas anotaciones pueden realizarse de manera difusa en función de la relevancia de esa tupla para la etiqueta en cuestión. De esta forma podríamos obtener índices de precisión y exhaustividad difusos, como en [MF08].

2. Con respecto al diseño de la *tag cloud*:

   - Permitir la consulta sobre un único término de una etiqueta, en lugar de la etiqueta completa, lo que es distinto a consultar a través del acoplamiento débil.
   - Estudiar si la agrupación en *clusters* de las etiquetas supone mejoras con respecto al diseño simple en que las relaciones entre conceptos se inducen gracias a los multitérminos.
   - Variar las formas de visualización de las etiquetas en la *tag cloud*. Analizar criterios tales como el color, dirección horizontal y/o vertical, orden de aparición de los términos, etc. con el fin de comprobar qué visualización resulta más atractiva a los usuarios y en cuál es más fácil identificar el contenido de la información representada.
   - 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 algunos autores afirman que se confunde la longitud con el tamaño de fuente, lo que a veces dificulta la identificación de los términos más relevantes.

3. Aplicar nuestra metodología sobre nuevos conjuntos de datos:

   Sería interesante emplear el sistema sobre una base de datos de noticias recuperadas de Internet. De esta forma se resaltarían los temas de mayor impacto y actualidad. La aplicación en las bases de datos de comercios daría muy buenos resultados: se resaltarían los artículos más demandados, las tendencias de los compradores, las mayores ventas, etc.

4. Ver cómo varían las *tag clouds* construidas sobre una base de datos cualquiera, cuando se aumenta o se disminuye el número de tuplas:

   Un aumento de tuplas se daría cuando se actualiza la base de datos, añadiendo nuevas entradas. En el caso de las noticias, se tendría una *tag cloud* dinámica, cuya evolución pondría de manifiesto los temas de actualidad en cada momento. Sin embargo, en otros tipos de datos podría ocurrir que ésta no variara mucho a través del tiempo, como podría ser en el caso de los datos sobre intervenciones quirúrgicas, en los que se necesitarían periodos mucho más prolongados para observar variación en las intervenciones realizadas con mayor frecuencia.

   Una razón de la disminución de tuplas puede ser la extracción de una muestra. Para *tag clouds* estáticas, como las construidas con datos médicos, sería interesante comprobar en qué tamaño de muestra se estabilizan, mostrando las mismas etiquetas que la generada para toda la población.

5. Estudiar un proceso para representar la *tag cloud* de todos los atributos textuales de la base de datos, conjuntamente:

   Esto podría realizarse asignando pesos diferentes según el término se encuentre o un campo o en otro de la base de datos, como proponían Koutrika et al. [Kou09a].

6. Combinar diferentes *tag clouds* obtenidas de diversas atributos de la base de datos:

   De esta forma podríamos reflejar las relaciones entre los atributos a través de las *tag clouds*.

7. Extender de la semántica de la estructura APO a través de ontologías:

   Una vez que se tiene la estructura APO de dominio, puede extenderse a una ontología enriquecida con términos adicionales de consulta extraídos de WordNet o Wikipedia. Dadas las particulares características de WordNet y su orientación lingüística, esta jerarquía contiene conceptos muy genéricos y abstractos, por lo que Wikipedia, debido a su carácter dinámico y consensuado, puede funcionar mejor para extender la semántica de la estructura.

8. Automatizar el preprocesamiento y postprocesamiento en otros idiomas aparte del inglés y el español:

   Automatizar primero el postprocesamiento en español cuando se disponga de un etiquetador adecuado para establecer la categoría gramatical de cada término. Posteriormente, diseñar un sistema de minería de textos multilingüe para automatizar todas las tareas y poder visualizar la *tag cloud* para distintos idiomas.

9. Uso del modelo OLAP y Data Warehouse:

   Creación de un modelo multidimensional, donde las dimensiones son los distintos atributos textuales, representadas utilizando un modelo OLAP. A partir del cubo OLAP puede construirse la *tag cloud* multitérmino como hacían Aouiche et al. [Aou09].

Apéndices

A

Uso de WordNet en el Preprocesamiento Semántico

En el Capítulo 5 se describió la metodología genérica para la limpieza de textos. En este apéndice, la adaptaremos para el uso de la forma de representación intermedia mediante WordNet, que es un diccionario léxico cuyo uso está muy extendido [Fel98].

Como ya se ha visto, en el preprocesamiento semántico lo que hacemos es agrupar los términos en conjuntos de sinónimos, para sustituir en el texto todos los sinónimos por el representante canónico de este conjunto. Toda esta tarea se realiza con la ayuda del diccionario electrónico WordNet.

WordNet organiza sus términos en *synsets*, que son los conjuntos de sinónimos. Una vez que se haya identificado el *synset* al que pertenece un término concreto, éste se sustituye en el texto por el representante canónico de ese *synset*. Apriori esta tarea parece sencilla, pero en realidad es un proceso bastante complejo que a su vez, requiere resolver otros problemas asociados, lo que lleva a cometer posibles errores para los que aún no se dispone de soluciones completamente efectivas, a pesar de haberse investigado mucho a este respecto.

Sabemos que una palabra con una misma forma sintáctica puede tener distinta categoría gramatical (nombre, verbo, etc.) y de esto depende su sentido, aunque incluso para una misma categoría gramatical, una palabra puede tener más de un sentido. A la tarea de etiquetar cada palabra con su categoría gramatical se la conoce como "*part-of-speech [POS] tagging*". Cada uno de los sentidos de una palabra hace que ésta esté incluida en un *synset* diferente , el cual puede contener varias palabras.

Las principales dificultades que nos encontramos en este procedimiento son:

1. Determinar la categoría gramatical de la palabra
2. Determinar su sentido correcto

A la tarea de concretar el sentido de una palabra se la conoce como "desambiguación (*Word Sense Disambiguation [WSD]*)".

A.1 Introducción a WordNet

La base de datos léxica para el idioma inglés, WordNet [Fel98], empezó a desarrollarse en el Laboratorio de Ciencia Cognitiva de la Universidad de Princeton, New Jersey, bajo la dirección del Catedrático en Psicología George A. Miller, en el año 1985.

WordNet posee un gran tamaño y su estructura lo convierte en una herramienta muy útil para el procesamiento de lenguaje natural y la lingüística computacional. WordNet agrupa nombres, verbos, adjetivos y adverbios en conjuntos de sinónimos llamados *synsets*, que están compuestos por términos que expresan un mismo concepto. Los *synsets* están conectados entre sí a través de relaciones conceptuales, semánticas y léxicas, formando una red. Esta red puede consultarse a través de una herramienta para obtener palabras y conceptos relacionados.

WordNet se compone de un conjunto de archivos lexicográficos, el código para convertirlos en una base de datos y las rutinas e interfaces de búsqueda para mostrar la información de la base de datos. Los archivos lexicográficos son los que organizan los nombres, verbos, adjetivos y adverbios en grupos de sinónimos y describen las relaciones entre dichos grupos.

Es muy importante conocer la forma en que WordNet organiza el conocimiento en la base de datos. Ya sabemos que la información en WordNet está organizada alrededor de agrupamientos lógicos llamados *synsets*. Cada uno de estos *synsets* se organiza en una lista de palabras individuales o colocaciones terminológicas (“wake up” , “screw driver”, etc.), y apuntadores que describen las relaciones entre un *synset* y los demás.

Una palabra puede aparecer en dos o más *synsets* y puede tener más de una categoría gramatical. Las palabras de un *synset* están agrupadas de tal forma que son intercambiables. Los apuntadores pueden describir relaciones de dos tipos: léxicas y semánticas. Las relaciones léxicas se generan entre las formas de las palabras y las semánticas entre los significados. Estas relaciones incluyen hiperonimia/hiponimia[1], antonimia, implicación, meronimia/holonimia[2], etc. Existen otros apuntadores para indicar diferentes relaciones.

Los verbos y los nombres se organizan en jerarquías basadas en relaciones de hiperonimia/hiponimia entre *synsets*.

Los adjetivos se organizan en grupos conteniendo *synsets* cabecera y *synsets* satélite. Cada *synset* satélite representa un concepto con significado similar al representado por el *synset* cabecera. Una forma de pensar acerca del agrupamiento de adjetivos, es visualizar una rueda, en la que *synset* cabecera actúa como centro y los *synsets* satélite son los radios. Dos o mas ruedas estarían conectadas a través de una relación de antonimia, que siguiendo el símil empleado, podría verse como un eje entre las ruedas. Existen excepciones a esta organización, como los adjetivos terminológicos (*“pertainyms”*), que son adjetivos relacionales y no se ajustan a la estructura descrita al no tener antónimos, por lo que, en la mayoría de los casos, el *synset* para ellos sólo contiene la palabra o la colocación y un apuntador léxico al adjetivo al que pertenece. Y los adjetivos participios, como se derivan de verbos, tienen apuntadores léxicos a estos verbos.

[1]Hiperónimo es aquel término cuyo significado incluye al de otro u otros términos, que son los hipónimos. Por ejemplo, pájaro es un hiperónimo respecto a los hipónimos jilguero y gorrión.

[2]Holonimia es una noción semántica que se opone a meronimia, del mismo modo en que se oponen el todo y la parte. Por ejemplo, bicicleta es un holónimo mientras que pedal, sillín, manillar son merónimos.

Los adverbios en ocasiones se derivan de los adjetivos y algunas veces tienen antónimos, por lo tanto, el *synset* para un adverbio suele contener un apuntador léxico al adjetivo del cual se deriva.

WordNet nos ofrece la información agrupada dependiendo de la categoría gramatical cuando realizamos una consulta. Y por cada categoría, muestra el conjunto de sentidos que tiene la palabra. Cada uno de estos sentidos se corresponderá con un *synset* y lleva asociada una definición y unos cuantos ejemplo para su comprensión. Cuando una palabra tiene más de un sentido en una misma forma gramatical, éstos aparecen ordenados según su frecuencia de uso. Esta característica de WordNet es muy importante en el proceso de desambiguación, ya que tomaremos como forma desambiguada, el primer sentido que nos ofrezca WordNet, que será el sentido más usado de esa palabra.

Términos y Conceptos Básicos en WordNet

A continuación, hacemos un breve repaso de la terminología básica de WordNet y los conceptos asociados a ellos, ya que la mayor parte de la terminología empleada en WordNet es propia del sistema e incluso otros términos generales tienen un significado específico en el contexto de WordNet.

En la lista de definiciones incluimos el nombre original del concepto.

- ***Synset* - Conjunto de sinónimos**. Es un conjunto de palabras que son intercambiables en algunos contextos sin cambiar el valor de verdad de la proposición en la que se encuentran.

- **Atributo (*attribute*)** - Es un nombre para el que los adjetivos expresan valores. Por ejemplo, el nombre “color” es un atributo, para el que los adjetivos “azul“ y ”blanco“ expresan valores.

- **Dominio (*domain*)** - Es una clasificación temática a la cual se ha enlazado un *synset* mediante un apuntador.

- **Término de dominio (*domain term*)** - Es un *synset* perteneciente a la clase de un determinado tema.

- **Forma base (*base form*)** - La forma base de una palabra o colocación terminológica, es aquella forma básica sobre la que se añaden inflexiones.
- **Lista de excepciones (*exception list*)** - Son transformaciones morfológicas de palabras que no son regulares, y por tanto no pueden procesarse de forma algorítmica.
- **Colocación terminológica (*collocation*)** - Una colocación terminológica en WordNet es una cadena con dos o más palabras conectadas por espacios en blanco o guiones. En la base de datos los espacios se representan como caracteres de subrayado ("_").
- **Formas relacionadas derivacionalmente (*derivationally related forms*)** - Son términos en diferentes categorías sintácticas que tienen la misma forma raíz y están semánticamente relacionados.
- **Adjetivo Participativo (*participial adjective*)** - Es un adjetivo que se deriva de un verbo.
- **Adjetivo Terminológico (*pertainym*)** - Es un adjetivo relacional. Los adjetivos de este tipo suelen definirse con frases tales como *"de o perteneciente a"* y no tienen antónimos. Un adjetivo terminológico puede apuntar a un nombre o a otro adjetivo terminológico.
- **Predicativo (*predicative*)** - Es un adjetivo que puede usarse únicamente en posiciones de predicado. Si X es un adjetivo predicativo sólo puede usarse en frases de tipo *"es X"* y nunca antes del nombre.
- **Postnominal** - Un adjetivo postnominal aparece únicamente justo a continuación del nombre que modifica.
- **Prenominal** - Es un adjetivo que únicamente puede aparecer antes del nombre que modifica.
- ***Cluster* de adjetivos (*adjective cluster*)** - Es un grupo de *synsets* adjetivos que están organizados alrededor de pares o triples de antónimos. Un *cluster* de adjetivos contiene dos o más *synsets* cabecera, que representan conceptos antónimos. Cada *synset* cabecera tiene uno o más *synsets* satélite.

- **Apuntador entre *clusters* (*cross-cluster pointer*)** - Es un apuntador semántico de un *cluster* de adjetivos a otro.
- **Apuntador léxico (*lexical pointer*)** - Un apuntador léxico indica una relación entre palabras de los *synsets*.
- **Apuntador semántico (*semantic pointer*)** - Un apuntador semántico indica una relación entre *synsets* (conceptos).
- **Synset cabecera (*head synset*)** - Es un cluster de adjetivos, el cual contiene al menos una palabra que posee un antónimo directo.
- **Synset satélite (*satellite synset*)** - Es el *synset* en un *cluster* de adjetivos, que representa un concepto similar en significado al concepto representado por el *synset* cabecera.
- **Grupo (*group*)** - Los grupos son sentidos de verbos similares en significado.
- **Implicación (*entailment*)** - Un verbo X implica a Y si X no puede realizarse a menos que se realice o se haya realizado Y.
- **Tropónimo (*troponym*)** - Es un verbo que denota la forma específica de otro verbo. Por ejemplo ”marchar“ es un tropónimo de ”andar“. X es tropónimo de Y, si hacer X es hacer Y de una determinada manera.
- **Holónimo (*holonym*)** - Es el nombre del todo al que hacen referencia los merónimos. Y es un holónimo de X, si X es parte de Y.
- **Merónimo (*meronym*)** - Es el nombre de un constituyente de *”la parte de“, ”la sustancia de“, ”el miembro de algo“*. X es un merónimo de Y, si X es parte de Y.
- **Hiperónimo (*hypernym*)** - El término general empleado para designar una clase completa de instancias específicas. Y es un hiperónimo de X, si X es algún tipo de Y.
- **Coordinados (*coordinate*)** - Términos coordinados son aquellos nombres o verbos que tienen el mismo hiperónimo.

- **Hipónimo (*hyponym*)** - El término específico usado para designar un miembro de una clase. X es un hipónimo de Y, si X es algún tipo de Y.
- **Subordinado (*subordinate*)** - Equivalente a hipónimo.
- **Superordinado (*superordinate*)** - Equivalente a hiperónimo.
- **Instancia (*instance*)** - Es un nombre propio cuyo referente es único (en contraposición a los nombres que hacen referencia a clases). Es una forma específica de hipónimo.
- **Monosémico (*monosemous*)** - Término que tiene un único sentido en una categoría sintáctica.
- **Polisémico (*polysemous*)** - Término que tiene más de un sentido en una categoría sintáctica.
- **Glosa o Definición (*gloss*)** - Cada *synset* contiene una glosa que consiste en una definición y ejemplos.
- **Lema (*lemma*)** - Es el texto ASCII en minúscula que aparece en los ficheros de índices de la base de datos de WordNet. Por lo general, se corresponde con la forma base de una palabra o de una colocación terminológica.
- **Fichero lexicográfico (*lexicographer file*)** - Es un fichero que contiene los datos de los *synsets* de WordNet, creados por expertos.
- **Identificador lexicográfico (*lex id*)** - Es un número decimal, que añadido a un lema, identifica el sentido de forma unívoca en un fichero lexicográfico.
- **Categoría gramatical (*part of speech - POS*)** - Es equivalente a categoría sintáctica. WordNet define las siguientes categorías gramaticales: nombre, verbo, adjetivo o adverbio.
- **Sentido (*sense*)** - Es un significado de una palabra en WordNet. Cada sentido de una palabra se encuentra en un *synset* diferente.

- **Clave de sentido (*sense key*)** - La clave de sentido proporciona la información necesaria para encontrar un sentido en la base de datos de WordNet. Una clave de sentido combina un campo lema y códigos para el tipo de *synset*, identificador lexicográfico, número de fichero lexicográfico y la información sobre la cabecera del *synset* satélite, en caso de que fuese necesario.

- **Contador polisémico (*polysemy count*)** - Es el número de sentidos de una palabra, en una categoría sintáctica en WordNet.

- **Concordancia semántica (*semantic concordance*)** - Se da cuando un conjunto de textos y un diccionario léxico como WordNet, están combinados de tal manera que cada sustantivo en el texto está enlazado con su sentido apropiado en el diccionario a través de una etiqueta semántica.

- **Etiqueta Semántica (*semantic tag*)** - Es un apuntador desde una palabra en un fichero de texto a un sentido específico de dicha palabra en la base de datos de WordNet. Una etiqueta semántica en una concordancia semántica se representa por una clave de sentido.

Análisis Morfológico

Las búsquedas en WordNet pueden realizarse sobre formas declinadas aunque WordNet, por lo general, almacene únicamente formas base. Esto se hace empleando un conjunto de funciones morfológicas ("*Morphy*"), que generan una forma que aparezca en WordNet aplicadas sobre la cadena de búsqueda.

Son dos los procesos que se aplican en este análisis morfológico para convertir la cadena:

1. En un proceso hay una serie de ficheros con una lista de excepciones, una por cada categoría sintáctica, en los que se encuentran algunas formas declinadas. Así, cuando se recibe una cadena a buscar, primero se comprueba si la palabra aparece en los ficheros de excepciones. De ser así, se sustituye la cadena por la forma base correspondiente. En caso de que la palabra no aparezca en la lista de excepciones, se aplica el otro proceso.

2. El otro proceso consiste en eliminar de las palabras individuales una serie de terminaciones, basadas en categorías sintácticas, incluidas en una lista, para transformar estas palabras en las formas apropiadas.

A continuación se describe con más detalle el contenido de las listas de excepciones y las reglas de eliminación de terminaciones.

Listas de Excepciones

Como hemos dicho, existe un fichero con una lista de excepciones para cada categoría sintáctica. Estas listas contienen transformaciones morfológicas que no pueden ser procesadas mediante un algoritmo al no ser regulares. El tipo de búsqueda que se realiza para buscar en las listas, es una búsqueda binaria. La lista de excepciones contiene, por cada línea, una forma declinada y una serie de formas base con las que se corresponde. Normalmente se requieren varias llamadas a *Morphy* para obtener todas las formas base de una palabra declinada.

Morphy también almacena algunas abreviaturas que se usan con frecuencia, aparte de excepciones. Y realiza un tratamiento distinto para las palabras terminadas con el sufijo "ful". Para procesar estas palabras, lo que hace es tomar la parte que precede a "ful" y, cuando el procesamiento ha finalizado, vuelve a añadirle este sufijo a la palabra.

Las palabras simples suelen tener un procesamiento sencillo; se buscan en WordNet y si no se obtienen resultados, se recurre a las listas de excepciones tras aplicarles las reglas de eliminación de terminaciones.

Las colocaciones sintácticas, en cambio, son más difíciles de procesar. Generalmente, lo que se hace es procesar las palabras que componen la colocación de forma individual. Esta aproximación suele funcionar con nombres, pero existen excepciones que serán incluidas en las listas de excepciones correspondientes.

Cuando las palabras en una colocación aparecen unidas mediante guiones, lo que es frecuente en inglés aunque no obedezca a regla alguna, WordNet interpreta los guiones como espacios en blanco y se procede de igual modo que en las colocaciones sin guión, pero sólo en el caso de que la forma con guión no se encuentre en WordNet.

Cuando existe una colocación sintáctica que contiene una preposición, el procesamiento es mucho más complicado. Primero se intenta buscar esta colocación en la lista de excepciones, como se haría con una palabra individual y en caso de no encontrarla en la lista, se comprueba si existe una preposición. Si es así, *Morphy* supone que la primera palabra de la colocación es un verbo y la última un nombre, obteniendo para ellas la forma base correspondiente y dejando el resto de palabras (palabras intermedias entre el verbo y el nombre) sin modificar. Si *Morphy* determina que no existe preposición, entonces se busca la forma base de todas las palabras de la colocación.

Reglas de eliminación de terminaciones

Podemos ver las reglas de eliminación de terminaciones empleadas por *Morphy* en la Tabla A.1.

Como vemos, no existen reglas aplicables a adverbios.

Si una palabra termina con alguno de los sufijos especificados en la tabla, el sufijo se elimina y se añade en su lugar la nueva terminación.

Una vez obtenida la forma base, ya es posible buscarla en WordNet.

Relaciones empleadas en WordNet

Como se ha visto, WordNet aporta las definiciones de los términos, los organiza en conjuntos de sinónimos y almacena las relaciones semánticas entre unos y otros.

En la Tabla A.2 se muestran las principales relaciones semánticas que se identifican, junto a una descripción de cada una.

A.2 Etiquetado de Categoría Gramatical

Recordemos que el etiquetado de la categoría gramatical lo realizábamos mediante un programa, desarrollado en la Universidad de Stanford y basado en los trabajos publicados por *"The Stanford Natural Language Processing Group"* [Tou00, Tou03], que asignaba una etiqueta a cada palabra indicando su categoría gramatical.

Las etiquetas empleadas por este etiquetador se corresponden con el conjunto de etiquetas *"Penn Treebank English POS Tag Set"* [Mar93].

POS	Suffix	Ending
NOUN	“s”	“”
NOUN	“ses”	“s”
NOUN	“xes”	“x”
NOUN	“zes”	“z”
NOUN	“ches”	“ch”
NOUN	“shes”	“sh”
NOUN	“men”	“man”
NOUN	“ies”	“y”
VERB	“s”	“”
VERB	“ies”	“y”
VERB	“es”	“e”
VERB	“es”	“”
VERB	“ed”	“e”
VERB	“ed”	“”
VERB	“ing”	“e”
VERB	“ing”	“”
ADJ	“er”	“”
ADJ	“est”	“”
ADJ	“er”	“e”
ADJ	“est”	“e”

Tabla A.1: Reglas de eliminación de terminaciones en *Morphy*

Pero WordNet sólo utiliza cuatro categorías gramaticales, que son nombres, verbos, adjetivos y adverbios, por lo que es necesario adaptar el conjunto de etiquetas *Penn Treebank* a WordNet.

En la Tabla A.3 se muestra la descripción y correspondencia entre los dos conjuntos de etiquetas (*Penn Treebank* y WordNet).

Cuando una etiqueta de *Penn Treebank* no tiene equivalencia en WordNet, dicha palabra no existe en WordNet, por lo que se considera que la etiqueta corresponde a una palabra vacía y la palabra se elimina.

Cuando aparecen palabras poco comunes, suelen reconocerse como nombres propios (*NP o NPS*), pero a veces son palabras técnicas o específicas que no se reconocen como tales, lo que es un inconveniente, ya que la categoría gramatical de estas palabras puede ser en realidad un verbo o un adjetivo y no necesariamente un nombre, que es la que se asignaría en WordNet.

Relación	Descripción
ANTONYM	Antónimo, relaciona *synsets* con significados opuestos
ENTAILMENT	Un verbo implica a otro si depende de éste para su realización
ENTAILED_BY	Un verbo es implicado por otro que depende de él para su realización
HYPERNYM	Hiperónimo, relaciona términos con otros que los contienen
HYPONYM	Hipónimo, relaciona un término con aquellos que contiene
SEE_ALSO	Enlaza términos relacionados
SIMILAR_TO	Indica cierta similitud entre los conceptos

Tabla A.2: Relaciones modeladas en WordNet

Al etiquetar las palabras con su categoría gramatical estamos eliminando todos los sentidos pertenecientes a otra categorías gramaticales, lo que facilita la tarea de encontrar el sentido adecuado a la hora de realizar la desambiguación.

A.3 Desambiguación

Una vez determinada la categoría gramatical de cada una de las palabras del texto, se procede a determinar su sentido. El algoritmo adaptado de Lesk [Ban02] es un método pensado específicamente para WordNet y que aprovecha al máximo su estructura, por lo que es el que hemos escogido entre los métodos de desambiguación presentados en la Sección 5.5.2.

Penn Treebank	Descripción	WordNet
CC	Coordinating conjunction	
CD	Cardinal number	
DT	Determiner	
EX	Existential there	
FW	Foreign word	
IN	Preposition or subordinating conjunction	
JJ	Adjective	ADJECTIVE
JJR	Adjective, comparative	ADJECTIVE
JJS	Adjective, superlative	ADJECTIVE
LS	List item marker	
MD	Modal	
NN	Noun, singular or mass	NOUN
NNS	Noun, plural	NOUN
NP	Proper noun, singular	NOUN
NPS	Proper noun, plural	NOUN
PDT	Predeterminer	
POS	Possessive ending	
PP	Personal pronoun	
PP$	Possessive pronoun	
RB	Adverb	ADVERB
RBR	Adverb, comparative	ADVERB
RBS	Adverb, superlative	ADVERB
RP	Particle	
SYM	Symbol	
TO	to	
UH	Interjection	
VB	Verb, base form	VERB
VBD	Verb, past tense	VERB
VBG	Verb, gerund or present participle	VERB
VBN	Verb, past participle	VERB
VBP	Verb, non-3rd person singular present	VERB
VBZ	Verb, 3rd person singular present	VERB
WDT	Wh-determiner	
WP	Wh-pronoun	
WP$	Possessive wh-pronoun	
WRB	Wh-adverb	ADVERB

Tabla A.3: Correspondencia entre *Penn Treebank Tag Set* y WordNet

B

Encuestas de los Experimentos

B.1 Encuesta de Comparación de dos *Tag Clouds*

(a) TC1

(b) TC2

Una *tag cloud* es una herramienta para representar el contenido de la información de un determinado lugar. Sirve para explorar este contenido y realizar búsquedas a través de las etiquetas que la componen, ya que cada etiqueta representada en la *tag cloud* es un enlace a las entradas que contienen esta etiqueta. Le presentamos dos *tag clouds* a las que llamaremos $TC1$ y $TC2$, las cuales enlazan la información de la base de datos de la revista "Security and Communication Network" de Wiley.

Experimente un rato con ambas herramientas y posteriormente, emita una valoración en la comparación de éstas en relación con cada uno de los aspectos que se detallan a continuación:

1. **Facilidad de uso**:

   1) $TC1$ es mucho mejor que $TC2$
   2) $TC1$ es un poco mejor que $TC2$
   3) $TC1$ es igual que $TC2$
   4) $TC1$ es un poco peor que $TC2$
   5) $TC1$ es mucho peor que $TC2$

2. **Cantidad global de información recuperada**:

   1) $TC1$ es mucho mejor que $TC2$
   2) $TC1$ es un poco mejor que $TC2$
   3) $TC1$ es igual que $TC2$
   4) $TC1$ es un poco peor que $TC2$
   5) $TC1$ es mucho peor que $TC2$

3. **Representación del contenido de la información**:

   1) $TC1$ es mucho mejor que $TC2$
   2) $TC1$ es un poco mejor que $TC2$
   3) $TC1$ es igual que $TC2$
   4) $TC1$ es un poco peor que $TC2$
   5) $TC1$ es mucho peor que $TC2$

4. **Facilidad para localizar un concepto relacionado con "vulneración del sistema"**:

   1) $TC1$ es mucho mejor que $TC2$
   2) $TC1$ es un poco mejor que $TC2$
   3) $TC1$ es igual que $TC2$
   4) $TC1$ es un poco peor que $TC2$
   5) $TC1$ es mucho peor que $TC2$

5. **Facilidad para localizar un concepto relacionado con "verificación de identidad"**:

   1) $TC1$ es mucho mejor que $TC2$
   2) $TC1$ es un poco mejor que $TC2$
   3) $TC1$ es igual que $TC2$
   4) $TC1$ es un poco peor que $TC2$
   5) $TC1$ es mucho peor que $TC2$

6. **Facilidad para localizar un concepto relacionado con "técnicas de cifrado y/o codificación"**:

   1) $TC1$ es mucho mejor que $TC2$
   2) $TC1$ es un poco mejor que $TC2$
   3) $TC1$ es igual que $TC2$
   4) $TC1$ es un poco peor que $TC2$
   5) $TC1$ es mucho peor que $TC2$

7. **Facilidad para localizar un concepto relacionado con "prevención de accesos no autorizados en una red"**:

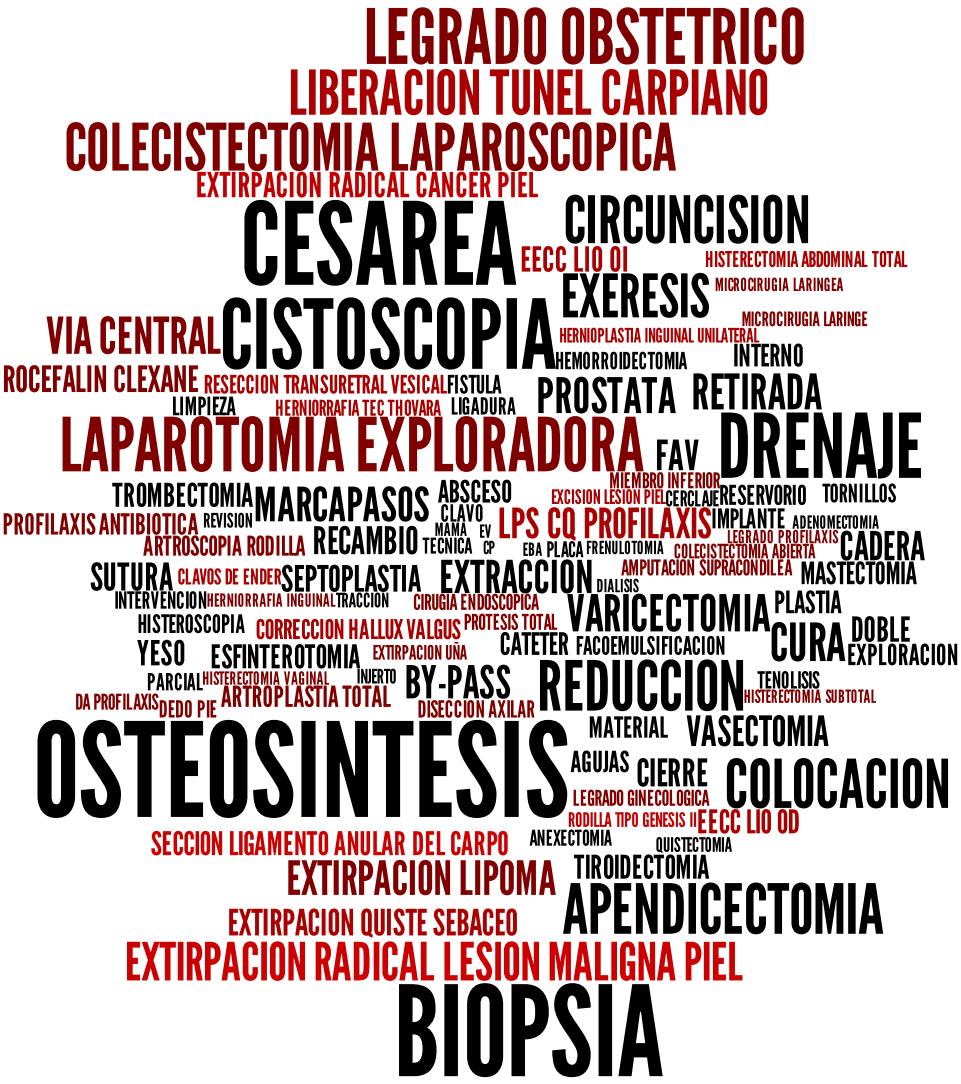
   1) $TC1$ es mucho mejor que $TC2$
   2) $TC1$ es un poco mejor que $TC2$

3) $TC1$ es igual que $TC2$

4) $TC1$ es un poco peor que $TC2$

5) $TC1$ es mucho peor que $TC2$

B.2 Encuesta de Satisfacción de la *Tag Cloud* con Datos Médicos



Solicitamos su colaboración para un estudio que estamos realizando en la Universidad de Granada.

En esta página le mostramos una herramienta que hemos construido de forma experimental sobre una base con datos reales de un hospital universitario y representa la información que se tiene en esa base sobre las intervenciones quirúrgicas realizadas en un determinado periodo de tiempo.

Esta herramienta se conoce como "*tag cloud*" y consiste en una visualización de texto con distintos tamaños de fuente según la frecuencia de los conceptos representados.

Suponga que usted desea realizar una consulta a través de una caja de texto para ver los registros de esta base de datos (u otra como ésta) en relación con una determinada intervención quirúrgica, pero desconoce el contenido y el vocabulario con que se expresa la información, lo que dificulta la obtención de resultados al realizar esta consulta.

A través de una *tag cloud* como la que le presentamos, podría identificar parte de este contenido, saber sobre qué intervenciones se tiene un mayor número de registros y bajo qué términos se pueden consultar estas intervenciones, asegurándose la obtención de resultados.

Para hacer una búsqueda de los registros que se tienen en relación a una determinada intervención, basta con pulsar en el término que la representa dentro de la tag cloud.

Le pedimos que experimente un rato con esta herramienta y posteriormente emita una valoración numérica del 1 al 5 para cada una de las afirmaciones que se realizan, donde 1="Completamente en desacuerdo" y 5="Completamente de acuerdo". Por último, le agradecemos que escriba algunas sugerencias sobre qué aspectos podemos mejorar o qué otras funciones desearía que tuviese.

1. **La *tag cloud* me parece intuitiva y de fácil uso**:

   a) 1

   b) 2

   c) 3

   d) 4

   e) 5

2. **La *tag cloud* presentada aporta información sobre el contenido de la base de datos**:

   a) 1

b) 2

c) 3

d) 4

e) 5

3. **La información recuperada con las etiquetas de la *tag cloud* es coherente con dichas etiquetas**:

   a) 1

   b) 2

   c) 3

   d) 4

   e) 5

4. **Una *tag cloud* como la presentada me ayudaría a realizar búsquedas en una base de datos médica**:

   a) 1

   b) 2

   c) 3

   d) 4

   e) 5

5. **Me resulta fácil identificar un concepto en la *tag cloud* relacionado con "una intervención quirúrgica para el nacimiento de un bebé"**:

   a) 1

   b) 2

   c) 3

   d) 4

   e) 5

6. **Me resulta fácil identificar un concepto en la *tag cloud* relacionado con "aborto quirúrgico o tratamiento tras aborto"**:

   a) 1

   b) 2

   c) 3

   d) 4

   e) 5

7. **Me resulta fácil identificar un concepto en la *tag cloud* relacionado con "operación quirúrgica que tiene por objeto la reconstrucción completa de una articulación destruida o anquilosada"**:

   a) 1

   b) 2

   c) 3

   d) 4

   e) 5

8. **Me resulta fácil identificar un concepto en la *tag cloud* relacionado con "técnica muy utilizada en la operación de cataratas"**:

   a) 1

   b) 2

   c) 3

   d) 4

   e) 5

9. **Sugerencias de mejora en la *tag cloud* presentada**:

Glosario

A

Atributo Propiedad de interés de una entidad que corresponde a una columna en la base de datos. Por ejemplo, atributos de la entidad "persona" son el "nombre","apellidos","edad",etc.

B

Base de conocimiento Tipo especial de base de datos que provee los medios para la recolección, organización y recuperación computarizada de conocimiento.

C

Cluster Agrupación.

Clustering Técnica enfocada en segmentar un conjunto en subconjuntos homogéneos.

Confianza Probabilidad de acertar en la estimación que se hace de un parámetro a través de un intervalo. Si la confianza es del 95 %, el verdadero valor del parámetro estará comprendido en el intervalo de confianza con un 95 % de probabilidad y 5 % será la probabilidad de equivocarnos en dicha estimación.

Conocimiento Conjunto de información almacenada.

Contraste bilateral Un contraste de hipótesis es bilateral cuando en la hipótesis alternativa el valor que se le asigna al parámetro es distinto que el que tiene en la hipótesis nula.

Contraste de hipótesis Procedimiento mediante el cual se trata de cuantificar las diferencias o discrepancias entre una hipótesis estadística y una realidad de la que se posee una información muestral, estableciendo una regla de decisión para juzgar si las discrepancias son excesivamente grandes y, por tanto, rechazar la hipótesis.

Contraste unilateral Un contraste de hipótesis es unilateral cuando en la hipótesis alternativa el valor que se le asigna al parámetro es mayor o menor que el que tiene en la hipótesis nula.

D

Data cloud (Nube de datos) 1. *Tag cloud* construida sobre una base de datos 2. *Tag cloud* donde las etiquetas son dígitos en lugar de palabras.

Desambiguación Proceso mediante el cual se identifica el sentido de una palabra usada en una oración.

Discretizar Dividir el todo en partes.

E

Entidad En bases de datos es la representación de un objeto o concepto del mundo real. Una entidad está constituida por uno o más atributos.

Estadístico Cualquier función de los valores de una muestra.

Estadístico experimental o de contraste Variable aleatoria cuyo valor para una muestra determinada nos permite tomar la decisión sobre la aceptación o el rechazo de la hipótesis nula.

Estructura APO Estructura-AP Ordenada.

Estructura WAP Estructura-AP Ponderada (Weighted AP-Structure).

Estructura WAPO Estructura-AP Ordenada Ponderada (Weighted AP-Ordered Structure).

Estructura-AP Estructura de *itemsets* frecuentes obtenidos mediante el algoritmo APriori [Agr94].

Exhaustividad Fracción de instancias relevantes que son recuperadas con la consulta. Se calcula como el número de resultados correctos entre el número de resultados que se debían haber devuelto con la consulta.

F

Folksonomía Resultado del etiquetado social o colaborativo. Clasificación de los contenidos por medio de etiquetas.

Forma intermedia Representación del conocimiento que captura lo esencial y permite trabajar con grandes colecciones de datos.

Fuente de información Diversos tipos de documentos que contienen datos útiles para satisface una demanda de información o conocimiento.

H

Herencia Mecanismo de los lenguajes de programación orientada a objetos basados en clases, por medio del cual una clase se deriva de otra de manera que extiende su funcionalidad.

Hipótesis alternativa Hipótesis que se plantea como alternativa a la hipótesis nula y que aceptaremos si, como consecuencia del contraste, rechazamos esta última.

Hipótesis nula Hipótesis que deseamos contrastar, considerada en principio como verdadera y que aceptaremos o rechazaremos como consecuencia del contraste.

I

IF Filtrado de Información (Information Filtering).

Intervalo de confianza Rango de valores calculado sobre una muestra en el cual se encuentra el verdadero valor del parámetro estimado con una determinada probabilidad. Dicha probabilidad se denomina "nivel de confianza".

IR Recuperación de Información (Information Retrieval).

Item-seq Conjunto ordenado o secuencia de elementos o términos.

Itemset Conjunto de términos o elementos.

L

Lema Forma que por convenio se acepta como representante de todas las formas flexionadas de una misma palabra.

Lematización Proceso lingüístico que consiste en, dada una forma flexionada (plural, femenino, conjugación, etc.), hallar el lema correspondiente.

M

Mediana Valor de la variable que divide la población en dos partes iguales. Ocupa la posición central en un conjunto de datos ordenados.

Moda Valor de la variable con mayor probabilidad.

Monotérmino Palabra simple o compuesta por un sólo término.

Multitérmino Palabra compuesta por dos o más términos.

N

Nivel de significación Probabilidad de rechazar la hipótesis nula cuando es cierta.

O

OLAP Procesamiento Analítico En Línea (On-Line Analytical Processing). Define una terminología que se basa en al análisis multidimensional de los datos y que le permite al usuario tener una visión más rápida e interactiva de los mismos.

Ontología Especificación explícita de una conceptualización. Definición del vocabulario de un área mediante un conjunto de términos básicos y relaciones entre dichos términos, así como las reglas que combinan términos y relaciones que amplían las definiciones dadas en el vocabulario.

P

PL Lenguaje de Procedimiento (Procedural Language).

PostgreSQL SGBD relacional orientado a objetos y libre.

Precisión Fracción de instancias recuperadas que son relevantes para la consulta. Se calcula como el número de resultados correctos entre el número total de resultados devueltos por la consulta.

p-valor Probabilidad de obtener un resultado al menos tan extremo como el que se ha obtenido en el estadístico de contraste calculado, suponiendo que la hipótesis nula es cierta. Cuando el p-valor es menor o igual que el nivel de significación, se rechaza la hipótesis nula, en caso contrario se acepta.

R

Retículo Un conjunto parcialmente ordenado se denomina retículo si satisface las propiedades: existencia del supremo por pares y existencia del ínfimo por pares. Ver Figura 2.3.

S

SGBS Sistemas Gestores de Bases de Datos.

Soporte El soporte de un *itemset* en una base de datos se define como la proporción de transacciones en la base de datos que contiene dicho *itemset*.

SQL Lenguaje Estructurado de Consulta (Structured Query Language).

T

Tag cloud (Nube de etiquetas) Visualización de etiquetas o marcas con distintos tamaños de fuente acordes a la importancia de cada etiqueta.

Tagging (Etiquetado) Acción que consiste en marcar una fuente de información con una determinada etiqueta o etiquetas.

TDA Tipo de Dato Abstracto.

Test estadístico Contraste de hipótesis.

Tokenización Proceso de descomposición de una cadena de texto en palabras, frases, símbolos u otros elementos significativos llamados "tokens".

Tupla Conjunto de elementos que se guardan en memoría de forma consecutiva. En bases de datos se define como una función finita que asocia unívocamente los nombres con algunos valores.

V

Varianza Valor que representa una medida de dispersión media de una variable respecto a su media.

W

Word cloud (Nube de palabras) Visualización de palabras extraídas del texto con distintos tamaños de fuente acordes a la frecuencia de cada palabra.

Bibliografía

[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. 25, 44

[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, pp 487–499. 1994. 6, 9, 57, 59, 115, 150, 153, 156, 157, 163, 182, 195, 311

[Agr95] Agrawal, R. y Srikant, R. Mining Sequential Patterns. En *Proceedings of the Eleventh International Conference on Data Engineering*, pp 3–14. 1995. 59, 155, 157

[Amm12] Ammari, A., Lau, L., y Dimitrova, V. Distributed Data Mining for User Sensemaking in Online Collaborative Spaces. En *Proceedings of the 26th Conference on Computer Supported Cooperative Work*, pp 1–10. 2012. 42

[Ang08] Angeletou, S., Sabou, M., y Motta, E. Semantically Enriching Folksonomies with FLOR. pp 65–80. 2008. 51

[Aou09] Aouiche, K., Lemire, D., y Godin, R. Web 2.0 OLAP: From Data Cubes to Tag Clouds. *Web Information Systems and Technologies*, pp 51–64, 2009. 31, 54, 284

[Ara97] Araujo, M.D., Navarro, G., y Ziviani, N. Large text searching allowing errors. pp 2–20, 1997. 160

[Ast09] Astrain, J., Echarte, F., Córdoba, A., y Villadangos, J. A Tag Clustering Method to Deal with Syntactic Variations on Collaborative Social Networks. *Web Engineering*, pp 434–441, 2009. 37

[Bal12] Balachandran, V., Balachandran, V.D.P., y Khemani, D. Interpretable and Reconfigurable Clustering of Document Datasets by Deriving Word-Based Rules. *Knowledge and Information Systems*, 32(3):475–503, 2012. 26, 230

[Ban02] Banerjee, S. y Pedersen, T. An Adapted Lesk Algorithm for Word Sense Disambiguation using WordNet. *Computational Linguistics and Intelligent Text Processing*, pp 136–145, 2002. 193, 206, 214, 298

[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*, pp 193–202. ACM, 2008. 35

[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. 17, 25, 51, 52

[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*, pp 703–704. ACM, 2008. 34

[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. 18, 63

[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. 23, 55

[Blu87] Blumer, A., Blumer, J., Haussler, D., McConnell, R., y Ehrenfeucht, A. Complete inverted files for efficient text retrieval and analysis. *Journal of the ACM*, 34(3):578–595, 1987. 9, 163, 164

[Bra00] Brants, T. TnT: a Statistical Part-of-Speech Tagger. En *Proceedings of the Sixth Conference on Applied Natural Language Processing*, pp 224–231. Association for Computational Linguistics, 2000. 191

[Can08] Cantador, I., Szomszor, M., Alani, H., Fernández, M., y Castells, P. Enriching Ontological User Profiles with Tagging History for Multi-Domain Recommendations. En *Proceedings of the 1st International Workshop on Collective Semantics: Collective Intelligence and the Semantic Web*, pp 5–20. 2008. 50, 51, 66

[Che09] Chen, Y.X., Santamaría, R., Butz, A., y Therón, R. TagClusters: Semantic Aggregation of Collaborative Tags beyond TagClouds. En *Proceedings of the 10th Intenational Symposium on Smart Graphics*, tomo 5531, pp 56–67. 2009. 55

[Cho10] Choudhury, S. y Breslin, J.G. Enriching Videos with Light Semantics. En *SEMAPRO 2010, The Fourth International Conference on Advances in Semantic Processing*, pp 126–131. 2010. 25, 42

[Cn09] Campaña, J. R., Martín-Bautista, M. J., Medina, J. M., y Vila, M. A. Semantic enrichment of database textual attributes. *Flexible Query Answering Systems*, pp 488–499, 2009. 2

[Cn11] Campaña, J.R. *Representación y Tratamiento Semántico de Información Imprecisa en Bases de Datos*. Tesis Doctoral, Department of Computer Sciencie and Artificial Intelligence. University of Granada, Spain, 2011. 204

[Cui10] Cui, W., Wu, Y., Liu, S., Wei, F., Zhou, M.X., y Qu, H. Context Preserving Dynamic Word Cloud Visualization. En *Pacific Visualization Symposium (PacificVis), 2010 IEEE*, pp 121–128. IEEE, 2010. 24

[DC08] Di Caro, L., Candan, K.S., y Sapino, M.L. Using tagFlake for Condensing Navigable Tag Hierarchies from Tag Clouds. En *Proceedings of the 14th International Conference on Knowledge Discovery and Data Mining*, pp 1069–1072. ACM, 2008. 55

[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*, pp 213–222. ACM, 2007. 25, 43, 65

[Dur12] Durao, F., Dolog, P., Leginus, M., y Lage, R. Simspectrum: A similarity based spectral clustering approach to generate a tag cloud. *Current Trends in Web Engineering*, pp 145–154, 2012. 25, 26, 33, 52, 231

[Fel98] Fellbaum, C. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA., 1998. 287, 288

[For01] Money Makes the World Go Round. *Fortune Magazine*, 2001. Http://money.cnn.com/magazines/fortune/. 22

[Fra00] 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. 44

[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*, pp 1087–1088. ACM, 2008. 34, 55

[Gia08] Giannakidou, E., Koutsonikola, V., Vakali, A., y Kompatsiaris, Y. Co-Clustering Tags and Social Data Sources. En *The 9th International*

Conference on Web-Age Information Management., pp 317–324. IEEE, 2008. 51

[Gim04] Giménez, J. y Márquez, L. SVMTool: A General POS Tagger Generator Based on Support Vector Machines. En *In Proceedings of the 4th International Conference on Language Resources and Evaluation*. Citeseer, 2004. 192

[Gol09] Golub, K., Moon, J., Tudhope, D., Jones, C., Matthews, B., PuzoD, B.B., y Lykke Nielsen, M. EnTag: Enhancing Social Tagging for Discovery. En *Proceedings of the 9th ACM/IEEE-CS joint conference on Digital libraries*, pp 163–172. ACM, 2009. 17

[Gra07] Grahl, M., Hotho, A., y Stumme, G. Conceptual Clustering of Social Bookmarking Sites. En *Proceedings of I-KNOW*, tomo 7, pp 5–7. 2007. 25

[Gru07] Gruber, T. Ontology of Folksonomy: A Mash-up of Apples and Oranges. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 3(1):1–11, 2007. 53

[GS09] García-Silva, A., Szomszor, M., Alani, H., y Corcho, O. Preliminary Results in Tag Disambiguation using DBpedia. En *Proceedings of the 1st International Workshop in Collective Knowledge Capturing and Representation*, pp 1–8. 2009. 51

[GS12] García-Silva, A., Corcho, O., Alani, H., y Gómez-Pérez, A. Review of the State of the Art: Discovering and Associating Semantics to Tags in Folksonomies. *The Knowledge Engineering Review*, 27(01):57–85, 2012. 49, 50

[Hah11] Hahmann, S. y Burghardt, D. Maple– a Web Map Service for Verbal Visualisation using Tag Clouds Generated from Map Feature Frequencies. *Advances in Cartography and GIScience. Volume 1*, pp 3–12, 2011. 23

[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*, pp 1314–1315. ACM, 2007. 34

[Han00] Han, Jiawei, Pei, Jian, y Yin, Yiwen. Mining frequent patterns without candidate generation. En *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pp 1–12. ACM, 2000. 154

[Hea08] Hearst, M.A. y Rosner, D. Tag clouds: Data Analysis Tool or Social Signaller? En *Hawaii International Conference on System Sciences (HICSS)*, pp 160–169. IEEE Computer Society, 2008. 27, 34, 64

[Hel10] Helic, D., Trattner, C., Strohmaier, M., y Andrews, K. On the Navigability of Social Tagging Systems. En *Proc. of 2010 IEEE International Conference on Social Computing*, pp 161–168. 2010. 30

[Hel11] Helic, D., Trattner, C., Strohmaier, M., y Andrews, K. Are Tag Clouds Useful for Navigation? A Network-Theoretic Analysis. *International Journal of Social Computing and Cyber-Physical Systems*, 1(1):33–55, 2011. 30

[Hey06] Heymann, P. y Garcia-Molina, H. Collaborative Creation of Communal Hierarchical Taxonomies in Social Tagging Systems. *Infolab: Technical Report. University of Stanford*, 2006. 25, 51

[Hip00] Hipp, Jochen, Güntzer, Ulrich, y Nakhaeizadeh, Gholamreza. Algorithms for association rule mining - a general survey and comparison. *SIGKDD Explor. Newsl.*, 2:58–64, 2000. 154, 182

[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*, pp 25–28. Citeseer, 2006. 14, 15, 16, 19, 29, 34, 41, 52, 63

[HM10] Hassan Montero, Y., Herrero-Solana, V., y Guerrero-Bote, V. Usabilidad de los Tag-Clouds: Estudio mediante Eye-Tracking. *Scire: representación y organización del conocimiento*, 16(1):15–33, 2010. 27, 34, 37, 65, 66

[How09] Howard, H. Knowledge Discovery in Databases. *Online Notes. Computer Science. University of Regina*, 2009. 115

[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. 18, 25, 53, 66

[Hsi12] Hsieh, C.C. y Cho, J. Finding Similar Items by Leveraging Social Tag Clouds. En *Annual ACM Symposium on Applied Computing (SAC)*. 2012. 15, 24

[Kap10] Kaptein, R. y Marx, M. Focused Retrieval and Result Aggregation with Political Data. *Information Retrieval*, 13(5):412–433, 2010. 25, 40, 44, 66, 197, 216, 219

[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. 35

[Ker06] Kerr, B. TagOrbitals: a Tag Index Visualization. En *Proceedings of the 37th International Conference and Exhibition on Computer Graphics and Interactive Techniques*, pp 158–162. 2006. 56

[Kha07] Khancome, C. y Boonjing, V. String matching using inverted list. En *The 1st Conference on Graduate Research, SRRU*, pp 104–115. Citeseer, 2007. 161, 162

[Kim10] Kim, H.L., Decker, S., y Breslin, J.G. Representing and Sharing Folksonomies with Semantics. *Journal of Information Science*, 36(1):57–72, 2010. 53

[Kim11] Kim, K.T., Ko, S., Elmqvist, N., y Ebert, D.S. WordBridge: Using Composite Tag Clouds in Node-Link Diagrams for Visualizing Content and Relations in Text Corpora. En *System Sciences (HICSS), 2011 44th Hawaii International Conference*, pp 1–8. IEEE, 2011. 23

[Kna10] Knautz, K., Soubusta, S., y Stock, W.G. Tag Clusters as Information Retrieval Interfaces. En *Hawai International Conference on System Sciencies (HICSS)*, pp 1–10. IEEE Computer Society, 2010. 37

[Koh10] Koh, K., Lee, B., Kim, B., y Seo, J. Maniwordle: Providing Flexible Control over Wordle. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1190–1197, 2010. 35

[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*, pp 1132–1135. ACM, 2009. 8, 21, 45, 46, 47, 48, 65, 283

[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*, pp 391–402. ACM, 2009. 21, 45, 47, 48, 65

[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*, pp 1204–1205. ACM, 2007. 15, 24, 39, 66

[Leo11] Leone, S., Geel, M., Müller, C., y Norrie, M.C. Exploiting Tag Clouds for Database Browsing and Querying. *Information Systems Evolution*, pp 15–28, 2011. 28, 45, 49, 65

[Les86] Lesk, M. Automatic Sense Disambiguation using Machine Readable Dictionaries: How to Tell a Pine Cone from an Ice Cream Cone. En

Proceedings of the 5th Annual International Conference on Systems Documentation, pp 24–26. ACM, 1986. 193

[Lin02] Lin, M.Y. y Lee, S.Y. Fast discovery of sequential patterns by memory indexing. *Data Warehousing and Knowledge Discovery*, pp 227–237, 2002. 156

[Loh09] Lohmann, S., Ziegler, J., y Tetzlaff, L. Comparison of Tag Cloud Layouts: Task-Related Performance and Visual Exploration. En *Proceedings of the 12th International Conference on Human-Computer Interaction: Part I*, pp 392–404. 2009. 34

[Mar93] Marcus, M., Marcinkiewicz, M., y Santorini, B. Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330, 1993. 205, 296

[Mar06] Marín, N., Martín-Bautista, M. J., Prados, M., y Vila, M. A. Enhancing Short Text Retrieval in Databases. *Flexible Query Answering Systems*, pp 613–624, 2006. 56, 57

[Mar12] Marinho, L.B., Hotho, A., Jáschke, R., Nanopoulos, A., Rendle, S., Schmidt-Thieme, L., Stumme, G., y Symeonidis, P. Social Tagging Systems. *Recommender Systems for Social Tagging Systems*, pp 3–15, 2012. 14, 16

[Mas12] Masada, T., Takasu, A., Shibata, Y., y Oguri, K. Clustering Documents with Maximal Substrings. *Enterprise Information Systems*, 102:19–34, 2012. 230

[MB06] Martín-Bautista, M. J., Prados, M., Vila, M. A., y Martínez-Folgoso, S. A Knowledge Representation for Short Texts Based on Frequent Itemsets. En *Proceedings of the 11th Conference of Information Processing and Management of Uncertainty (IPMU), Paris*, pp 1065–1070. 2006. 4

[MB08] Martín-Bautista, M. J., Vila, M. A., y Martínez-Folgoso, S. A New Semantic Representation for Short Texts. En *Data Warehousing and Knowledge Discovery*, tomo 5182, pp 347–356. 2008. 4, 56, 59

[MF08] Martínez-Folgoso, S. *Una Solución Semántica al Tratamiento de Atributos Textuales en un Modelo Relacional Orientado a Objetos: Implementación en Software Libre*. Tesis Doctoral, Department of Computer Sciencie and Artificial Intelligence. University of Granada, Spain, 2008. 4, 56, 57, 67, 72, 77, 206, 247, 248, 282

[Mil76] Milgram, S. y Jodelet, D. Psychological Maps of Paris. *Environmental Psychology*, pp 104–124, 1976. 22

[Mor06] Morville, P. y Rosenfeld, L. *Information Architecture for the World Wide Web*. O'Reilly Media, 2006. 34

[Mor12] Morik, K., Kaspari, A., Wurst, M., y Skirzynski, M. Multi-Objective Frequent Termset Clustering. *Knowledge and information systems*, 30(3):715–738, 2012. 26, 33

[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)*, pp 1917–1920. 2006. 24, 25, 43, 66

[Pap10] Papadopoulos, S., Kompatsiaris, Y., y Vakali, A. A Graph-Based Clustering Scheme for Identifying Related Tags in Folksonomies. *Data Warehousing and Knowledge Discovery*, pp 65–76, 2010. 26

[Pat11] Patil, M., Thankachan, S.V., Shah, R., Hon, W.K., Vitter, J.S., y Chandrasekaran, S. Inverted indexes for phrases and strings. En *Proceedings of the 34th International ACM SIGIR Conference on Research And Development in Information Retrieval*, pp 555–564. ACM, 2011. 161

[Pei01] Pei, J., Han, J., Mortazavi-Asl, B., Pinto, H., Chen, Q., Dayal, U., y Hsu, M.C. Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth. En *Proceedings of the 17th International Conference on Data Engineering*, pp 215–224. 2001. 156

[Por80] Porter, M. An Algorithm for Suffix Stripping. *Program: Electronic Library and Information Systems*, 14(3):130–137, 1980. 188

[Pro08] Provost, J. Improved Document Summarization and Tag Clouds via Singular Value Decomposition. 2008. 37

[Qia12] Qiao, M. y Zhang, d. Efficiently matching frequent patterns based on bitmap inverted files built from closed itemsets. *International Journal on Artificial Intelligence Tools*, 21(03), 2012. 160

[Rab12] Raban, D.R., Danan, A., Ronen, I., y Guy, I. Impression Formation in Corporate People Tagging. En *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems*, pp 569–578. 2012. 18

[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)*, pp 998–1001. ACM, 2007. 15, 27, 35

[Sal75] Salton, G., Wong, A., y Yang, C.S. A Vector Space Model for Automatic Indexing. *Communications of the ACM*, 18(11):613–620, 1975. 29

[Sav95] Savasere, Ashoka, Omiecinski, Edward, y Navathe, Shamkant B. An efficient algorithm for mining association rules in large databases. En *Proceedings of the 21th International Conference on Very Large Data Bases*, pp 432–444. 1995. 153, 182

[Sch99] Schmid, H. Improvements in Part-of-Speech Tagging with an Application to German. *Natural Language Processing using Very Large Corpora*, 11:13–26, 1999. 192

[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*, pp 2037–2040. ACM, 2009. 34

[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. 22, 23, 55

[Sim10] Simon, R., Sadilek, C., Korb, J., Baldauf, M., y Haslhofer, B. Tag Clouds and Old Maps: Annotations as Linked Spatiotemporal Data in the Cultural Heritage Domain. p 12, 2010. 23

[Sin08] Sinclair, J. y Cardew-Hall, M. The Folksonomy Tag Cloud: When is it Useful? *Journal of Information Science*, 34:15–30, 2008. 15, 18, 28, 29, 30, 40, 63

[Sko11] Skoutas, D. y Alrifai, M. Tag Clouds Revisited. En *Proceedings of the 20th ACM international conference on Information and knowledge management*, pp 221–230. ACM, 2011. 33, 38, 39, 66

[Spe07] Specia, L. y Motta, E. Integrating Folksonomies with the Semantic Web. En *Proceedings of the 4th European Conference on the Semantic Web: Research and Applications*, pp 624–639. 2007. 50, 51

[Sri96] Srikant, R. y Agrawal, R. Mining sequential patterns: Generalizations and performance improvements. *Advances in Database Technology*, pp 1–17, 1996. 156

[Str10] Strohmaier, M., Körner, C., y Kern, R. Why Do Users Tag? Detecting Users'Motivation for Tagging in Social Tagging Systems. En *International AAAI Conference on Weblogs and Social Media (ICWSM2010), Washington, DC, USA*, pp 339–342. 2010. 40

[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, pp 661–666. ACM, 2003. 114

[Tom94] Tomasic, A., Garcia-Molina, H., y Shoens, K. *Incremental Updates of Inverted Lists for Text Document Retrieval*, tomo 23. ACM, 1994. 159

[Tou00] Toutanova, K. y Manning, C. Enriching the Knowledge Sources used in a Maximum Entropy Part-of-Speech Tagger. En *Proceedings of the 2000 Joint SIGDAT conference on Empirical methods in Natural Language Processing and Very Large Corpora: Held in Conjunction with the 38th Annual Meeting of the Association for Computational Linguistics*, tomo 13, pp 63–70. Association for Computational Linguistics, 2000. 204, 296

[Tou03] Toutanova, K., Klein, D., Manning, C., y Singer, Y. Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. En *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, tomo 1, pp 173–180. Association for Computational Linguistics, 2003. 192, 204, 296

[TP12] Torres-Parejo, U., Campaña, J. R., Vila, M. A., y Delgado, M. Text Retrieval and Visualization in Databases Using Tag Clouds. En *Advances in Computational Intelligence.*, tomo 297 de *Communications in Computer and Information Science*, pp 390–399. Springer, 2012. 66

[TP13a] Torres-Parejo, U., Campaña, J.R., Delgado, M., y Vila, M.A. MTCIR: A Multi-Term Tag Cloud Information Retrieval System. *Expert Systems with Applications*, 40:5448–5455, 2013. 214, 272, 277, 281

[TP13b] Torres-Parejo, U., Campaña, J.R., Vila, M.A., y Delgado, M. A Theoretical Model for the Automatic Generation of Tag Clouds. *Knowledge and Information Systems*, 651:1–33, 2013. 147, 213, 216, 231, 233, 276, 281

[Tra11] Trattner, C., Helic, D., y Strohmaier, M. On the Construction of Efficiently Navigable Tag Clouds Using Knowledge From Structured Web Content. *Journa of Universal Computer Science*, 17(4):565–582, 2011. 26, 30, 53

[Vas04] Vasilescu, F., Langlais, P., y Lapalme, G. Evaluating Variants of the Lesk Approach for Disambiguating Words. En *Proceedings of the Conference of Language Resources and Evaluations (LREC)*, pp 633–636. 2004. 193, 205

[Ven11] Venetis, P., Koutrika, G., y Garcia-Molina, H. On the Selection of Tags for Tag Clouds. En *Proceedings of the fourth ACM international conference on Web search and data mining*, pp 835–844. ACM, 2011. 31, 33, 231, 258, 272

[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. 24, 35

[Vié08] Viégas, F. B. y Wattenberg, M. TIMELINES Tag clouds and the case for vernacular visualization. *Interactions*, 15:49–52, 2008. 20, 22, 24

[Vié09] Viégas, F. B., Wattenberg, M., y Feinberg, J. Participatory Visualization with Wordle. *IEEE Transactions on Visualization and Computer Graphics*, 15:1137–1144, 2009. 24, 35

[Wan10] Wang, J., Clements, M., Yang, J., de Vries, A.P., y Reinders, M.J.T. Personalization of Tagging Systems. *Information processing & management*, 46(1):58–70, 2010. 16

[Wat08] Watters, D. y Chicago, IL. Meaningful Clouds: Towards a Novel Interface for Document Visualization. *Online Notes. University of Chicago*, 2008. 25, 44

[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*, pp 426–435. ACM, 2006. 18

[Xex09] Xexéo, G., Morgado, F., y Fiuza, P. Automatically Generated Tag Clouds. *XXIV Simpósio Brasileiro de Banco de Datos*, 2009. 3

[Yag10] Yager, R. y Reformat, M. Tagging and Fuzzy Sets. *Intelligent Systems: From Theory to Practice*, pp 1–17, 2010. 14

[Yu06] Yu, T.Z. y Chien, L.F. *Automatic Organization of User-Generated Tags from the Web*. Tesis Doctoral, National Taiwan University, 2006. 25

[Zak97] Zaki, M. J., Parthasarathy, S., Ogihara, M., y Li, W. New algorithms for fast discovery of association rules. En *3rd Intl. Conf. on Knowledge Discovery and Data Mining*, tomo 20, pp 283–286. 1997. 154, 182

[Zak01] Zaki, M.J. Spade: An efficient algorithm for mining frequent sequences. *Machine Learning*, 42(1):31–60, 2001. 156, 182

[Zha03] Zhao, Q. y Bhowmick, S.S. Sequential pattern mining: A survey. *ITechnical Report CAIS Nayang Technological University Singapore*, pp 1–26, 2003. 157

[Zub12] Zubiaga, A. Enhancing Navigation on Wikipedia with Social Tags. *Arxiv preprint arXiv:1202.5469*, 2012. 17, 24