



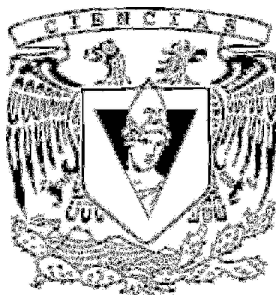
UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO

FACULTAD DE CIENCIAS

**“Conquiro: Un metabuscador basado en
clustering”**

T E S I S
QUE PARA OBTENER EL TÍTULO DE:
LICENCIADA EN CIENCIAS DE LA COMPUTACIÓN

P R E S E N T A:
YESICA YADIRA CASTELLANOS MEDINA



FACULTAD DE CIENCIAS
UNAM

Tutora : DRA. MARÍA DEL SOCORRO VARGAS VERA

Asesor : DR. PEDRO EDUARDO MIRAMONTES VIDAL

2006



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

1. Datos del alumno
Castellanos
Medina
Yesica Yadira
26 17 30 15
Universidad Nacional Autonoma de Mexico
Facultad de Ciencias
Ciencias de la Computación
094190339

2. Datos de tutor
Dra
María del Socorro
Vargas
Vera

3. Datos del sinodal 1
Dr
Pedro Eduardo
Miramontes
Vidal

4. Datos del sinodal 2
M en IO
María de Luz
Gasca
Soto

5. Datos del sinodal 3
M en C
María Guadalupe Elena
Ibargüengoitia
González

6. Datos del sinodal 4
Dra
Sofía Natalia
Galicía
Haro

7. Datos del trabajo escrito
Conquiro: Un metabuscador basado en clustering
137 p
2006

Agradecimientos

Quiero expresar mi gratitud a todas las personas que enriquecieron este trabajo con sus comentarios e ideas. En especial quiero agradecer a la doctora María Vargas-Vera su paciencia, sus enseñanzas y consejos, ya que gracias a ella aprendí el trabajo que se necesita realizar en un proyecto de investigación.

Por otro lado, quiero agradecer a Dawid Weiss por haber respondido a todas mis preguntas; y a Daniel Crabtree por haberme proporcionado su base de ejemplos.

A mis padres y a mi esposo Eliot.

Contenido

| | |
|---|----|
| Agradecimientos | 7 |
| Prefacio | 9 |
| Introducción | 11 |
| Motivación | 11 |
| Objetivos | 13 |
| Contribuciones | 13 |
| Resumen | 14 |
| Organización | 14 |
| Convenciones utilizadas | 15 |
| Capítulo 1. Búsqueda de información en la Web | 17 |
| 1.1 Motores de búsqueda | 17 |
| 1.2 Directorios de la Web | 20 |
| 1.3 Metabuscadores | 22 |
| 1.4 Agrupamiento de los resultados de la búsqueda (<i>clustering</i>) | 24 |
| 1.4.1 Scatter / Gather | 25 |
| 1.4.2 Grouper | 27 |
| 1.4.3 Carrot 2 | 28 |
| 1.4.4 Vivísimo | 29 |
| 1.4.5 Comparación de los sistemas analizados | 31 |
| Capítulo 2. Agrupamiento (<i>clustering</i>) de documentos | 33 |
| 2.1 Definición general | 33 |
| 2.2 Agrupamiento (<i>clustering</i>) en IR | 34 |
| 2.3 Modelo de espacio de vectores (VSM) y la representación de documentos | 35 |
| 2.3.1 Limpieza de los documentos | 36 |
| 2.3.2 Asignar pesos a los términos | 37 |
| 2.3.3 Medidas de semejanza | 39 |
| 2.3.4 Reducción de la dimensionalidad | 40 |
| 2.3.5 Normalización de los vectores | 41 |
| 2.3.6 Ejemplo utilizando el modelo VSM | 41 |
| 2.4 Algoritmos de <i>clustering</i> (algoritmos de agrupamiento) | 44 |
| 2.4.1 Algoritmos de Partición | 44 |
| 2.4.2 Algoritmos de Traslape (<i>Overlapping</i>) | 49 |
| 2.4.3 Algoritmos jerárquicos | 52 |
| 2.5 Métodos para etiquetar los grupos | 64 |
| Capítulo 3. El metabuscador Conquiro | 69 |
| 3.1 Introducción | 69 |
| 3.2 Características de Conquiro | 69 |
| 3.3 Arquitectura de Conquiro | 71 |
| 3.3.1 Interfaz | 72 |
| 3.3.2 Módulo de procesamiento de la consulta | 74 |
| 3.3.3 Módulo de procesamiento de documentos | 75 |
| 3.3.4 Módulo de <i>clustering</i> (agrupamiento) | 76 |
| 3.3.5 Módulo de generación de resultados | 77 |
| 3.3.6 Módulo de visualización de resultados | 77 |
| 3.3.7 La respuesta de Conquiro a una consulta de usuario, paso a paso | 79 |

| | |
|--|-----|
| Capítulo 4. Evaluación y experimentos | 93 |
| 4.1 Métodos de Evaluación | 93 |
| 4.1.1 <i>Precision y recall</i> | 94 |
| 4.1.2 <i>Gold Standard</i> | 96 |
| 4.1.3 <i>Evaluación de usuarios</i> | 98 |
| 4.2 Caso de estudio 1: Documentos de la Web | 98 |
| 4.2.1 <i>Experimentos</i> | 100 |
| 4.2.2 <i>Resultados</i> | 101 |
| 4.3 Caso de estudio 2: <i>Newsgroups</i> | 111 |
| 4.3.1 <i>Datos</i> | 111 |
| 4.3.2 <i>Evaluación de usuarios</i> | 111 |
| 4.3.3 <i>Resultados</i> | 112 |
| | |
| Capítulo 5. Conclusiones y trabajo futuro | 117 |
| 5.1 Conclusiones | 117 |
| 5.2 Trabajo futuro | 118 |
| | |
| Apéndices | 121 |
| A. Palabras Comunes (<i>stop words</i>) | 121 |
| B. <i>Gold Standard</i> de las consultas realizadas a Conquiro | 124 |
| <i>Consulta “star”</i> | 124 |
| <i>Consulta “jaguar”</i> | 125 |
| <i>Consulta “salsa”</i> | 126 |
| <i>Consulta “apple”</i> | 126 |
| C. Evaluación de usuario | 127 |
| | |
| Bibliografía | 133 |

Prefacio

El rápido crecimiento en la cantidad de información disponible a través de la Internet provoca que las herramientas para buscar información estén en constante mejora; de ahí que en estos últimos años se han realizado diversos trabajos, en los cuales se han propuesto diversas soluciones para mejorar estas herramientas y así permitir a los usuarios buscar de manera eficiente cualquier tipo de información (texto, imágenes, sonidos y video).

El objetivo de este trabajo es diseñar un sistema que permita a los usuarios buscar información sin necesidad de examinar grandes listas de documentos. Este sistema es un metabuscador llamado *Conquiro*, el cual aplica la técnica de agrupamiento (*clustering*) a los resultados de un motor de búsqueda.

Introducción

Motivación

Desde su nacimiento en el año de 1990 la *World Wide Web* (Web) ha tenido un crecimiento exponencial, hasta convertirse en el espacio de información más grande del mundo. La búsqueda de información relevante en la Web se ha convertido en una tarea difícil, debido a que ésta ha tenido un crecimiento desmedido y a que las personas pueden publicar cualquier tipo de información en la Web, lo cual dificulta al usuario llegar a la información que está buscando.

Las herramientas más comúnmente usadas para buscar información en la Web son los motores de búsqueda y los directorios. Los motores de búsqueda permiten al usuario hacer una consulta y aquéllos la responden con un conjunto de referencias a documentos (páginas de la Web). Por otro lado, los directorios son colecciones de referencias a documentos de la Web, las cuales fueron organizadas manualmente en una jerarquía de categorías. El usuario puede buscar información, explorando las categorías hasta encontrar lo que necesita.

Aunque el desempeño de los motores de búsqueda mejora día con día, la búsqueda de información puede ser tediosa, por ejemplo, Zamir [Zamir, 1999] menciona las razones por las cuales buscar información con los motores de búsqueda no siempre es exitoso:

- 1) Los motores de búsqueda ordenan los resultados de la búsqueda de acuerdo con la relevancia que tengan con la consulta. Estos esquemas de ordenamiento (*ranking*) trabajan bien cuando el usuario formula consultas bien definidas. Si embargo, los usuarios formulan a menudo consultas muy cortas o ambiguas, lo cual ocasiona que el motor de búsqueda regrese una gran cantidad de documentos que no son de su interés.
- 2) Como consecuencia del punto anterior, los resultados que regresan los motores de búsqueda contienen miles o millones de documentos.

Para ayudar al usuario a visualizar rápidamente la información relevante obtenida de una consulta a un motor de búsqueda, se propone construir un sistema que agrupe los documentos por temas (tópicos), usando la técnica de *agrupamiento* (*clustering*). El *agrupamiento* de documentos en recuperación de la información (*information retrieval*) consiste en encontrar un rubro para un conjunto de documentos de tal manera que los que pertenecen al mismo *grupo* (*cluster*) son similares entre sí y diferentes de los que pertenecen a grupos distintos.

La agrupación de los resultados en *grupos* (*clusters*) permite al usuario explorar un conjunto grande de documentos eficientemente y además, con una descripción apropiada de los *grupos* (*clusters*), el usuario podrá identificar el tema de su interés de manera rápida. Los *grupos* podrían eventualmente contener grupos más pequeños, los cuales a su vez contendrían más grupos y así sucesivamente; este tipo de *agrupamiento*

es llamado *jerárquico* (*hierarchical clustering*). Si por el contrario los *grupos* no contienen otros, entonces se le llama *agrupamiento no jerárquico* (*non-hierarchical clustering*).

Las ventajas de utilizar la técnica de agrupamiento son:

- El número de grupos es menor que el número de documentos que cada uno de ellos contiene. Esto permite explorar los resultados de manera rápida.
- La técnica de *agrupamiento* muestra los diferentes temas de la colección de documentos en grupos separados, lo cual permite al usuario examinar los grupos que estén relacionados con los temas que le interesan.

El trabajo previo para aplicar la técnica de *agrupamiento* (*clustering*) de una colección de documentos [Pirolli *et al.*, 1996; Zamir y Etzioni, 1999; Weiss, 2001] se ha enfocado en los siguientes puntos:

- Colocar en un mismo *grupo* (*cluster*) los documentos similares.
- Presentar los *grupos* (*clusters*) de documentos de tal forma que el usuario pueda encontrar de manera rápida la información que necesita.

De este trabajo previo han surgido sistemas importantes como: *Scatter/Gather*, *Grouper*, *Carrot2* y *Visisimo*. De los cuales sólo *Carrot2* tiene código abierto. Sin embargo, se decidió desarrollar el sistema *Conquiroy*¹ por las siguientes razones:

- 1) Tener un sistema que permita evaluar la eficiencia de los algoritmos² de *agrupamiento*, como *K-Means*, *Bisecting K-Means*, *HAC (single)*, *HAC (complete)*, *HAC (complete) dendrogram pruning* y *Suffix Tree Clustering*, para organizar una colección de documentos.
- 2) Tener un sistema donde se pueda evaluar la eficacia de diferentes métodos que construyan descripciones del contenido de los grupos. Esta parte es muy importante, ya que una buena descripción ayuda al usuario a identificar el tema del grupo.
- 3) En un futuro se desea extender el sistema, agregándole las siguientes funciones:
 - Buscar y agrupar información en varios idiomas.
 - Agrupar por temas los mensajes de un grupo de noticias (*newsgroup*).
 - Organizar la información de una base externa de documentos.
 - Utilizar los mejores métodos para crear las descripciones de los grupos.
 - Realizar tanto *agrupamiento* no supervisado (*unsupervised clustering*) como supervisado (*supervised clustering*)
 - Utilizar la técnica de *expansión de consultas* (*query expansion*) para refinar la consulta del usuario.

¹ Se optó por este nombre porque en latín significa “buscar”.

² Un algoritmo es una secuencia finita de pasos que nos llevan a la solución de un problema en un tiempo determinado.

- Enviar la consulta del usuario a varios motores de búsqueda.

Es importante decir que cuando el sistema fue diseñado aún no se tenía conocimiento de *Carrot2*, por lo cual este proyecto no utilizó su *framework*.

Objetivos

Esta tesis presenta el sistema *Conquiro* como una herramienta para que el usuario busque información de manera eficiente; tiene como principal componente un conjunto de algoritmos de *agrupamiento*, probados con documentos de la Web para determinar su eficiencia, es decir, cuál o cuáles algoritmos producen mejores resultados.

Otro componente importante del sistema es un conjunto de métodos, los cuales crean descripciones de los grupos de documentos. Estos métodos fueron probados para determinar su eficacia, es decir, determinar cuál o cuáles métodos crean descripciones de los grupos que ayuden al usuario a identificar los temas de los grupos.

Se considera que *Conquiro* no sólo debe ser una herramienta que organice documentos de la Web, sino también que pueda organizar por temas de discusión los mensajes de un *newsgroup*. Por esta razón se decidió probar la eficiencia de los algoritmos de *agrupamiento* para agrupar los mensajes de un *newsgroup*.

Los algoritmos de *agrupamiento* (*clustering*) que *Conquiro* utiliza son: *K-Means*, *Bisecting K-Means*, *HAC (single)*, *HAC (complete)*, *HAC (complete) dendrogram pruning*, *HAC (UPGMA)* y *Suffix Tree Clustering*.

Contribuciones

En este trabajo se desarrolló *Conquiro*, el cual es un sistema que agrupa la información obtenida de un motor de búsqueda por temas. Para realizar esta agrupación *Conquiro* utiliza diversos algoritmos de *agrupamiento* (*clustering*) jerárquico y no jerárquico. Los algoritmos de *agrupamiento* jerárquico son: *Bisecting K-Means*, *HAC (single)*, *HAC (complete)*, *HAC (complete) dendrogram pruning* y *HAC (UPGMA)*. Los algoritmos de *agrupamiento* no jerárquico son: *K-Means* y *Suffix Tree Clustering*.

Este trabajo contribuirá a:

1. Encontrar los mejores algoritmos de *agrupamiento*. Esto se realizó mediante la comparación de diversos algoritmos de *agrupamiento* (*clustering*) con base en las medidas de *precision* y *recall* (evaluación cuantitativa), las cuales están definidas en la sección 4.1.1. De esta comparación se encontró que los tres mejores algoritmos son: *HAC (dendrogram pruning)*, *Suffix Tree Clustering* y *Bisecting K-Means*.

2. Comparar los diversos métodos para construir las etiquetas de los grupos con base en una evaluación de usuario (evaluación cualitativa, la cual está descrita en la sección 4.1) de los siguientes métodos: *Inverse Document Frequency* (Frecuencia Inversa de Documentos), *Frequent and Predictive Words* (Palabras Frecuentes y Predictivas), *Phrases* (Frasas) y *Common Term in the Cluster* (Término Común en el Grupo), el cual fue desarrollado para este trabajo. Hasta donde sabemos no se ha creado un método para construir etiquetas de los grupos como el método *Common Term in the Cluster*.

Resumen

Conquiro fue desarrollado como un sistema experimental para probar varios algoritmos de *agrupamiento* y ver su efectividad en diferentes casos de estudio (documentos de la Web y *newsgroups*). Además de estos algoritmos, *Conquiro* también utiliza diversos métodos para crear las descripciones de los grupos, los cuales fueron evaluados para determinar su eficiencia. Una vez que se ha establecido tanto la efectividad de los algoritmos de *agrupamiento* como la eficiencia de los métodos que crean las descripciones de los grupos, se propone extender *Conquiro* agregándole otras características, como por ejemplo que busque y agrupe información en varios idiomas; que maneje métodos adicionales para crear las descripciones de los grupos, o que maneje más algoritmos de *agrupamiento*, entre otras mejoras las cuales pueden consultarse en la sección 5.2.

Entre los principales retos de investigación en un sistema de manejo de información como *Conquiro*, están los métodos que crean las descripciones de los grupos, los cuales son una de las partes más importantes de *Conquiro* debido a que la utilidad del sistema se verá aumentada si los grupos tienen una buena descripción. Por esta razón incluimos en las contribuciones el algoritmo llamado *Common Term in the Cluster*, el cual está descrito en la sección 2.5.

Otro de los retos enfrentados es contar con una metodología para evaluar el sistema que no esté basada en un *gold standard*. Esto se debe a que éste es construido por un humano y es una actividad que consume mucho tiempo, en particular cuando se está trabajando con colecciones grandes de documentos.

Finalmente, es indispensable que *Conquiro* tenga algoritmos de *agrupamiento* más eficientes, cuya complejidad sea lineal, debido a que esto incrementa la eficiencia del proceso de *agrupamiento*.

Organización

La parte restante de esta tesis está organizada de la siguiente manera. El capítulo 1 contiene una breve descripción de las herramientas que existen para buscar información en la Web y además se hace una comparación de los sistemas que aplican la técnica de *agrupamiento* (*clustering*) a una colección de documentos.

El capítulo 2 describe a detalle el proceso de *agrupamiento de documentos* (*document clustering*), abordando los siguientes temas: definición de *agrupamiento* (*clustering*), el *agrupamiento* (*clustering*) en IR, el *modelo de espacio de vectores* (*vector space model*), los algoritmos de *agrupamiento* (*clustering*) y los métodos para construir las etiquetas de los *clusters* (grupos).

En el capítulo 3 se presenta el sistema **Conquiro**. Se explica los motivos por los cuales se desarrolló y las características del sistema; se compara con los sistemas descritos en la sección 1.4 y, además, se describe la arquitectura de **Conquiro** explicando cada uno de los módulos que lo conforman; finalmente se presenta un ejemplo que muestra el proceso que realiza **Conquiro** para responder a una consulta de usuario.

El capítulo 4 explica los experimentos, los resultados y las evaluaciones para los siguientes casos de estudio:

- a) Agrupación de los resultados obtenidos de un motor de búsqueda.
- b) Agrupación de los mensajes de un *newsgroup*.

El capítulo 5 presenta las conclusiones de este trabajo y los retos para el futuro.

Convenciones utilizadas

Se han utilizado las siguientes convenciones para el texto:

- Letra *itálica* para poner énfasis en conceptos que están en otro idioma y en conceptos importantes manejados en el texto.
- Letra **negrita** cuando un nuevo término es definido.
- Las figuras y los cuadros están enumerados de acuerdo con el orden en el que aparecen en el capítulo, por ejemplo “Figura 2.8” denota la figura 8 del capítulo 2.

Capítulo 1. Búsqueda de información en la Web

La Web es un espacio global donde las personas comparten información de todo tipo (texto, música, imágenes, video). Ha tenido tanta popularidad que la cantidad de información que contiene crece muy rápidamente; a principios del año 2005 se estimó que contiene más de 11.5 billones de páginas indexadas por los motores de búsqueda [Gulli y Signorini, 2005], lo cual ocasiona que la búsqueda de información sea una tarea difícil.

Para resolver el problema de la búsqueda de información en la Web, se han creado diversas herramientas, que pueden buscar información en archivos de texto, patrones en imágenes, sonido y video. En las siguientes secciones se hablará del uso de estas herramientas para la búsqueda de información en archivos de texto.

1.1 Motores de búsqueda

Los motores de búsqueda, desde su aparición a mediados de la década de los noventa, se han convertido en la herramienta más usada para buscar información en la Web. Se utilizan de la siguiente manera: el usuario hace una consulta donde se especifica la información que desean recibir y el motor de búsqueda regresa un conjunto de resultados, los cuales contienen el título, una descripción breve del documento de la Web (*snippet*) y su URL (Uniform Resource Locator: dirección única que se asigna a cada uno de los recursos disponibles en Internet); la figura 1.1 muestra un ejemplo de la lista de resultados.

The screenshot shows a Google search interface with the search bar containing the word "star". Below the search bar, the results are listed. The first result is "Star Magazine - The #1 Celebrity News Magazine". The second result is "The Toronto Star", which is annotated with arrows pointing to its title, snippet, and URL. The third result is "The Star Online 10th Anniversary". The fourth result is "Star Trek". The fifth result is "Home : ENERGY STAR".

Figura 1.1 Lista de resultados producida por el motor de búsqueda

Los motores de búsqueda contienen las siguientes partes [Hu y Chen, 2001]:

- 1) La araña (*spider*)
- 2) El programa de indexación
- 3) El programa que busca y ordena la información.

La araña (*spider*)

Es un programa que automáticamente busca varios sitios web y recolecta los documentos para almacenarlos en una base de documentos. Los contenidos de éstos son examinados para ver si se encuentran nuevos urls que puedan ser utilizados como puntos para ser explorados. Para hacer más rápida la recolección de documentos de la Web, se envían varias arañas para que recorran varios sitios al mismo tiempo. Las arañas regresan a los sitios periódicamente para verificar si ha habido cambios.

El programa de indexación

Este programa examina la información de la base de documentos y construye una estructura (índice) donde se realicen búsquedas de manera eficiente.

El programa que busca y ordena la información

Este programa analiza la consulta y la compara con los índices para encontrar los documentos relevantes. Para determinar el orden en el que éstos serán presentados al usuario, el motor de búsqueda utiliza un algoritmo que haga este ordenamiento (algoritmo de *ranking*).

La figura 1.2 muestra la arquitectura general de un motor de búsqueda.

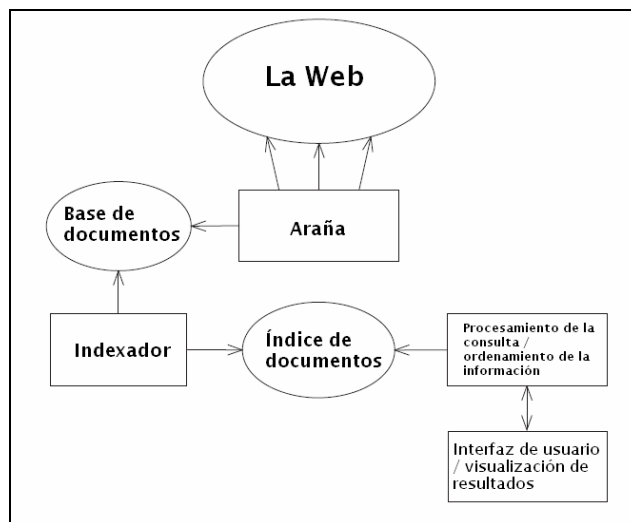


Figura 1.2 Arquitectura de un motor de búsqueda

Las ventajas y desventajas de los motores de búsqueda son [Netskills]:

- **Relevancia.** Los motores de búsqueda son buenos para obtener grandes cantidades de información; sin embargo ésta podría no ser relevante para nuestras necesidades, o de no tan buena calidad.

- **Búsqueda de información.** Los motores de búsqueda son una buena opción si se tiene una idea clara de la información que se está buscando.

Los motores de búsqueda más reconocidos son Google [Google, 2006], Yahoo [Yahoo, 2006], Ask Jeeves [Ask, 2006] y MSN [MSN, 2006]. Las figuras 1.3 y 1.4 muestran los resultados que Ask devuelve para la consulta “apple”; y los resultados que MSN devuelve para la consulta “salsa”, respectivamente.

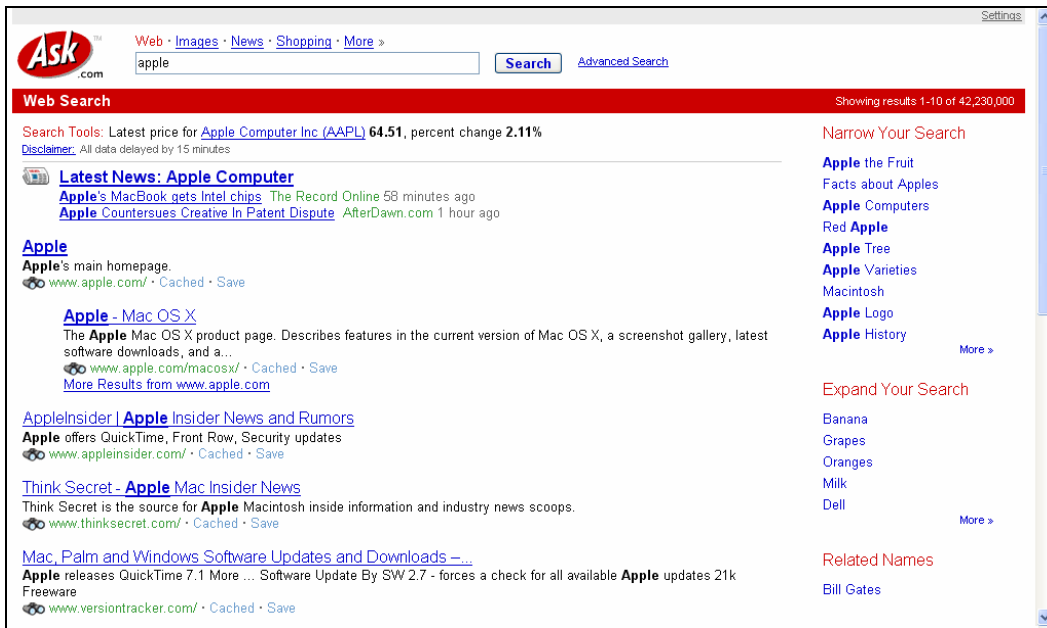


Figura 1.3 Resultados de Ask para la consulta “apple”.

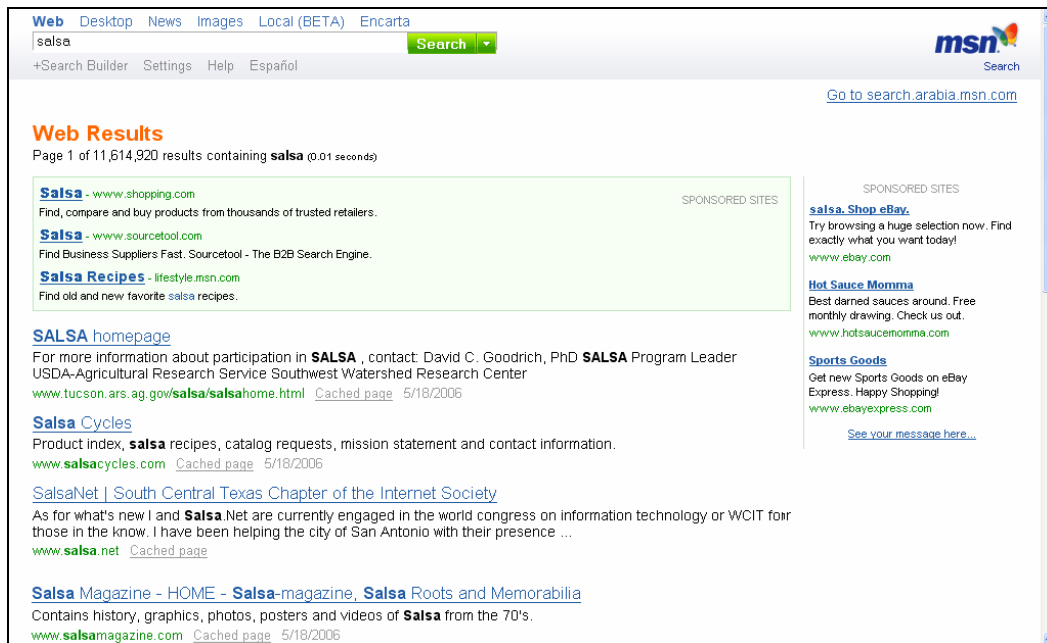


Figura 1.4 Resultados de MSN para la consulta “salsa”.

1.2 Directorios de la Web

Otra herramienta para buscar información son los directorios de la Web;¹ éstos surgieron, al igual que los motores de búsqueda, en la década de los noventa; se tiene conocimiento de que el primer directorio fue Yahoo (año de 1994).

Los directorios son largas listas de sitios web, organizadas en categorías. Esto permite a los usuarios hacer búsquedas por palabras o bien explorar la jerarquía de categorías. La diferencia entre los motores de búsqueda y los directorios es que la información que contienen los directorios es recopilada por personas, las cuales visitan los sitios e ingresan la información manualmente en la base de datos.

Las ventajas que tienen los directorios son las siguientes [Wroblewski, 2003]:

- La estructura de la información permite refinar la búsqueda, es decir, iniciar en las categorías más generales e ir moviéndose a categorías más específicas.
- Como las categorías son asignadas manualmente, la organización de los documentos es más adecuada que las herramientas que hacen organización automática. Además, la calidad de la información es alta porque la información pasó por un proceso de selección, en el cual se integraron los sitios cuya información se considera que aporta un valor real.
- No hay necesidad de formular consultas, ya que se puede explorar la estructura del directorio de manera fácil.

Las desventajas que tienen los directorios son las siguientes [Wroblewski, 2003]:

- La lenta indexación (debido al hecho de que se realiza manualmente).
- Bases de datos de tamaño pequeño. El cuadro 1.1 muestra los tamaños, es decir, el número de páginas o urls de las bases de los principales directorios.
- Los catálogos no están actualizados, debido a las razones expuestas en el primer punto.
- En la estructura predefinida del directorio podrían no considerarse temas muy específicos.

| Servicio | Editores | Tamaño de la base |
|----------------|---------------------|-------------------|
| Yahoo | Usuarios y editores | 3,000,000 |
| Open Directory | 59,000 editores | 3,800,000 |
| LookSmart | Selección | 2,300,000 |

Cuadro 1.1 Comparación de los directorios más usados [Search Engine Showdown, 2006]

¹ En adelante se hará referencia a ellos como directorios

Los directorios son una buena opción cuando la información buscada sea general, se quiera explorar la estructura del directorio, o bien si se necesita información de alta calidad. Los directorios más populares son Yahoo [Yahoo, 2006], dmoz [dmoz, 2006] y LookSmart [LookSmart, 2006]. Las figuras 1.5 y 1.6 muestran los resultados que devuelve el directorio de Yahoo para la consulta “apple”; y los resultados que dmoz devuelve para la consulta “salsa”, respectivamente.

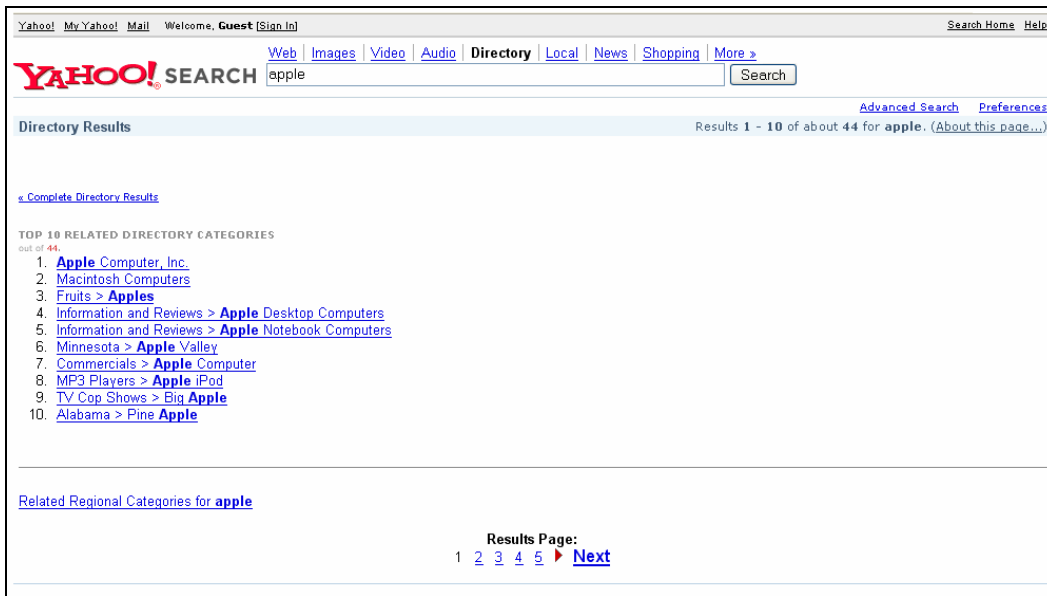


Figura 1.5 Resultados del directorio Yahoo para la consulta “apple”.

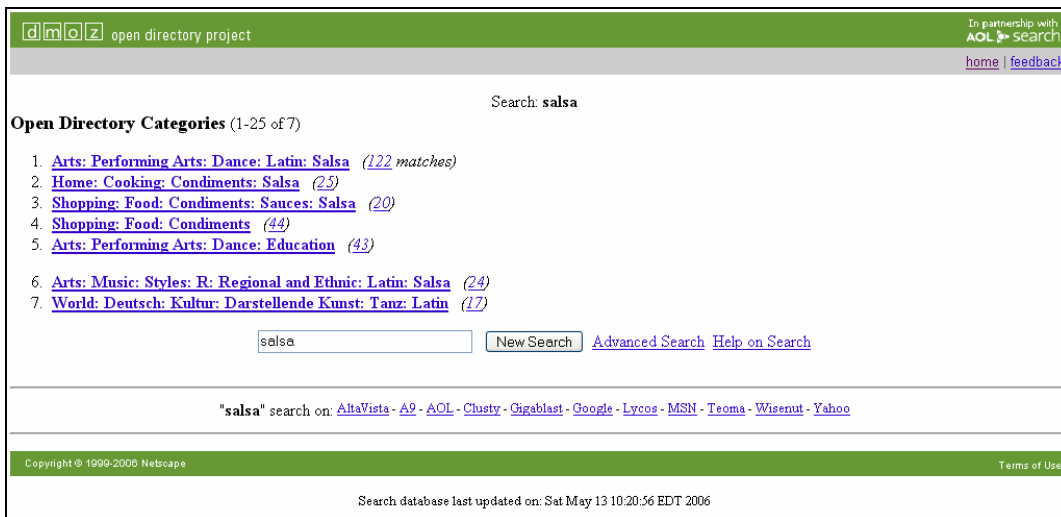


Figura 1.6 Resultados del directorio dmoz para la consulta “salsa”.

1.3 Metabuscadores

En la década de los noventa, cuando los usuarios hacían la misma consulta a los diferentes motores de búsqueda, éstos mostraban diferentes resultados, por lo cual no era fácil encontrar la información deseada. A raíz de esta situación surgió la necesidad de usar varios motores de búsqueda simultáneamente. En 1995 Erik Selberg, de la Universidad de Washington, desarrolló el primer metabuscador, llamado MetaCrawler, el cual buscaba información en los motores de búsqueda Lycos, Altavista, Yahoo, Excite, WebCrawler e Infoseek.

Un metabuscador es un sistema que toma una consulta de usuario y la envía a varios motores de búsqueda. Después el metabuscador recolecta las respuestas de los motores de búsqueda y presenta los resultados al usuario. Los metabuscadores también pueden buscar en otras fuentes de información como directorios o servicios de noticias.

Los metabuscadores permiten al usuario obtener información de diversas herramientas de búsqueda sin tener que consultar a cada una de forma individual. A diferencia de los motores de búsqueda, los metabuscadores no crean bases de datos propias, sino que utilizan las de otras herramientas de búsqueda. La figura 1.7 muestra la estructura de un metabuscador, que consta de tres partes [Hu y Chen, 2001]:

- Envío de consulta (*Dispatch*): Determina a qué motores de búsqueda será enviada la consulta.
- Interfaz (*Interface*): Adapta el formato de la consulta para que corresponda con el formato que el motor de búsqueda utiliza.
- Vista de resultados (*Display*): Integra los resultados regresados por los diferentes recursos consultados para ser mostrados al usuario.

Las ventajas de los metabuscadores son:

- Permiten al usuario buscar en varios recursos al mismo tiempo.
- Muestran resultados relevantes la mayoría de las veces, ya que remueven los que están duplicados y, como manejan gran cantidad de información, ponen un límite al número de resultados recuperados.

Las desventajas de los metabuscadores son:

- Algunos no manejan sintaxis avanzadas de búsqueda debido a que los motores de búsqueda manejan diferentes sintaxis. Esto ocasiona dificultades para encontrar la información que se necesita.

Se recomienda utilizar los motores de búsqueda cuando:

- La consulta sea muy específica.

- Haya problemas en localizar la información con un motor de búsqueda o directorio específico.

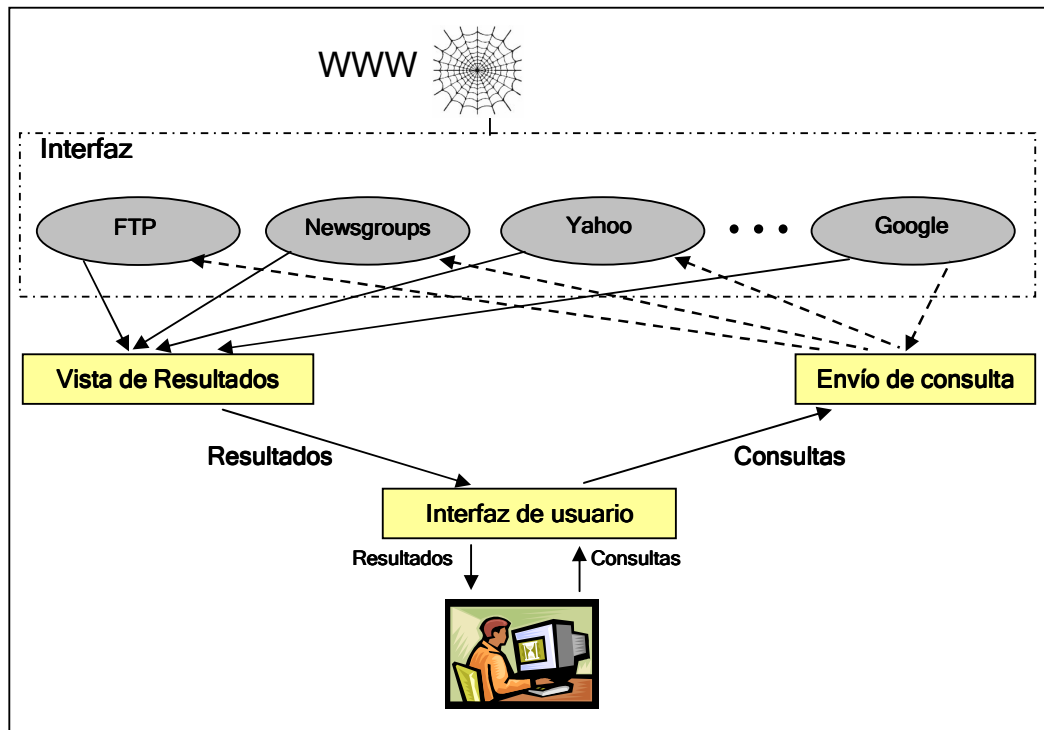


Figura 1.7 Estructura de un motor de búsqueda

Los motores de búsqueda más conocidos son: Dogpile [Dogpile, 2006], Mamma [Mamma, 2006], Ixquick [Ixquick, 2006]. En las figuras 1.8 y 1.9 se muestran los resultados que Dogpile devuelve para la consulta “apple”; y los que Mamma devuelve para “salsa”, respectivamente.

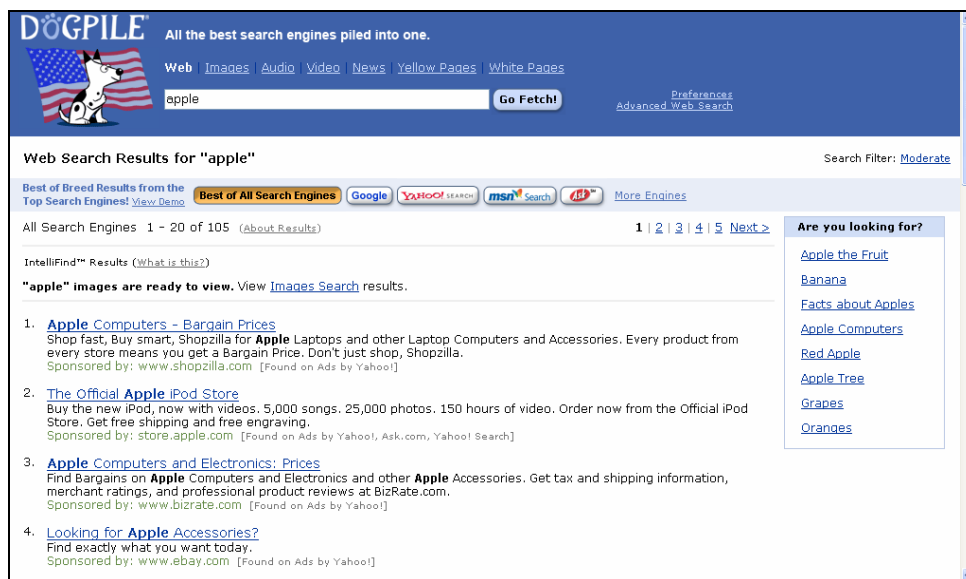


Figura 1.8 Resultados de Dogpile para la consulta “apple”

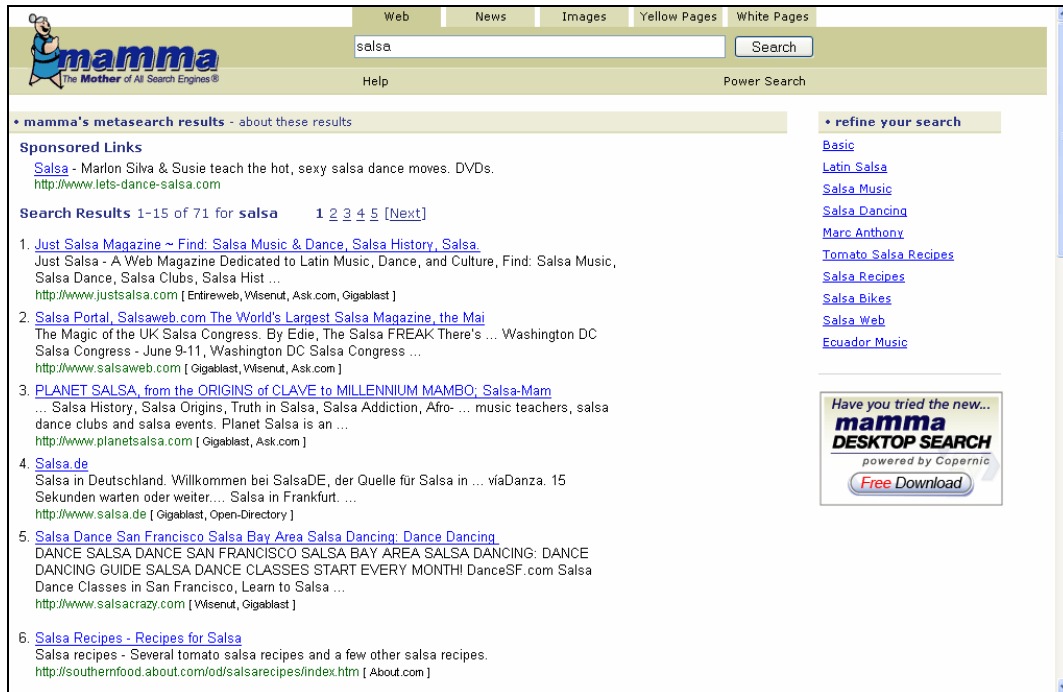


Figura 1.9 Resultados de Mamma para la consulta “salsa”

1.4 Agrupamiento de los resultados de la búsqueda (*clustering*)

Como hemos señalado, la cantidad de información que la Web contiene cada día va creciendo de manera exponencial y las herramientas de búsqueda muestran listas ordenadas de miles o millones de documentos de la Web. Estos documentos pueden relacionarse con varios temas y los documentos que el usuario está buscando suelen estar, en el peor de los casos, hasta el final de la lista. Obtener el conjunto de documentos que tengan la información deseada requiere una consulta que defina de manera precisa el contenido de los documentos.

Una solución a este problema es mostrar a los usuarios la lista de temas presentes en los resultados de la búsqueda. Esto le permite consultar los de su interés y así agilizar la búsqueda de información. Para generar esta lista de temas, se necesita agrupar los documentos. Esta técnica es llamada *clustering* (*agrupamiento*) y en el siguiente capítulo se definirá más a detalle.

Las ventajas de utilizar la técnica de *agrupamiento* (*clustering*) para organizar los documentos de la Web son las siguientes [Lang, 2003]:

- La organización de los resultados permite explorar un gran conjunto de datos.
- La asignación apropiada de etiquetas a los grupos permite al usuario descubrir los principales temas del conjunto de resultados e identificar el de su interés.

- Con los resultados agrupados por temas, el usuario puede examinar mayor cantidad de información relevante, lo cual no podía hacer cuando se le mostraba una lista de miles de documentos.
- El usuario puede revisar los documentos de la categoría de su interés.

Han sido realizados varios trabajos notables para agrupar grandes cantidades de información en grupos. A continuación se dará una breve explicación de los más importantes.

1.4.1. Scatter / Gather

Scatter / Gather es una técnica de exploración basada en *agrupamiento (clustering)* para grandes colecciones de texto; se basa en colocar documentos similares en el mismo *grupo*. Igualmente formar *grupos* de documentos en una jerarquía. Para cada *grupo*, en cada nivel de la jerarquía, se presenta al usuario un sumario que describe el contenido de los documentos que el *grupo* contiene. El usuario puede seleccionar (*gather*) el o los *grupos* que le parezcan interesantes. Este conjunto de *grupos* son reorganizados (*scatter*) para generar *grupos* más refinados de documentos, es decir, cada vez que se seleccionen uno o más *grupos*, éstos serán organizados de tal forma que se obtengan *grupos* más pequeños y más detallados; eventualmente se puede llegar a *grupos* con un solo documento [Pirolli *et al.*, 1996].

A continuación se muestra un ejemplo del uso de *scatter/gather*.²

Se realizó la consulta “*star*” y se recuperaron 250 documentos; *scatter/gather* colocó estos documentos en 5 *grupos*. La figura 1.10 muestra los grupos obtenidos, sus tamaños (número de documentos que contienen), la lista de términos que aparecen frecuentemente en los grupos (*topical terms*) y la lista de los títulos de los documentos. De la lista en la figura 1.10 el *Cluster 2* es un grupo que tiene 68 documentos relacionados con estrellas de televisión y cine. Si se indica a *scatter/gather* reorganizar este grupo, se obtienen otros tres, que se muestran en la figura 1.11. Los temas de los tres nuevos grupos son: personas que son estrellas del deporte (grupo 1); estrellas de cine, televisión y teatro (grupo 2) y músicos (grupo 3).

Con esta técnica se puede obtener información más detallada, que permitirá al usuario encontrar la información que le interesa sin necesidad de examinar grandes cantidades de documentos.

² Estos ejemplos provienen de <http://www.sims.berkeley.edu/~hearst/images/sg-example1.html>.

| | |
|---|--|
| Cluster 1 Size: 8 | key army war francis spangle banner air song scott word poem british |
| <input type="radio"/> Star-Spangled Banner, The <input type="radio"/> Key, Francis Scott <input type="radio"/> Fort McHenry <input type="radio"/> Arnold, Henry Harley <input type="radio"/> Mitchell, Andrew | Lista de documentos Topical terms |
| Cluster 2 Size: 68 | film play career win television role record award york popular stage p |
| <input type="radio"/> Burstyn, Ellen <input type="radio"/> Stanwyck, Barbara <input type="radio"/> Berle, Milton <input type="radio"/> Zukor, Adolph <input type="radio"/> Bankhead, Tallulah | |
| Cluster 3 Size: 97 | bright magnitude cluster constellation line type contain period spectr |
| <input type="radio"/> star <input type="radio"/> Galaxy, The <input type="radio"/> extragalactic systems <input type="radio"/> interstellar matter <input type="radio"/> cluster star | |
| Cluster 4 Size: 67 | astronomer observatory astronomy position measure celestial telesco |
| <input type="radio"/> astronomy and astrophysics <input type="radio"/> astrometry <input type="radio"/> Agena <input type="radio"/> astronomical catalogs and atlases <input type="radio"/> Hubble, Sir William | |
| Cluster 5 Size: 10 | family specie flower animal arm plant shape leaf brittle tube foot hor |
| <input type="radio"/> blazing star <input type="radio"/> brittle star <input type="radio"/> bishop's-cap <input type="radio"/> feather star | |

Figura 1.10 Grupos obtenidos por *scatter/gather* para la consulta "star".

| | |
|---|---|
| Cluster 1 Size: 14 | player league hit game national set bat average season history baseba |
| <input type="radio"/> Musial, Stan <input type="radio"/> Bench, Johnny <input type="radio"/> Carew, Rod <input type="radio"/> Robertson, Oscar <input type="radio"/> Beliveau, Jean <input type="radio"/> Casper, Billy <input type="radio"/> Chinese checkers <input type="radio"/> Best, George <input type="radio"/> Beamon, Bob | |
| Cluster 2 Size: 47 | role stage broadway comedy performance actress production musical |
| <input type="radio"/> Burstyn, Ellen <input type="radio"/> Stanwyck, Barbara <input type="radio"/> Berle, Milton <input type="radio"/> Bankhead, Tallulah <input type="radio"/> Murphy, Eddie <input type="radio"/> Walsh, Raoul <input type="radio"/> Martin, Mary <input type="radio"/> Zukor, Adolph <input type="radio"/> Cosby, Bill | |
| Cluster 3 Size: 7 | music country jazz folk pop paul cowboy leader williams hampton boy |
| <input type="radio"/> Williams, Hank <input type="radio"/> Crosby, Bing <input type="radio"/> Campbell, Glen <input type="radio"/> Belafonte, Harry <input type="radio"/> Shore, Dinah <input type="radio"/> Denver, John <input type="radio"/> Hampton, Lionel | |

Figura 1.11 Reorganización.

1.4.2. Grouper

Grouper fue la primera aplicación de la técnica de *agrupamiento* (*clustering*) para la información obtenida de un motor de búsqueda; fue integrado al metabuscador HuskySearch.

Grouper fue desarrollado en la Universidad de Washington por Zamir y Etzioni [Zamir y Etzioni, 1999]; utiliza el algoritmo *Suffix Tree Clustering* para agrupar documentos que compartan frases en común. Este algoritmo permite que los documentos puedan pertenecer a más de un grupo. Es importante señalar que este sistema ya no está disponible en línea.

En la figura 1.12 se muestra la interfaz de Grouper y en la figura 1.13, un ejemplo de los resultados obtenidos para la consulta “Clinton”.

Figura 1.12. Interfaz de Grouper [Zamir, 1999]

| Cluster | Size | Shared Phrases and Sample Document Titles |
|-----------------------------------|------|--|
| 1 View Results | 37 | <p>Monica Lewinsky (32%), Clinton's scandals (16%), Kenneth Starr Investigation (14%), Hillary Clinton (14%)</p> <ul style="list-style-type: none"> ● Joke Post: Clinton Lewinsky Jokes ● The Bill Clinton Information Gateway ● Bill Clinton, Monica Lewinsky and Kenneth Starr - the saga of Bill and Monica. |
| 2 View Results | 20 | <p>Clinton a positive or negative (20%), Clinton/Gore (20%), Presidential Election (20%), election of (20%)</p> <ul style="list-style-type: none"> ● Republicans for Clinton ● Clinton, Bill - Project Vote Smart ● Clinton Record, The |
| 3 View Results | 8 | <p>Jones's (63%), documents (50%), special (50%); President (37%), Report (37%), legal (37%), Paula (37%)</p> <ul style="list-style-type: none"> ● Jones v. Clinton Special Report ● Paula Jones Legal Fund ● JONES vs CLINTON |

Figura 1.13. Resultados obtenidos por Grouper para la consulta “Clinton” [Zamir, 1999]

1.4.3. Carrot 2

Carrot2 [Carrot2, 2006] es un sistema de código abierto basado en Grouper que fue desarrollado, en la Universidad Tecnológica de Poznan, por Dawid Weiss [Weiss, 2001]; aplica la técnica de *agrupamiento (clustering)* para agrupar información en idioma inglés y polaco.

Este sistema recupera información de varias fuentes de datos, como Google, Yahoo, All the web y BBC news, la procesa y la muestra al usuario. La figura 1.14 muestra la interfaz de Carrot2 y la figura 1.15., un ejemplo de los resultados obtenidos para la consulta “Clinton”. Para esta consulta se especificó al sistema recuperar la información de Yahoo y utilizar el algoritmo Lingo para agrupar la información.

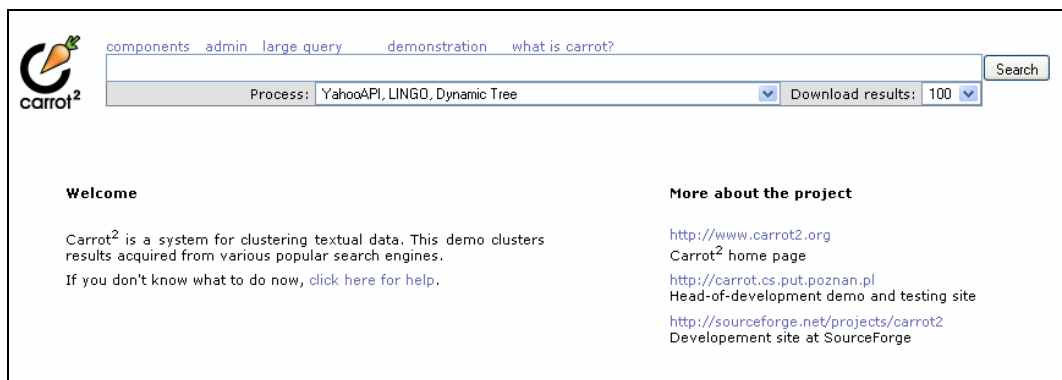


Figura 1.14. Interfaz de Carrot2.

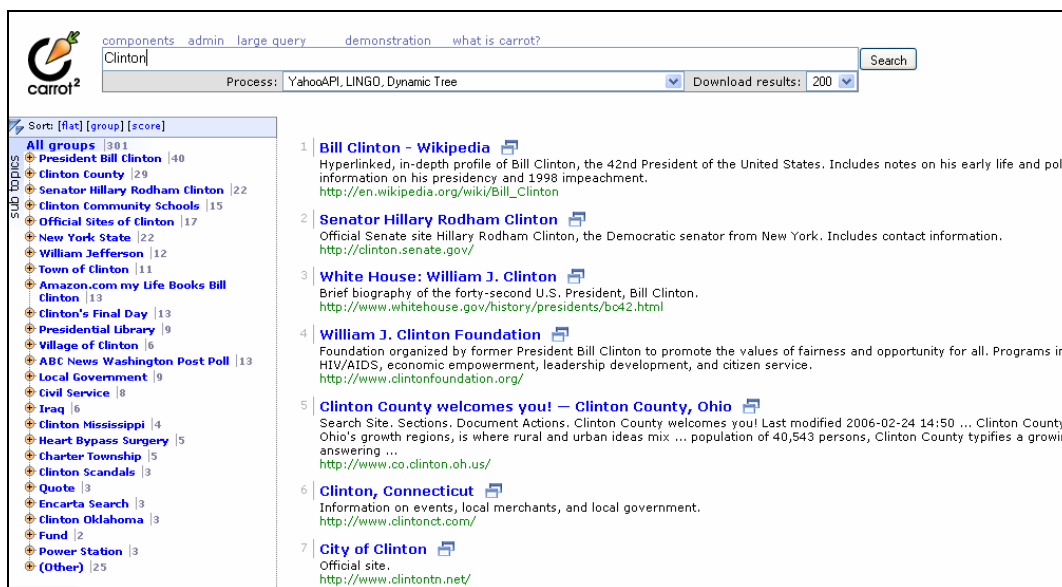


Figura 1.15. Resultados obtenidos para la consulta “Clinton”.

1.4.4. Vivísimo

Vivísimo [Vivísimo, 2006] es un metabuscador comercial que utiliza la técnica de *agrupamiento (clustering)*. Este metabuscador es conocido por su habilidad para producir jerarquías de alta calidad en los resultados de la búsqueda. Desafortunadamente no hay información del algoritmo que este metabuscador utiliza. Los autores dan un panorama general de cómo funciona Vivísimo:³

Nosotros usamos un algoritmo heurístico especialmente desarrollado para agrupar los documentos de texto. Este algoritmo está basado en una idea vieja de inteligencia artificial: un buen grupo o agrupación de documentos, es aquella que tiene una buena y clara descripción. Así que, más que formar grupos y decidir cómo describirlos, nosotros sólo formamos en primer lugar grupos bien descritos.

La versión mejorada de Vivísimo se llama Clusty [Clusty, 2006]. Clusty tiene nuevas características y una nueva interfaz que Vivísimo no tiene. La figura 1.16 muestra la imagen con la interfaz de Clusty; la 1.17, la interfaz de Vivísimo y la figura 1.18, un ejemplo de los resultados obtenidos para la consulta “jaguar”.

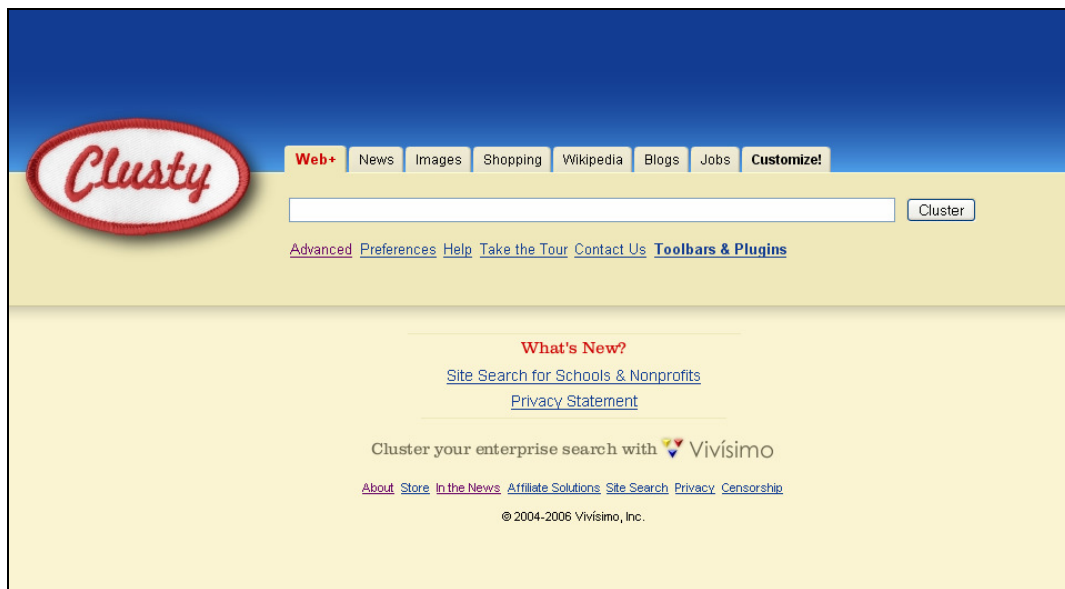


Figura 1.16. Interfaz de Clusty.

³ <http://vivisimo.com/html/faq>.



Figura 1.17. Interfaz de Vivísimo.

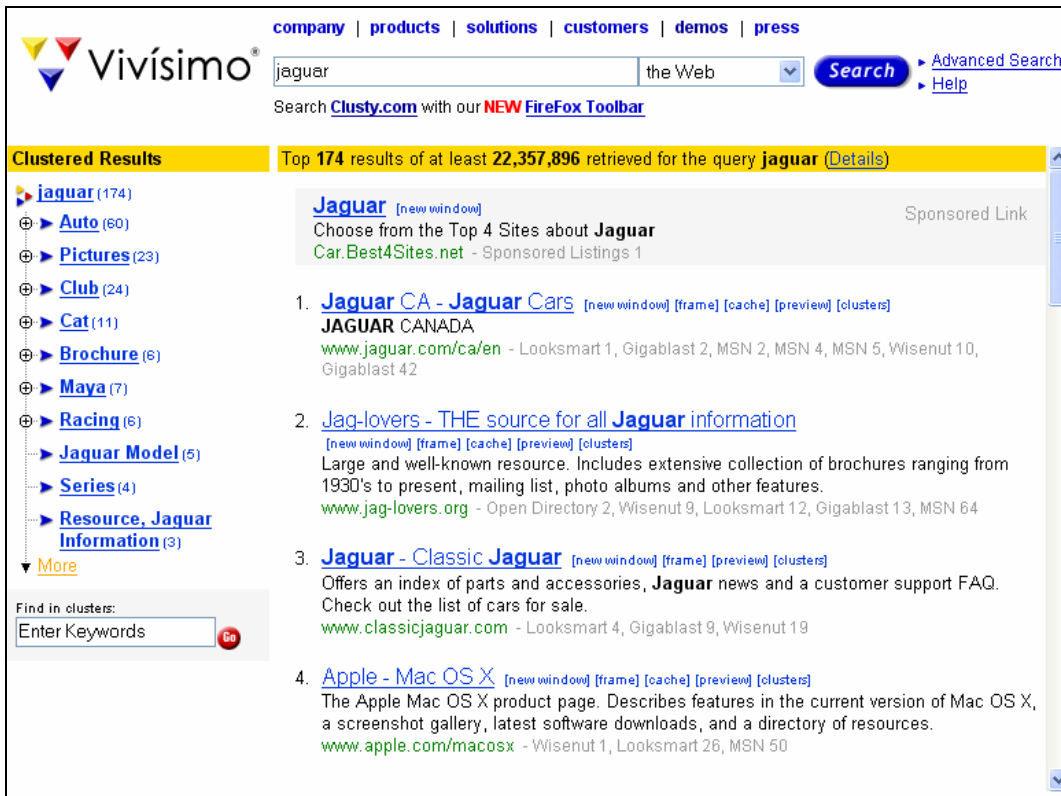


Figura 1.18. Resultados obtenidos para la consulta “jaguar”.

1.4.5. Comparación de los sistemas analizados

Las secciones anteriores presentaron los sistemas más importantes que aplican la técnica de *agrupamiento* a un conjunto de documentos. Esta sección muestra una comparación de cada uno de estos sistemas, exponiendo sus características más relevantes.

El cuadro 1.2 enlista las características de cada uno de estos sistemas, de las cuales se obtienen las siguientes características en común y diferencias que hay entre ellos:

- *Scatter/Gather* y *Vivísimo* son sistemas que aplican la técnica de *agrupamiento* a una colección de documentos que no necesariamente provienen de un motor de búsqueda.
- *Scatter/Gather* puede agrupar hasta 5000 documentos cortos.
- *Scatter/Gather* es el único sistema que permite el *reagrupamiento* (*reclustering*), es decir, el usuario elige los grupos que son de su interés y el sistema reagrupa los documentos de cada grupo con el fin de obtener otros más pequeños y más detallados.
- Los sistemas que realizan *agrupamiento* jerárquico son: *Scatter/Gather*, *Carrot2* y *Vivísimo*.
- Los sistemas que realizan *agrupamiento* no jerárquico son: *Groupier* y *Carrot2*.
- *Carrot2* es el único sistema que puede utilizar varios algoritmos de *agrupamiento*.
- En cuanto a las etiquetas de los grupos la mayoría de los sistemas excepto *Scatter/Gather* utilizan frases.
- En cuanto al idioma de los documentos *Vivísimo* es el único sistema que maneja varios idiomas (más de dos); *Carrot2* maneja sólo dos idiomas y el resto sólo manejan documentos en idioma inglés.
- Los sistemas que están disponibles en la Web son: *Vivísimo* [Vivísimo, 2006] y *Carrot2* [Carrot2, 2006].

De la información anterior surge la pregunta ¿cuál de estos sistemas debo de usar?; la respuesta no es sencilla ya que depende de las necesidades del usuario.

Actualmente de los sistemas listados el usuario tiene sólo dos opciones: *Vivísimo* y *Carrot2*. De estos dos se recomienda utilizar *Vivísimo* a las personas que no tengan conocimiento de la técnica de *agrupamiento* porque su interfaz para realizar una consulta y mostrar los resultados es bastante amigable. *Carrot2* es recomendada para personas que tienen conocimiento de la técnica de *agrupamiento* porque este sistema presenta la opción de poder elegir el algoritmo a usar, lo cual permite al usuario

comparar las agrupaciones realizadas por los distintos algoritmos y así escoger el de su preferencia. La interfaz de *Carrot2* que muestra los resultados es bastante amigable.

| Sistema | Características |
|------------------------|--|
| Scatter /Gather | <ol style="list-style-type: none"> 1. Aplica la técnica de <i>agrupamiento</i> a grandes colecciones de texto (es capaz de agrupar 5000 documentos cortos [Hearst y Pedersen, 1996]). 2. Este sistema trabaja con documentos en inglés. 3. Realiza <i>agrupamiento</i> jerárquico 4. Permite el <i>reagrupamiento</i>, es decir, el usuario selecciona los grupos de su interés y el sistema los reagrupa. 5. Sólo utiliza un algoritmo de <i>agrupamiento</i> (llamado <i>Buckshot</i>). 6. Utiliza como etiquetas de los grupos los términos que aparezcan frecuentemente en el grupo. |
| Groupier | <ol style="list-style-type: none"> 1. Aplica la técnica de <i>agrupamiento</i> a los resultados obtenidos de varios motores de búsqueda. 2. Los resultados que recupera del motor de búsqueda están en inglés. 3. Realiza <i>agrupamiento</i> no jerárquico. 4. No permite el <i>reagrupamiento</i>. 5. Sólo utiliza un algoritmo de <i>agrupamiento</i>. 6. Utiliza frases como etiquetas de los grupos. |
| Carrot2 | <ol style="list-style-type: none"> 1. Aplica la técnica de <i>agrupamiento</i> a los resultados de los motores de búsqueda como Yahoo, Google, All The Web y BBC News. 2. Los resultados que recupera del motor de búsqueda están en inglés o en polaco. 3. Recupera de 50 a 200 resultados de los motores de búsqueda. 4. Utiliza varios algoritmos de <i>agrupamiento</i>. 5. Realiza <i>agrupamiento</i> jerárquico y no jerárquico. 6. Utiliza frases como etiquetas de los grupos. 7. El sistema está disponible en la Web. |
| Vivísimo | <ol style="list-style-type: none"> 1. Aplica la técnica de <i>agrupamiento</i> a los resultados obtenidos de varios motores de búsqueda. 2. Los resultados que recupera del motor de búsqueda pueden estar en varios idiomas. 3. Recupera de 100 a 500 resultados de los motores de búsqueda. 4. Utiliza sólo un algoritmo de <i>agrupamiento</i>. 5. Hace <i>agrupamiento</i> jerárquico. 6. El sistema está disponible en la Web. |

Cuadro 1.2. Características de los sistemas más importantes que utilizan la técnica de *agrupamiento*.

Capítulo 2. Agrupamiento (clustering) de documentos

En este capítulo se explica qué es la técnica del *agrupamiento (clustering)* de documentos y en qué consiste este proceso, desde la representación de documentos usando el *modelo de espacio de vectores (vector space model)*; el *agrupamiento* de los documentos realizado por un algoritmo y la asignación de etiquetas que describen el contenido de cada uno.

2.1 Definición general

Clustering es una técnica para agrupar datos.

Sea $D = \{d_1, d_2, \dots, d_n\}$ un conjunto de datos y $\delta(d_i, d_j)$ la medida de semejanza entre d_i y d_j para $i \neq j$, $1 \leq i, j \leq n$. *Clustering* se define como la tarea de encontrar la descomposición (partición) de D en K grupos¹ (*clusters*) $C = \{c_1, \dots, c_k\}$ tal, que cada dato es asignado a un grupo y los datos que pertenecen al mismo grupo son similares entre sí (en relación con la medida de semejanza δ) [Lang, 2003].

De acuerdo con Halkidi [Halkidi *et al.*, 2001] y Jain [Jain *et al.*, 1999], la técnica de *Agrupamiento* tiene aplicaciones en diversas áreas, como:

- **Negocios:** Ayuda a los especialistas en mercadotecnia a descubrir patrones de compras en los clientes.
- **Biología:** Define taxonomías y clasifica genes de acuerdo con su funcionalidad.
- **Análisis espacial de datos:** Automatiza el proceso de análisis de los datos espaciales (identificando y extrayendo características y patrones interesantes), debido a que por naturaleza es un proceso costoso y a que se tienen disponibles grandes cantidades de datos espaciales que pueden ser obtenidos de las imágenes de satélites, equipo médico o sistemas de información geográfica (GIS, por sus siglas en inglés).
- **Recuperación de Información (IR, por sus siglas en inglés):** Ayuda a construir una taxonomía de una colección de documentos, es decir, clasificar los documentos de acuerdo con su contenido.

Este trabajo es un ejemplo de la aplicación de la técnica de *Agrupamiento* en IR.

¹ De aquí en adelante utilizaremos la palabra grupo en vez de la palabra *cluster* y la palabra agrupamiento en lugar de la palabra *clustering*.

2.2 Agrupamiento (*clustering*) en IR

El uso de la técnica de *Agrupamiento* en IR está basada en la hipótesis de Rijsbergen [Rijsbergen, 1979]: “los documentos cercanamente relacionados tienden a ser relevantes a la misma petición (consulta)”. Esto significa que los documentos relevantes son más similares entre sí que los no relevantes. Si esta hipótesis se cumple para una colección de documentos, entonces esto podría hacer la recuperación más efectiva porque, una vez que se identifique el grupo de interés, éste sólo contendrá documentos relevantes [Anton y Croft, 1996].

La técnica de *Agrupamiento* también puede usarse como una herramienta para que el usuario examine los grupos, lo cual le permite ver en diferentes niveles de detalle la información. Esto es útil cuando el usuario no puede expresar en forma clara la información que está buscando.

El proceso para hacer el *agrupamiento* de documentos queda descrito en la siguiente figura.

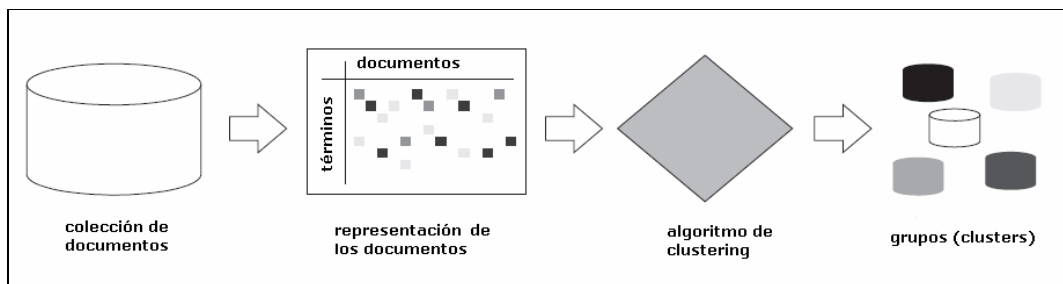


Figura 2.1 Proceso de *agrupamiento* de documentos [Lang, 2003].

El proceso consta de los siguientes pasos:

- 1) Obtener la colección de documentos.
- 2) Elegir una representación apropiada de ellos.
- 3) Agrupar los documentos con algún algoritmo de *agrupamiento*.
- 4) Obtener los grupos de documentos.

En el sistema *Conquiro*, los pasos del proceso anterior se realizan de la siguiente manera: la colección de documentos se obtiene del motor de búsqueda Google dada una consulta; se utiliza el modelo de espacio de vectores (*Vector Space Model*, VSM por sus siglas en inglés), que es el más comúnmente usado para hacer la representación de los documentos; se tiene la opción de realizar el *agrupamiento* de documentos con siete algoritmos diferentes; y finalmente se asigna una etiqueta a los grupos generados por el algoritmo.

En las siguientes secciones se explica en qué consiste el modelo de espacio de vectores (VSM),² los algoritmos de *agrupamiento* y los métodos para etiquetar los grupos de documentos utilizados en este trabajo.

² De ahora en adelante, para hacer referencia al modelo de espacio de vectores, se utilizará VSM.

2.3 Modelo de espacio de vectores (VSM) y la representación de documentos

Los modelos clásicos en IR consideran que cada documento puede ser descrito por un conjunto de palabras clave llamadas términos índice (*index term*). Un **término índice** es una palabra cuyo significado ayuda a identificar el tema principal del documento. Los modelos en IR basados en esta idea son: el lógico (*boolean*), el probabilístico y el VSM [Baeza-Yates y Ribeiro-Neto, 1999].

El VSM es un modelo que trata a los documentos como vectores de números, los cuales contienen valores que corresponden a la ocurrencia de los términos índice (a los que llamaremos términos) en sus respectivos documentos. Sea t el número de términos y n el número de documentos en la colección. Entonces todos los documentos D_i , $i = 1, \dots, n$ pueden ser representados como vectores t -dimensionales:

$$D_i = [a_{i1}, a_{i2}, \dots, a_{it}]$$

donde los coeficientes a_{ik} para $k = 1, \dots, t$, representan el peso del término k en el documento D_i , respectivamente. Los documentos y los términos forman la **matriz de términos por documentos** $M_{t \times n}$. Los renglones de ésta representan los términos y las columnas, los documentos.

El peso del término en el documento se refiere a la importancia que éste tiene para representar el contenido del documento. Si el peso de un término es cero, significa que no está presente en el documento. Hay varios métodos para asignar pesos a los términos, entre los más comunes están *tf* y *tf-idf*, los cuales serán explicados en la sección 2.3.2.

El modelo VSM permite explotar las relaciones geométricas entre los vectores de los documentos para modelar las semejanzas y diferencias en el contenido [Berry *et al.*, 1999]. Las medidas más comunes usadas para medir la semejanza entre los vectores son: distancia euclidiana y la semejanza de coseno (*cosine similarity*), las cuales quedan explicadas en la sección 2.3.3.

Cada dimensión del vector equivale a un término³ (palabra) distinto en la colección de documentos. Debido a la naturaleza de los documentos de texto, el número de términos distintos puede ser extremadamente grande, lo cual implica tener vectores de dimensiones muy grandes. Como no todos los términos son útiles para describir los contenidos de los documentos, es necesario limpiar los documentos y aplicar métodos que reduzcan el número de términos utilizados en el vector. En la sección 2.3.1 se explica cómo se realiza la limpieza de los documentos; y en la 2.3.4, el método DF (*Document Frequency thresholding*) para reducir la dimensionalidad del vector.

³ La palabra “término” y “palabra” se usan de manera indistinta.

2.3.1 Limpieza de los documentos

La limpieza de los documentos es una tarea importante, debido a que puede influir en el desempeño de los algoritmos de *agrupamiento (clustering)*. En esta etapa se puede disminuir el número de términos de los vectores, lo cual reduce el tiempo de ejecución y mejora la calidad de los términos. Para hacer la selección de los términos, es necesario hacer varias operaciones sobre el texto.

En este trabajo fueron utilizados documentos en inglés, por lo cual los métodos para la limpieza del texto están enfocados a este idioma.

Para limpiar los documentos se realizan los siguientes pasos:

- 1) Eliminar el código HTML del documento, los URL y las direcciones de e-mail.
- 2) Eliminar caracteres que no sean letras (símbolos de puntuación, números).
- 3) Eliminar las palabras que aparezcan en la lista de palabras comunes (*stop words*).
- 4) Cambiar las letras mayúsculas a minúsculas.
- 5) Aplicar al texto la técnica de reducción de términos a la raíz (técnica de *stemming*).

El código HTML, los URL y las direcciones de e-mail se eliminan porque no aportan información acerca del contenido del documento. Respecto del punto 2, hay que considerar algunos casos como:

- a. El algoritmo *Suffix Tree Clustering* extrae frases del texto, por lo cual es necesario preservar en el texto los siguientes símbolos ‘.’, ‘?’ y ‘!’, ya que éstos son considerados como delimitadores de oraciones.
- b. En inglés hay palabras que utilizan guión (ejemplo, *thirty-one*). Para este caso el guión se preserva.
- c. Los números no aportan ninguna información acerca del contenido de los documentos, exceptuando en los casos de fechas históricas (1000 b. C.), para referirse al precio de algún objeto (25 c).

Hay palabras que ocurren en todos los documentos, independientemente de su tema, debido a esto la información que aportan acerca del contenido del documento es nula. Las palabras con estas características son llamadas **palabras comunes** o *stop words*. Algunas de ellas son los artículos (“a”, “the”), las preposiciones (“in”), las conjunciones (“and”, “but”), algunos verbos (“to be”), los adjetivos y los adverbios.

El apéndice A muestra la lista de las palabras comunes que utiliza el sistema *Conquiro*, la mayor parte fue tomada de la lista construida por Salton and Buckley;⁴ el resto provino de la lista utilizada en Carrot2 [Wroblewski, 2003].

La técnica de reducción de términos a la raíz (técnica de *stemming*) consiste en extraer sufijos y prefijos, de tal forma que las palabras que literalmente son diferentes

⁴ <http://www.lextek.com/manuals/onix/stopwords2.html>

pero con una raíz común pueden ser consideradas como un solo término a partir de su raíz (*stem*). Por ejemplo las palabras “*clusters*”, “*clustering*” y “*clustered*” tienen como raíz “*cluster*”. El uso de esta técnica permite sustituir las variaciones de las palabras a una forma representativa, de tal manera que el tamaño del vocabulario se reduce sin afectar el contenido semántico. Es conveniente resaltar que los algoritmos que hacen esta reducción están basados en reglas simples para remover los prefijos y sufijos y no necesariamente producen palabras lingüísticamente correctas, por ejemplo, “*computing*”, “*computation*” son reducidas a “*comput*”, aun cuando la palabra correcta sea “*compute*” [Lang, 2003].

El algoritmo para la reducción de términos a la raíz, utilizado en este trabajo para el idioma inglés, fue el de Porter [Porter, 1980].⁵

2.3.2 Asignar pesos a los términos

Ya señalamos que el peso de un término implica la importancia que éste tiene para representar el contenido del documento y distinguirlo de otros. El proceso para calcular el peso de un término es llamado **asignación de pesos** (*term weighting*). Hay varios métodos para calcular los pesos de los términos; entre los más usados están *tf* y *tf-idf*.

Método *tf*

El método *tf* (del inglés *term frequency*) está basado en la idea de que los términos que aparecen frecuentemente en un documento son útiles para describir el contenido de éste. Con base en esto, mide la frecuencia de los términos en los documentos. La frecuencia del término *i* en el documento *j* se calcula de la siguiente manera:

$$tf_{ij} = n_{ij}$$

donde n_{ij} es el número de veces que el término *i* aparece en el documento *j*.

Método *tf-idf*

Cuando los términos con frecuencias altas aparecen en la mayoría de los documentos de la colección, no hay manera de distinguir entre los contenidos de los documentos, por lo cual se necesita un factor que asigne mayor peso a los términos que aparezcan en pocos documentos y menor peso a los que aparezcan en muchos. El nombre de este factor es frecuencia inversa de documento (*Inverse Document Frequency*, *idf* por sus siglas en inglés).

El factor IDF del término *j* se calcula de la siguiente manera:

⁵ Ésta es la página oficial del algoritmo de Porter: <http://www.tartarus.org/martin/PorterStemmer/>

$$idf_j = \log\left(\frac{N}{df_j}\right)$$

donde N es el número de documentos en la colección y df_j es la frecuencia de documentos (*document frequency*) del término j , es decir, el número de documentos de la colección que contienen el término j .

El método **tf-idf** (del inglés *Term Frequency-Inverse Document Frequency*) es la combinación de frecuencia del término (*tf*) y la frecuencia inversa de documento (*idf*) del término. La idea de esta combinación es asignar un peso alto al término que ocurra frecuentemente en un documento, pero que aparezca en pocos documentos de la colección.

Sea m el número de términos en la colección de documentos t_1, \dots, t_m y N el número de documentos de la colección d_1, \dots, d_N . Para cada término t_j en el documento d_i , *tf-idf* es definido por Salton [Salton, 1989] como:

$$w_{ij} = tf_{ij} * idf_j$$

donde tf_{ij} es la frecuencia del término j en el documento i ; e idf_j es la frecuencia inversa de documento del término j .

La gráfica de la figura 2.2 muestra cómo se comporta el valor de *idf* en función del valor de *df*. Cuando un término aparece en todos los documentos, es decir *df* es igual a N , entonces *idf* es igual a 0 porque este término no aporta información para diferenciar los contenidos. Sin embargo, conforme el valor de *df* sea menor que N , es decir, que el término aparezca en pocos documentos, el valor de *idf* es alto.

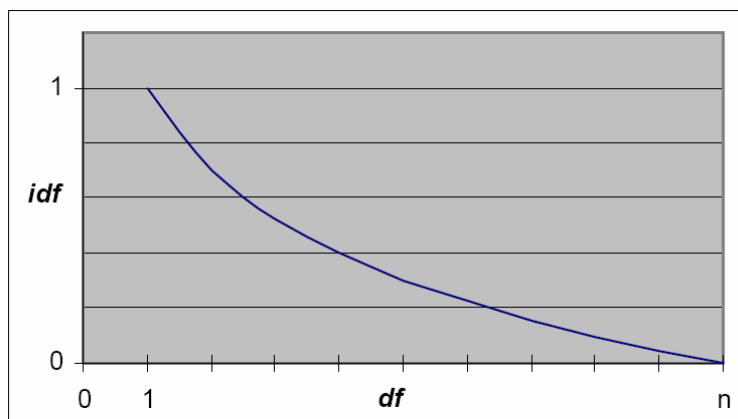


Figura 2.2 *IDF* en función de *DF* [Wroblewski, 2003]

2.3.3 Medidas de semejanza

La representación vectorial de los documentos permite calcular la distancia o la semejanza entre dos vectores. Este cálculo entre dos vectores *n*-dimensionales puede hacerse de varias maneras. Las medidas más usadas son las siguientes [Salton, 1989]:

| Medidas $\text{sim}(X, Y)$ | Evaluación para vectores de términos binarios | Evaluación para vectores no binarios |
|-------------------------------|--|---|
| Producto interno | $X \cap Y$ | $\sum_{i=1}^t x_i \cdot y_i$ |
| Coefficiente de Dice | $2 \frac{ X \cap Y }{ X + Y }$ | $\frac{2 \sum_{i=1}^t x_i y_i}{\sum_{i=1}^t x_i^2 + \sum_{i=1}^t y_i^2}$ |
| Semejanza de coseno | $\frac{ X \cap Y }{ X ^{1/2} \cdot Y ^{1/2}}$ | $\frac{\sum_{i=1}^t x_i y_i}{\sqrt{\sum_{i=1}^t x_i^2 \cdot \sum_{i=1}^t y_i^2}}$ |
| Coefficiente de Jaccard | $\frac{ X \cap Y }{ X + Y - X \cap Y }$ | $\frac{\sum_{i=1}^t x_i y_i}{\sum_{i=1}^t x_i^2 + \sum_{i=1}^t y_i^2 - \sum_{i=1}^t x_i y_i}$ |

Figura 2.3 Las medidas más usadas para calcular la distancia o la semejanza entre dos vectores

Salton [Salton, 1989] menciona que la elección de la medida para calcular la distancia o la semejanza para alguna aplicación en particular carece de consideraciones teóricas, por lo cual se deja a criterio de cada persona. Sin embargo, se ha encontrado que la distancia euclidiana es a menudo inapropiada para la agrupación de los documentos, por lo cual se sugiere utilizar la semejanza de coseno (*cosine similarity*)⁶ como medida [Shyu *et al.*, 2004] [Strehl *et al.*, 2000].

Este trabajo utiliza las medidas de semejanza de coseno y distancia euclidiana con el fin de comprobar que, con la semejanza de coseno, se obtienen mejores agrupaciones de documentos. La sección 4.2.1 muestra los resultados obtenidos con estas medidas.

La **semejanza de coseno** de dos vectores *a* y *b* se define como:

$$\cos(a, b) = \frac{a \cdot b}{\|a\| \|b\|} = \frac{\sum_{i=1}^n a_i b_i}{\sqrt{\sum_{i=1}^n a_i^2} \sqrt{\sum_{i=1}^n b_i^2}}$$

Los valores de la semejanza de coseno siempre están entre 0 y 1.

⁶ De ahora en adelante se utilizará semejanza de coseno

La **distancia euclidiana** entre dos vectores a y b se define como:

$$d(a,b) = \sqrt{\sum_{i=1}^n (b_i - a_i)^2}$$

Widdows [Widdows, 2004] menciona que la semejanza es la función inversa a la distancia porque la semejanza corresponde a la proximidad y la no semejanza, a la distancia; es decir, a menor distancia mayor semejanza y a mayor distancia menor semejanza. Supongamos que para un conjunto A tenemos una función de semejanza $sim: A \times A \rightarrow \mathbb{R}$, la cual da valores altos para $sim(a,b)$ si a y b son semejantes, y valores pequeños si a y b son muy diferentes el uno del otro. Si aplicamos la función sim a dos puntos muy distantes uno del otro (la distancia entre ellos es muy grande), entonces la función le asignará valores de semejanza pequeños; por el contrario, si estos puntos son muy cercanos (la distancia entre ellos es pequeña), entonces la función sim les asignará valores de semejanza grandes.

2.3.4 Reducción de la dimensionalidad

El número de términos distintos que los vectores de los documentos tienen pueden ser cientos de miles para una colección de documentos de tamaño moderado, por lo cual es deseable reducir el número de términos utilizados en los vectores de los documentos, pero sin afectar la precisión del *agrupamiento*. Esta reducción es posible utilizando métodos de selección automática de características (*Automatic Feature Selection methods*), que eliminan los términos no informativos de acuerdo a su importancia estadística en la colección. Yang y Pedersen [Yang y Pedersen, 1997] proponen 5 métodos para la selección automática de características (términos); entre ellos está el método *DF* (*Document Frequency thresholding*), el cual fue utilizado en este trabajo.

DF (*Document frequency thresholding*)

Como se mencionó en la sección 2.3.2, la frecuencia del documento (*Document Frequency*) es el número de documentos en los que el término aparece. Para cada término del espacio de características⁷, se calcula el *DF*; y los términos cuyo valor *DF* sea menor que un umbral predeterminado son eliminados del espacio de características. La idea es eliminar los términos que afectan el desempeño de los algoritmos de *agrupamiento*.

Antes de aplicar el método *DF* es necesario hacer la limpieza de los documentos. El método puede utilizar diferentes umbrales, pero si el umbral tiene un valor alto, es posible que haya documentos que tengan todos sus términos debajo de este umbral y éstos sean eliminados. Para evitar remover todos los términos de un documento, se aplica la siguiente regla: aplicar el umbral a los términos de los documentos sólo si no se genera un documento vacío. El inconveniente de esta regla es que si se utilizan todos los términos del documento, se corre el riesgo de dejar términos que generen ruido en el

⁷ El espacio de características es el conjunto de términos que pueden ser usados para formar los vectores de los documentos.

proceso de *agrupamiento* (por ejemplo, términos que no aporten información acerca del contenido), por lo cual se propone la siguiente modificación a la regla anterior: Si al aplicar el umbral el documento conserva menos de 30% de sus términos, entonces hay que reducir el umbral una unidad hasta que se obtenga al menos 30% de los términos. Con esta información se podrá conservar los términos que aporten información al contenido del documento.

Este método es una técnica simple para reducir el vocabulario (número de términos); sin embargo debe ser usado con cuidado, ya que hay términos que tienen un valor bajo de DF y que se consideran como medianamente informativos y su eliminación podría afectar el desempeño de los algoritmos de *agrupamiento*.

En la sección 4.2.2 se muestra la influencia que la reducción de la dimensionalidad tiene en el desempeño de los algoritmos.

2.3.5 Normalización de los vectores

Los documentos en la colección pueden tener varios tamaños; para los de gran tamaño, los componentes de sus vectores tienen valores altos comparados con los vectores de los documentos de menor tamaño, lo cual podría causar que los algoritmos de *agrupamiento* produzcan soluciones incorrectas. Para solucionar este problema, los vectores de los documentos se normalizan [Lang, 2003].

2.3.6 Ejemplo utilizando el modelo VSM

En el siguiente ejemplo se muestra cómo una colección simple de 6 documentos puede ser descrita por 11 términos usando el modelo VSM.

La figura 2.4 (a) muestra la colección de documentos, conformada por 6 títulos de páginas de la Web, que va a ser utilizada.

Para seleccionar los términos que describirán el contenido de los 6 documentos, se necesita realizar el proceso de limpieza, en el cual serán eliminados los términos que no aportan información respecto del contenido de los documentos. Para la colección de la figura 2.4 (a), fueron eliminados los caracteres que no son letras y las palabras que aparecieron en la lista de palabras comunes del apéndice A; además se convirtieron la letras mayúsculas a minúsculas, como se indica en la sección 2.3.1.

Los términos que quedaron después de esta eliminación fueron: *big, apple, circus, home, recipesource, autum, punch, recipes, custard, recipe, place y lime*. A esta lista de términos se aplica la técnica de reducción de términos a la raíz. Como resultado se obtuvo la lista que muestra la figura 2.4 (b), donde los términos *recipe y recipes* quedaron representados con el término *recip*, raíz que la técnica les asignó. Los términos *apple, circus y recipesource* serán representados por los términos *appl, circu y recipesourc*, respectivamente.

Los términos que describirán el contenido de la colección son los obtenidos como resultado de aplicar la técnica de reducción de términos a la raíz. En este caso, esos términos se muestran en la figura 2.4 (b).

Una vez obtenidos los términos que describirán el contenido de los documentos de la colección, ya se puede construir los vectores de los documentos (también llamados vectores de características). El vector de características del documento D_i se construye asignando a cada término el peso que éste tiene en el documento D_i para $i=1,\dots,6$. Para este ejemplo el peso de los términos fue calculado con el método *tf-idf*.

La figura 2.4 (c) muestra los vectores de características de los 6 documentos de la colección. A manera de ejemplo se explicará cómo se construyó el vector de características del documento $D3$. En este documento los únicos términos que aparecen son *apple* y *recipes*, representados por los términos *appl* y *recip*, respectivamente. El peso de los términos que no están presentes en el documento es 0.00 y el peso del término *appl* se calcula de la siguiente manera:

$$w_{applD3} = tf_{applD3} * \log\left(\frac{N}{df_{appl}}\right) = 1 * \log\left(\frac{6}{6}\right) = 0$$

donde w_{applD3} es el peso del término *appl* en el documento $D3$; tf_{applD3} es la frecuencia del término *appl* en $D3$, la cual es 1. N es el número de documentos en la colección y df_{appl} es el número de documentos que contienen el término *appl*, en este caso es 6. Debido a que el peso del término *appl* es cero, la entrada dos del vector de características del documento $D3$ es cero.

El peso del término *recip* es 0.30, por lo cual la entrada ocho del vector es distinta de cero. El cálculo del peso de este término es el siguiente:

$$w_{recipD3} = 1 * \log\left(\frac{6}{3}\right) = 0.30$$

Una vez construidos los vectores de características de los documentos, la matriz de términos por documentos $M_{11 \times 6}$ se construye de la siguiente manera: en la entrada m_{ij} se pone el peso que el término i tiene en el documento j para $i=1,\dots,11$ y $j=1,\dots,6$. En otras palabras, las columnas de M representan los vectores de cada uno de los documentos. La figura 2.4 (d) muestra la matriz de términos por documentos de este ejemplo.

Para normalizar esta matriz se normaliza cada una de sus columnas. La figura 2.4 (e) muestra la matriz de términos por documentos de este ejemplo normalizada.

a) $d = 6$ documentos, los cuales son títulos de páginas de web

- D1: Big apple Circus – Home
- D2: RecipeSource: Autum apple Punch
- D3: Apple recipes
- D4: Welcome to the Big apple Circus
- D5: Custard apple recipes
- D6: The Recipe Place | apple LIME PUNCH

b) $t = 11$ términos

- T1: big
- T2: appl (e)
- T3: circu (s)
- T4: home
- T5: recipesource (recipsourc)
- T6: autum
- T7: punch
- T8: recip (e, es)
- T9: custard
- T10: place
- T11: lime

c) Vectores de los documentos

- $v1 = (0.48, 0.00, 0.48, 0.78, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00)$
- $v2 = (0.00, 0.00, 0.00, 0.00, 0.78, 0.78, 0.48, 0.00, 0.00, 0.00, 0.00)$
- $v3 = (0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.30, 0.00, 0.00, 0.00)$
- $v4 = (0.48, 0.00, 0.48, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00)$
- $v5 = (0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.30, 0.78, 0.00)$
- $v6 = (0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.48, 0.30, 0.00, 0.78, 0.78)$

d) Matriz de términos por documentos

$$M_{11 \times 6} = \begin{pmatrix} 0.48 & 0.00 & 0.00 & 0.48 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.48 & 0.00 & 0.00 & 0.48 & 0.00 & 0.00 \\ 0.78 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.78 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.78 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.48 & 0.00 & 0.00 & 0.00 & 0.48 \\ 0.00 & 0.00 & 0.30 & 0.00 & 0.30 & 0.30 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.78 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.78 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.78 \end{pmatrix}$$

e) Matriz de términos por documentos normalizada

$$M_{11 \times 6} = \begin{pmatrix} 0.4642 & 0.000 & 0.000 & 0.7071 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.4642 & 0.000 & 0.000 & 0.7071 & 0.000 & 0.000 \\ 0.7543 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.6484 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.6484 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.3990 & 0.000 & 0.000 & 0.000 & 0.3871 \\ 0.000 & 0.000 & 1.000 & 0.000 & 0.3590 & 0.2420 \\ 0.000 & 0.000 & 0.000 & 0.000 & 0.9333 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.6291 \\ 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.6291 \end{pmatrix}$$

Figura 2.4 La construcción de la matriz de términos por documentos

2.4 Algoritmos de *clustering* (algoritmos de agrupamiento)

Los algoritmos *agrupamiento* pueden ser clasificados de diferentes maneras. Can y Ozkarahan [Can y Ozkarahan, 1990] proponen la siguiente clasificación de acuerdo con la manera en la que los documentos son distribuidos en los grupos:

- **Algoritmos de Partición.**
- **Algoritmos de Traslape (*overlapping*).**
- **Algoritmos Jerárquicos**

Consideraremos a los Algoritmos de Partición y a los de Traslape como **algoritmos de agrupación no jerárquicos** porque producen una lista de grupos; en otras palabras, estos algoritmos no generan una jerarquía (árbol) de grupos.

2.4.1 Algoritmos de Partición

Los algoritmos de partición dividen el conjunto de objetos en grupos disjuntos de tal forma que cada objeto esté exactamente en un grupo. Los algoritmos dentro de esta categoría son *K-Means*, *Spherical K-Means* y otras variantes de *K-Means*.

Algoritmo *K-Means*

K-Means es el algoritmo más usado en el *agrupamiento* de documentos; está basado en datos numéricos, por lo que si se quiere aplicar para agrupar documentos se necesita usar el VSM.

El algoritmo está basado en la idea de que el *centroide* (media⁸ de un grupo de puntos) representa un grupo [Steinbach *et al.*, 2000]. El algoritmo funciona de la siguiente manera, dado un conjunto de K centroides, encontrar para cada vector de documento cuál es su centroide más cercano, según la distancia euclidiana. Los vectores de documentos cercanos a un centroide forman un nuevo grupo, por lo cual al finalizar este paso se tendrán K nuevos grupos, a los cuales se calcula su centroide de la siguiente manera: calcular la media de los vectores de los documentos del grupo. Estos pasos se repiten hasta que no haya cambios en los grupos. En otras palabras, el algoritmo termina cuando haya convergencia.

K-Means queda descrito de manera formal en el algoritmo 1.

⁸ La media de un grupo S se define como $\frac{1}{|S|} \sum_{p \in S} p$, donde $|S|$ es el número de elementos en el grupo S .

Algoritmo 1 El algoritmo *K-Means*

Input: número de grupos k , conjunto de n vectores de documentos

- 1: Selecciona k vectores de documentos como centroides iniciales
 - 2: **repeat**
 - 3: Forma k grupos asignando cada vector de documento a su centroide más cercano
 - 4: Actualizar el centroide de cada grupo
 - 5: **until** que no haya cambios
-

La figura 2.5 muestra de manera gráfica cómo *K-Means* encuentra tres grupos a partir de tres centroides en 4 iteraciones. La figura muestra cómo se van moviendo los centroides en cada iteración del algoritmo. Los centroides están representados por el símbolo '+' y los puntos que pertenecen al mismo grupo tienen la misma figura. La ventaja de *K-Means* es que su complejidad es de $O(nkt)$, donde k es el número deseado de grupos, n es el número de puntos o documentos y t es el número de iteraciones [Wang, 2005].

Desafortunadamente el desempeño de este algoritmo no es muy bueno cuando se aplica a una colección de documentos [Zamir y Etzioni, 1998]. Sus principales desventajas son:

1. Se necesita saber, por adelantado, el número de grupos en los que se dividen la colección de documentos (generalmente no hay forma de saber cuántos grupos existen).
2. No se especifica cómo hacer la elección inicial de los centroides, por lo cual generalmente se hace de manera aleatoria. Es importante señalar que la calidad de los grupos obtenidos depende en gran medida de la elección inicial de los centroides.
3. Se obtienen buenos resultados sólo cuando los datos pueden ser agrupados en grupos con formas esféricas.

Steinbach [Steinbach *et al.*, 2000] menciona que hay varias formas de mejorar el desempeño del algoritmo *K-Means*, pero para mantener las cosas simples él propone seleccionar los centroides aleatoriamente y actualizarlos de manera incremental en vez de hacer la actualización al final; es decir, cada vez que un punto es asignado a un grupo, el centroide de este grupo se actualiza. Steinbach dice que la actualización incremental de centroides produce mejores resultados. Por esta razón se decidió utilizar el algoritmo *K-Means* con esta modificación en el sistema *Conquiro*.

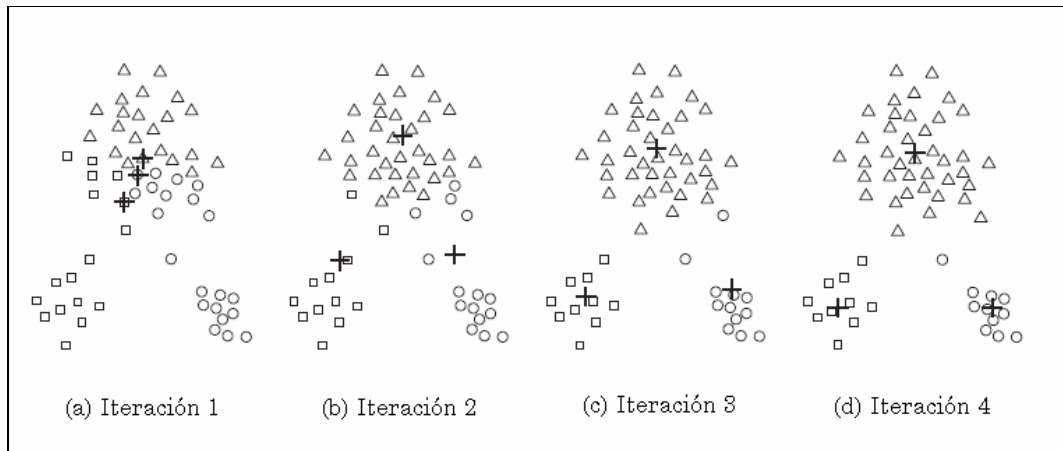


Figura 2.5 Iteraciones del algoritmo *K-Means* para generar 3 grupos [Tan et al., 2005]

Algoritmo *Spherical K-Means*

La versión clásica de *K-Means* utiliza la distancia euclidiana; sin embargo, esta medida es a menudo inapropiada para el *agrupamiento* de documentos. La medida que comúnmente se usa en IR es la semejanza de coseno. El algoritmo de *K-Means* puede ser adaptado para usar la semejanza de coseno para originar el algoritmo *Spherical K-Means*; el nombre se debe a que el algoritmo utiliza vectores que están en la esfera unitaria [Dhillon y Modha, 2001].

El algoritmo funciona de la siguiente manera, al inicio se crea K grupos (particiones) de manera arbitraria, a los cuales llamaremos particiones antiguas, y se calcula el centroide para cada grupo (vector de concepto). Para cada uno de los vectores de documentos se encuentra el centroide más cercano, según la semejanza de coseno. Los vectores cercanos a un centroide forman un nuevo grupo (partición), por lo cual al finalizar este paso se tendrá K nuevos grupos (particiones), a los cuales llamaremos particiones actuales. Se calculan los centroides de las particiones actuales, luego se evalúan las particiones antiguas y particiones actuales. El algoritmo termina cuando la diferencia de ambas evaluaciones es menor que un umbral; en caso contrario las particiones actuales se utilizarán como particiones antiguas y se vuelve a calcular las actuales.

Antes de describir el algoritmo de manera formal se darán algunas definiciones.

Sea n el número de documentos y d el número de palabras, los vectores de los documentos se representan como $\{x_1, x_2, \dots, x_n\}$ donde cada $x_i \in \mathbb{R}^d$. Sea $\pi_1, \pi_2, \dots, \pi_k$ una partición de los vectores de documentos en k grupos disjuntos, tales que

$$\bigcup_{j=1}^k \pi_j = \{x_1, x_2, \dots, x_n\} \text{ donde } \pi_j \cap \pi_l = \emptyset \text{ si } j \neq l$$

Para cada grupo $1 \leq j \leq k$, el **vector media** o el **centroide** de los vectores de documentos contenidos en el grupo π_j es

$$m_j = \frac{1}{n_j} \sum_{x \in \pi_j} \mathbf{x}$$

donde n_j es el número de vectores de documentos en π_j . Obsérvese que el vector m_j no necesita ser unitario; para capturar su dirección se define el siguiente **vector de concepto** como:

$$c_j = \frac{m_j}{\|m_j\|}$$

El vector de concepto c_j tiene la siguiente propiedad: para cualquier vector unitario \mathbf{z} en \mathbb{R}^d ; por la desigualdad de Cauchy-Schwarz se tiene que

$$\sum_{x \in \pi_j} \mathbf{x}^T \mathbf{z} \leq \sum_{x \in \pi_j} \mathbf{x}^T c_j$$

Entonces, el vector de concepto podría verse como el vector que es cercano respecto de la semejanza de coseno a todos los vectores en el grupo π_j .

Se define la **coherencia** o **calidad** de cada grupo $\pi_j, 1 \leq j \leq k$ como

$$\sum_{x \in \pi_j} \mathbf{x}^T c_j$$

Para medir la calidad de una partición dada $\{\pi_j\}_{j=1}^k$ se usa la siguiente **función objetivo**:

$$Q(\{\pi_j\}_{j=1}^k) = \sum_{j=1}^k \sum_{x \in \pi_j} \mathbf{x}^T c_j$$

El propósito es encontrar una agrupación que maximice el valor de la función objetivo descrita.

El algoritmo *Spherical K-Means* es un proceso iterativo que genera una secuencia de particiones

$$\{\pi_l^{(0)}\}_{l=1}^k, \{\pi_l^{(1)}\}_{l=1}^k, \dots, \{\pi_l^{(t)}\}_{l=1}^k \text{ con } Q(\{\pi_j^{(t+1)}\}_{j=1}^k) \geq Q(\{\pi_j^{(t)}\}_{j=1}^k)$$

Para resaltar la relación entre las particiones $\{\pi_l^{(t)}\}_{l=1}^k$ y $\{\pi_l^{(t+1)}\}_{l=1}^k$ se denota $\{\pi_l^{(t+1)}\}_{l=1}^k$ por $nextKM(\{\pi_l^{(t)}\}_{l=1}^k)$. Con estas definiciones se describe de manera formal *Spherical K-Means* en el algoritmo 2.

La complejidad del algoritmo es de $O(k\tau)$ donde k es el número de grupos y τ es el número de iteraciones del algoritmo [Dhillon *et al.*, 2001].

La ventaja del algoritmo es que se ha encontrado que obtiene buenos resultados para grandes cantidades de documentos; sin embargo produce resultados pobres cuando el tamaño de los grupos es pequeño (25-30 documentos) [Dhillon *et al.*, 2002].

Este algoritmo está incluido en el sistema y es utilizado cuando se especifica realizar el *agrupamiento* de documentos con *K-Means*, pero utilizando como medida la semejanza de coseno (parámetro *Distance/Similarity*, para mayor referencia consultar la sección 4.2).

Algoritmo 2 El algoritmo *Spherical K-Means*

Input: número de grupos k , n vectores de documentos, tolerancia $tol > 0$

1: Iniciar con una partición arbitraria $\{\pi_j^{(0)}\}_{j=1}^k$ y los vectores de concepto $\{c_j^{(0)}\}_{j=1}^k$ asociados con la partición. Poner el índice de la iteración $t = 0$.

2: **for each** vector de documento $x_i, 1 \leq i \leq n$

3: Encontrar el vector de concepto más cercano a x_i respecto de la semejanza de coseno.

4: Calcular la nueva partición $\{\pi_j^{(t+1)}\}_{j=1}^k = nextKM\left(\{\pi_j^{(t)}\}_{j=1}^k\right)$ inducida por los vectores de conceptos anteriores $\{c_j^{(t)}\}_{j=1}^k$:

$$\pi_j^{(t+1)} = \left\{ x \in \{x_i\}_{i=1}^n : x^T c_j^{(t)} > x^T c_l^{(t)}, 1 \leq l \leq n, l \neq j \right\}, 1 \leq j \leq k$$

En otras palabras, $\pi_j^{(t+1)}$ es el conjunto de todos los vectores de documentos cercanos al vector de concepto $c_j^{(t)}$.

En caso de que un vector de documento sea cercano a más de un vector de concepto entonces éste es asignado aleatoriamente a una de las particiones representadas por estos vectores de concepto.

5: **end**

6: Calcular los nuevos vectores de concepto correspondientes a las particiones calculadas en el paso 2:

$$c_j^{(t+1)} = \frac{m_j^{(t+1)}}{\|m_j^{(t+1)}\|}, 1 \leq j \leq k$$

donde $m_j^{(t+1)}$ es el centroide o media de los vectores de documentos del grupo $\pi_j^{(t+1)}$.

7: **if** $\left[Q\left(nextKM\left(\{\pi_l^{(t)}\}_{l=1}^k\right)\right) - Q\left(\{\pi_l^{(t)}\}_{l=1}^k\right) > tol \right]$ **then**

8: $t = t + 1$

9: continúa con el paso 2

10: **else**

11: fin del algoritmo

12: **end**

2.4.2 Algoritmos de Traslape (*overlapping*)

Los algoritmos de traslape (*overlapping*) dividen el conjunto de objetos en grupos de tal modo que un objeto puede pertenecer a más de un grupo. Un ejemplo de este tipo de algoritmo es el *Suffix Tree Clustering*.

Algoritmo *Suffix Tree Clustering*

Este algoritmo fue diseñado con el propósito de agrupar documentos. Zamir y Etzioni [Zamir y Etzioni, 1998] consideran que los métodos de *agrupamiento (clustering)* deben cumplir con los siguientes requisitos:

- 1) **Relevancia:** El método debe producir grupos que contengan documentos relevantes a la consulta del usuario.
- 2) **Sumarios navegables (*browsable summaries*):** El usuario necesita determinar a simple vista si el contenido del grupo es de su interés, por lo cual los métodos necesitan presentar los resultados de tal manera que el usuario pueda explorarlos sin tener que ver listas interminables de documentos.
- 3) **Traslape:** Existen documentos que están relacionados con varios temas, por lo cual es importante evitar asignar los documentos a un solo grupo.
- 4) ***Snippet-tolerance*:** Los métodos deben producir grupos de gran calidad, incluso cuando sólo tengan acceso a los *snippets* (resúmenes de las páginas) proporcionados por los motores de búsqueda.
- 5) **Velocidad:** El algoritmo de *agrupamiento (clustering)* debe ser capaz de agrupar cientos de *snippets* en pocos segundos.
- 6) **Incrementabilidad:** Para ahorrar tiempo, el método debe de procesar cada *snippet* conforme lo va recibiendo.

El algoritmo *Suffix Tree Clustering (STC)* por sus siglas en inglés fue diseñado por Zamir y Etzioni [Zamir y Etzioni, 1998] con el objetivo de cumplir con estos requisitos. Este algoritmo es incremental, novedoso y su complejidad es de $O(n)$, donde n es el número de documentos. *STC* no trata al documento como un conjunto de términos (VSM), sino como una cadena de caracteres, con el fin de crear grupos de documentos que tengan una frase en común.

La idea del algoritmo es agrupar documentos que compartan frases (secuencia ordenada de palabras) en común. El algoritmo utiliza un árbol de sufijos generalizado para construir el índice de frases. El árbol se recorre y los nodos que tengan un conjunto de documentos en común se unen. Las etiquetas de los *grupos* son las frases más relevantes.

El algoritmo *STC* está compuesto por tres fases; en la primera se encuentran las frases comunes y los documentos que las comparten. Esto se realiza con la estructura de datos llamada **árbol de sufijos generalizado (*generalized suffix tree*)** [Weiss, 2001],

que es un árbol de sufijos (*suffix tree*) [Ukkonen, 1995] para un conjunto de cadenas. Esta estructura es construida en tiempo lineal respecto del número de documentos y contiene todas las frases de la colección y los documentos que las comparten; además éstos pueden ser agregados incrementalmente al árbol. La figura 2.6 muestra un ejemplo de árbol de sufijos generalizado para las cadenas “*cat ate cheese*”, “*mouse ate cheese too*” y “*cat ate mouse too*”.

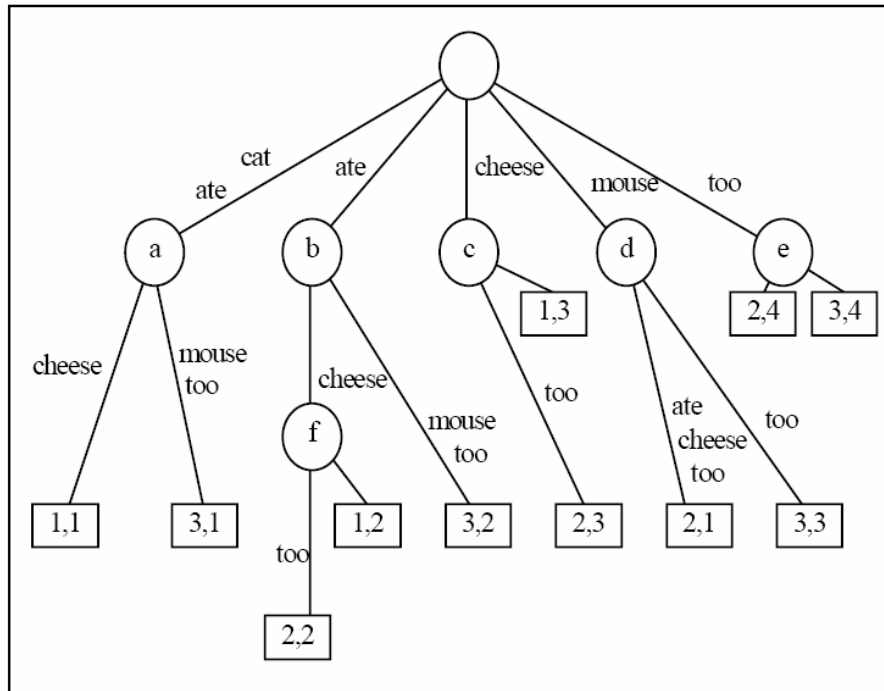


Figura 2.6 Árbol generalizado de sufijos de las cadenas “*cat ate cheese*”, “*mouse ate cheese too*” y “*cat ate mouse too*” [Zamir y Etzioni, 1998].

En la segunda fase se crea una lista de grupos, llamada grupos base, de la siguiente manera: todas las frases que ocurren en más de un documento son consideradas como grupos base. A cada uno de estos grupos se asigna una puntuación. La puntuación toma en cuenta el número de documentos que el grupo contiene y el número de palabras que contenga la frase, cuya calificación no sea cero (las palabras que no estén en la lista de palabras comunes *stop words* y que aparezcan en más de tres documentos). Los grupos cuya puntuación no exceda cierto umbral son rechazados.

En la tercera fase se crea una gráfica de grupos donde los nodos representan los grupos base y las aristas entre los grupos (nodos) indican que tienen documentos en común. Las subgráficas de esta gráfica representan los grupos resultados y las frases asociadas con estos grupos son usadas como descripciones. La figura 2.7 muestra la gráfica generada por el algoritmo para las cadenas de la figura 2.6.

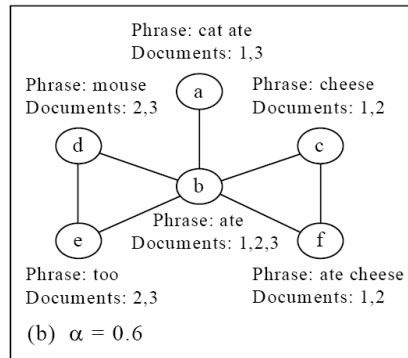


Figura 2.7 Gráfica generada por el algoritmo *STC* para las cadenas “cat ate cheese”, “mouse ate cheese too” y “cat ate mouse too”.

El algoritmo 3 muestra el pseudocódigo del *STC* [Weiss, 2001].

La ventaja del algoritmo *STC* es su velocidad, ya que su complejidad es lineal y procesa incrementalmente los documentos. Sus autores señalan que da mejores resultados que los algoritmos basados en el *VSM* [Zamir y Etzioni, 1998]. La desventaja que *STC* tiene es que genera una lista de grupos, lo cual no permite que el usuario visualice los temas en diferentes niveles de detalle.

Algoritmo 3 Pseudocódigo del algoritmo *Suffix Tree Clustering*

```

1: Dividir el texto en oraciones que contengan palabras;
2: /* Fase 1. Creación del árbol generalizado de sufijos de todas las oraciones */
3: for each documento
4:   for each oración
5:     Hacer la reducción de términos a la raíz de las palabras.
6:     /* Las palabras que están en la lista de términos comunes (stop words) o que aparecen en
       * más de cierto porcentaje de documentos o en menos de cierto número de documentos
       * son ignoradas. Llamaremos a estas palabras términos comunes.
       */
7:     If tamaño de la oración > 0
8:       /* elimina las oraciones que inicien o terminen con términos comunes */
9:       Inserta las oraciones y todas sus subcadenas en el árbol de sufijos
       generalizado, actualizando los nodos internos con el índice del
       documento actual;
10:    end
11:  end
12: end
13: /* Fase 2. Construcción de la lista de grupos base */
14: for each nodo en el árbol
15:   if el número de documentos en el nodo > 2
16:     If calificación del nodo > umbral
17:       Agrega el nodo a la lista de grupos base.
18:     end
19:   end
20: end
21: /* Fase 3. Unión de los grupos base */
22: Construir una gráfica donde los nodos sean grupos base y haya una arista entre
    el nodo A y B, si y sólo si el número de documentos que tienen en común A y B
    es más grande que un umbral.

```

2.4.3 Algoritmos jerárquicos

Los algoritmos jerárquicos crean una jerarquía de grupos que se representa con un árbol donde el nodo raíz es el grupo que contiene todos los documentos de la colección. Entre los algoritmos de esta categoría están *Bisecting K-Means* y los métodos de *agrupamiento* jerárquicos.

Algoritmo *Bisecting K-Means*

Bisecting K-Means es una variante del algoritmo *K-Means*. Este algoritmo inicia con un solo grupo que contiene todos los vectores de documentos; este grupo se va dividiendo en grupos más pequeños de la siguiente manera: se selecciona el grupo más grande, se divide en dos subgrupos usando el algoritmo *K-Means* (paso de bisección), se ejecuta el paso de bisección α veces; de las α particiones (cada partición tiene dos subgrupos) obtenidas se escoge la que tenga el más bajo *SSE* (*Sum of the Squared Error*) o la máxima cohesión total; estos pasos se repiten hasta que se haya producido K grupos.

Bisecting K-Means está descrito de manera formal en el algoritmo 4.

Algoritmo 4 El algoritmo *Bisecting K-Means*

Input: k número de grupos, n vectores de documentos, ITER número de veces que se ejecuta *K-means*.

- 1: Inicializar la lista de grupos con el grupo que contiene los n vectores de documentos.
 - 2: **repeat**
 - 3: Remover un grupo de la lista
 - 4: **for** $i = 1$ **to** ITER **do**
 - 5: Obtener 2 subgrupos utilizando el algoritmo *K-Means*
 - 6: **end for**
 - 7: Selecciona los subgrupos que tengan un bajo *SSE* o la máxima cohesión total
 - 8: Agrega estos subgrupos a la lista de grupos
-

Tan [Tan *et al.*, 2005] define *SSE* como la función que mide la distancia euclidiana al centroide más cercano y entonces calcula la suma del error ajustado. La función *SSE* se define como:

$$SSE = \sum_{i=1}^K \sum_{x \in C_i} dist(c_i, x)^2$$

donde *dist* es la distancia euclidiana entre dos objetos y c_i es el centroide (media) del grupo tal como se definió en *K-Means*.

Dados dos diferentes conjuntos de grupos, generados por dos diferentes ejecuciones de *K-Means*, se elige aquella con el menor *SSE*, ya que los centroides de esta agrupación son una buena representación de los puntos del grupo.

Para el caso de semejanza de coseno se define la función análoga a *SSE*, que maximiza la semejanza de los puntos con el centroide del grupo; esta función es conocida como la **cohesión** del grupo. Se define la **cohesión total** de la siguiente manera:

$$\text{Cohesión total} = \sum_{i=1}^K \sum_{x \in C_i} \cos(x, c_i)$$

donde c_i es el centroide (media) del grupo tal como se definió en *K-Means*.

Steinbach [Steinbach *et al.*, 2000] menciona que *Bisecting K-Means* puede producir una lista o una jerarquía de grupos; sin embargo, estrictamente hablando este algoritmo es de partición de tipo jerárquico de división (más adelante se explica en qué consisten los métodos jerárquicos de división). Para este trabajo se utilizó *Bisecting K-Means* para construir una jerarquía de grupos binaria, donde el número de grupos (nodos) es $n + (n - 1)$ y n es el número de datos. La figura 2.8 muestra un ejemplo del árbol que se genera con este algoritmo para el conjunto de datos 0,1,2,3,4,5,6,7,8,9,10,11.

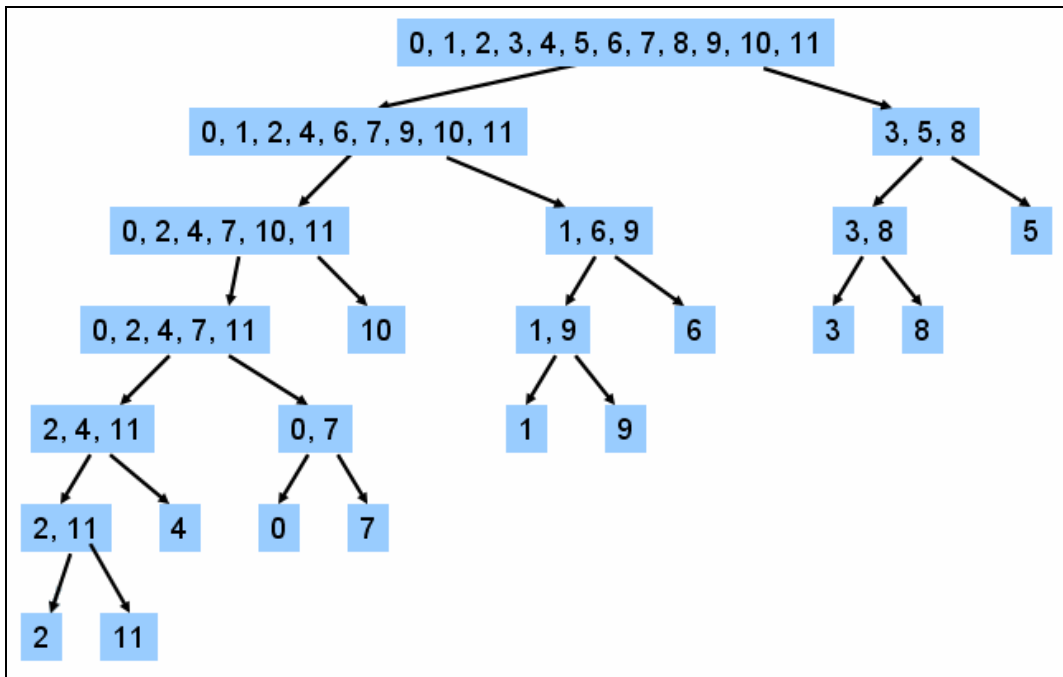


Figura 2.8 Árbol generado con *Bisecting K-Means*.

La complejidad del algoritmo es lineal en el número de documentos.

Métodos de agrupamiento jerárquicos

Los métodos jerárquicos construyen una jerarquía de grupos; en otras palabras, un árbol de grupos, conocido como **dendrograma**; la figura 2.9 muestra en el inciso *a)* el dendrograma; y en el inciso *b)* el árbol binario de este dendrograma. Este árbol es

binario porque cada nodo tiene dos hijos. Los métodos de *agrupamiento* jerárquicos se basan en el VSM para agrupar la información.

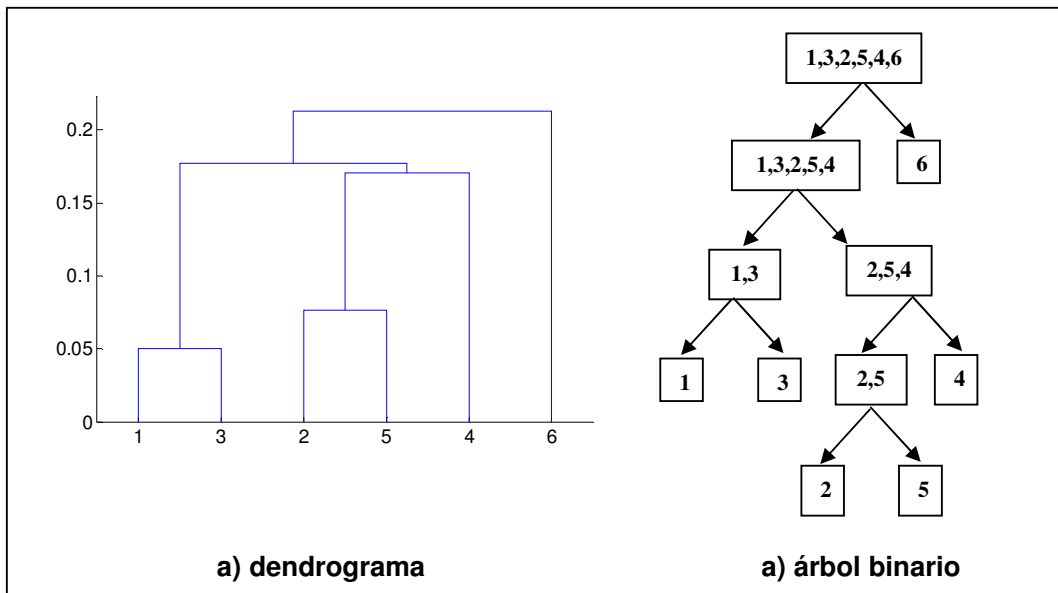


Figura 2.9 Dendrograma generado por los métodos de *Agrupamiento* jerárquicos.

Los métodos de *agrupamiento* jerárquicos se dividen en dos grupos [Jain y Dubes, 1988]:

- **División**
- **Aglomeración**

Los métodos de **división jerárquicos** inician con un grupo que contiene todos los vectores de documentos y divide sucesivamente los grupos resultantes hasta obtener grupos con solo un vector de documento. Es importante señalar que estos métodos no son muy usados.

Los métodos de **aglomeración jerárquicos** (*Hierarchical Agglomerative Clustering, HAC*⁹ por sus siglas en inglés) inician con grupos que contienen solo un vector de documento que sucesivamente une el par de grupos más similares hasta que todos los grupos se unen en un solo grupo, el cual es el nivel más alto de la jerarquía (la raíz del árbol). El algoritmo 5 describe de manera formal la construcción de la jerarquía de los métodos de aglomeración.

La **matriz de semejanzas** que utiliza el algoritmo es una matriz cuadrada de $n \times n$, donde n es el número de documentos. Ésta contiene las distancias o semejanzas¹⁰ entre los vectores de documentos i y j . Las medidas de distancia o semejanza cumplen

⁹ De ahora en adelante se utilizará esta abreviación para hacer referencia a este tipo de algoritmos.

¹⁰ Para calcular la distancia o semejanza entre puntos se puede utilizar diferentes medidas; sin embargo para la aplicación desarrollada en este trabajo para calcular la distancia se utilizó distancia euclidiana; y para calcular la semejanza, la semejanza de coseno.

con la propiedad de simetría, es decir, $\text{Distancia}(X,Y) = \text{Distancia}(Y,X)$, por lo cual la matriz de semejanzas es triangular inferior [Frakes y Baeza-Yates, 1992]. La figura 2.10 muestra un ejemplo de la matriz de semejanzas.

$$M = \begin{pmatrix} 0 & 0 & 0 & \dots & 0 \\ a_{21} & 0 & 0 & \dots & 0 \\ a_{31} & a_{32} & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{n(n-1)} & 0 \end{pmatrix}$$

Figura 2.10 Matriz de semejanzas.

Algoritmo 5 Método Jerárquico de Aglomeración

Input: conjunto de vectores de documentos D

Output: árbol T

- 1: $C := \{\{o\} | o \in D\}$ y $T = \emptyset$
 - 2: **while** $|C| > 1$ **do**
 - 3: Selecciona los grupos (a,b) más cercanos tal que $a,b \in C$
 - 4: Crea un nuevo grupo $c = a \cup b$ y sea a y b hijos de c en T
 - 5: $C := C \cup \{c\}$;
 - 6: $C := C \setminus \{a,b\}$;
 - 7: **foreach** $x \in C$ **do**
 - 8: Calcular las distancias (semejanzas) entre x y c ;
 - 9: **return** T
-

El algoritmo 5 tiene tres puntos principales: la selección de los grupos más cercanos (paso 3), la creación del nuevo grupo (paso 4) y el cálculo de las distancias o semejanzas entre dos grupos (paso 7). En el paso 3 los grupos más cercanos son los que tienen el valor más pequeño (distancia euclidiana) o el valor más grande (semejanza de coseno) en la matriz M de semejanzas; es decir, si el grupo i y j son los más cercanos, entonces la entrada m_{ij} tendrá el valor más pequeño en distancia euclidiana o el valor más grande respecto de la semejanza de coseno.¹¹ En el paso 4 la creación del nuevo grupo implica reducir la dimensión de la matriz de semejanzas M en 1. Esta reducción se hace de la siguiente manera: supongamos que se tienen 5 grupos ($C1, C2, \dots, C5$), cuya matriz de semejanzas M se muestra en el inciso a) de la figura 2.11. Los grupos $C2$

¹¹ Para dejar clara esta idea, consúltese la sección 2.3.3.

y $C5$ son los más cercanos, entonces se unen para formar un nuevo grupo. Para agregar el nuevo grupo $C2C5$ a la matriz de semejanzas M , se eliminan los renglones y columnas correspondientes a los grupos $C2$ y $C5$ (inciso b) figura 2.11) y se agrega la columna y renglón correspondientes al grupo $C2C5$ (inciso c) figura 2.11); la distancia o semejanza que el grupo $C2C5$ tiene con los demás grupos se calcula en el paso 7.

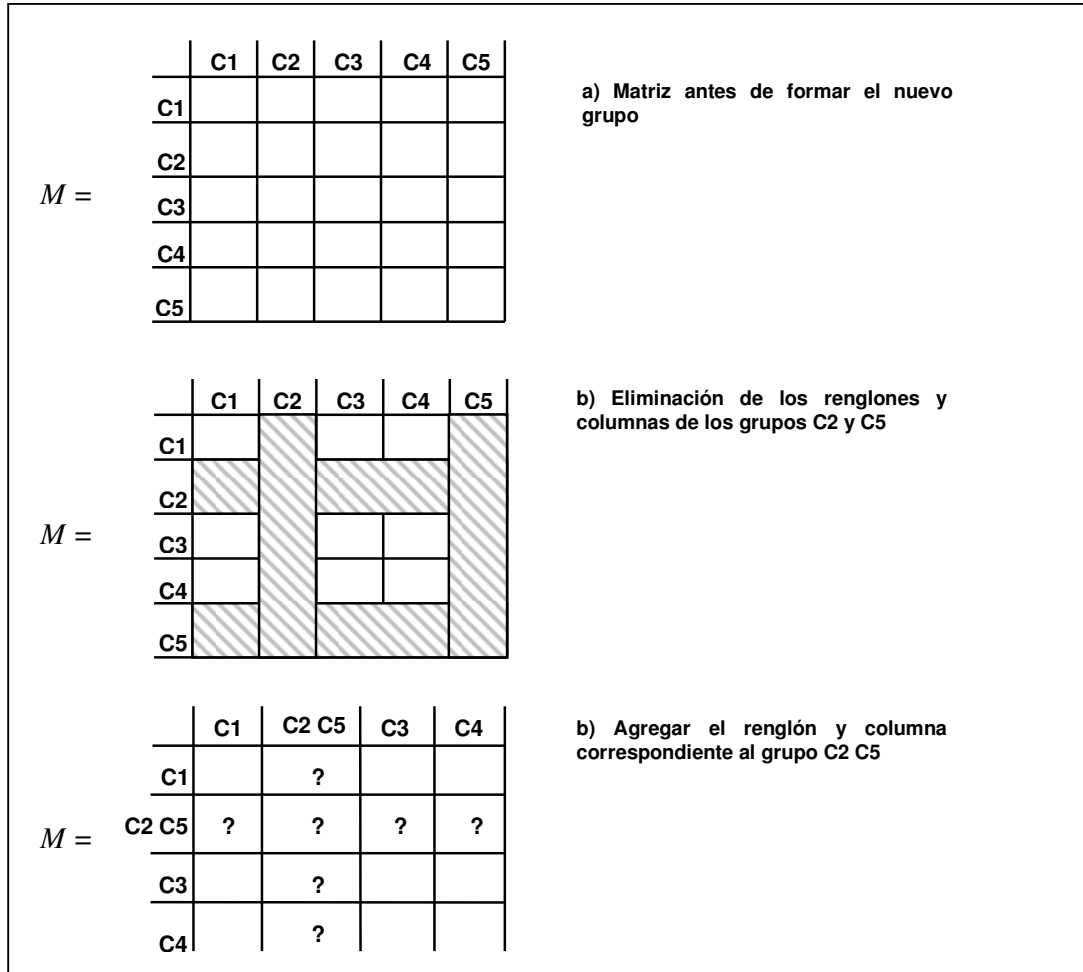


Figura 2.11 Reducción de la dimensión de la matriz de semejanza M cuando se crea un nuevo grupo.

En el paso 7 para calcular la distancia o la semejanza que el nuevo grupo tiene con el resto, se utiliza alguno de los siguientes métodos de ligado: **ligado simple** (*single linkage*), **ligado completo** (*complete linkage*), **UPGMA**, entre otros.¹² Una vez calculadas las distancias o semejanzas del nuevo grupo con el resto, la matriz de semejanzas se actualiza.

De acuerdo con los anteriores métodos de ligado, tenemos los siguientes tipos de algoritmos *HAC*: *HAC (single)*, *HAC (complete)*, *HAC (UPGMA)*. El algoritmo 5 se aplica a los tipos de *HAC*, excepto por el paso 7, que varía dependiendo del método de ligado que se utilice.

A continuación se explica en qué consiste cada uno de los métodos de ligado.

¹² Para mayor referencia, véase Dubes y Jain, 1988.

Ligado simple (*single linkage*): También es conocido como “el vecino más cercano”. En este procedimiento la distancia entre dos grupos se define como la mínima distancia entre ellos, es decir, la distancia entre los vectores de documentos más cercanos. En la figura 2.12 se muestra gráficamente.

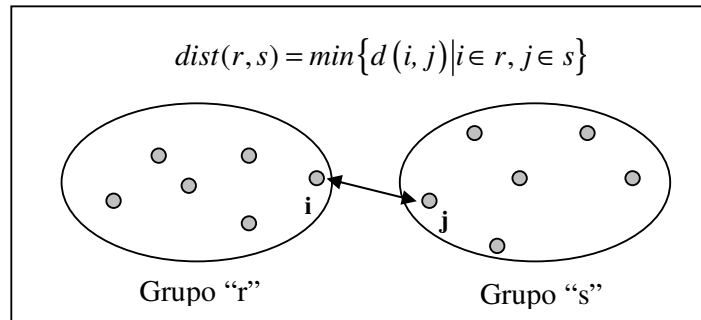


Figura 2.12 Ligado simple.

Si en vez de utilizar la distancia, se utiliza la semejanza de coseno la fórmula cambia a

$$\text{semejanza}(r, s) = \max\{\cos(i, j) | i \in r, j \in s\}$$

Tan [Tan *et al.*, 2005] menciona que este método maneja grupos de formas no elípticas; sin embargo es sensible al ruido.

La figura 2.13 muestra los resultados de aplicar este procedimiento a un conjunto de seis puntos. El inciso *a*) de la figura muestra los grupos anidados como una secuencia de elipses anidadas, los números asociados con las elipses indican el orden en que fueron agrupados los datos. El inciso *b*) muestra la representación de los grupos en un dendrograma; la altura del dendrograma refleja la distancia entre dos grupos.

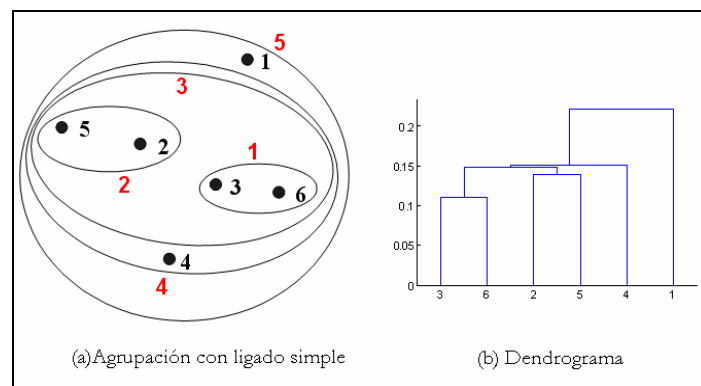


Figura 2.13 Agrupación de seis puntos con el método de ligado simple [Tan *et al.*, 2005].

Ligado completo (*complete linkage*): También conocido como “el vecino más lejano”, este procedimiento es el opuesto del ligado simple porque la distancia entre dos grupos se define como la máxima distancia entre ellos. La figura 2.14 lo muestra gráficamente.

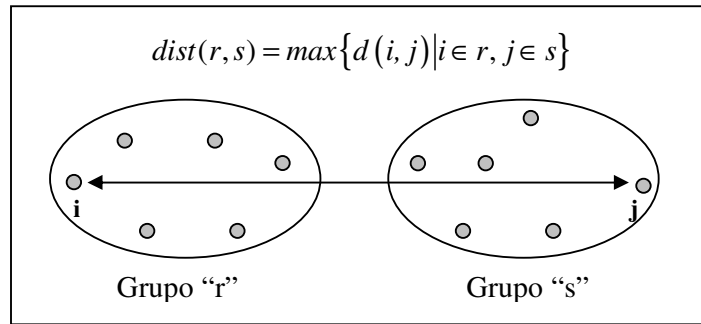


Figura 2.14 Ligado completo

Si en vez de utilizar la distancia se utiliza la semejanza de coseno la fórmula cambia a

$$\text{semejanza}(r, s) = \min\{\cos(i, j) | i \in r, j \in s\}$$

Con este método cada miembro del grupo tiende a ser más cercano a todos los miembros del mismo grupo y más lejano a los de otros grupos, por lo cual, al unir dos grupos sus elementos son muy cercanos entre sí. Debido a esto, el ligado completo genera grupos más compactos.

Tan [Tan *et al.*, 2005] menciona que este método es menos susceptible al ruido, sin embargo fragmenta grupos grandes.

La figura 2.15 muestra los resultados de aplicar este procedimiento al mismo conjunto de puntos de la figura 2.13. El inciso *a)* de la figura 2.15 muestra los grupos anidados como una secuencia de elipses anidadas, los números asociados con las elipses indican el orden en que fueron agrupados los datos. El inciso *b)* muestra la representación de los grupos en un dendrograma, la altura del dendrograma refleja la distancia entre dos grupos.

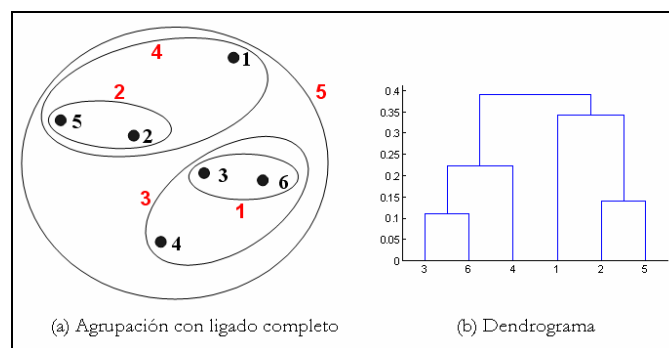


Figura 2.15 Agrupación de seis puntos con el método de ligado completo [Tan *et al.*, 2005].

UPGMA (*Unweighted Pair-Group Method using Arithmetic Averages*): También es conocido como “promedio de grupo (*group average*)”. En este procedimiento la distancia entre dos grupos se define como el promedio de las distancias entre todos los pares de puntos de los dos grupos. La distancia entre dos grupos r y s cuyos tamaños son n_r y n_s se define en la fórmula de la figura 2.16.

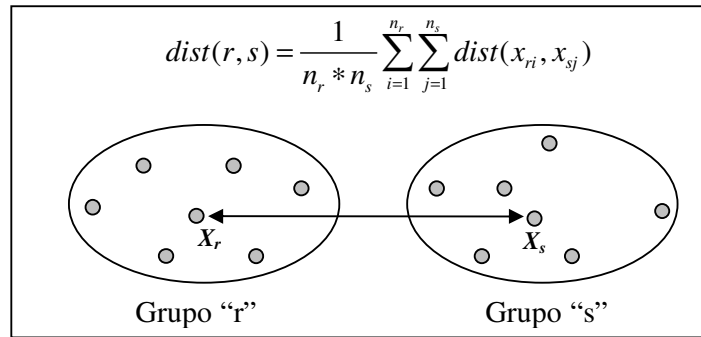


Figura 2.16 Método UPGMA.

Si en vez de utilizar la distancia se utiliza la semejanza de coseno entonces sustituir en la fórmula *dist* por *cos*.

La figura 2.17 muestra los resultados de aplicar este procedimiento al mismo conjunto de puntos de la figura 2.13. El inciso a) de la figura 2.17 muestra los grupos anidados como una secuencia de elipses anidadas, los números asociados con las elipses indican el orden en que fueron agrupados los datos. El inciso b) muestra la representación de los grupos en un dendrograma, la altura del dendrograma refleja la distancia entre dos grupos.

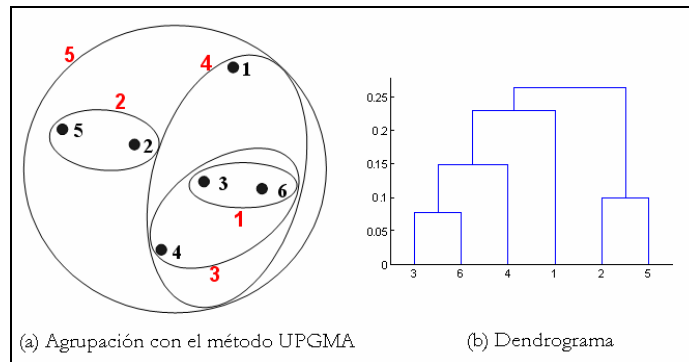


Figura 2.17 Agrupación de seis puntos con el método UPGMA [Tan *et al.*, 2005].

La complejidad de *HAC (single)* y *HAC (UPGMA)* es de $O(n^2)$, mientras que el *HAC (complete)* es de $O(n^3)$ [Zamir y Etzioni, 1998].

La ventaja que tienen los algoritmos *HAC* es que la forma de organizar los grupos permite visualizar la información en diferentes niveles de detalle, de lo general a lo más específico. Sin embargo tiene varias desventajas: estos algoritmos son lentos cuando se utilizan para organizar grandes colecciones de documentos debido a su complejidad computacional. Otra desventaja es que el algoritmo no permite hacer ajustes una vez que se unen dos grupos, lo cual ocasiona que el algoritmo en algunos casos pueda producir resultados pobres. Finalmente, las jerarquías que estos algoritmos generan son binarias; desafortunadamente para el caso de documentos, un tema puede

tener más de dos subtemas, esto genera que las jerarquías tengan varios niveles de profundidad, lo cual no facilita su exploración.

Maarek [Maarek *et al.*, 2000] propone un algoritmo óptimo para realizar un agrupamiento efímero (*ephemeral¹³ clustering*) de documentos, el cual se basa en *HAC (complete)* y genera jerarquías más compactas. Se hará referencia a este algoritmo como *HAC (complete) dendrogram pruning*.

Algoritmo HAC (complete) dendrogram pruning

Este algoritmo genera una estructura más compacta que el dendrograma binario generado por los algoritmos *HAC*. La figura 2.18 muestra un dendrograma binario (lado izquierdo) y el dendrograma modificado con menos niveles generado por el algoritmo (lado derecho). Para evidenciar que el dendrograma generado por este algoritmo es más compacto en la figura 2.19 se muestran los árboles correspondientes a los dendrogramas de la figura 2.18.

Antes de describir el algoritmo es necesario dar algunas definiciones.

Los grupos se consideran como un conjunto de documentos.

Por simplicidad se asume que dos documentos distintos no tienen semejanza igual a 1.0, por lo cual está en uno de los siguientes diez valores: 0, 0.1, 0.2, ..., 0.9.

La semejanza s se redondea de la siguiente manera: $\lfloor 10s \rfloor / 10$ donde $\lfloor 10s \rfloor$ denota el mayor entero menor o igual que $10s$.

El **valor de semejanza** de un grupo c se define como la mínima semejanza entre un par de miembros del grupo c .

La **semejanza entre un par de grupos** (c_1, c_2) se define como la mínima semejanza entre dos miembros de la unión de c_1 y c_2 , lo cual corresponde con el ligado completo.

La matriz de semejanzas contiene el valor de semejanza que cada par de grupos no marcados¹⁴ (c_1, c_2) tiene. En particular, la diagonal de la matriz corresponde al valor de semejanza del grupo c .

Se utiliza diez cubos (*buckets*) llamados 0,0.1, ..., 0.9. En el cubo θ están los pares de grupos (c_1, c_2) cuyo valor de semejanza es θ . El “orden” de los grupos en el cubo es arbitrario y los cubos se actualizan en cada iteración. En el proceso de actualización se utiliza una matriz llamada “**matriz de apuntadores actuales**”, la cual apunta a la posición que el par de grupos (c_1, c_2) tiene en el cubo.

En el algoritmo 6 se describe formalmente el algoritmo *HAC (complete) dendrogram pruning*.

¹³ Maarek llama agrupación efímera al utilizar los resultados de los algoritmos de agrupación para hacer exploración interactiva.

¹⁴ Los grupos no marcados son los que se pueden elegir para formar un nuevo grupo.

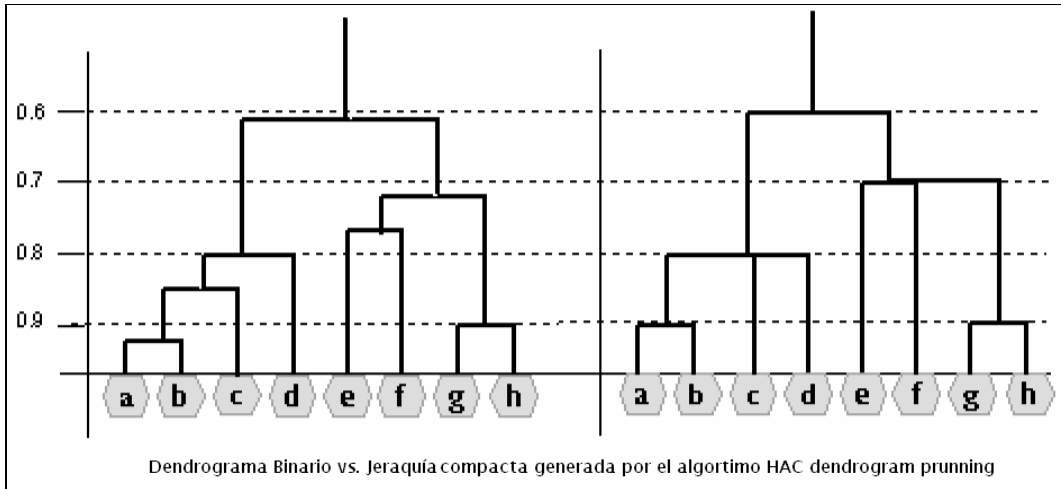


Figura 2.18 Dendrograma binario vs. Dendrograma Compacto generado por el algoritmo [Maarek *et al.*, 2000].

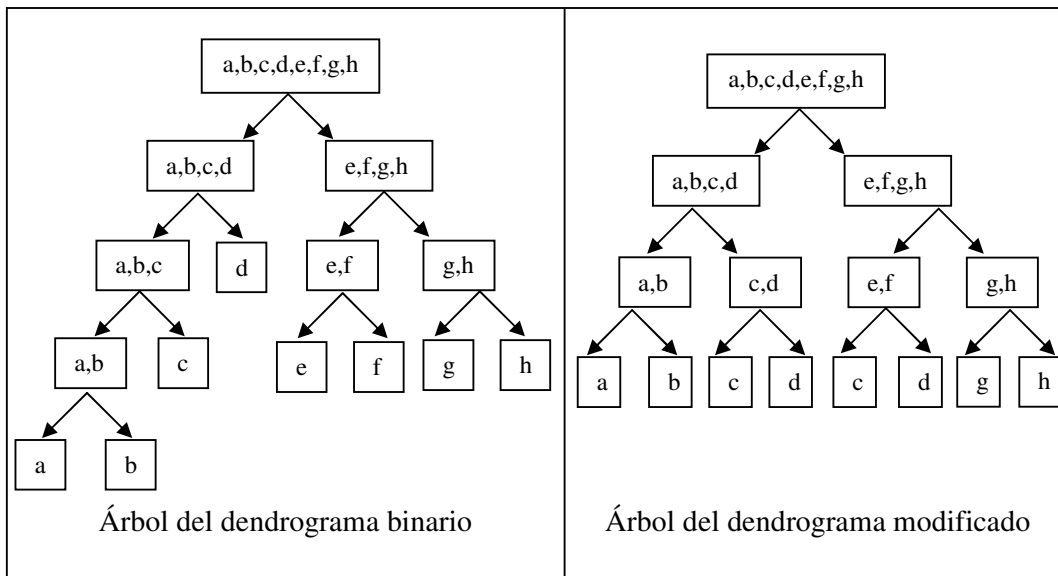


Figura 2.19 Árboles de los dendrogramas de la figura 2.18.

La complejidad del algoritmo es de $O(n^2)$. El análisis que Maarek hace es el siguiente:

- **Paso 1.** La inicialización toma: $O(n^2)$ para inicializar la matriz de semejanzas, $O(n^2)$ para inicializar los cubos y $O(n^2)$ para inicializar la matriz de apuntadores actuales.
- **Paso 2.** Este paso toma $O(1)$.
- **Paso 3.** Este paso toma $O(n)$.

- **Paso 4.** Hay sólo $n-1$ iteraciones, debido a que en cada iteración el número de grupos no marcados decrece en uno.

Entonces, la inicialización es de $O(n^2)$ y hay un número lineal de iteraciones donde cada una toma una cantidad de tiempo lineal. Por lo tanto, el tiempo total del algoritmo es de $O(n^2)$.

Algoritmo 6 HAC (complete) dendrogram pruning

- 1: Asignar cada uno de los n documentos a un grupo. Estos grupos son considerados como no marcados. La matriz de semejanzas de $n \times n$ se inicializa con el valor de semejanza entre cada par de documentos. Cada valor de semejanza se redondea. Los cubos se inicializan colocando cada par de grupos (c_1, c_2) en el cubo apropiado. La matriz de apuntadores actuales se actualiza indicando en qué posición del cubo se encuentra cada uno de los grupos (c_1, c_2) .
 - 2: Del cubo no vacío con el valor más grande de θ tomar el primer par (c_1, c_2) . Considerar los siguientes tres casos:
 - Caso 1: Los valores de semejanza del par (c_1, c_2) , del grupo c_1 y del grupo c_2 son iguales. En este caso, c_1 y c_2 se unen en un grupo. El nuevo grupo no marcado c_1c_2 es creado y los grupos c_1 y c_2 son marcados y descartados. Los hijos de c_1c_2 consisten de los grupos que fueron hijos de c_1 o c_2 .
 - Caso 2: El valor de semejanza del par (c_1, c_2) es menor que el mínimo de los valores de semejanza del grupo c_1 y del grupo c_2 . En este caso, los grupos c_1 y c_2 son marcados, el grupo c_1c_2 es creado, c_1 y c_2 son asignados como los hijos del nuevo grupo.
 - Caso 3: El valor de semejanza del par (c_1, c_2) es igual al mínimo, pero menor que el máximo de los valores de semejanza del grupo c_1 y del grupo c_2 . Se asume sin pérdida de generalidad que el valor de semejanza del grupo c_1 es menor que el del grupo c_2 . En este caso, el nuevo grupo no marcado c_1c_2 es creado, los grupos c_1 y c_2 son marcados y el grupo c_1 es descartado. Los hijos del nuevo grupo son el grupo c_2 y los hijos del grupo c_1 .
 - 3: Actualizar las estructuras de datos de la siguiente manera:
 - (a) El valor de semejanza del nuevo grupo c_1c_2 con cada uno de los *clusters* c no marcados, excepto c_1c_2 , se calcula tomando el mínimo valor de los valores de semejanza de (c_1, c) , (c_2, c) y (c_1, c_2) . Entonces se añade la columna y renglón correspondiente al grupo c_1c_2 y se eliminan las columnas y renglones correspondientes a los grupos c_1 y c_2 en la matriz de semejanzas. El efecto en la matriz de semejanzas es la reducción de su dimensionalidad en uno.
 - (b) Una vez que se calcularon los valores de semejanza de (c_1c_2, c) , el par (c_1c_2, c) se agrega al cubo b que le corresponda. La matriz de apuntadores actuales se actualiza de tal manera que las entradas (c_1c_2, c) y (c, c_1c_2) apunten a la posición que el par (c_1c_2, c) ocupa en el cubo b . Esto implica añadir una columna y renglón para el grupo c_1c_2 . Remover (c_1, c) y (c_2, c) de los cubos correspondientes usando la información de la matriz de apuntadores actuales. Finalmente, las columnas y renglones de c_1 y los renglones y columnas de c_2 son removidos de la matriz de apuntadores actuales. El efecto es reducir la dimensión de la matriz de apuntadores actuales en uno.
 - 4: Terminar cuando sólo haya un grupo no marcado.
-

2.5 Métodos para etiquetar los grupos

Para dar un mayor valor agregado a la organización de documentos en grupos, es necesario que éstos tengan etiquetas que describan el tema del grupo para que los usuarios puedan determinar fácilmente si los documentos del grupo son de su interés.

Desafortunadamente los primeros algoritmos de *agrupamiento* (*HAC*, *K-Means*, *Bisecting K-Means*), a diferencia de los más recientes (*STC*, *LINGO*,¹⁵ entre otros) no crean etiquetas para los grupos, por lo cual es necesario utilizar un método para crearlas.

Los siguientes métodos¹⁶ fueron utilizados para etiquetar los grupos generados por los algoritmos *HAC*, *K-Means* y *Bisecting K-Means*:

- 1.- *Inverse Document Frequency* (Frecuencia Inversa de Documentos).
- 2.- *Frequent and Predictive Words* (Palabras Frecuentes y Predictivas).
- 3.- *Common Term in the Cluster* (Término común en el grupo).

Estos métodos utilizan los términos de los documentos de cada uno de los grupos para construir su etiqueta. Se utilizaron términos en las etiquetas para compararlas con las etiquetas que el *Suffix Tree Clustering* genera utilizando frases.

Método *Inverse Document Frequency*

Este método, propuesto por Tonella [Tonella *et al.*, 2003b], se basa en la idea de que los términos que ocurren uniformemente en los documentos de la colección no son útiles para distinguir los documentos con contenido similar de los que no están relacionados. Por ello, define *Inverse Document Frequency* como:

$$Freq_k = Cl_k \log \left(\frac{|C|}{|C_k|} \right)$$

donde Cl_k es la suma de frecuencias del término k en los documentos del grupo Cl , $|C|$ es el número de grupos obtenidos por el algoritmo de *agrupamiento* y $|C_k|$ es el número de grupos que contienen el término k .

Los términos comunes al grupo Cl tienen un valor alto en $Freq_k$; por el contrario, los que son comunes a todos los grupos tienen un valor pequeño de $Freq_k$. Por lo tanto, los términos con valor alto en $Freq_k$ son utilizados como etiquetas del grupo.

¹⁵ La descripción del algoritmo se puede encontrar en el trabajo de Wroblewski [Wroblewski, 2003].

¹⁶ De ahora en adelante se utilizará el nombre en inglés de estos métodos para hacer referencia a ellos, debido a que así es como se utilizan en la interfaz del sistema.

Método *Frequent and Predictive Words*

Este método, propuesto por Popescul y Ungar [Popescul y Ungar, 2000], selecciona como etiquetas de los grupos los términos que ocurran frecuentemente y que efectivamente los distinga de los demás.

La selección de estos términos se basa en el producto de la frecuencia local y la predictividad:

$$P(\text{término}|\text{grupo}) \times \frac{P(\text{término}|\text{grupo})}{P(\text{término})}$$

donde $P(\text{término}|\text{grupo})$ es la frecuencia del término en un grupo dado y $P(\text{término})$ es la frecuencia del término en una categoría más general o de toda la colección. En el diseño del sistema *Conquiro*, se utilizó la frecuencia de toda la colección.

Los términos que tengan una predictividad alta son buenos para distinguir un grupo de otro.

Método *Common Term in the Cluster*

El metabuscador CREDO¹⁷ etiqueta los grupos que genera con un solo término; sin embargo no se encontró información acerca de cómo lo hace, por lo cual se realizaron varias consultas y se observó que el término que usa como etiqueta es aquel que está en la mayoría de los documentos del grupo, pero también que hay mucha repetición de etiquetas. Con base en estas observaciones se desarrolló el método llamado *Common Term in the Cluster*.

El método *Common Term in the Cluster* utiliza como etiqueta del grupo el término común a la mayoría de los documentos que éste contiene. Se utiliza este término porque es el más representativo del contenido del grupo, debido a que los documentos en el grupo están relacionados entre sí y, si el término aparece en la mayoría de los documentos, es muy probable que represente su tema. Sin embargo el término común no es la mejor opción como etiqueta en los siguientes casos:

- Un grupo que tiene un documento: en este caso es recomendable usar el término que ocurra más frecuentemente, el cual será el mejor representante del contenido del grupo.
- Un grupo que contiene documentos con el mismo contenido (en los resultados del motor de búsqueda se observó que una página aparecía repetida varias veces pero con diferentes URL): en este caso el término común no es una buena opción porque todos los del grupo ocurren en todos los documentos; entonces surge la pregunta ¿cuál de todos los términos es el mejor representante del

¹⁷ <http://credo.fub.it/>

contenido del grupo?; al igual que en el caso anterior, el término que ocurra más frecuentemente en el grupo será el más representativo del contenido.

- Un grupo que tiene dos documentos: para este caso se observó que muchos de los términos ocurren en ambos documentos, por lo cual la mejor opción como etiqueta es el que ocurra más frecuentemente.

En resumen, el término que será utilizado como etiqueta del grupo es aquel que sea común a la mayoría de los documentos, o bien el que sea más frecuente en el grupo, según sea el caso.

Para disminuir la repetición de etiquetas en los grupos, el método hace lo siguiente: para el grupo que va a ser etiquetado tiene una lista de términos que pueden ser utilizados como etiquetas. El término que será utilizado como etiqueta del grupo actual será el que no haya sido utilizado como etiqueta de otros grupos. En caso de que todos los términos de la lista ya hayan sido usados, entonces se utiliza el primer término de la lista como etiqueta del grupo actual.

Para este método fue diseñado un algoritmo que se describe de manera formal en el algoritmo 7.

En el algoritmo se utiliza la estructura de datos lista, la cual va a contener la siguiente información por cada uno de los términos del grupo:¹⁸

- El número de documentos del grupo que contienen el término.
- La frecuencia del término en el grupo, es decir, la suma de las frecuencias del término en cada uno de los documentos del grupo.

Otra estructura utilizada en el algoritmo es el *hash*. Éste contiene los términos que ya han sido utilizados como etiquetas de otros grupos y además, la etiqueta que se asigne al grupo actual.

La complejidad del algoritmo es $O(n(m \log m))$, donde n es el número de grupos y m es el número de términos del grupo. El análisis que se hizo fue el siguiente:

1. Insertar en una lista m términos tiene una complejidad de $O(m)$.
2. Ordenar la lista tiene una complejidad de $O(m \log m)$ (Si se utiliza el algoritmo *Quicksort*).
3. Buscar en el *hash* tiene una complejidad de $O(m)$.
4. Insertar en el *hash* tiene una complejidad de $O(1)$.

¹⁸ Los términos del grupo se obtienen del centroide porque éste es el representante del grupo.

Algoritmo 7. Common Term in the Cluster

Input: grupo C , Hash de etiquetas H

Output: etiqueta del grupo cl

```
1:  $cl = \text{null}$ 
2: for each término  $t$  del centroide del grupo  $C$ , cuyo peso sea  $> 0$ 
3:   Calcular el número de documentos en el grupo que contienen  $t$ .
4:   Obtener la frecuencia de  $t$  en el grupo.
5:   Agregar  $t$  con esta información a la lista  $L$ .
6: end
7: if  $|C| > 2$  and los documentos del grupo no son iguales entre sí.
8:   Ordenar  $L$  por número de documentos en forma descendente.
9: else
10:  Ordenar  $L$  por frecuencia en forma descendente.
11: end
12:
13: while  $cl == \text{null}$  and  $L$  tenga términos do
14:   if  $t$  no está en  $H$  then  $cl = t$ 
15: end
16:
17: if  $cl == \text{null}$  then  $cl = \text{head}(L)$ 
18:
19: agregar  $cl$  a  $H$ 
```

Capítulo 3. El metabuscador Conquiro

Para este trabajo se desarrolló el sistema *Conquiro*, el cual es un metabuscador que aplica la técnica de *agrupamiento (clustering)* a los resultados obtenidos de un motor de búsqueda. Este capítulo explica las razones por las cuales se desarrolló, sus características y arquitectura; y, paso a paso, cómo *Conquiro* da respuesta a una consulta de usuario.

3.1 Introducción

Durante los últimos años la información disponible en la Web ha crecido tanto, que se estima que actualmente podría contener billones de páginas, de las cuales 11.5 billones han sido indexadas por los motores de búsqueda [Gulli y Signorini, 2005]. Por esta razón aproximadamente 80% de los usuarios utiliza motores de búsqueda y otras herramientas para buscar información en la Web [Jansen y Spink, 2005].

Un problema para los usuarios es la larga lista de resultados que el motor de búsqueda regresa, además de que contiene páginas de diferentes temas; por lo cual el usuario examina sólo unas cuantas páginas de esta lista para ver si encuentra la información que está buscando. En la mayoría de los casos el usuario no va a encontrar la información que necesita.

Surge entonces la necesidad de proporcionar al usuario una herramienta que le permita visualizar de manera rápida la información relevante y no relevante contenida en la lista de resultados del motor de búsqueda.

Conquiro, que en latín significa buscar, es un sistema desarrollado con el objetivo de ayudar al usuario a encontrar la información que necesita rápidamente. Para cumplir con su objetivo, *Conquiro* recupera los resultados de una consulta, los organiza en grupos utilizando la técnica de *agrupamiento (clustering)* y los muestra al usuario de manera que éste pueda explorar el contenido de cada uno de los grupos.

3.2 Características de Conquiro

En esta sección se describe las principales características de *Conquiro* y se compara con los sistemas descritos en la sección 1.4.

Las características de *Conquiro* son:

- Es un sistema que aplica la técnica de *agrupamiento* a los resultados de Google.
- Los resultados que recupera de Google están en inglés.

- Recupera hasta 200 resultados del motor de búsqueda.
- Utiliza siete algoritmos de *agrupamiento* (*K-Means*, *Bisecting K-Means*, *HAC (single)*, *HAC (complete)*, *HAC (complete) dendrogram pruning*, *HAC (UPGMA)* y *Suffix Tree Clustering*).
- Permite configurar el proceso de *agrupamiento*, asignando valores a un conjunto de parámetros. Esto permite al usuario elegir el método de asignación de pesos (*Term weighting*), la medida de distancia o semejanza, el número de grupos y el algoritmo con el que se va a hacer el agrupamiento. Además, el usuario puede elegir utilizar como documentos los sumarios que Google proporciona de las páginas o el texto completo de la página, o bien seleccionar párrafos de la página.
- Realiza *agrupamiento* jerárquico y no jerárquico.
- Utiliza cuatro métodos para asignar una descripción (etiqueta) a los grupos.

Conquiro es un sistema que tiene diferencias y semejanzas con los sistemas descritos en la sección 1.4. A continuación se expondrán cada una de ellas.

Semejanzas:

1. **Conquiro**, al igual que *Grouper*, *Carrot2* y *Vivísimo*, agrupa los resultados de un motor de búsqueda.
2. **Conquiro**, al igual que *Carrot2*, utiliza varios algoritmos para hacer el agrupamiento de documentos; y además ambos realizan *agrupamiento* jerárquico y no jerárquico.
3. **Conquiro**, al igual que *Scatter/Gather* y *Grouper*, agrupa documentos en inglés.

Diferencias:

1. **Conquiro** es el único de estos sistemas que permite configurar el proceso de *agrupamiento* de los documentos a través de una serie de parámetros.
2. **Conquiro** es el único de estos sistemas que utiliza más de un método para crear las descripciones de los grupos.
3. **Conquiro** no hace *reagrupamiento* como *Scatter/Gather*.
4. **Conquiro** no obtiene los resultados de varios motores de búsqueda, como *Grouper*, *Carrot2* y *Vivísimo*.
5. **Conquiro** no tiene la opción para indicarle que recupere menos de 200 documentos, como en el caso de *Carrot2* y *Vivísimo*.

3.3 Arquitectura de Conquiro

La arquitectura del sistema *Conquiro* (figura 3.1) está compuesta por seis módulos:

Interfaz: Es una ventana donde el usuario captura la información de su consulta; ésta puede contener una o más palabras en inglés.

Procesamiento de la consulta: Este módulo se encarga de enviar la consulta a Google y recuperar los resultados.

Procesamiento de documentos: Este módulo se encarga de limpiar los documentos, construir los vectores de éstos y de extraer la información necesaria para llevar a cabo el proceso de *agrupamiento*.

Clustering (Agrupamiento): Este módulo se encarga de realizar el proceso de *agrupamiento* de acuerdo con la configuración del usuario.

Generación de resultados: Este módulo se encarga de completar el XML¹ generado en el módulo de *Clustering (Agrupamiento)* para que el módulo de visualización de resultados los muestre al usuario.

Visualización de resultados: Este módulo se encarga de presentar los resultados de la consulta al usuario. La interfaz que usa *Conquiro* para mostrar los resultados permite al usuario explorar el contenido de cada grupo.

El sistema se desarrolló sobre la plataforma Windows; como servidor de web se utiliza Apache, como sistema manejador de base de datos MySQL y como lenguajes de programación se utilizó Perl y Java. En las siguientes secciones se explicará con detalle cada uno de estos módulos y se mostrará con un ejemplo cómo el sistema da respuesta a la consulta del usuario.

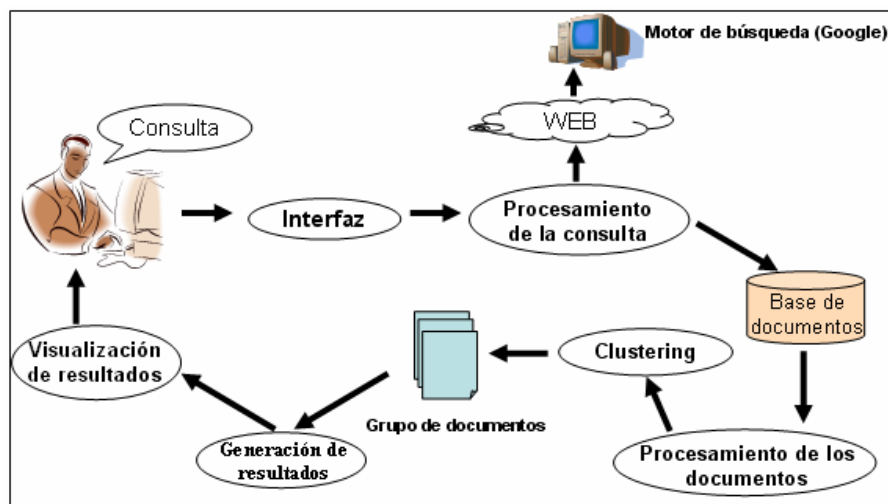


Figura 3.1. Arquitectura del sistema *Conquiro*.²

¹ XML significa *Extensible Markup Language*; este lenguaje es utilizado para estructurar y describir datos de forma que puedan ser entendidos o interpretados por diferentes aplicaciones. Para una mayor referencia acerca de XML, consúltese: <http://www.w3.org/XML/>.

3.3.1 Interfaz

La interfaz es una ventana (figura 3.2) que contiene un campo de texto donde se va a escribir la consulta y dos tipos de parámetros: los de *agrupamiento* (*Clustering Parameters*) y los parámetros para etiquetar los grupos (*Labeling Parameters*). Con estos dos tipos se configura el proceso de *agrupamiento* de los documentos.

En esta ventana el usuario escribe su consulta en inglés y asigna valores a los dos tipos de parámetros. Cuando el usuario da clic en el botón *search*, un CGI³ escrito en Perl proporciona esta información a los módulos de procesamiento de la consulta, procesamiento de los documentos y *agrupamiento*.

Los parámetros de *agrupamiento* que se manejan en la interfaz son los siguientes:

- **Algorithm:** Este parámetro indica el algoritmo que se va a utilizar para hacer el *agrupamiento* de los documentos. Se muestran siete opciones: *K-Means*, *Bisecting K-Means*, *HAC (single)*, *HAC (complete)*, *HAC (complete) dendrogram pruning*, *HAC (UPGMA)* y *Suffix Tree Clustering*.
- **Use as documents:** Este parámetro indica al sistema qué información de las páginas, obtenidas de Google, va a usar en el proceso de *agrupamiento* como documentos. Este parámetro tiene tres opciones:
 - a) *Summaries (snippets)*: Es el sumario de la página que Google proporciona.
 - b) *All text*: Es el texto completo de la página
 - c) *Selected paragraphs*: Del texto completo de la página se toman los párrafos que contengan los términos de la consulta.
- **Term weighting:** Este parámetro indica el método para asignar pesos a los términos cuando se utilice como representación de los documentos el modelo VSM. Este parámetro tiene dos opciones:
 - a) *tf*: Es el método *Term Frequency*.
 - b) *tf-idf*: Es el método *Term Frequency-Inverse Document Frequency*.
- **Term threshold:** Este parámetro indica el valor del umbral que se va a utilizar en el método descrito en la sección 2.3.4.
- **Distance/Similarity:** Este parámetro indica la medida de distancia (semejanza) que se va a utilizar en el VSM. Este parámetro tiene dos opciones:
 - a) *Cosine similarity*: Es la medida de *Semejanza de coseno*.
 - b) *Euclidean distance*: Es la medida de *Distancia euclidiana*.

² La notación usada en la figura es la siguiente: las elipses representan procesos y los cilindros representan datos.

³ CGI (Common Gateway Interface) es un estándar para la transferencia de datos entre el servidor de la Web y una aplicación externa.

- **Number of clusters o Repeat Bisecting Step:** Cuando se va a utilizar como algoritmo de *agrupamiento K-Means*, en este parámetro se especifica el número de *grupos* que se desea obtener con este algoritmo. En caso contrario si se va a utilizar *Bisecting K-Means*, entonces en este parámetro se especifica el número de veces que se va a ejecutar el paso de bisección.

Los parámetros para etiquetar los grupos que se manejan en la interfaz son los siguientes:

- **Method:** Este parámetro indica el método a utilizar para crear las etiquetas de los *grupos*. Este parámetro tiene las siguientes opciones:
 - a) *Inverse Document Frequency*
 - b) *Frequent and Predictive Words*
 - c) *Common Term in the Cluster*
 - d) *Phrases*

Los primeros tres métodos son utilizados para asignar las etiquetas de los grupos creados por los algoritmos: *K-Means*, *Bisecting K-Means*, *HAC (single)*, *HAC (complete)*, *HAC (complete) dendrogram pruning* y *HAC (UPGMA)*. El último método es utilizado por el algoritmo *Suffix Tree Clustering*.

- **Number of terms in the label:** Este parámetro indica el número de términos que la etiqueta va a contener. Este parámetro aplica para los primeros tres métodos del parámetro *Method*.

Es importante señalar que cuando el usuario elige algún algoritmo o método para etiquetar los grupos, algunos parámetros se deshabilitan ya sea porque no se utilizan o bien porque se sugiere un valor predeterminado que no debe ser modificado.

The screenshot shows the 'CONQUIRO metasearch' interface. At the top, there is a navigation bar with 'Advanced search', 'History of queries', and 'Help'. Below this is a search section with a 'Find Results' input field, a 'Search' button, and a 'Source' dropdown menu set to 'Google'. The main content area is divided into two sections: 'Clustering Parameters' and 'Labeling Parameters'. The 'Clustering Parameters' section includes: 'Algorithm' (Bisecting K-means), 'Use as documents' (Summaries (snippets)), 'Term weighting' (TF), 'Term threshold' (empty input), 'Distance / Similarity' (Cosine similarity), and 'Number of clusters' (empty input). The 'Labeling Parameters' section includes: 'Method' (Inverse Document Frequency) and 'Number of terms in label' (empty input).

Figura 3.2 Interfaz del sistema *Conquiro*.

3.3.2 Módulo de procesamiento de la consulta

Este módulo se encarga de:

- Enviar la consulta del usuario a Google [Google, 2006] para obtener los resultados de la búsqueda.
- De los resultados de la búsqueda extrae la siguiente información:
 - a) Título
 - b) URL
 - c) Contenido de la página de la Web: el contenido de la página que se va a utilizar depende del parámetro *Use as documents*.
- Con la información obtenida de los resultados de la búsqueda, se crea una base de documentos, la cual será utilizada por los módulos de Procesamiento de documentos y visualización de resultados.

Para enviar la consulta a Google y obtener los resultados de la búsqueda, se utiliza un programa en Perl que emplea el API de Google⁴ [Calishain y Dornfest, 2004].

Para almacenar el título, contenido y URL de cada una de las páginas de la Web de los resultados, se utiliza un programa en Perl, el cual antes de guardar la información en la base de documentos, asigna un identificador único a la página.

El contenido de la página web que se va a utilizar queda especificado en el parámetro *Use as documents*. Las opciones de este parámetro son: Sumarios (*snippets*), Todo el texto (*All text*) o Párrafos seleccionados (*Selected paragraphs*).

En la opción *Sumarios (snippets)* se utiliza la descripción corta de la página web que Google proporciona, como se muestra en la figura 3.3. En la opción *Todo el texto* se descarga el texto completo de la página web. En la opción *Párrafos seleccionados* se elige del texto completo los párrafos que contengan la(s) palabra(s) de la consulta. Es importante decir que **Conquiuro** tiene funcionando las tres opciones.

Zamir y Etzioni [Zamir y Etzioni, 1998] sugieren utilizar las descripciones cortas que proporciona Google en vez del texto completo, debido a que descargar el texto completo de los documentos consume tiempo y muchos usuarios no están dispuestos a esperar. Debido a esto se decidió utilizar en este trabajo la opción *Sumarios (snippets)*.

Al sumario de la página se concatena su título porque se considera que al agregar el título se crea un sumario que describe mejor el contenido de la página [Zhang y Dong, 2004].

⁴ <http://www.google.com/apis/>

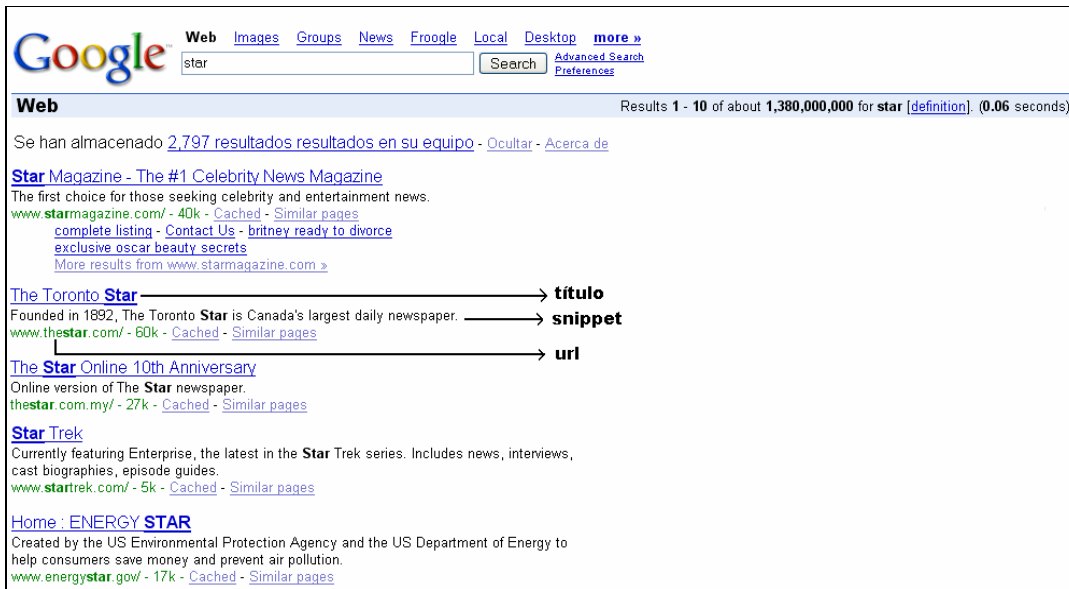


Figura 3.3. Título, *Snippet* y URL en los resultados de Google.

3.3.3 Módulo de procesamiento de documentos

El módulo de procesamiento de los documentos tiene como funciones:

- 1) Para cada uno de los documentos de la base de documentos:
 - Limpiar el documento como se describe en la sección 2.3.1.
 - Aplicar el método DF, descrito en la sección 2.3.4, para reducir el número de términos que serán utilizados en los vectores de documentos. El valor del umbral que se utilizará en el método será el que el usuario haya especificado en el parámetro *Term threshold*.
 - Crear una base de datos con los términos que quedaron después de aplicar el método DF donde se asigna a cada término un identificador único.
 - Asignar pesos a los términos de cada uno de los documentos con el método que el usuario especificó en el parámetro *Term weighting* (consultar sección 2.3.2).
- 2) Construir la matriz de términos por documentos como se describe en la sección 2.3.

Para el caso del algoritmo *Suffix Tree Clustering*, este módulo sólo limpia los documentos porque el algoritmo se encarga de procesarlos de tal manera que pueda extraer frases de éstos.

Este módulo está implementado en Perl, ya que este lenguaje tiene un manejo eficiente de cadenas, lo cual permite realizar eficientemente el procesamiento de los documentos.

3.3.4. Módulo de clustering (agrupamiento)

El módulo de *clustering* (agrupamiento) se encarga de:

- Normalizar la matriz de términos por documentos;⁵ es decir, que la norma de cada uno de los vectores de documentos (columnas de la matriz) sea 1.
- Realizar el agrupamiento con el algoritmo que el usuario especificó en el parámetro *Algorithm*, el cual agrupa los documentos y etiqueta cada uno de los grupos generados.
- Crear un archivo que contenga la siguiente información de cada grupo: número de grupo y la lista de identificadores de cada uno de los documentos que el grupo contiene.
- Crear un archivo XML que represente la estructura jerárquica de los grupos.

Los algoritmos de *K-Means*, *Bisecting K-Means*, *HAC (single)*, *HAC (complete)*, *HAC (complete) dendrogram pruning* y *HAC (UPGMA)* están implementados en Matlab.⁶ Para el caso de *K-Means* se modificó el código de Kardi Teknomo,⁷ para que los centroides se actualicen incrementalmente como lo propone Steinbach [Steinbach *et al.*, 2000] y para el caso de los algoritmos jerárquicos *HAC*, se modificó el código proporcionado en DCPR Matlab Toolbox,⁸ para que utilizaran la semejanza de coseno y para que el algoritmo genere un árbol de grupos, que va a ser utilizado para el proceso de asignar las etiquetas a los grupos y generar el XML que represente esta estructura.

El algoritmo *Suffix Tree Clustering* está basado en el código abierto⁹ de Carrot2 [Carrot2, 2006] y está implementado en Java.

Conquiro maneja tanto algoritmos jerárquicos como no jerárquicos. La salida de los algoritmos jerárquicos es un árbol de grupos; mientras que la de los algoritmos no jerárquicos es una lista de grupos, entonces por así convenir se transformó la lista en un árbol *n-ario* donde la raíz tiene como hijos todos los elementos de la lista; la figura 3.4 muestra cómo es el árbol. Esto permite utilizar el mismo código XML para mostrar los resultados de los algoritmos a los usuarios.

El XML representa la estructura jerárquica de los grupos y por cada uno se tiene un campo que corresponde con la etiqueta; ésta contiene frases, para los resultados del algoritmo *Suffix Tree Clustering* y para los demás algoritmos de *agrupamiento*, se tiene el índice o la lista de índices de los términos que forman la etiqueta del grupo. Para este

⁵ En la sección 2.3.6 se muestra un ejemplo donde la matriz de términos por documentos se normaliza.

⁶ La versión de Matlab utilizada fue la 7.0.

⁷ http://people.revoledu.com/kardi/tutorial/kMean/matlab_kMeans.htm

⁸ <http://neural.cs.nthu.edu.tw/jang/matlab/toolbox/DCPR/>

⁹ <http://www.carrot2.org/website/xml/index.xml>

último caso, los índices son reemplazados por los términos correspondientes en el módulo de generación de resultados.

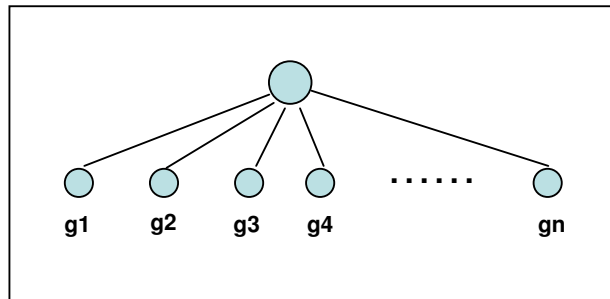


Figura 3.4 Árbol *n*-ario de documentos.

3.3.5 Módulo de generación de resultados

Este módulo se encarga de las siguientes tareas:

- Asignar un identificador numérico a la consulta actual.
- Crear una base de grupos y documentos con el archivo que se genera en el módulo de *clustering* (*agrupamiento*), el cual contiene por cada grupo la lista de los documentos que lo componen.
- Asignar los términos correspondientes a los índices que forman la etiqueta de los grupos.

Este módulo está implementado en Perl debido a que el manejo eficiente de cadenas permite hacer los cambios en el XML de manera eficiente.

3.3.6 Módulo de visualización de resultados

Este módulo se encarga de mostrar a los usuarios los resultados de la búsqueda en grupos de documentos.

La interfaz que *Conquiro* utiliza consta de dos áreas: grupos encontrados (*Clusters found*) y resultados del grupo (*Results of cluster*). En el área de grupos encontrados se muestra el árbol que contiene los grupos que el algoritmo de *agrupamiento* generó. En el área de resultados del grupo se muestra la lista de documentos que contiene el grupo.

La figura 3.5 muestra el árbol¹⁰ de grupos generado para los documentos de la consulta “apple” y la lista de documentos, donde por cada documento se muestra la siguiente información: título, resumen del documento (página) y URL.

Esta interfaz funciona de la siguiente manera: cuando se hace clic en cualquiera de los grupos, varios CGI escritos en Perl se encargan de consultar la base de documentos y grupos para obtener la lista de los identificadores de los documentos que el grupo contiene; después obtiene de la base de documentos el título, sumario y URL de estos identificadores; finalmente muestra esta lista de documentos al usuario. Para el caso de la raíz del árbol, se muestran todos los documentos que se recuperaron para la consulta.

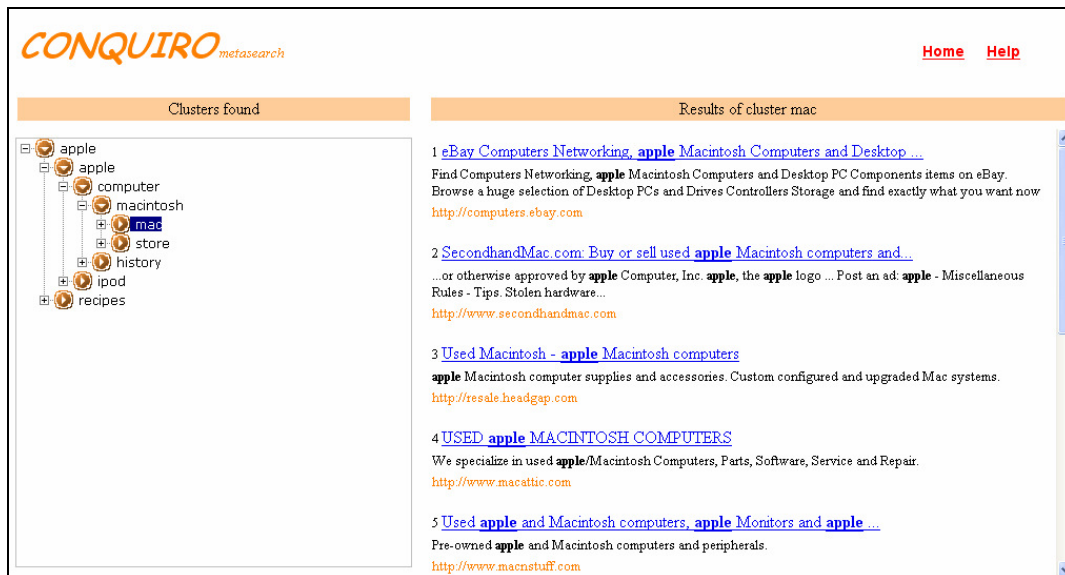


Figura 3.5 Resultado de la consulta “apple”.

Otra parte que el módulo de visualización de resultados tiene es un histórico de consultas donde se puede ver la lista de consultas al sistema, junto con sus resultados.

La figura 3.6 muestra el histórico de consultas donde se presenta la siguiente información: el número de consulta, las palabras de búsqueda, el motor de búsqueda que se utilizó (sólo se integró Google), los valores utilizados en los parámetros de agrupamiento y los valores que se utilizaron en los parámetros para etiquetar los grupos; como información extra se muestra el tiempo de ejecución del algoritmo y el número de términos utilizados para la matriz de semejanza, la fecha de la consulta y la liga donde se pueden ver los resultados. En el histórico se pueden ver las consultas (en total 103) que se hicieron para los experimentos de la secciones 4.2 y 4.3.

¹⁰ Este árbol es generado por un programa en JavaScript, el cual lee el XML correspondiente a esta consulta.

CONQUIRO query history [Home](#)

This page shows the list of some queries that was processed by conquiroy metasearch. Click the link "view" of some query to see its page of results. If you want to eliminate some query form the history then click the link "delete".

Short version Detailed version

| Query number | search words | clustering parameters | Cluster labeling parameters | Link |
|--------------|--------------|--|--|--|
| 1 | star | Algorithm : K-Means Use as documents : summaries Term Weighting : TF Term threshold : 2 Distance/Similarity : euclidean dist. Number of clusters : 63 | method : Inverse Document Frequency #terms in label : 3 | view delete |
| 2 | star | Algorithm : Bisecting K-means Use as documents : summaries Term Weighting : TF Term threshold : 2 Distance/Similarity : euclidean dist. Repeat Bisecting step : 5 | method : Inverse Document Frequency #terms in label : 3 | view delete |
| 3 | star | Algorithm : HAC(SINGLE) Use as documents : summaries Term Weighting : TF Term threshold : 2 Distance/Similarity : euclidean dist. Number of clusters : 0 | method : Inverse Document Frequency #terms in label : 3 | view delete |

Figura 3.6. Histórico de consultas.

3.3.7. La respuesta de Conquiroy a una consulta de usuario, paso a paso

En el siguiente ejemplo se mostrará paso a paso cómo **Conquiroy** da respuesta a la consulta de usuario “jaguar”.

1. El usuario escribe “jaguar” en el campo *Find Results* y asigna los siguientes valores a los parámetros de *agrupamiento* y a los parámetros para etiquetar los grupos:

Parámetros de agrupamiento

Algorithm : HAC (complete) dendrogram pruning

Use as documents: Summaries (snippets)

Term weighting: tf-idf

Term threshold: 2

Distance/Similarity: Cosine similarity

Parámetros para etiquetar los grupos

Method: Common Term in the Cluster

La figura 3.7 muestra la pantalla que contiene esta información.

The screenshot shows the CONQUIRO metasearch interface. At the top, there is a logo for CONQUIRO metasearch and a navigation bar with 'Advanced search', 'History of queries', and 'Help'. Below this, there is a search input field containing 'jaguar' and a 'Search' button. A 'Source' dropdown menu is set to 'Google'. Two main sections are visible: 'Clustering Parameters' and 'Labeling Parameters'. The 'Clustering Parameters' section includes: Algorithm (HAC (COMPLETE) dendrogram pruning), Use as documents (Summaries (snippets)), Term weighting (TF-IDF), Term threshold (2), Distance / Similarity (Cosine similarity), and Number of clusters (0). The 'Labeling Parameters' section includes: Method (Common Term in the Cluster) and Number of terms in label (1).

Figura 3.7. Consulta del usuario.

Una vez que el usuario completa su consulta y da clic en el botón *search*, el CGI de la interfaz proporciona la siguiente información al módulo de procesamiento de la consulta: consulta y el valor del campo *Use as documents* (en este caso es *Summaries*).

2. El módulo de procesamiento de la consulta realiza las siguientes tareas:

a) Enviar la consulta a Google utilizando la siguiente instrucción:¹¹

```
my $results = $google_search ->
    doGoogleSearch( key, query, start, maxResults,
                   filter, restrict, safeSearch,
                   lr, ie, oe);
```

En el parámetro *query* se pone la consulta del usuario; en el parámetro *start* se le asigna el número de resultado *a partir* del cual hay que empezar a contar los *maxResults* que se van a devolver; en el parámetro *maxResults* se le asigna el valor de 10 para indicar al API que devuelva 10 resultados y en el parámetro *lr* se utiliza el valor “lang_en” para indicar al API que los resultados deben estar en idioma inglés.

Debido a que el API de Google sólo regresa 10 resultados por búsqueda, se necesita ejecutar 20 veces la instrucción anterior para obtener los 200 documentos.

b) Los resultados de la consulta se recuperan de la siguiente manera:

¹¹ Para mayor información acerca de los parámetros, consúltese el libro Calishain y Dornfest [Calishain y Dornfest, 2004] o bien <http://www.google.com/apis/>

```

foreach my $result ( @{$results->{resultElements}} )
{
    my $desc = $result->{snippet};

    ProcesarResult ($clus_opt, \ $num_url, $result->{URL}, $desc,
                  $result->{title}, $db);
}

```

Para cada uno de los resultados se obtiene el *snippet* (porque así lo especificó el usuario), el URL y el título, para que la función *ProcesarResult* asigne un identificador único al documento, concatene el título al *snippet* e inserte esta información en la base de documentos.

3. Una vez almacenados los resultados de la búsqueda en la base de documentos, se le proporciona al módulo de Procesamiento de documentos los valores de los parámetros *Term weighting*, *Term threshold* y *Distance/Similarity*.
4. El módulo de procesamiento de documentos realiza las siguientes tareas:
 - a) Eliminar del contenido de la página el código HTML, URL, direcciones de correo, caracteres que no son letras y además las mayúsculas se convierten a minúsculas.

La siguiente figura muestra el documento antes de que se realice la eliminación de términos.

Jaguar -- Kids' Planet -- Defenders of Wildlife.
Images of Jaguars: Jaguar in Water [72k jpg]; Jaguar [62k gif]. ... Endangered DESCRIPTION:
The jaguar is one of the most majestic and mysterious animals in nature. ...

Figura 3.8. Documento antes de la eliminación.

La siguiente figura muestra el documento cuando se elimina el código HTML.

Jaguar -- Kids' Planet -- Defenders of Wildlife.Images of Jaguars: Jaguar in Water [72k jpg]; Jaguar [62k gif]. Endangered DESCRIPTION: The jaguar is one of the most majestic and mysterious animals in nature.

Figura 3.9. Documento sin código HTML.

Finalmente en la siguiente figura se muestra el documento cuando se eliminan los caracteres que no son letras y cuando las letras mayúsculas se convierten a minúsculas.

jaguar kids planet defenders of wildlife images of jaguars jaguar in water 72k jpg jaguar 62k gif endangered description the jaguar is one of the most majestic and mysterious animals in nature

Figura 3.10. Documento después de la eliminación.

- b) Eliminar los términos que aparezcan en la lista de palabras comunes.

Al eliminar las palabras comunes del documento de la figura 3.10 tenemos lo siguiente:

```
jaguar kids planet defenders wildlife images jaguars jaguar water 72k jaguar
62k endangered description jaguar majestic mysterious animals nature
```

Figura 3.11. Documento sin palabras comunes.

- c) Aplicar la técnica de reducción de términos a la raíz (*stemming*).

Al aplicar esta técnica al documento de la figura 3.11 se obtiene lo siguiente:

```
jaguar kid planet defend wildlif imag jaguar jaguar water k jaguar k endang
descript jaguar majest mysteri anim natur
```

Figura 3.12 Documento después de aplicar la reducción de términos a la raíz.

Es importante señalar que para este punto se utilizó un módulo de Perl *Lingua::Stem::En* que tiene implementado el algoritmo de Porter [Porter, 1980].

- d) Aplicar el método de reducción de dimensionalidad¹² (utilizando el método DF, descrito en la sección 2.3.4) eliminando los términos que aparezcan en menos de X documentos, donde X es el valor especificado en el parámetro *Term threshold*.

Una vez limpiados los 200 documentos, se obtuvieron 530 términos que aparecieron en más documentos de los especificados en el parámetro *Term threshold*. Si no se hubiera hecho esta reducción de la dimensionalidad, se hubiera obtenido en total 1379 términos.

- e) Crear una base de datos que contenga los 530 términos a los cuales se asigna un identificador único.
- f) Generar los vectores de características de cada uno de los documentos¹³.
- Calcular el peso de cada uno de los 530 términos (de acuerdo con el método especificado en el parámetro *Term weighting*) en el documento.
 - Construir el vector de características del documento. La figura 3.13 muestra el vector de características del documento de la figura 3.12. La dimensión de este vector es de 530.

¹² La reducción de la dimensionalidad tiene el fin de eliminar los términos que no aporten información para describir el contenido de los documentos.

¹³ En la sección 2.3.6 se da una explicación más detallada de cómo se construyen los vectores de los documentos.

$v=(1.7202,1.6232,1.8451, 1.8451,2.0212,1.8451,2.0212,1.4771,1.7202,1.8451,0.1060,0,\dots,0)$

Figura 3.13. Vector de características.

g) Crear la matriz de términos por documentos.

$$M_{530 \times 200} = \begin{pmatrix} 1.7202 & \dots & 0 \\ 1.6232 & & 0 \\ 1.8451 & & 0.0636 \\ 1.8451 & \dots & 0 \\ 0 & & 1.2808 \\ \vdots & & \vdots \\ 0 & \dots & 0 \end{pmatrix}$$

Figura 3.14. Matriz de términos por documentos.

5. Se ejecuta el algoritmo *HAC (complete) dendrogram pruning*, el cual fue elegido por el usuario, utilizando las instrucciones en Perl de la figura 3.15.

En el bloque SWITCH_FUN se indica para cada algoritmo (*K-Means*, *HAC (single)*, *HAC (complete)*, *HAC (dendrogram) pruning*, *HAC (UPGMA)*) la función que se va a ejecutar en Matlab.

La función especificada se ejecuta a través del comando *System*.

```

SWITCH_FUN:
{
  #K-MEANS
  if( $metodo == 1 )
  {
    #nombre de la funcion mas sus parametros
    $exec_algo = "kmeans_system('$pkmeans','$par_dist','$metclab','$ntermlab')";
    last SWITCH_FUN;
  }
  .
  .
  .

  #HAC (COMPLETE) dendrogram pruning
  if( $metodo == 4 )
  {
    #nombre de la funcion mas sus parametros
    $exec_algo = "agglo_system('4','$par_dist','$conreb','$metclab','$ntermlab')";
    last SWITCH_FUN;
  }
}

#indica a Matlab que se ejecute en modo terminal
my $opciones = "-nosplash -nodesktop -r";

#al terminar Matlab esta instruccion cierra la terminal que se abrio
my $fin = ";exit";

#ejecucion de los algoritmos
my @args = ();
if( $metodo != 8){ @args = ("matlab ".$opciones ".$exec_algo".$fin); }
else{ @args = ("java ".$STclustering "); }
system(@args);

```

Figura 3.15 Código que ejecuta los algoritmos de *agrupamiento*.

6. La función *agglo_system* realiza las siguientes tareas:

- a) Lee el archivo de la matriz de términos por documentos, utilizando la siguiente instrucción:

```
mdat = load('mdat.dat');
```

- b) La matriz de términos por documentos se normaliza

$$M_{530 \times 200} = \begin{pmatrix} 0.3016 & \dots & 0 \\ 0.2846 & & 0 \\ 0.3235 & & 0.0118 \\ 0.3235 & \dots & 0 \\ 0 & & 0.2376 \\ \vdots & & \vdots \\ 0 & \dots & 0 \end{pmatrix}$$

Figura 3.16 Matriz de términos por documentos normalizada.

- c) A partir de la matriz de términos por documentos se crea la de semejanzas (la semejanza entre los documentos fue calculada con el método especificado en el parámetro *Distance/Similarity*)

$$S_{200 \times 200} = \begin{pmatrix} Nan & 0.0001 & 0.0001 & \dots & 0.0002 \\ 0.0001 & Nan & 0.0000 & \dots & 0.0001 \\ 0.0001 & 0.0000 & Nan & \dots & 0.0425 \\ \vdots & \vdots & \ddots & \ddots & \\ 0.0001 & 0.0002 & \dots & 0.0000 & Nan \end{pmatrix}$$

Figura 3.17 Matriz de semejanzas.

En la matriz de la figura 3.17, la diagonal tiene *Nan* para que, en las operaciones de las matrices, no se tome en cuenta la diagonal debido a que se está considerando la semejanza entre dos documentos distintos.

- d) Para el caso del algoritmo *HAC (complete) dendrogram pruning* se hace el redondeo de las semejanzas entre los documentos, entonces la matriz de semejanzas queda de la siguiente manera:

$$S_{200 \times 200} = \begin{pmatrix} Nan & 0 & 0 & \dots & 0 \\ 0 & Nan & 0 & \dots & 0 \\ 0 & 0 & Nan & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \\ 0 & 0 & \dots & 0 & Nan \end{pmatrix}$$

Figura 3.18 Matriz de semejanzas de la figura anterior cuando se le aplica el redondeo.

Aunque la matriz de semejanzas mostrada en la figura 3.18 tenga entradas en cero, es importante aclarar que no son todas (hay entradas con valores de 0.4, 0.1, ..., 0.9), debido al tamaño no se muestran las demás.

- e) Asigna cada uno de los vectores de características de los documentos a un grupo.
- f) Para llenar los cubos (*buckets*) se utilizará la matriz de semejanzas para saber la semejanza que hay entre cada par de grupos.

Los cubos se llenan de la siguiente manera: los pares cuya semejanza sea 0 se ponen en el cubo 0; los pares cuya semejanza sea 0.1, en el cubo 0.1, y así sucesivamente se llenan los cubos restantes hasta llegar al cubo que contenga pares cuya semejanza sea 0.9. La figura 3.19 muestra un ejemplo del contenido de los cubos para la consulta "jaguar".

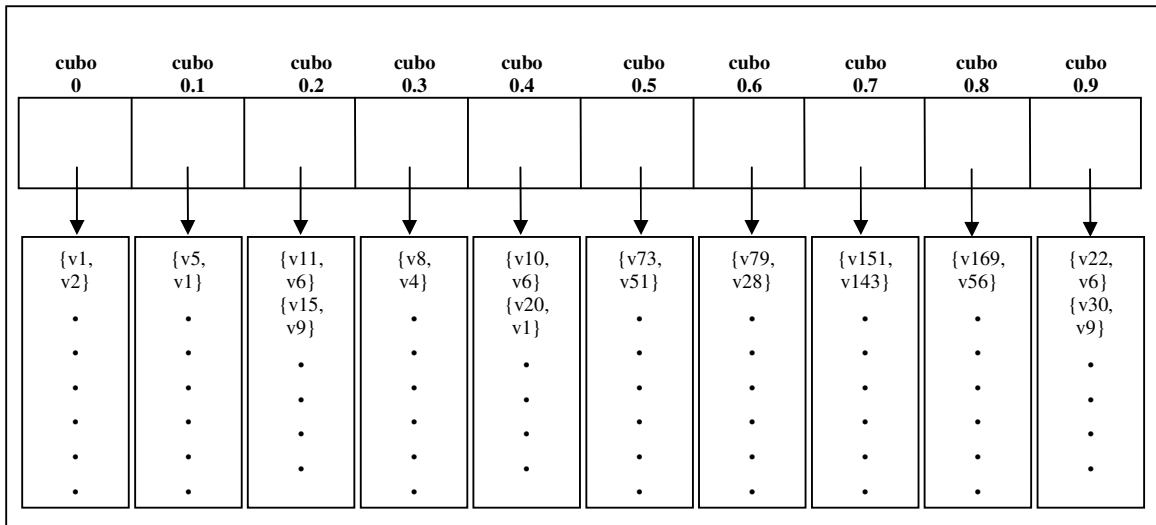


Figura 3.19 Los cubos para la consulta “jaguar”.

- g) Una vez llenos, se crea una matriz que contiene el número de *bucket* al que cada par de grupos *i, j* pertenece.

$$\text{buckposMat}_{200 \times 200} = \begin{pmatrix} \text{Nan} & 1 & 1 & \dots & 1 \\ 1 & \text{Nan} & 1 & \dots & 1 \\ 1 & 1 & \text{Nan} & \dots & 1 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 1 & 1 & \dots & 1 & \text{Nan} \end{pmatrix}$$

Figura 3.20 Matriz que contiene las posiciones de los grupos.

- h) Del cubo 0.9 se toma la primera pareja de grupos: {v22, v6}.
- i) Se calcula la semejanza de coseno de los siguientes grupos:
- a) Grupo que contiene el vector de características del documento 22, representador por v22.
 - b) Grupo que contiene el vector de características del documento 6, representado por v6.
 - c) Grupo que contiene los vectores de características del documento 22 y del documento 6

La semejanza de coseno del caso (a) y del caso (b) es 1 porque es la semejanza del vector consigo mismo.

La semejanza de coseno del caso (c) es 0.9.

- j) Una vez obtenidos estos valores se verifica que se cumpla alguno de los tres casos del algoritmo *HAC (complete) dendrogram pruning*.

- k) Para estos valores se cumple el caso 2 que dice: “El valor de semejanza del par de grupos (c_1, c_2) es menor que el mínimo de los valores de semejanza del grupo c_1 y del grupo c_2 . En este caso, los grupos c_1 y c_2 son asignados como hijos del nuevo grupo.”
- l) Se crea el nuevo grupo que tiene como hijos a los que contienen los vectores de características del documento 22 y del documento 6, como se muestra en la figura 3.21.

La información que tiene el nuevo grupo es la siguiente:

- Una lista de documentos : { 22 , 6 }
- Número total de documentos : 2
- Vector de suma de frecuencias¹⁴ : $v_{22} + v_6$
- Centroide = $\frac{\text{vector de suma de frecuencias}}{2}$
- Label : “ ” (cadena vacía)
- Hijos : { 22 , 6 }

Los campos vector de suma de frecuencias, centroide y label son utilizados cuando se crea la etiqueta del grupo.

El campo lista de documentos es utilizado para indicar que documentos están en el grupo.

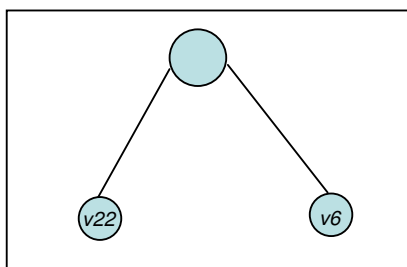


Figura 3.21 Nuevo grupo creado.

- m) Se elimina del cubo 0.9 el grupo {v22, v6} y los grupos que contengan a v22 y a v6. La figura 3.22 muestra que del cubo 0.2 se quita la pareja {v11,v6} y del cubo 0.4, la {v10,v6}.

¹⁴ El vector de suma de frecuencias del grupo es la suma de los vectores de características de los documentos que pertenecen al grupo.

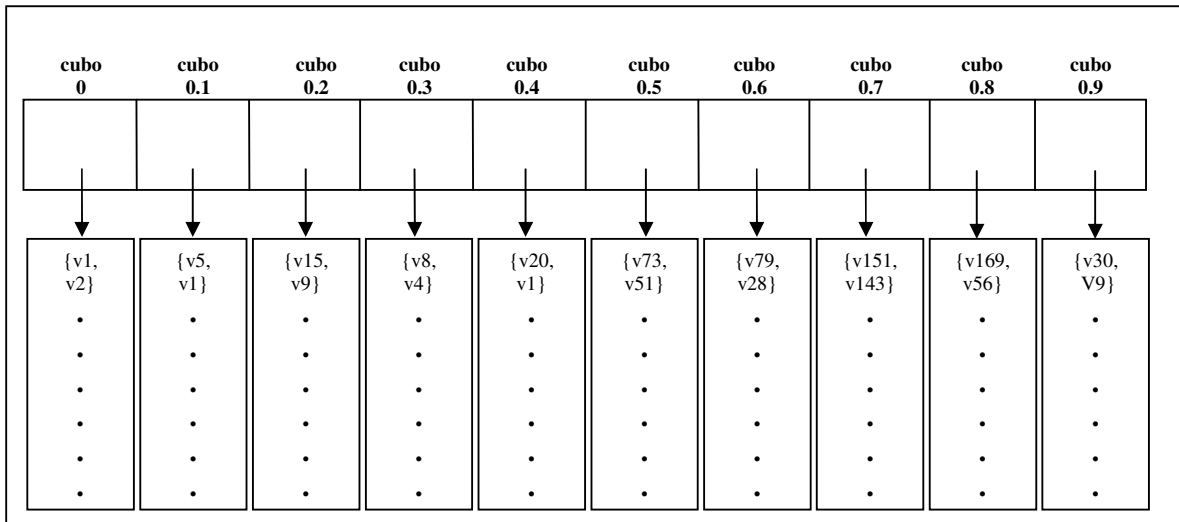


Figura 3.22 Eliminación de parejas de grupos.

- n) Se actualiza la matriz de semejanzas de la siguiente manera:
- Se remueven las columnas y renglones correspondientes al documento 22 y al documento 6.
 - Se agrega una nueva columna y renglón correspondiente al nuevo grupo.
 - Se calcula la semejanza de coseno del nuevo grupo con los grupos restantes y se pone el valor en las entradas correspondientes de la matriz.
 - La matriz de semejanzas obtenida reduce su dimensión en 1.
- o) Agregar los pares de grupos del nuevo grupo con el resto a los cubos correspondientes según el valor de la semejanza.
- p) Se actualiza la matriz de la figura 3.20 de la siguiente manera:
- Se remueven las columnas y renglones correspondientes al documento 22 y al documento 6.
 - Se agrega una nueva columna y renglón correspondiente al nuevo grupo.
 - Para cada par del nuevo grupo con el resto de los grupos se pone el número de cubo al que pertenecen.
- q) Se repiten los pasos (h) hasta el (p) hasta que se obtenga un solo grupo.
- r) Al finalizar el algoritmo, se tiene una estructura jerárquica de grupos como la siguiente:

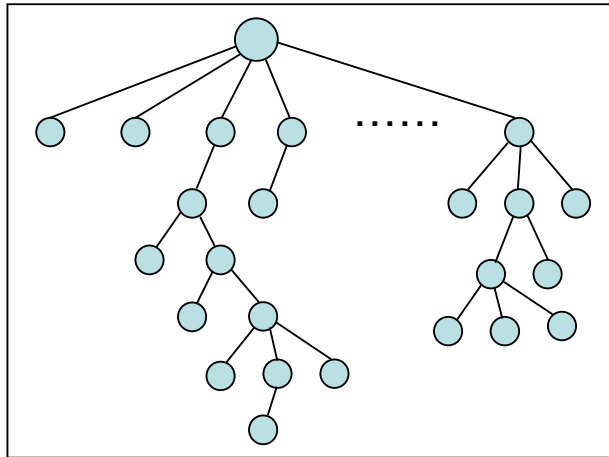


Figura 3.23 Estructura de grupos generada por el algoritmo *HAC (complete) dendrogram pruning*.

- s) Una vez obtenida la estructura de los grupos, para cada uno se crea una etiqueta que describe el contenido de los documentos del grupo. En este caso el usuario eligió construir estas etiquetas con el método *Common Term in the Cluster*, explicado en la sección 2.5. Para cada grupo este método elige al término que describa mejor el contenido de los documentos del grupo.
- t) Una vez que todos los grupos tienen su etiqueta, se genera un archivo con el XML de la estructura del árbol.
- u) Una vez creado el XML de la estructura, se crea un archivo que tiene por cada nodo de la estructura la lista de documentos que cada uno contiene. La figura 3.24 muestra una porción del archivo generado para esta consulta. En la parte izquierda está el identificador del nodo y en la derecha está la lista de documentos que el nodo contiene. Es importante recordar que el nodo con identificador 1 (la raíz de la jerarquía) contiene en su lista todos los documentos de la colección.

| Número de grupo | Lista de documentos |
|-----------------|---------------------|
| 2 | d164, d116 |
| 3 | d189, d109 |
| 4 | d197, d95 |
| 5 | d208, d94 |
| 6 | d144, d129 |
| 7 | d142, d92 |
| 8 | d150, d148 |
| 9 | d145, d19 |
| 10 | d100, d84 |
| 11 | d111, d61 |
| 12 | d70, d59 |
| 13 | d78, d45 |
| 14 | d138, d112 |

Figura 3.24 Lista de los nodos con la lista de de los documentos que contienen.

7. Cuando el proceso de *agrupamiento* termina, el módulo de generación de resultados hace las siguientes tareas:

- Verifica si la consulta existe para asignar el número de consulta correspondiente. En caso de que la consulta no se haya hecho, se le asigna el consecutivo de los números de consulta registrados en el histórico de consultas; en caso contrario, se le asigna el número que se le había asignado en el histórico.

Es importante señalar que el sistema tiene un histórico de consultas para ver las consultas al sistema y los resultados de cada una. Por lo cual es importante llevar un control.

- Si la consulta ya se había realizado, entonces se limpian las tablas de la base de datos que contienen los resultados de la consulta anterior y se actualiza la fecha que indica cuándo se realizó la consulta; esto tiene el fin de sustituir los resultados anteriores con los nuevos e indicar la fecha de la nueva consulta. En caso de que la consulta no se hubiera realizado, entonces se crean las tablas que van a contener los resultados y se agrega la siguiente información al histórico de las consultas:
 - a) Número de consulta.
 - b) Consulta del usuario.
 - c) Fecha en la que se hizo la consulta.
 - d) Motor de búsqueda utilizado.
 - e) Los valores utilizados en los parámetros de *agrupamiento*.
 - f) Los valores utilizados en los parámetros para etiquetar los grupos.
 - g) Tiempo de ejecución del algoritmo de *agrupamiento* y la dimensión de los vectores de características de los documentos.

8. El módulo de visualización de resultados muestra los resultados de la consulta al usuario. En la figura 3.25 aparece la ventana que se presenta al usuario con los resultados de la consulta.

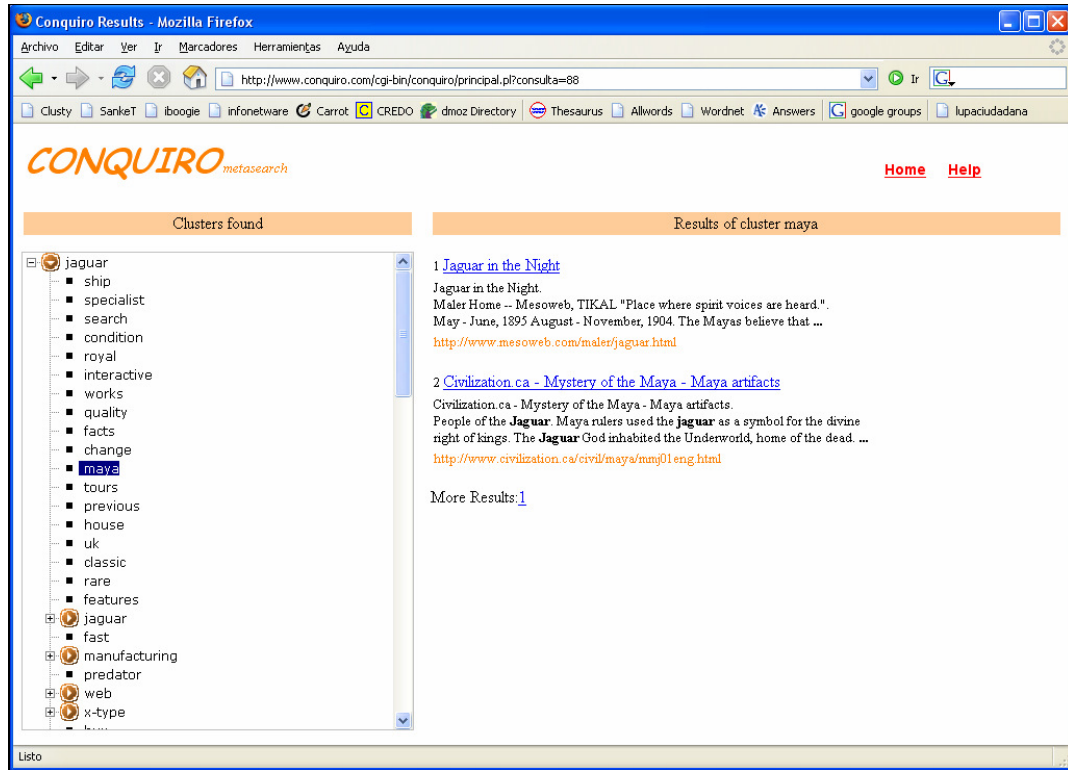


Figura 3.25 Resultados de la consulta "jaguar".

Capítulo 4. Evaluación y experimentos

En este capítulo son descritos los experimentos que se realizaron para el caso de agrupar documentos de la Web y para el caso de agrupar mensajes de un *newsgroup*, así como los resultados obtenidos y los métodos de evaluación utilizados para calificar los resultados.

4.1 Métodos de Evaluación

Un sistema de recuperación de información (SRI) generalmente produce un conjunto de documentos en respuesta a una consulta de usuario. Estos documentos necesitan ser examinados y evaluados.

Para entender el proceso de evaluación de un SRI, es necesario dar respuesta a tres preguntas: ¿cuál es el objetivo de la evaluación?, ¿qué se va a evaluar? y ¿cómo se va a evaluar?

El objetivo de evaluar los sistemas de recuperación de información es medir el beneficio de utilizar un sistema, qué tan bien realiza las tareas para las cuales fue diseñado y compararlo con otros de su tipo.

Rijsbergen [Rijsbergen, 1979] contesta la segunda pregunta citando a Cleverdon, quien en 1966 listó seis medidas: la cobertura de la colección, es decir, el grado en el cual el sistema incluye el material relevante; el tiempo de respuesta del sistema a una petición de búsqueda; la forma de presentación de los resultados; el esfuerzo realizado por el usuario para obtener las respuestas a su petición de búsqueda; el *recall* del sistema y el *precision* del sistema. Este autor opina que las cuatro primeras medidas son fácilmente calculables y las dos últimas miden la **efectividad del sistema**, es decir, su capacidad para recuperar los documentos relevantes a la consulta del usuario.

La respuesta a la tercera pregunta está basada en el concepto de relevancia; esto se debe a que evaluar la efectividad del sistema de recuperación no es una tarea fácil, ya que es difícil establecer criterios para determinar cuándo un documento es relevante o no, porque la relevancia del documento está en función de las necesidades de información de la persona o de su grado de conocimiento de la materia. A causa de estas dificultades han sido creados diversos métodos de evaluación, en los cuales los autores dan su definición subjetiva de relevancia.

Para evaluar la efectividad de *Conquiro* fueron realizadas dos tipos de evaluaciones: cuantitativa¹ y cualitativa.² Para la evaluación cuantitativa se utilizó el método de *Gold Standard* (descrito a detalle en la sección 4.1.2), que usa las medidas de *Precision* y *Recall* (descritas en la sección 4.1.1). Para la evaluación cuantitativa se

¹ La evaluación cuantitativa es aquella donde es calificada la efectividad del sistema.

² La evaluación cualitativa es aquella donde se juzga o valora la efectividad del sistema.

utilizó el método de *Evaluación de usuario* (descrito en 4.1.3), el cual se basa en cuestionarios.

Para una mayor referencia acerca de la evaluación en recuperación de información (IR por sus siglas en inglés), se recomienda consultar a Rijsbergen [Rijsbergen, 1979], Wu [Wu y Sonnenwald, 1999] y Mandl [Mandl, 2005].

4.1.1 *Precision y recall*

En IR, *precision* y *recall* son dos medidas básicas usadas en la evaluación de sistemas de recuperación de información. Éstas están basadas en dos conjuntos: los documentos recuperados por el sistema dada una consulta y los documentos relevantes a la consulta; la figura 4.1 muestra gráficamente estos conjuntos.

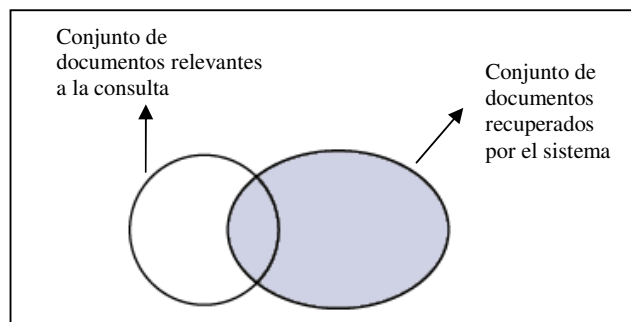


Figura 4.1 Documentos relevantes y documentos recuperados [Jizba, 2000].

En la figura 4.2 se puede ver que en los dos conjuntos anteriores hay tres tipos de documentos: los relevantes no recuperados, los relevantes recuperados y los documentos irrelevantes recuperados.

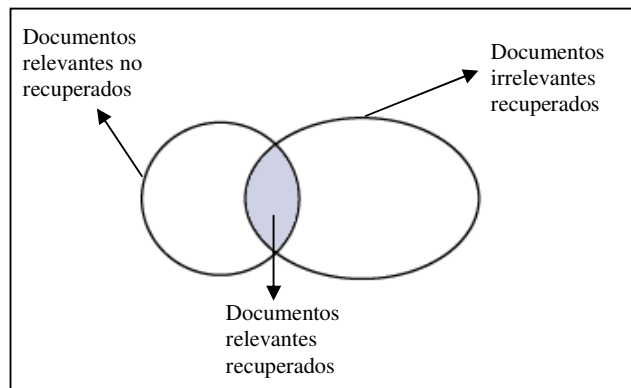


Figura 4.2 Documentos relevantes e irrelevantes [Jizba, 2000].

Con base en lo anterior se define *precision* y *recall* de la siguiente manera:

Precision indica que proporción de los documentos recuperados son relevantes. Este valor usualmente es expresado como porcentaje. La figura 4.3 muestra su cálculo.

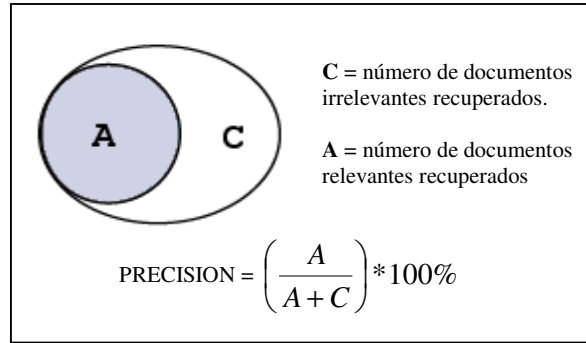


Figura 4.3 *Precision* [Jizba, 2000].

De lo anterior se puede concluir que *precision* es un indicador de la cantidad de “ruido” que tiene la información recuperada por el sistema.

Recall indica la proporción de los documentos relevantes que fueron recuperados. Este valor usualmente es expresado como porcentaje. La figura 4.4 muestra una descripción gráfica del concepto.

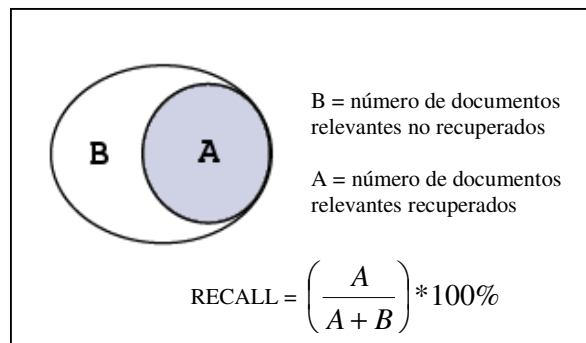


Figura 4.4 *Recall* [Jizba, 2000].

Precision y *recall* son inversamente proporcionales, es decir, si se recuperaron n documentos relevantes, donde n es menor que el total de documentos relevantes, entonces el *precision* será 100% y *recall* tendrá un valor bajo. Si por el contrario se recuperan todos los documentos relevantes, pero también documentos irrelevantes, entonces *recall* será 100% y *precision* tendrá un valor bajo. La figura 4.5 muestra gráficamente esta relación.³

La relación inversamente proporcional entre ambas medidas es inevitable [Buckland y Gey, 1994].

³ En la figura se puede ver que los valores de *precision* y *recall* están en el rango de 0.0 - 1.0 (0% - 100%).

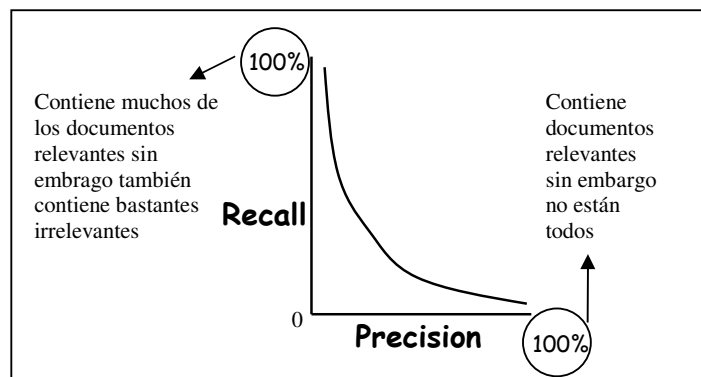


Figura 4.5 Relación entre precisión y *recall*.

Es importante señalar que un *recall* alto no siempre es necesario, debido a que las personas comúnmente no quieren todos los documentos relevantes; a menudo prefieren sólo uno o algunos de ellos. Sin embargo, obtener un *recall* alto siempre es deseable [Buckland y Gey, 1994].

Calcular estas dos medidas no es fácil, ya que determinar qué documentos son relevantes es cuestión de criterio, es decir, lo relevante para una persona podría no serlo para otra.

4.1.2 Gold Standard

El *Gold Standard* es un método general de evaluación, en el cual una solución “ideal” o de referencia al problema es construida manualmente por uno o más expertos [Tonella *et al.*, 2003a]. Para el caso de *agrupamiento (clustering)* de documentos, esto significa crear una agrupación ideal para un conjunto de documentos. Con base en la solución ideal el sistema se evalúa con las medidas de *precision* y *recall*.

Este método tiene varias desventajas, la primera es que no funciona para grandes cantidades de datos debido a que hacer la construcción manual del *Gold Standard* requeriría de mucho tiempo y esfuerzo. La segunda es que no es fácil definir una “buena” clasificación, incluso desde el punto de vista humano, ya que puede haber varias posibles agrupaciones [Anton y Croft, 1996].

El método de evaluación basado en *Gold Standard* utilizado en esta tesis fue el que Crabtree [Crabtree, 2004] presenta en su trabajo. En este método, Crabtree sugiere una modificación a las fórmulas de *precision* y *recall*. El método consiste en utilizar un *Gold Standard* para asignar los documentos de la colección a categorías, las cuales son llamadas **categorías asignadas por usuarios**. Éstas deben ser generales, por ejemplo, si hay documentos relacionados con ‘clubes de carros’, ‘partes de carros’ y ‘vendedores de carros’, entonces los documentos deberían ser asignados a la categoría carros, que es la más general. Cuando en algunos casos no se identifique la categoría a la que pertenecen los documentos entonces éstos serán asignados a la categoría “otros (*others*)”. Es importante señalar que los documentos sólo pueden ser asignados a una sola categoría.

Antes de calcular *precision* y *recall*, es necesario identificar qué categoría representa cada uno de los grupos generados por el sistema, los cuales son llamados

grupos resultado. La categoría que el grupo resultado representa es a la que pertenecen la mayoría de los documentos que éste contiene.

La figura 4.6 muestra un ejemplo, los grupos A y B son las categorías asignadas por los usuarios, la letra F representa la categoría “otros” y los C, D y E son los grupos resultado. Los números de cada grupo representan la cantidad de documentos.

La categoría del grupo resultado C y D es A porque en ambos grupos la mayoría de sus documentos (35 en C y 25 en D) pertenecen a la categoría A. El grupo resultado E pertenece a la categoría B porque todos sus documentos (25) son de esta categoría.

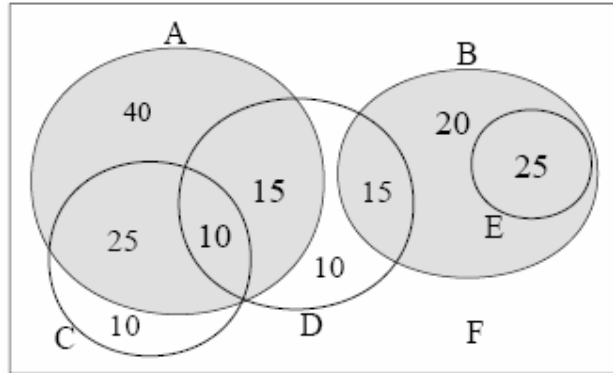


Figura 4.6 Ejemplo de evaluación [Crabtree, 2004].

Precision se define como el número total de documentos que representan la categoría de cada grupo resultado, dividido entre el número total de documentos en los grupos resultado. Como muestra se va a calcular el valor de *precision* del ejemplo de la figura 4.6.

$$precision = \frac{(35 + 25 + 25)}{(45 + 50 + 25)} = \frac{85}{120} = 71\%$$

En el numerador está el número total de documentos que representan la categoría de cada uno de los grupos, donde 35 documentos representan la categoría del grupo C; 25, la del grupo D; y 25 documentos representan la categoría del grupo E. En el denominador está el total de documentos de los grupos C, D y E.

Recall se define como el número total de documentos distintos que representan la categoría de cada grupo resultado, dividido entre el total de documentos que contienen las categorías asignadas por los usuarios. Como muestra calculará el valor de *recall* del ejemplo de la figura 4.6.

$$recall = \frac{(35 + 15 + 25)}{(90 + 60)} = \frac{75}{150} = 50\%$$

En el numerador está el número total de documentos distintos que representan la categoría de cada uno de los grupos, donde 35 documentos son del grupo C; 15 son del

grupo D, ya que los otros 10 documentos fueron utilizados en el grupo C y 25 del grupo E. En el denominador está el total de documentos de las categorías A y B.

En este método los documentos de la categoría “Otros” no son utilizados en el cálculo de *precision* y *recall*.

4.1.3 Evaluación de usuarios

Las evaluaciones de usuario usualmente se basan en algún cuestionario, un conjunto de preguntas para obtener información de su opinión. Esta técnica podría ser la mejor forma de evaluar el desempeño de un sistema cuando éste es usado y apreciado por los usuarios por los resultados que éste genera [Weiss, 2001]. A pesar de esto, la técnica tiene varias desventajas. Primeramente, los usuarios deben involucrarse en el proceso de evaluación y no simplemente contestar por contestar, en la evaluación realizada para el caso de estudio 2 (presentado en la sección 4.3), los usuarios se mostraban inquietos después de estar 35 minutos contestando el cuestionario y además tuvieron dificultades con el idioma, ya que *Conquiro* trabaja con documentos en inglés; en segundo lugar, para este tipo de evaluaciones el número de participantes debería ser estadísticamente significativo, es decir, usualmente se necesita aplicar más de 12 o 13 cuestionarios para derivar conclusiones del análisis estadístico [Weiss, 2001]; sin embargo por falta de tiempo en este trabajo no fue posible realizar este número de cuestionarios.

4.2 Caso de estudio 1: Documentos de la Web

La organización de la información en la Web ha sido tratada en diversos trabajos [Cutting *et al.*, 1992; Ferragina y Gulli, 2005; Zamir y Etzioni, 1999; Zeng *et al.*, 2004 y Zhang y Dong, 2004]. Estos trabajos proponen diversos algoritmos para agrupar los documentos; sin embargo este trabajo propone comparar la agrupación de los algoritmos más conocidos en *agrupamiento (clustering)*: *K-Means*, *Bisecting K-Means*, *HAC (single)*, *HAC (complete)*, *HAC (complete) dendrogram pruning*, *HAC (UPGMA)* y *Suffix Tree Clustering*, que es un algoritmo más reciente, creado por Zamir y Etzioni [Zamir y Etzioni, 1998].

Con esta comparación se pretende analizar cuál es el mejor algoritmo para agrupar la información en la Web. Para los experimentos de este caso de estudio fueron utilizados varios parámetros, los cuales se considera pueden afectar el desempeño de los algoritmos. Son los parámetros para el *agrupamiento* y para etiquetar los grupos. Las figuras 4.7 y 4.8 muestran la interfaz del sistema donde están ambos tipos de parámetros.

The screenshot shows the 'CONQUIRO metasearch' interface with the 'Advanced search' tab selected. At the top right, there are links for 'History of queries' and 'Help'. Below the search bar, the 'Source' is set to 'Google'. The 'Clustering Parameters' section includes:

- Algorithm: K-Means
- Use as documents: Summaries (snippets)
- Term weighting: TF
- Term threshold: (empty text box)
- Distance / Similarity: Cosine similarity
- Number of clusters: (empty text box)

 The 'Labeling Parameters' section includes:

- Method: Inverse Document Frequency
- Number of terms in label: (empty text box)

Figura 4.7 Parámetros de *agrupamiento* con el parámetro *Number of clusters* y parámetros para etiquetar los grupos.

This screenshot is similar to Figure 4.7 but with different clustering parameters. The 'Clustering Parameters' section includes:

- Algorithm: Bisecting K-means
- Use as documents: Summaries (snippets)
- Term weighting: TF
- Term threshold: (empty text box)
- Distance / Similarity: Cosine similarity
- Repeat Bisecting Step: (empty text box)

 The 'Labeling Parameters' section remains the same as in Figure 4.7:

- Method: Inverse Document Frequency
- Number of terms in label: (empty text box)

Figura 4.8 Parámetros de *agrupamiento* con el parámetro *Repeat Bisecting Step* y parámetros para etiquetar los grupos.

Los parámetros de *agrupamiento*⁴ son los siguientes:

Term weighting: Es el método (*tf* o *tf-idf*) a utilizar para asignar pesos a los términos.

Term threshold: Es el umbral a utilizar en el método descrito en la sección 2.3.4.

Distance/Similarity: Es el método de distancia o semejanza (distancia euclidiana o semejanza de coseno) a utilizar para la construcción de la matriz de semejanza.

Number of clusters: Este parámetro es el número de *clusters* que *K-Means* va a obtener; corresponde con la cantidad de categorías que tiene el *Gold Standard* sin contar la categoría “Otros”.

⁴ En la sección 3.3.1 hay una descripción más detallada de los parámetros de *agrupamiento* y de los parámetros para etiquetar los grupos.

Repeat Bisecting Step: Este parámetro indica el número de veces que *Bisecting K-Means* va a ejecutar *K-Means*. Para los experimentos el valor de este parámetro fue 5.

Los parámetros para etiquetar los grupos son los siguientes:

Method: El método (*Inverse Document Frequency, Frequent and Predictive Words, Common Term in the Cluster, Phrases*) que se va a utilizar para etiquetar los grupos.

Number of terms in label: El número de términos o palabras que va a tener la etiqueta. Este parámetro no aplica cuando el método es frases.

En las siguientes secciones se explica con detalle los experimentos que se realizaron y se muestran los resultados de las evaluaciones de éstos.

4.2.1 Experimentos

Los experimentos se realizaron usando los documentos (título y el sumario que Google proporciona de la página de la Web) en inglés de las consultas que se muestran en el cuadro 4.1.

| Query | Idioma | Tipo | Número de documentos |
|--------|--------|---------|----------------------|
| star | inglés | ambigua | 189 |
| jaguar | inglés | ambigua | 210 |
| salsa | inglés | ambigua | 198 |
| apple | inglés | ambigua | 180 |

Cuadro 4.1 Consultas utilizadas en los experimentos.⁵

Los documentos de las consultas “*star*” y “*apple*” fueron obtenidos por el sistema *Conquiro* de Google [Google, 2001] y los documentos de la consulta “*jaguar*” y “*salsa*” provinieron de la base de datos que Crabtree⁶ [Crabtree, 2004] utilizó para su trabajo.

Antes de agrupar los documentos con alguno de los algoritmos, excepto para *Suffix Tree Clustering*, el proceso fue el siguiente:

1. Procesamiento de los documentos, como se describe en la sección 2.3.1.
2. Eliminación de términos con el método descrito en la sección 2.3.4, cuyo umbral sea igual al valor del parámetro *Term threshold*. Los valores utilizados para este parámetro fueron 2, 3 y 5.
3. Asignación de pesos a los términos del documento con el método especificado en el parámetro *Term weighting*, que está descrito en la sección 2.3.2.
4. Construcción de la matriz de semejanza como se describe en la sección 2.2 con la medida especificada en el parámetro *Distance/Similarity*.

⁵ Estas consultas son ambiguas porque tienen muchos significados.

⁶ <http://www.danielcrabtree.com/research/wi05/rawdata.zip>

Para el caso de *Suffix Tree Clustering* el único paso que se lleva a cabo es el 1.

Una vez que se agrupan los documentos, cada uno de los grupos generados por los algoritmos son etiquetados. El método de frases es utilizado para *Suffix Tree Clustering* y los métodos de Frecuencia Inversa de Documentos (*Inverse Document Frequency*), Palabras Frecuentes y Predictivas (*Frequent and Predictive Words*) y Término Común en el Grupo (*Common Term in the Cluster*) para el resto de los algoritmos.

Para calcular el *precision* y *recall*⁷ de los algoritmos, se utilizó el *Gold Standard*. El *Gold Standard* de las consultas “jaguar” y “salsa” fue obtenido de la base de Crabtree [Crabtree, 2004] y el de las consultas “star” y “apple” fue construido manualmente de la siguiente manera: para asignar las categorías a cada documento se consultó el directorio de Yahoo [Yahoo, 2006] y el de Google [Google, 2006]. El *Gold Standard* de las cuatro consultas (“star”, “jaguar”, “salsa” y “apple”) que se realizaron con el sistema *Conquiro* está en el apéndice B.

4.2.2 Resultados

La influencia que cada uno de los parámetros de *agrupamiento* tiene en el desempeño de los algoritmos se refleja en los valores de *precision* y *recall*. Los experimentos que se hicieron con los documentos de las consultas “star” y “jaguar” lo demuestran.

Los experimentos consistieron en:⁸

- 1) Comparar los valores de *precision* y *recall* de los algoritmos cuando se utiliza como medida la semejanza de coseno o la distancia euclidiana (parámetro *Distance/Similarity*).
- 2) Comparar los valores de *precision* y *recall* de los algoritmos cuando se utilizan como métodos de asignación de pesos a los términos *tf* o *tf-idf* (parámetro *Term weighting*)
- 3) Comparar los valores de *precision* y *recall* para distintos valores en el parámetro *Term threshold*.

En estos experimentos el algoritmo *HAC (complete) dendrogram pruning* sólo fue utilizado con semejanza de coseno.

Para el experimento 1, los valores utilizados en los parámetros de *agrupamiento* y en los parámetros para etiquetar los grupos aparecen en el cuadro 4.2.

⁷ El cálculo de *precision* y *recall* se hizo automáticamente con un módulo de estadísticas independiente de Conquiro.

⁸ En los resultados de estas comparaciones no se muestra el *precision* y *recall* del algoritmo *Suffix Tree Clustering* porque los parámetros utilizados no aplican para éste.

| Parámetros de Agrupamiento | Parámetros para etiquetar grupos |
|----------------------------|-------------------------------------|
| Term Weighting : tf | Method : Inverse Document Frequency |
| Term threshold : 2 | Number of terms in label : 3 |

Cuadro 4.2 Parámetros utilizados en el experimento 1.

Los resultados de *precision* y *recall* de este experimento están en los cuadros 4.3 a 4.6, donde las dos primeras corresponden a la consulta “*star*” y las restantes a la consulta “*jaguar*”.

| Algorithm | Distance/Similarity | Precision | Recall |
|----------------------------------|---------------------|-----------|--------|
| HAC(complete) dendrogram pruning | cosine similarity | 68% | 84% |
| K-Means | cosine similarity | 56% | 56% |
| HAC(single) | cosine similarity | 51% | 97% |
| HAC(UPGMA) | cosine similarity | 49% | 86% |
| HAC(complete) | cosine similarity | 46% | 84% |
| Bisecting K-means | cosine similarity | 43% | 86% |

Cuadro 4.3 *Precision* y *Recall* de los algoritmos cuando se utiliza como medida la semejanza de coseno (consulta “*star*”).

| Algorithm | Distance/Similarity | Precision | Recall |
|-------------------|---------------------|-----------|--------|
| K-Means | euclidean distance | 65% | 65% |
| HAC(SINGLE) | euclidean distance | 51% | 97% |
| HAC(UPGMA) | euclidean distance | 49% | 86% |
| HAC(COMPLETE) | euclidean distance | 46% | 84% |
| Bisecting K-means | euclidean distance | 40% | 98% |

Cuadro 4.4 *Precision* y *Recall* de los algoritmos cuando se utiliza como medida la distancia euclidiana (consulta “*star*”).

| Algorithm | Distance/Similarity | Precision | Recall |
|----------------------------------|---------------------|-----------|--------|
| HAC(COMPLETE) dendrogram pruning | cosine similarity | 68% | 93% |
| K-Means | cosine similarity | 63% | 64% |
| Bisecting K-means | cosine similarity | 47% | 98% |
| HAC(COMPLETE) | cosine similarity | 47% | 94% |
| HAC(UPGMA) | cosine similarity | 45% | 96% |
| HAC(SINGLE) | cosine similarity | 42% | 99% |

Cuadro 4.5 *Precision* y *Recall* de los algoritmos cuando se utiliza como medida la semejanza de coseno (consulta “*jaguar*”).

| Algorithm | Distance/Similarity | Precision | Recall |
|-------------------|---------------------|-----------|--------|
| K-Means | euclidean distance | 65% | 65% |
| HAC(COMPLETE) | euclidean distance | 47% | 97% |
| HAC(UPGMA) | euclidean distance | 45% | 98% |
| HAC(SINGLE) | euclidean distance | 43% | 99% |
| Bisecting K-means | euclidean distance | 38% | 99% |

Cuadro 4.6 *Precision* y *Recall* de los algoritmos cuando se utiliza como medida la distancia euclidiana (consulta “*jaguar*”).

De los experimentos anteriores se tiene que los algoritmos que obtienen mayor *precision* son HAC (complete) dendrogram pruning, con semejanza de coseno y K-

Means con distancia euclidiana; sin embargo en ambas consultas se obtiene una mayor *precision* para la mayoría de los algoritmos cuando se utiliza como medida la *semejanza de coseno*. Estos resultados confirman que la semejanza de coseno es apropiada para la agrupación de documentos [Shyu *et al.*, 2004; Strehl *et al.*, 2000], por lo cual ésta será utilizada en los experimentos 2 y 3.

Para el experimento 2, respecto de los métodos *tf* y *tf-idf*, los valores utilizados en los parámetros de *agrupamiento* y en los parámetros para etiquetar los grupos están en el cuadro 4.7.

| Parámetros de Agrupamiento | Parámetros para etiquetar grupos |
|----------------------------|-------------------------------------|
| Term threshold : 2 | Method : Inverse Document Frequency |
| | Number of terms in label : 3 |

Cuadro 4.7 Parámetros utilizados en el experimento 2.

Los resultados de *precision* y *recall* de este experimento aparecen en los cuadros 4.8 a 4.11, donde las dos primeras corresponden a la consulta “*star*” y las restantes a “*jaguar*”.

| Algorithm | Distance/Similarity | Term Weighting | Precision | Recall |
|----------------------------------|---------------------|----------------|-----------|--------|
| HAC(COMPLETE) dendrogram pruning | cosine similarity | tf-idf | 78% | 79% |
| HAC(UPGMA) | cosine similarity | tf-idf | 60% | 83% |
| Bisecting K-means | cosine similarity | tf-idf | 58% | 83% |
| K-Means | cosine similarity | tf-idf | 51% | 52% |
| HAC(SINGLE) | cosine similarity | tf-idf | 49% | 87% |
| HAC(COMPLETE) | cosine similarity | tf-idf | 47% | 82% |

Cuadro 4.8 *Precision* y *Recall* de los algoritmos cuando se utiliza el método *tf-idf* (consulta “*star*”).

| Algorithm | Distance/Similarity | Term Weighting | Precision | Recall |
|----------------------------------|---------------------|----------------|-----------|--------|
| HAC(COMPLETE) dendrogram pruning | cosine similarity | tf | 68% | 84% |
| K-Means | cosine similarity | tf | 56% | 56% |
| HAC(SINGLE) | cosine similarity | tf | 51% | 97% |
| HAC(UPGMA) | cosine similarity | tf | 49% | 86% |
| HAC(COMPLETE) | cosine similarity | tf | 46% | 84% |
| Bisecting K-means | cosine similarity | tf | 43% | 86% |

Cuadro 4.9 *Precision* y *Recall* de los algoritmos cuando se utiliza el método *tf* (consulta “*star*”).

| Algorithm | Distance/Similarity | Term Weighting | Precision | Recall |
|----------------------------------|---------------------|----------------|-----------|--------|
| HAC(COMPLETE) dendrogram pruning | cosine similarity | tf-idf | 89% | 88% |
| Bisecting K-means | cosine similarity | tf-idf | 72% | 94% |
| HAC(UPGMA) | cosine similarity | tf-idf | 61% | 93% |
| K-Means | cosine similarity | tf-idf | 50% | 50% |
| HAC(COMPLETE) | cosine similarity | tf-idf | 48% | 92% |
| HAC(SINGLE) | cosine similarity | tf-idf | 44% | 97% |

Cuadro 4.10 *Precision* y *Recall* de los algoritmos cuando se utiliza el método *tf-idf* (consulta “*jaguar*”).

| Algorithm | Distance/Similarity | Term Weighting | Precision | Recall |
|----------------------------------|---------------------|----------------|-----------|--------|
| HAC(COMPLETE) dendrogram pruning | cosine similarity | tf | 68% | 93% |
| K-Means | cosine similarity | tf | 63% | 64% |
| Bisecting K-means | cosine similarity | tf | 47% | 98% |
| HAC(COMPLETE) | cosine similarity | tf | 47% | 94% |
| HAC(UPGMA) | cosine similarity | tf | 45% | 96% |
| HAC(SINGLE) | cosine similarity | tf | 42% | 99% |

Cuadro 4.11 *Precision y Recall* de los algoritmos cuando se utiliza el método *tf* (consulta “jaguar”).

En los resultados de ambas consultas se puede observar los valores más altos de *precision* se obtienen con el método *tf-idf*. El algoritmo *HAC (complete) dendrogram pruning* tiene valores de *precision* altos con ambos métodos de asignación de pesos. En el resto de los algoritmos se observa que sus valores de *precision* varían dependiendo del método de asignación de pesos.

Debido a que con el método *tf-idf* se obtienen valores altos de *precision*, éste será usado en el experimento 3.

Para el experimento 3, respecto de los valores en el parámetro *Term threshold*, los valores utilizados en los parámetros para etiquetar los grupos aparecen en el cuadro 4.12.

| Parámetros para etiquetar grupos |
|-------------------------------------|
| Method : Inverse Document Frequency |
| Number of terms in label : 3 |

Cuadro 4.12 Parámetros utilizados en el experimento 3.

Los resultados de *precision y recall* de este experimento están en los cuadros 4.13 y 4.14, que corresponden a la consulta “*star*” y a la consulta “*jaguar*”, respectivamente.

En los resultados de ambas consultas se puede observar que, dependiendo del valor que tenga el parámetro *Term threshold*, los algoritmos aumentan o disminuyen su valor de *precision*, lo cual ocasiona que sus resultados sean mejores que otros algoritmos, o bien no tan buenos. La razón de este comportamiento es que, con este parámetro, se están eliminando términos del vector; los cuadros 4.15 y 4.16 muestran el número de términos que el vector de los documentos tiene por cada uno de los valores asignados al parámetro *Term threshold* para las consultas “*star*” y “*jaguar*”.

| Algorithm | Distance/Similarity | Term Weighting | Term Threshold | Precision | Recall |
|----------------------------------|---------------------|----------------|----------------|-----------|--------|
| HAC(COMPLETE) dendrogram pruning | cosine similarity | tf-idf | 5 | 79% | 75% |
| Bisecting K-means | cosine similarity | tf-idf | 5 | 60% | 83% |
| HAC(UPGMA) | cosine similarity | tf-idf | 5 | 59% | 82% |
| K-Means | cosine similarity | tf-idf | 5 | 55% | 56% |
| HAC(SINGLE) | cosine similarity | tf-idf | 5 | 55% | 87% |
| HAC(COMPLETE) | cosine similarity | tf-idf | 5 | 46% | 79% |
| HAC(COMPLETE) dendrogram pruning | cosine similarity | tf-idf | 3 | 76% | 75% |
| HAC(UPGMA) | cosine similarity | tf-idf | 3 | 60% | 82% |
| Bisecting K-means | cosine similarity | tf-idf | 3 | 59% | 87% |
| K-Means | cosine similarity | tf-idf | 3 | 54% | 55% |
| HAC(SINGLE) | cosine similarity | tf-idf | 3 | 53% | 87% |
| HAC(COMPLETE) | cosine similarity | tf-idf | 3 | 45% | 81% |
| HAC(COMPLETE) dendrogram pruning | cosine similarity | tf-idf | 2 | 78% | 79% |
| HAC(UPGMA) | cosine similarity | tf-idf | 2 | 60% | 83% |
| Bisecting K-means | cosine similarity | tf-idf | 2 | 58% | 83% |
| K-Means | cosine similarity | tf-idf | 2 | 51% | 52% |
| HAC(SINGLE) | cosine similarity | tf-idf | 2 | 49% | 87% |
| HAC(COMPLETE) | cosine similarity | tf-idf | 2 | 47% | 82% |

Cuadro 4.13 *Precision y Recall* de los algoritmos para diversos valores del parámetro *Term Threshold* (consulta “star”).

| Algorithm | Distance/Similarity | Term Weighting | Term Threshold | Precision | Recall |
|----------------------------------|---------------------|----------------|----------------|-----------|--------|
| HAC(COMPLETE) dendrogram pruning | cosine similarity | tf-idf | 5 | 89% | 88% |
| Bisecting K-means | cosine similarity | tf-idf | 5 | 74% | 95% |
| K-Means | cosine similarity | tf-idf | 5 | 65% | 65% |
| HAC(UPGMA) | cosine similarity | tf-idf | 5 | 62% | 94% |
| HAC(COMPLETE) | cosine similarity | tf-idf | 5 | 52% | 91% |
| HAC(SINGLE) | cosine similarity | tf-idf | 5 | 45% | 97% |
| HAC(COMPLETE) dendrogram pruning | cosine similarity | tf-idf | 3 | 89% | 86% |
| Bisecting K-means | cosine similarity | tf-idf | 3 | 70% | 94% |
| HAC(UPGMA) | cosine similarity | tf-idf | 3 | 64% | 94% |
| K-Means | cosine similarity | tf-idf | 3 | 60% | 61% |
| HAC(COMPLETE) | cosine similarity | tf-idf | 3 | 48% | 91% |
| HAC(SINGLE) | cosine similarity | tf-idf | 3 | 45% | 97% |
| HAC(COMPLETE) dendrogram pruning | cosine similarity | tf-idf | 2 | 89% | 88% |
| Bisecting K-means | cosine similarity | tf-idf | 2 | 72% | 94% |
| HAC(UPGMA) | cosine similarity | tf-idf | 2 | 61% | 93% |
| K-Means | cosine similarity | tf-idf | 2 | 50% | 50% |
| HAC(COMPLETE) | cosine similarity | tf-idf | 2 | 48% | 92% |
| HAC(SINGLE) | cosine similarity | tf-idf | 2 | 44% | 97% |

Cuadro 4.14 *Precision y Recall* de los algoritmos para diversos valores del parámetro *Term Threshold* (consulta “jaguar”).

| Term Weighting | Term Threshold | Número de términos en el vector |
|----------------|----------------|---------------------------------|
| tf-idf | 5 | 255 |
| tf-idf | 3 | 291 |
| tf-idf | 2 | 415 |

Cuadro 4.15 Número de términos del vector en la consulta “star” para diferentes valores en el parámetro *Term Threshold*

| Term Weighting | Term Threshold | Número de términos en el vector |
|----------------|----------------|---------------------------------|
| tf-idf | 5 | 362 |
| tf-idf | 3 | 398 |
| tf-idf | 2 | 530 |

Cuadro 4.16 Número de términos del vector en la consulta “jaguar” para diferentes valores en el parámetro *Term Threshold*

En las evaluaciones de las consultas “star” y “jaguar” (cuadros 4.13 y 4.14) se observa que hay valores buenos de *precision* y *recall* para los tres diferentes valores del parámetro *Term threshold*, pero con *Term threshold* igual a 2 se obtienen valores de *precision* aceptablemente altos con un *recall* alto; sin embargo es importante aclarar que es preferible obtener un *precision* alto, aunque el *recall* no lo sea tanto.

De acuerdo con lo anterior, los tres algoritmos con los mejores resultados en la evaluación están en el cuadro 4.17.

| Algorithm | Precision | Recall |
|----------------------------------|-----------|--------|
| HAC(COMPLETE) dendrogram pruning | 89% | 88% |
| Bisecting K-means | 72% | 94% |
| HAC(UPGMA) | 61% | 93% |

Cuadro 4.17 Los algoritmos con mejor valor de *precision*.

Ahora si comparamos estos algoritmos con los valores de *precision* del algoritmo *Suffix Tree Clustering*, obtenemos el siguiente cuadro:

| Algorithm | Precision | Recall |
|----------------------------------|-----------|--------|
| HAC(COMPLETE) dendrogram pruning | 89% | 88% |
| Suffix Tree Clustering | 88% | 54% |
| Bisecting K-means | 72% | 94% |
| HAC(UPGMA) | 61% | 93% |

Cuadro 4.18 Comparación del algoritmo *Suffix Tree Clustering* con los algoritmos más comunes en agrupamiento (*clustering*)

De estas comparaciones concluimos que *Suffix Tree Clustering* es el segundo mejor algoritmo, seguido por *Bisecting K-Means* y *HAC (UPGMA)* y que *HAC (complete) dendrogram pruning* es el mejor de todos. Para confirmar estos resultados, realizamos más experimentos con los documentos de las consultas “salsa” y “apple”.

Los experimentos fueron realizados con los siguientes valores en los parámetros de *agrupamiento* y en los parámetros para etiquetar los grupos.

| Parámetros de Agrupamiento | Parámetros para etiquetar grupos |
|--|-------------------------------------|
| Term Weighting : tf-idf | Method : Inverse Document Frequency |
| Term threshold : 2 | Number of terms in label : 3 |
| Distance/Similarity: cosine similarity | |

Cuadro 4.19 Parámetros utilizados en los experimentos finales

En los cuadros 4.20 y 4.21 se muestran los resultados de estos experimentos.

| Algorithm | Precision | Recall |
|----------------------------------|-----------|--------|
| HAC(COMPLETE) dendrogram pruning | 82% | 83% |
| Suffix Tree Clustering | 74% | 53% |
| Bisecting K-means | 67% | 94% |
| HAC(UPGMA) | 59% | 94% |

Cuadro 4.20 Evaluación de *Precision* y *Recall* para la consulta “salsa”.

| Algorithm | Precision | Recall |
|----------------------------------|------------|------------|
| HAC(COMPLETE) dendrogram pruning | 91% | 92% |
| Bisecting K-means | 76% | 98% |
| Suffix Tree Clustering | 71% | 65% |
| K-Means | 69% | 69% |
| HAC(UPGMA) | 68% | 96% |

Cuadro 4.21 Evaluación de *Precision* y *Recall* para la consulta “apple”.

De los resultados mostrados en los cuadros anteriores se confirma que *HAC (complete) dendrogram pruning*, *Suffix Tree Clustering* y *Bisecting K-Means* son los mejores algoritmos porque tienen un *precision* más alto que los demás. Zamir y Etzioni [Zamir y Etzioni, 1998] afirman que con *Suffix Tree Clustering* se obtienen mejores resultados que con los algoritmos basados en el VSM; sin embargo los experimentos realizados contradicen esta afirmación, ya que se obtuvieron mejores resultados con *HAC (complete) dendrogram pruning*; y para el caso de la consulta “apple”, *Bisecting K-Means* tuvo un mejor desempeño que *Suffix Tree Clustering*.

A pesar de que *HAC (complete) dendrogram pruning* obtuvo mejores evaluaciones que el resto de los algoritmos, su complejidad cuadrática hace que los algoritmos *Suffix Tree Clustering* y *Bisecting K-Means* sean más atractivos para agrupar colecciones de documentos grandes.

Por otro lado, Steinbach [Steinbach *et al.*, 2000] afirma que con *Bisecting K-Means* se obtienen mejores resultados que con *K-Means* y *HAC (UPGMA)*; los experimentos anteriores confirman esta afirmación.

Dubes y Jain [Dubes y Jain, 1988] mencionan que se considera que *K-Means* no es tan bueno como los algoritmos *HAC*; sin embargo en los experimentos realizados anteriormente se observa que, dependiendo de los parámetros utilizados, *K-Means* tiene una mejor evaluación que los algoritmos *HAC (single)*, *HAC (complete)*, *HAC (UPGMA)* e incluso *Bisecting K-Means*.

De lo anterior se puede concluir que los parámetros de *agrupamiento* tienen una gran influencia en el desempeño de los algoritmos, por lo cual es importante asignar a los parámetros los valores con los cuales el algoritmo tenga un desempeño óptimo.

Además de evaluar *precision* y *recall*, es importante evaluar que las etiquetas de los grupos ayuden al usuario a identificar el tema de los documentos del grupo. Debido a la importancia que las etiquetas tienen en el proceso de *agrupamiento*, en este trabajo se hizo una evaluación cualitativa de cuatro métodos (*Inverse Document Frequency*, *Frequent and Predictive Words*, *Common Term in the Cluster* y *Frases*) para asignar etiquetas a los grupos generados para la consulta “*star*”.

Los métodos *Inverse Document Frequency*, *Frequent and Predictive Words* y *Common Term in the Cluster* fueron utilizados para crear las etiquetas de los grupos generados por el algoritmos *HAC (complete) dendrogram pruning*; y el método de frases es utilizado por el algoritmo *Suffix Tree Clustering* para crear las etiquetas de los grupos que genera.

La evaluación de estos métodos consistió en verificar si la etiqueta efectivamente ayuda al usuario a identificar el tema de los documentos de cada uno de los grupos generados. Las figuras 4.9 a la 4.13 muestran las etiquetas que cada método asignó a los grupos generados para la consulta “*star*”.

De los cuatro métodos evaluados se encontró que las etiquetas generadas por el método de *Frases* y el *Common Term in the Cluster* ayudan a identificar el tema de los documentos que los grupos contienen. Las etiquetas del método de *Frases* se caracterizan por contener la frase común a la mayoría de los documentos del grupo; y las etiquetas del método *Common Term in the Cluster*, por contener el término que es común a la mayoría de los documentos del grupo.

Estas observaciones coinciden con lo que Treeratpituk y Callan [Treeratpituk y Callan, 2006] mencionan en su trabajo: “una lista de términos no es una buena elección como etiqueta de un grupo porque el usuario necesita inferir el concepto de la lista de términos.”.

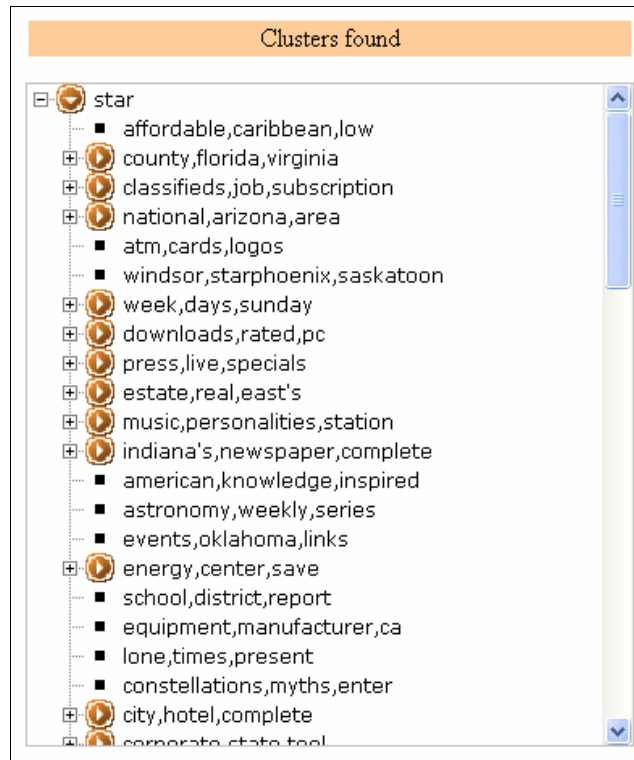


Figura 4.9 Grupos etiquetados con el método *Inverse Document Frequency*.
(Se utilizó *HAC (complete) dendrogram prunnig*)

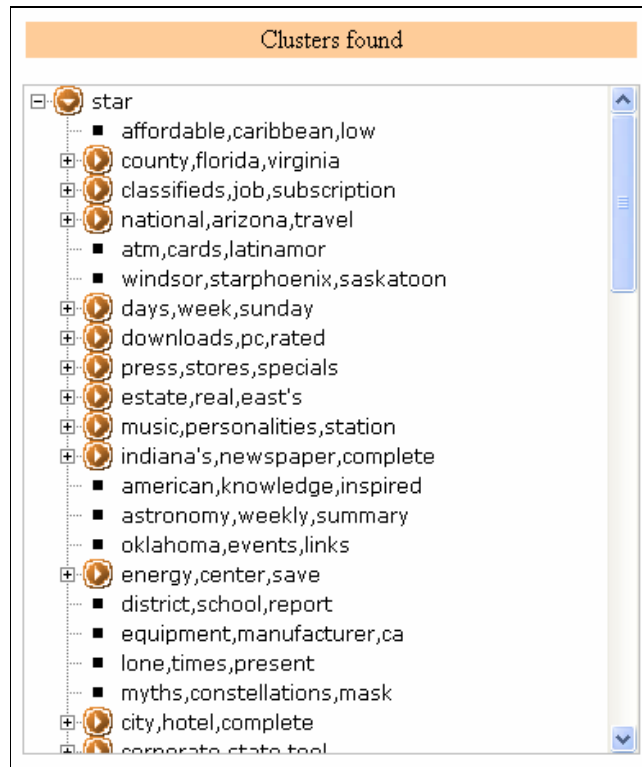


Figura 4.10 Grupos etiquetados con el método *Frequent and Predictive Words*.
(Se utilizó *HAC (complete) dendrogram prunnig*)

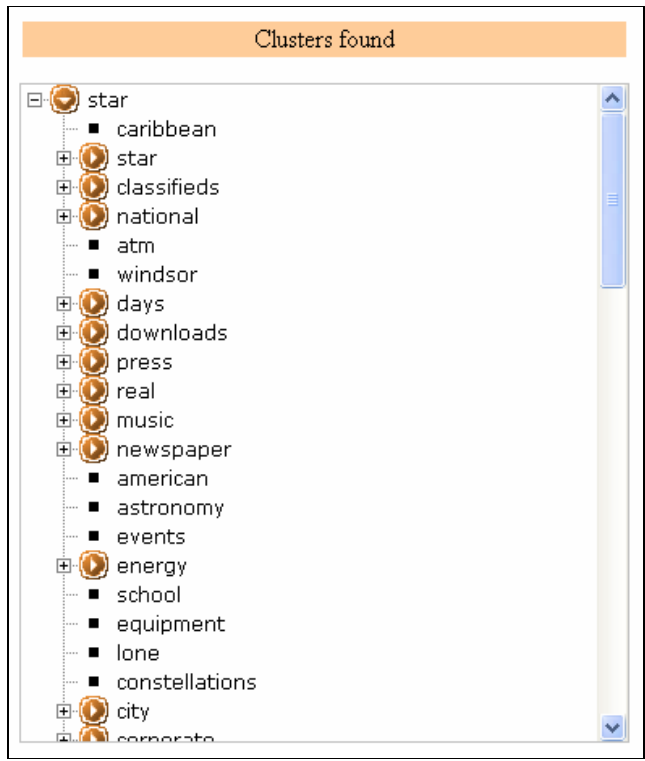


Figura 4.11 Grupos etiquetados con el método *Common Term in the Cluster*.
(Se utilizó *HAC (complete) dendrogram prunnig*)

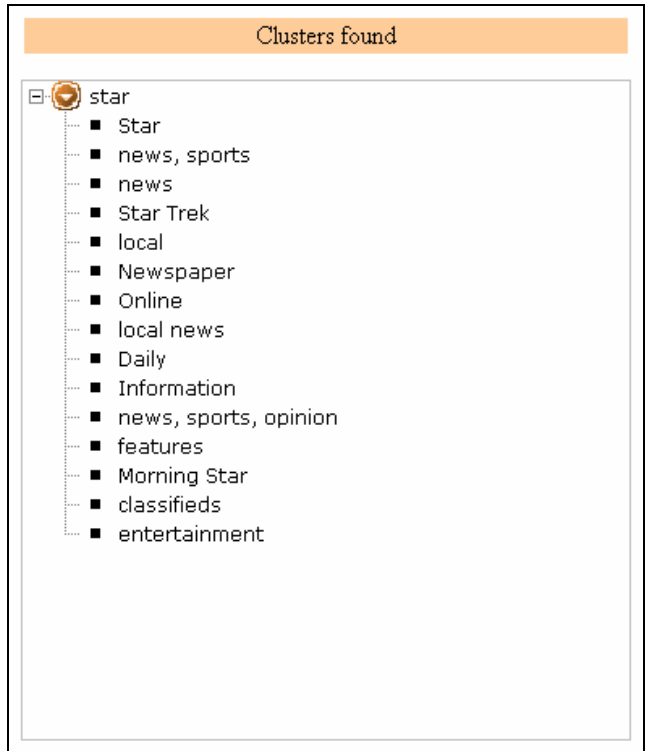


Figura 4.12 Grupos etiquetados con el método de Frases. (Se utilizó *Suffix Tree Clustering*).

4.3 Caso de estudio 2: *Newsgroups*

En esta sección se presenta el *agrupamiento* de mensajes de un *newsgroup* con el algoritmo *HAC (complete) dendrogram pruning*; los grupos generados fueron etiquetados con el método *Common Term in the Cluster*.

Los grupos de mensajes son presentados a los usuarios para su evaluación. El objetivo de realizar esta evaluación es comprobar si los algoritmos de *agrupamiento (clustering)* agrupan eficientemente los mensajes por tema de discusión.

4.3.1 *Datos*

Para este caso de estudio fueron utilizados 100 mensajes del *newsgroup misc.forsale*,⁹ en los cuales se venden, compran o rentan diversos artículos. La información que utilizó el algoritmo para hacer la agrupación fue el asunto del mensaje (*subject*) y el mensaje completo.

4.3.2 *Evaluación de usuarios*

La evaluación de usuario consistió en evaluar la agrupación que el algoritmo *HAC (complete) dendrogram pruning* hizo de los mensajes y si las etiquetas de los grupos les dio idea del contenido de los mensajes del grupo; las etiquetas fueron creadas con el método *Common Term in the Cluster*.

Para hacer la evaluación de la agrupación que el algoritmo hizo, se pidió a 5 usuarios que calificaran cada uno de los grupos generados por el algoritmo (71 grupos) de acuerdo con la siguiente escala:

| | | |
|----------|---|---|
| Todos | = | Todos los mensajes del grupo son similares entre sí. |
| No todos | = | No todos los mensajes del grupo son similares entre sí. |
| Ninguno | = | Ninguno de los mensajes del grupo son similares entre sí. |

El objetivo es evaluar si el algoritmo está formando grupos de mensajes similares entre sí, es decir, que tratan sobre el mismo tema. Para los mensajes utilizados en esta evaluación, significa que los mensajes de un grupo estén relacionados con la compra o venta de artículos parecidos entre sí.

Para la evaluación de las etiquetas de los grupos, se pidió a los usuarios, una vez que vieron los mensajes del grupo, que calificaran si la etiqueta les dio idea del contenido de los mensajes de acuerdo con la siguiente escala: mucho, poco o nada. Esta evaluación fue para cada uno de los 71 grupos de mensajes.

El apéndice C muestra la forma de evaluación que se les aplicó a los usuarios.

⁹ Los mensajes del *newsgroup* fueron tomados de una colección de mensajes de 20 *newsgroups*, ordenados por fecha y de los cuales fueron removidos encabezados y duplicados. Los mensajes pueden ser encontrados en la siguiente dirección: <http://people.csail.mit.edu/jrennie/20Newsgroups/> .

4.3.3 Resultados

Los resultados presentados en esta sección muestran cómo los usuarios perciben que el algoritmo agrupó los mensajes y etiquetó estos grupos.

Para el caso de la agrupación de los mensajes los usuarios consideran que 72% de los grupos tienen todos sus mensajes similares entre sí; 24% de los grupos tienen mensajes que no son similares entre sí; y no todos los mensajes de 4% de los *clusters* son similares entre sí. Las figuras 4.13 y 4.14 muestran estos resultados.

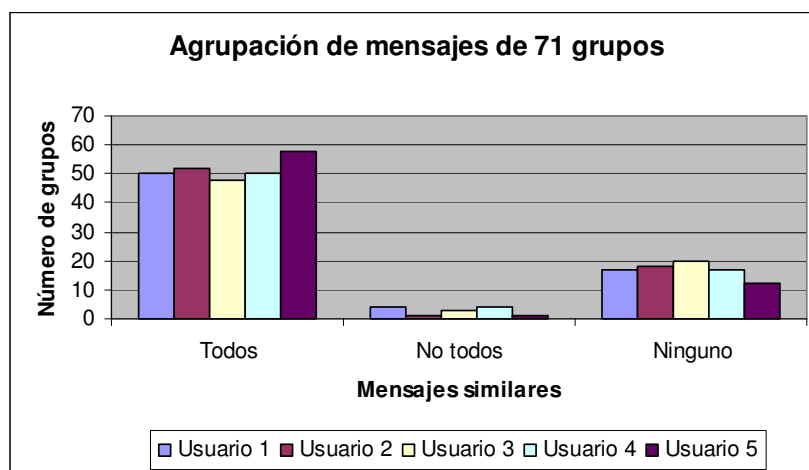


Figura 4.13 Evaluación de la agrupación de los mensajes por usuarios.

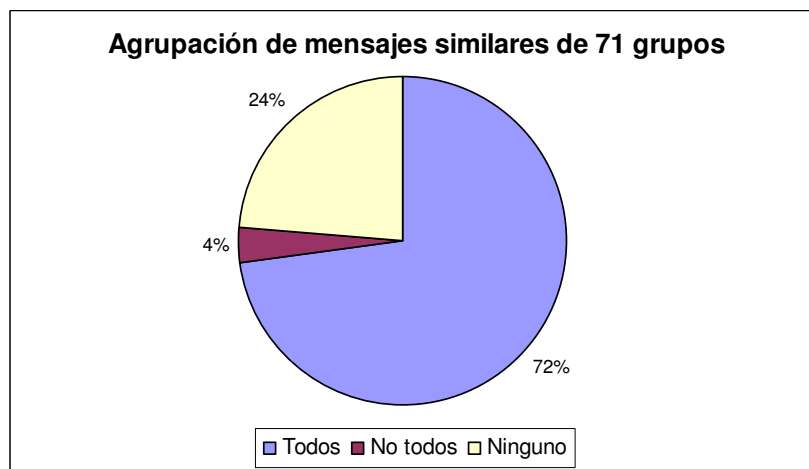


Figura 4.14 Evaluación de la agrupación de los mensajes.

Para confirmar qué tan confiables son estas evaluaciones, se valoró si los usuarios calificaron bien la agrupación que el algoritmo hizo de los mensajes. De esta evaluación se obtuvo que 93% de los grupos fue bien evaluado y 7% no. La figura 4.15 muestra esta información.

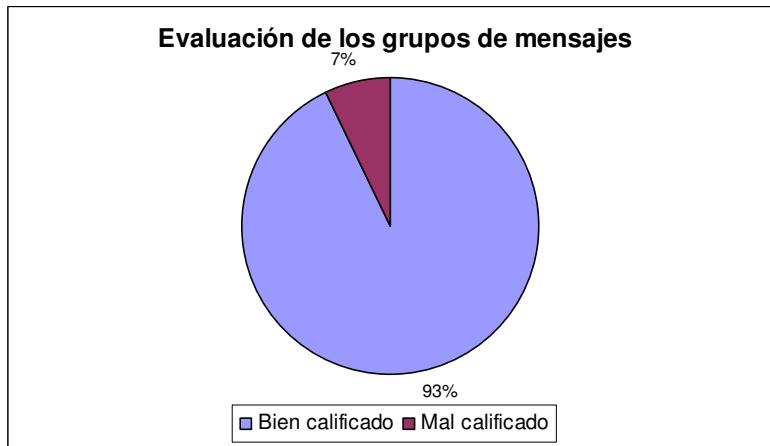


Figura 4.15 Calificación de la evaluación de los usuarios.

Respecto de las etiquetas de los grupos, los usuarios consideran que 44% de los grupos tienen etiquetas que corresponden con el contenido de sus mensajes; 25% tienen etiquetas que no dan idea del contenido de sus mensajes; 17% tienen etiquetas que dan más o menos idea del contenido de sus mensajes y 14% de los grupos tienen etiquetas que dan poca idea del contenido de sus mensajes. Las figuras 4.16 y 4.17 representan gráficamente esta información.

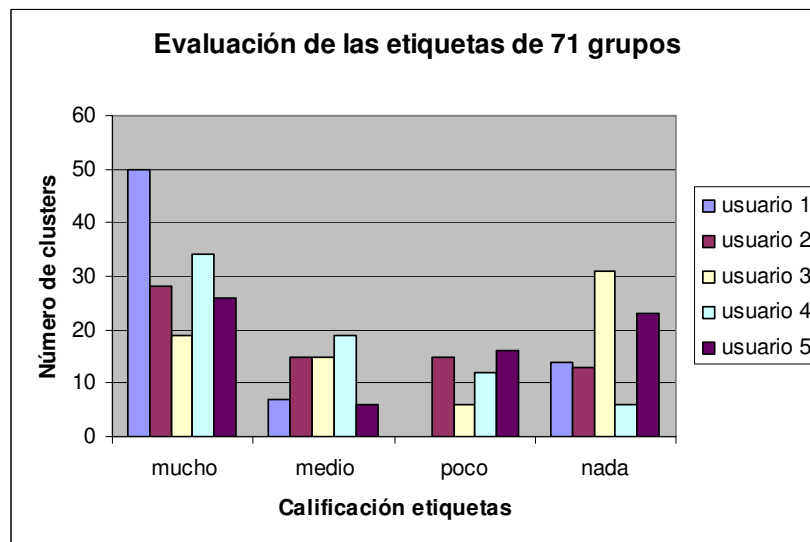


Figura 4.16 Evaluación de las etiquetas de los grupos por usuario.

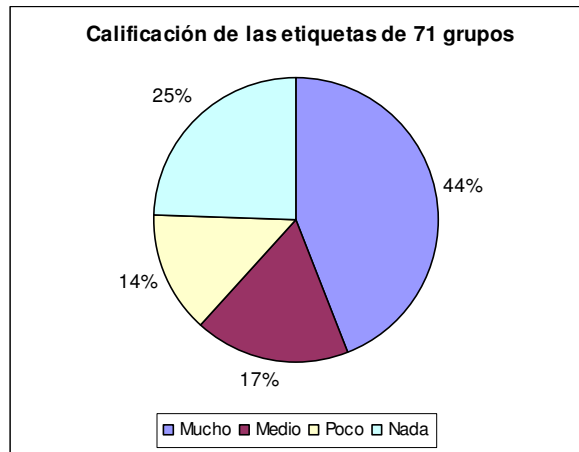


Figura 4.17 Evaluación de las etiquetas de los grupos.

Los datos anteriores indican que la agrupación que el algoritmo hizo de los mensajes fue buena y las etiquetas de la mayoría de los grupos cumplieron con dar idea al usuario del contenido de los mensajes.

Para finalizar, la figura 4.18 muestra los temas que los usuarios identificaron en los 100 mensajes del *newsgroup*.

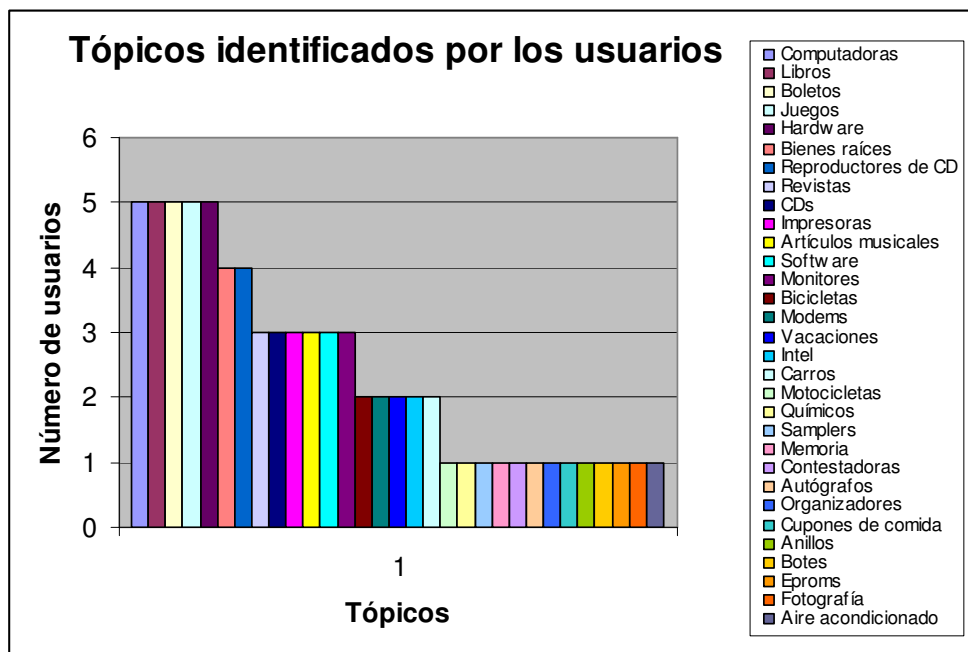


Figura 4.18 Temas que los usuarios identificaron en los mensajes.

En la figura anterior se observa que los temas que más identificaron los usuarios fueron: computadoras, libros, boletos, juegos, *hardware*, bienes raíces y reproductores de CD. Muchos temas no fueron identificados por los usuarios debido a las siguientes razones:

1. Los usuarios son hispanohablantes y, aunque tienen una buena comprensión del inglés, había artículos cuyos nombres los usuarios no conocían.
2. Como el algoritmo generó muchos grupos, después de cierto tiempo de estar evaluando, los usuarios estaban cansados.

Capítulo 5. Conclusiones y trabajo futuro

5.1 Conclusiones

El crecimiento exponencial que ha tenido la Web en estos últimos años ha creado la necesidad de proporcionar herramientas que permitan a los usuarios buscar información de manera eficiente.

Una solución propuesta para este problema es aplicar la técnica de *agrupamiento* (*clustering*) a los resultados que un motor de búsqueda genera. Esto tiene el fin de agrupar los documentos por temas para que el usuario encuentre la información que necesita de manera eficiente.

En el trabajo de investigación desarrollado para aplicar la técnica de *agrupamiento* a los resultados de un motor de búsqueda, se ha propuesto diversos algoritmos, las preguntas que surgen son:

1. ¿Con cuál de estos algoritmos se obtiene una buena agrupación de los resultados?
2. ¿Cuál de estos algoritmos genera resultados fáciles de explorar por el usuario?

Para contestar a la primera pregunta, en este trabajo se hizo una comparación de siete algoritmos (*K-Means*, *Bisecting K-Means*, *HAC (single)*, *HAC (complete)*, *HAC (complete) dendrogram pruning*, *HAC (UPGMA)* y *Suffix Tree Clustering*), de los cuales seis están basados en el VSM. En estas comparaciones se encontró que los parámetros (*Term weighting*, *Term threshold*, *Distance/Similarity*) afectan el desempeño de los algoritmos (qué tan bien se hizo la agrupación), por lo cual para comparar los algoritmos basados en el VSM con el *Suffix Tree Clustering*, se asignaron a los parámetros los valores donde estos algoritmos tuvieron mejor desempeño.

Los resultados de esta comparación muestran que el algoritmo *HAC (complete) dendrogram pruning* organizó mejor los documentos que el *Suffix Tree Clustering*, pero como la complejidad de este algoritmo es cuadrática no es el apropiado para colecciones de documentos muy grandes, por lo cual el algoritmo recomendado para la agrupación de los resultados de un motor de búsqueda es el *Suffix Tree Clustering*, cuya complejidad es lineal respecto del número de documentos.

Para contestar la segunda pregunta fueron analizados los resultados arrojados por los siete algoritmos y se encontró que *HAC (single)*, *HAC (complete)*, *HAC (UPGMA)* y *Bisecting K-Means* generaron árboles de grupos muy profundos, lo cual dificulta al usuario buscar la información de su interés, ya que tiene que examinar demasiados nodos para localizar el tema, además de que hay problemas para mostrar este árbol en un navegador. Este problema no se presenta con el árbol de grupos generados con *HAC (complete) dendrogram pruning*, debido a que es menos profundo.

Este tipo de árbol permite al usuario ver la información con diferentes niveles de detalle.

Los algoritmos *K-Means* y *Suffix Tree Clustering* producen una lista de grupos. La ventaja de generar los grupos de esta manera es que los usuarios pueden fácilmente explorar los grupos, sin embargo tiene la desventaja de que la información no puede verse a diferentes niveles de detalle como en los algoritmos que generan un árbol de grupos.

Una parte importante de la agrupación de los documentos son las descripciones de los grupos, porque éstas ayudan al usuario a identificar el tema de los documentos que cada grupo contiene y de esta manera identificar el tema de su interés. Desafortunadamente, a diferencia de *Suffix Tree Clustering*, los algoritmos basados en el VSM no asignan descripciones a los grupos que generan, por lo cual es necesario utilizar algún método que elabore estas descripciones. En este trabajo fueron utilizados tres métodos que usan términos para crear las descripciones de los grupos; dos utilizan una lista de términos en las descripciones y el otro utiliza el que sea común a los documentos del grupo.

De las descripciones que usan términos y las que usan frases, surge la pregunta: ¿cuál es la mejor para describir el contenido de un grupo? Para contestarla se compararon los tres métodos que usan términos contra el método del algoritmo *Suffix Tree Clustering*, que usa frases. De esta comparación se encontró que un término describe mejor el contenido del grupo que una lista; sin embargo, si se compara la descripción que utiliza un término contra la que utiliza una frase, la frase proporciona al usuario más información para inferir el tema del grupo, por lo tanto parecería que la mejor elección de etiqueta de un grupo son las frases. Sin embargo, no hay un consenso entre los investigadores, ya que hay quienes proponen usar términos, mientras otros proponen frases; por ejemplo, Sanderson y Croft [Sanderson y Croft, 1999] argumentan que las etiquetas que tienen un solo término son más fácilmente entendidas por el usuario, es decir, que éste puede identificar de manera más fácil el tema del *grupo*.

La técnica de *agrupamiento (clustering)* no sólo puede ser aplicada para agrupar los resultados de un motor de búsqueda, sino también para los mensajes de un *newsgroup*. En este trabajo se hizo esta aplicación y se observó que, al agrupar los mensajes, se ayuda al usuario a identificar los diversos temas discutidos en los mensajes.

5.2 Trabajo futuro

Conquiro es un sistema creado con el propósito de facilitar al usuario la búsqueda de información en la Web. Se consideró inicialmente que organizara los documentos en inglés que Google regresa como resultado de una consulta del usuario. Los resultados de la evaluación del usuario mostraron que *Conquiro* fue útil para encontrar la información; sin embargo el sistema tiene limitantes que provocan que el usuario no encuentre la información tan fácilmente como es deseable. Por ello se propone como trabajo futuro agregar al sistema una serie de características agrupadas en: interfaz, métodos para crear las etiquetas de los grupos de documentos y sistema.

Interfaz

- Agregar opciones de configuración para indicar al sistema cómo van a ser mostrados los resultados. Las opciones son las siguientes:
 - a) **Mostrar los resultados como los genera el algoritmo de agrupamiento (*clustering*)**: En esta opción los resultados se mostrarían como el sistema lo hace actualmente.
 - b) **Reagrupar los grupos seleccionados**: En esta opción el usuario selecciona los grupos de su interés a través de un *checkbox* y el sistema los reagrupa siguiendo la idea de *scatter/gather*. Con esta opción el usuario podrá ver con diferentes niveles de detalle la información que le interesa.
- Agregar una opción para que el sistema organice los documentos de una base externa. Esto permitirá organizar una cantidad arbitraria de información, actualmente está restringido a organizar hasta 200 documentos.
- Agregar la opción de consultar *newsgroups* para agrupar los mensajes por tema.
- Agregar al sistema dos opciones para buscar información:
 - a) Búsqueda simple: esta opción ofrecerá al usuario una interfaz parecida a la de Google, donde sólo escriba su consulta y el sistema agrupe los resultados con el mejor algoritmo de *agrupamiento*.
 - b) Búsqueda avanzada: esta opción es la que el sistema ofrece actualmente; el usuario puede configurar el proceso de *agrupamiento*.

Métodos para crear las etiquetas de los grupos de documentos

- Investigar nuevos métodos. Uno de los métodos que se propone utilizar es el *centroid-based-sumarization* [Radev *et al.*, 2000] para crear un sumario de los grupos generados para la consulta. Este método consiste en calcular el centroide del grupo, el cual contiene un grupo de palabras comunes a todos los documentos del grupo. Una vez calculado el centroide del grupo se calcula la semejanza que hay entre cada oración del grupo con el centroide; se evalúa las oraciones y las que tengan una puntuación más alta serán utilizadas en el sumario.
- Hacer una más detallada evaluación cualitativa de los métodos.
- Mejorar el algoritmo *Common Term in the Cluster*.

Sistema

- Hacer que el sistema pueda procesar documentos en español. Esta modificación consiste en agregarle lo siguiente:

- 1) El algoritmo que haga reducción de términos a la raíz (*stemming*) para el idioma español.
 - 2) La lista de palabras comunes (*stopwords*) para el idioma español.
- Enviar la consulta del usuario a más motores de búsqueda, como Yahoo, Altavista, etcétera.
 - Cambiar la arquitectura del sistema a una arquitectura distribuida para que el proceso de obtención de documentos y *agrupamiento* sea eficiente.
 - Agregar más algoritmos de *agrupamiento*, cuyo tiempo de ejecución sea lineal.
 - Ligar el sistema con WordNet¹ para aplicar la técnica de *expansión de consultas*² (*query expansion*) para refinar la consulta y así mejorar las evaluaciones en *precision* y *recall* [Klink *et al.*, 2002].
 - Mejorar la calidad del *agrupamiento* permitiendo que **Conquiro** cambie de ser un sistema que maneja *agrupamiento no supervisado* (*unsupervised clustering*) a uno que maneje *agrupamiento supervisado* (*supervised clustering*); es decir, permitir que **Conquiro** reciba retroalimentación del usuario para hacer los cambios necesarios en el *agrupamiento* de los documentos. Para hacer este cambio se necesita que **Conquiro** cuente con una interfaz que permita al usuario elegir los grupos que le interesan e indicar al sistema la siguiente información:
 - 1) Este documento no pertenece a este grupo
 - 2) Mover este documento a este grupo.
 - 3) Estos documentos deben (no deben) estar juntos

Con la información anterior el sistema hace un *reagrupamiento* de los documentos. Es importante señalar que estos cambios no son para todos los documentos, sino para aquellos que en particular estén mal agrupados [Cohn *et al.*, 2003].

La ventaja de esta mejora es que se incrementa la calidad del *agrupamiento* y se pueden hacer evaluaciones más precisas, aunque requiera datos de entrenamiento adicionales [Zeng *et al.*, 2004].

¹ <http://wordnet.princeton.edu/>

² La técnica de *expansión de consulta* consiste en construir una consulta nueva de la anterior, añadiéndole sinónimos o términos relacionados en la taxonomía (por ejemplo las de WordNet) u otro tipo de términos semánticamente relacionados. Para mayor referencia sobre esta técnica consúltese Efthimiadis, 1996.

Apéndices

A. Palabras Comunes (*stop words*)

| | | | | |
|-------------|------------|---------------|-------------|-----------|
| a | at | changes | either | get |
| a's | aside | clearly | eleven | gets |
| able | ask | click | else | getting |
| about | asking | co | elsewhere | gif |
| above | associated | con | empty | give |
| according | at | com | enough | given |
| accordingly | aug | come | entirely | gives |
| across | available | comes | especiallly | go |
| actually | away | concerning | et | goes |
| after | awfully | consequently | etc | going |
| afterwards | b | consider | even | gone |
| again | back | considering | ever | got |
| against | based | contain | every | gotten |
| ain't | be | containing | everybody | greetings |
| al | became | contains | everyone | h |
| all | because | corresponding | everything | had |
| allow | become | could | everywhere | hadn't |
| allows | becomes | couldn | ex | happens |
| almost | becoming | couldn't | exactly | hardly |
| alone | been | course | example | has |
| along | before | cry | except | hasn't |
| already | beforehand | currently | f | have |
| also | behind | d | far | haven't |
| although | being | dec | fax | having |
| always | believe | definitely | feb | he |
| am | below | describe | few | he's |
| among | beside | described | fifteen | hello |
| amongst | besides | despite | fifth | help |
| amongst | best | detail | fill | hence |
| amount | better | did | find | her |
| an | between | didn't | first | here |
| and | beyond | different | five | here's |
| another | both | do | followed | hereafter |
| any | bottom | does | following | hereby |
| anybody | brief | doesn't | follows | herein |
| anyhow | but | doing | for | hereupon |
| anyone | by | don't | former | hers |
| anything | c | done | formerly | herself |
| anyway | c'mon | dot | forth | hi |
| anyways | c's | down | forty | him |
| anywhere | call | downwards | found | himself |
| apart | came | dr | four | his |
| appear | can | due | frames | hither |
| appreciate | can't | during | fri | http |
| appropriate | cannot | e | from | hopefully |
| apr | cant | e-mail | front | how |
| are | cause | each | full | howbeit |
| aren't | causes | edu | further | however |
| around | certain | eg | furthermore | hundred |
| as | certainly | eight | g | i |

| | | | | |
|-----------|-------------|--------------|--------------|------------|
| i'd | lets | nov | really | sensible |
| i'll | like | novel | reasonably | sent |
| i'm | liked | now | regarding | sep |
| i've | link | nowhere | regardless | serious |
| ie | likely | o | regards | seriously |
| if | little | obviously | relatively | seven |
| ignored | look | oct | respectively | several |
| ii | looking | of | right | shall |
| iii | looks | off | s | she |
| immediate | ltd | often | said | should |
| in | m | oh | same | shouldn't |
| inasmuch | made | ok | sat | show |
| inc | mail | okay | saw | side |
| indeed | main | old | say | since |
| indicate | mainly | on | saying | sincere |
| indicated | make | once | placed | site |
| indicates | many | one | please | six |
| inner | mar | ones | plus | sixty |
| interest | may | only | possible | so |
| insofar | maybe | onto | presumably | some |
| instead | me | or | previous | somebody |
| into | mean | other | probably | somehow |
| inward | meanwhile | others | prof | someone |
| is | merely | otherwise | provides | something |
| isn't | might | ought | put | sometime |
| it | mine | our | q | sometimes |
| it'd | mon | ours | que | somewhat |
| it'll | more | ourselves | quite | somewhere |
| it's | moreover | out | qv | soon |
| its | most | outside | r | sorry |
| itself | mostly | over | rather | specified |
| iv | move | overall | rd | specify |
| ix | much | own | re | specifying |
| j | must | p | really | still |
| jan | my | page | reasonably | sub |
| java | myself | part | regarding | such |
| jpeg | n | particular | regardless | sup |
| jpg | name | particularly | regards | sure |
| jul | namely | per | relatively | system |
| jun | near | perhaps | respectively | t |
| just | nearly | pl | right | t's |
| k | necessary | placed | s | take |
| keep | need | placed | said | taken |
| keeps | needs | please | same | tel |
| kept | neither | plus | sat | tell |
| know | never | possible | saw | ten |
| knows | nevertheles | presumably | say | tends |
| known | s | previous | saying | th |
| l | new | probably | says | than |
| la | next | prof | second | thank |
| last | nine | provides | secondly | thanks |
| lately | no | put | see | thanx |
| later | nobody | q | seeing | that |
| latter | non | que | seem | that's |
| latterly | none | quite | seemed | thats |
| least | noone | qv | seeming | the |
| less | nor | r | seems | their |
| lest | normally | rather | seen | theirs |
| let | not | rd | self | them |
| let's | nothing | re | selves | themselves |

| | | | | |
|---------------|------------|------------|--|--|
| then | use | why | | |
| thence | used | will | | |
| there | useful | willing | | |
| there's | uses | wish | | |
| thereafter | using | with | | |
| thereby | usually | within | | |
| therefore | uucp | without | | |
| therein | v | won't | | |
| theres | value | wonder | | |
| thereupon | various | would | | |
| these | very | wouldn't | | |
| they | vi | x | | |
| they'd | via | xi | | |
| they'll | vii | xii | | |
| they're | viii | xiii | | |
| they've | viz | xiv | | |
| thick | vs | xv | | |
| thin | w | y | | |
| think | www | yes | | |
| third | want | yet | | |
| this | wants | you | | |
| thorough | was | you'd | | |
| thoroughly | wasn't | you'll | | |
| those | way | you're | | |
| though | we | you've | | |
| three | we'd | your | | |
| through | we'll | yours | | |
| throughout | we're | yourself | | |
| thru | we've | yourselves | | |
| thu | wed | z | | |
| thus | welcome | zero | | |
| to | well | | | |
| together | went | | | |
| too | were | | | |
| took | weren't | | | |
| top | what | | | |
| toward | what's | | | |
| towards | whats | | | |
| tried | whatever | | | |
| tries | when | | | |
| truly | whence | | | |
| try | whenever | | | |
| trying | where | | | |
| tue | where's | | | |
| twelve | whereafter | | | |
| twenty | whereas | | | |
| twice | whereby | | | |
| two | wherein | | | |
| u | whereupon | | | |
| un | wherever | | | |
| under | whether | | | |
| unfortunately | which | | | |
| unless | while | | | |
| unlikely | whither | | | |
| until | who | | | |
| unto | who's | | | |
| up | whoever | | | |
| upon | whole | | | |
| url | whom | | | |
| us | whose | | | |

B. Gold Standard de las consultas realizadas a Conquiro

Consulta "star"

| Número de categoría | Nombre de Categoría | Número de documentos |
|---------------------|--|----------------------|
| 1 | <i>News and Media (Newspapers, Magazines)</i> | 76 |
| 2 | <i>Energy</i> | 3 |
| 3 | <i>Star Trek</i> | 10 |
| 4 | <i>Nonprofit organizations</i> | 1 |
| 5 | <i>Astronomy</i> | 14 |
| 6 | <i>Real Estate</i> | 1 |
| 7 | <i>Finance and Investment</i> | 1 |
| 8 | <i>Employment</i> | 3 |
| 9 | <i>Car</i> | 1 |
| 10 | <i>Airlines</i> | 3 |
| 11 | <i>StarOffice</i> | 1 |
| 12 | <i>Music</i> | 8 |
| 13 | <i>Chinese language Word Processor</i> | 1 |
| 14 | <i>Others</i> | 2 |
| 15 | <i>Downloads</i> | 4 |
| 16 | <i>Financial Services</i> | 2 |
| 17 | <i>Web hosting</i> | 1 |
| 18 | <i>Government Organizations</i> | 2 |
| 19 | <i>Video</i> | 2 |
| 20 | <i>History</i> | 2 |
| 21 | <i>Health</i> | 1 |
| 22 | <i>Games</i> | 3 |
| 23 | <i>Television</i> | 2 |
| 24 | <i>Solenoidal Tracker At Rhic (STAR) experiment</i> | 1 |
| 25 | <i>Photography</i> | 2 |
| 26 | <i>Star registry</i> | 1 |
| 27 | <i>Cruises</i> | 2 |
| 28 | <i>Kosher Certification</i> | 1 |
| 29 | <i>Costumes</i> | 1 |
| 30 | <i>Casinos</i> | 2 |
| 31 | <i>MSI (Micro-Star International)</i> | 2 |
| 32 | <i>Reading</i> | 1 |
| 33 | <i>Resources for Teachers</i> | 1 |
| 34 | <i>Minnesota Government (state, minnesota, government)</i> | 1 |
| 35 | <i>Internships, students</i> | 1 |
| 36 | <i>Cooking, chef, magazine</i> | 1 |
| 37 | <i>Standardized Testing and Reporting (STAR)</i> | 1 |
| 38 | <i>Star Micronics Worldwide Gateway</i> | 1 |
| 39 | <i>Translation services</i> | 1 |
| 40 | <i>Entertainment portal</i> | 1 |
| 41 | <i>Entertainment search engine</i> | 1 |
| 42 | <i>Non English</i> | 1 |
| 43 | <i>horse racing</i> | 1 |
| 44 | <i>Silver Star mountain resort</i> | 1 |
| 45 | <i>Communications and Networking</i> | 1 |
| 46 | <i>Technology provider</i> | 1 |
| 47 | <i>Web site tools</i> | 1 |
| 48 | <i>CFD/CAE software</i> | 1 |
| 49 | <i>Shaw supermarket</i> | 1 |

| | | |
|----|--|---|
| 50 | <i>Minnesota STAR program</i> | 1 |
| 51 | <i>Office of Real Property Services (New York State)</i> | 1 |
| 52 | <i>Computer technical support</i> | 1 |
| 53 | <i>Ticket agent</i> | 1 |
| 54 | <i>Alaska state information</i> | 1 |
| 55 | <i>Jamaica Entertainment</i> | 1 |
| 56 | <i>Collaborative Weblogs</i> | 1 |
| 57 | <i>Webcam</i> | 1 |
| 58 | <i>IndiaStar Review of Books</i> | 1 |
| 59 | <i>Maritime Museum of San Diego</i> | 1 |
| 60 | <i>Web design</i> | 1 |
| 61 | <i>Fitness equipment</i> | 1 |
| 62 | <i>Windsurfing</i> | 1 |
| 63 | <i>Roses</i> | 1 |
| 64 | <i>Publishing</i> | 1 |

Consulta "jaguar"

| Número de categoría | Nombre de Categoría | Número de documentos |
|----------------------------|----------------------------|-----------------------------|
| 1 | <i>Aircraft</i> | 4 |
| 2 | <i>Animal</i> | 29 |
| 3 | <i>Art</i> | 1 |
| 4 | <i>Bat</i> | 1 |
| 5 | <i>Boat</i> | 1 |
| 6 | <i>Car</i> | 77 |
| 7 | <i>Chemistry</i> | 2 |
| 8 | <i>Coffee</i> | 1 |
| 9 | <i>Database</i> | 1 |
| 10 | <i>Games</i> | 29 |
| 11 | <i>Health</i> | 1 |
| 12 | <i>Hotel</i> | 1 |
| 13 | <i>Links</i> | 1 |
| 14 | <i>Macintosh</i> | 36 |
| 15 | <i>Maya</i> | 3 |
| 16 | <i>Model</i> | 1 |
| 17 | <i>Movie</i> | 1 |
| 18 | <i>Music</i> | 1 |
| 19 | <i>Network</i> | 1 |
| 20 | <i>Non English</i> | 1 |
| 21 | <i>Physics</i> | 2 |
| 22 | <i>Pizza</i> | 1 |
| 23 | <i>Power supply</i> | 1 |
| 24 | <i>Snake</i> | 1 |
| 25 | <i>Software</i> | 1 |
| 26 | <i>Star Trek</i> | 1 |
| 27 | <i>Store</i> | 2 |
| 28 | <i>Technology</i> | 1 |
| 29 | <i>Teeshirt</i> | 1 |
| 30 | <i>Others</i> | 1 |
| 31 | <i>Watch</i> | 1 |
| 32 | <i>Web Design</i> | 2 |
| 33 | <i>Web Hosting</i> | 1 |
| 34 | <i>Xfiles</i> | 1 |

Consulta "salsa"

| Número de categoría | Nombre de Categoría | Número de documentos |
|---------------------|---------------------|----------------------|
| 1 | <i>Bike</i> | 1 |
| 2 | <i>Casino</i> | 1 |
| 3 | <i>Club</i> | 4 |
| 4 | <i>DNA</i> | 1 |
| 5 | <i>Dance</i> | 71 |
| 6 | <i>Drama</i> | 1 |
| 7 | <i>Fabric</i> | 1 |
| 8 | <i>Food</i> | 53 |
| 9 | <i>Foreign</i> | 19 |
| 10 | <i>Hockey</i> | 1 |
| 11 | <i>Hotel</i> | 1 |
| 12 | <i>Internet</i> | 3 |
| 13 | <i>Kids</i> | 1 |
| 14 | <i>Language</i> | 2 |
| 15 | <i>Medicine</i> | 1 |
| 16 | <i>Memorabilia</i> | 1 |
| 17 | <i>Mexico</i> | 2 |
| 18 | <i>Music</i> | 28 |
| 19 | <i>Peace</i> | 1 |
| 20 | <i>Software</i> | 4 |
| 21 | <i>Web</i> | 1 |

Consulta "apple"

| Número de categoría | Nombre de Categoría | Número de documentos |
|---------------------|----------------------------------|----------------------|
| 1 | <i>Apple Macintosh computers</i> | 12 |
| 2 | <i>Circus</i> | 2 |
| 3 | <i>Corps</i> | 6 |
| 4 | <i>Emulator</i> | 13 |
| 5 | <i>Grower</i> | 5 |
| 6 | <i>History</i> | 11 |
| 7 | <i>Ipod</i> | 21 |
| 8 | <i>MAC OS</i> | 17 |
| 9 | <i>MacBook Pro</i> | 3 |
| 10 | <i>Music</i> | 9 |
| 11 | <i>QuickTime</i> | 6 |
| 12 | <i>Recipes</i> | 56 |
| 13 | <i>Store</i> | 11 |
| 14 | <i>Theatre</i> | 4 |
| 15 | <i>Travel</i> | 4 |

C. Evaluación de usuario

Este apéndice muestra el cuestionario de evaluación contestado por los usuarios. En las figuras C.1 a la C.5 aparecen las dos partes de la evaluación: las instrucciones y el cuestionario.

CONQUIRO *metasearch*

Evaluación de Usuario

Conquiro es un sistema que agrupa los resultados de una búsqueda en Web. En este estudio evaluarás la agrupación que hizo Conquiro de 100 mensajes tomados del newsgroup misc.forsale. En estos mensajes las personas ofrecen o solicitan algún producto para su venta o compra.

Instrucciones

Al iniciar la evaluación aparecerán dos ventanas, una que contiene la agrupación de los mensajes que realizó el sistema Conquiro y la otra contiene el formulario de evaluación.

Ventana con los resultados

Del lado izquierdo se muestran los grupos (clusters) en los que el sistema organizó los mensajes. Al dar click en la etiqueta del grupo los mensajes que éste contiene serán mostrados del lado derecho. Sólo se muestra el título y autor del mensaje, para ver el mensaje completo hay que dar click en el título.

| Clusters found | Results of cluster sales |
|--|---|
| <ul style="list-style-type: none">sales<ul style="list-style-type: none">roomtestelvisdigitalguitarvacationchemicalsoffer<ul style="list-style-type: none">applemsexcellentintelprintersupersamplerd-22cdgamesincludessaledrivechannel | <ol style="list-style-type: none">1 Need APARTMENT/ROOM in BOSTON misc.forsale - by David Wilson /conquiro/msg98.html2 GRE & GRE Economics Test Books for SALE misc.forsale - by Keith Frederick /conquiro/msg87.html3 1956 Elvis autograph misc.forsale - by g perry /conquiro/msg78.html4 Casio Digital Diary misc.forsale - by Brent Kirkwood /conquiro/msg58.html5 Boss Guitar Pedal misc.forsale - by Harry Powell Watson /conquiro/msg92.html |

Figura C.1 Instrucciones de la evaluación (parte 1).

Ventana con el formulario


En este deberás de capturar los datos de tu evaluación.

CONQUIRO *metasearch*

Formulario de evaluación

1. Nombre:

2. ¿Cuál es tu experiencia en realizar búsquedas en Web ?



3. ¿Qué buscadores que organicen la información en grupos (clusters) has utilizado ?
(Si no has utilizado ninguno deja la caja en blanco.)

Comenzar

Figura C.2 Instrucciones de evaluación (parte 2).

Formulario de evaluación

1. Nombre:

2. ¿Cuál es tu experiencia en realizar búsquedas en Web ?

▼

3. ¿Qué buscadores que organicen la información en grupos (clusters) has utilizado ?
(Si no has utilizado ninguno deja la caja en blanco.)

4. Para cada uno de los grupos identifica lo siguientes puntos:

a) Califica si los mensajes del grupo son similares entre si de acuerdo con la siguiente escala:

Todos = Todos los mensajes del grupo son similares entre si.

No todos = No todos los mensajes del grupo son similares entre si.

Ninguno = Ninguno de los mensajes del grupo son similares entre si.

Definición: Un mensaje es similar a otro si ambos tratan o están relacionados con el mismo tema.

b) Califica que tan bien la etiqueta del grupo te dio idea del contenido de los mensajes de acuerdo con la siguiente escala:

Figura C.3 Cuestionario de evaluación (parte 1).

(mucho, poco, nada).

Sugerencia : para contestar este punto se recomienda leer los mensajes e identificar que artículo están vendiendo o comprando.

NOTA: Hay grupos cuyas etiquetas tienen el caracter '>' es decir, grupo1 > grupo2, esto lo que indica es que el grupo1 tiene como subgrupo al grupo2.

| Número de grupo | Etiqueta del grupo | Número de mensajes que tratan el mismo tema | Considero que la etiqueta representa el contenido de los mensajes |
|-----------------|--------------------|---|--|
| 1 | room | <input checked="" type="radio"/> todos <input type="radio"/> no todos <input type="radio"/> ninguno | <input checked="" type="radio"/> mucho <input type="radio"/> medio <input type="radio"/> poco <input type="radio"/> nada |
| 2 | test | <input checked="" type="radio"/> todos <input type="radio"/> no todos <input type="radio"/> ninguno | <input checked="" type="radio"/> mucho <input type="radio"/> medio <input type="radio"/> poco <input type="radio"/> nada |
| 3 | elvis | <input checked="" type="radio"/> todos <input type="radio"/> no todos <input type="radio"/> ninguno | <input checked="" type="radio"/> mucho <input type="radio"/> medio <input type="radio"/> poco <input type="radio"/> nada |
| 4 | digital | <input checked="" type="radio"/> todos <input type="radio"/> no todos <input type="radio"/> ninguno | <input checked="" type="radio"/> mucho <input type="radio"/> medio <input type="radio"/> poco <input type="radio"/> nada |
| 5 | guitar | <input checked="" type="radio"/> todos <input type="radio"/> no todos <input type="radio"/> ninguno | <input checked="" type="radio"/> mucho <input type="radio"/> medio <input type="radio"/> poco <input type="radio"/> nada |
| 6 | vacation | <input checked="" type="radio"/> todos <input type="radio"/> no todos <input type="radio"/> ninguno | <input checked="" type="radio"/> mucho <input type="radio"/> medio <input type="radio"/> poco <input type="radio"/> nada |
| 7 | chemicals | <input checked="" type="radio"/> todos <input type="radio"/> no todos <input type="radio"/> ninguno | <input checked="" type="radio"/> mucho <input type="radio"/> medio <input type="radio"/> poco <input type="radio"/> nada |

Figura C.4 Cuestionario de evaluación (parte 2).

| | | | | | | | | |
|----|-------------------|--|--------------------------------|-------------------------------|--|-----------------------------|----------------------------|----------------------------|
| 65 | usa > dec | <input checked="" type="radio"/> todos | <input type="radio"/> no todos | <input type="radio"/> ninguno | <input checked="" type="radio"/> mucho | <input type="radio"/> medio | <input type="radio"/> poco | <input type="radio"/> nada |
| 66 | usa > heavy | <input checked="" type="radio"/> todos | <input type="radio"/> no todos | <input type="radio"/> ninguno | <input checked="" type="radio"/> mucho | <input type="radio"/> medio | <input type="radio"/> poco | <input type="radio"/> nada |
| 67 | traded | <input checked="" type="radio"/> todos | <input type="radio"/> no todos | <input type="radio"/> ninguno | <input checked="" type="radio"/> mucho | <input type="radio"/> medio | <input type="radio"/> poco | <input type="radio"/> nada |
| 68 | traded > genesis | <input checked="" type="radio"/> todos | <input type="radio"/> no todos | <input type="radio"/> ninguno | <input checked="" type="radio"/> mucho | <input type="radio"/> medio | <input type="radio"/> poco | <input type="radio"/> nada |
| 69 | traded > baseball | <input checked="" type="radio"/> todos | <input type="radio"/> no todos | <input type="radio"/> ninguno | <input checked="" type="radio"/> mucho | <input type="radio"/> medio | <input type="radio"/> poco | <input type="radio"/> nada |
| 70 | talking | <input checked="" type="radio"/> todos | <input type="radio"/> no todos | <input type="radio"/> ninguno | <input checked="" type="radio"/> mucho | <input type="radio"/> medio | <input type="radio"/> poco | <input type="radio"/> nada |
| 71 | fancy | <input checked="" type="radio"/> todos | <input type="radio"/> no todos | <input type="radio"/> ninguno | <input checked="" type="radio"/> mucho | <input type="radio"/> medio | <input type="radio"/> poco | <input type="radio"/> nada |

5. De acuerdo a los mensajes que leiste, escribe los subtemas que puedes identificar en éstos.
Sugerencia: Algunas de las etiquetas que el sistema asignó a los grupos pueden ayudarte.

Fin de la Evaluación.

Enviar

Figura C.5 Cuestionario de evaluación (parte 3).

Bibliografía

Anton L. y Croft W. B.

1996 An Evaluation of Techniques for Clustering Search Results. Technical Report IR-76, Department of Computer Science, University of Massachusetts, Amherst.

Ask

2006 <http://www.ask.com/> (Mayo de 2006)

Baeza-Yates R. y Ribeiro-Neto B. A.

1999 Modern Information Retrieval. Addison-Wesley, first edition.

Berry M. W., Drmac Z. y Jessup R. R.

1999 Matrices, Vector Spaces, and Information Retrieval. SIAM Review, 41(2):335-362.

Buckland M. y Gey F.

1994 The Relationship between Recall and Precision. Journal of American Society for Information Science, 45(1):12-19.

Calishain T. y Dornfest R.

2004 Google. Los mejores trucos. Anaya Multimedia, primera edición.

Can F. y Ozkarahan E. A.

1990 Concepts and Effectiveness of the Cover-Coefficient-Based Clustering Methodology for Text Databases. ACM Transactions on Databases Systems, 14(5):483-517.

Carrot2

2006 <http://carrot.cs.put.poznan.pl/carrot2-remote-controller/index.jsp> (Mayo de 2006)

Clusty

2006 <http://clusty.com/> (Mayo de 2006)

Cohn D., Caruana R. y McCallum A.

2003 Semi-supervised Clustering with User Feedback. TR2003-1892, Cornell University.

Crabtree D.

2004 Improvements to Web Page Clustering Methods. BSc thesis, Victoria University of Wellington.

Cutting D. R., Karper D. R., Pedersen J. O. y Turkey J. W.

1992 Scatter/Gather: A Cluster-based Approach to Browsing Large Document Collections. Proceedings of the 15th Annual International ACM/SIGIR Conference, Copenhagen.

Dhillon I. S. y Modha D. S.

2002 Concept Decompositions for Large Sparse Text Data using Clustering. Machine Learning, 42(1):143-175.

Dhillon I. S., Fan J. y Guan Y.

2001 Efficient Clustering of Very Large Document Collections. Data Mining for Scientific and Engineering Applications, pp. 357-381.

Dhillon I. S., Guan Y. y Kogan J.
2002 Iterative Clustering of High Dimensional Text Data Augmented by Local Search. Proceedings of The Second IEEE International Conference on Data Mining, pp. 131-38.

dmoz
2006 <http://dmoz.org/> (Mayo de 2006)

Dogpile
2006 <http://www.dogpile.com> (Mayo de 2006)

Dubes R. C. y Jain A. K.
1988 Algorithms for Clustering Data. Prentice Hall.

Efthimiadis E. N.
1996 Query Expansion. Annual Review of Information Science and Technology, 31:121-187.

Ferragina P. y Gulli A.
2005 A Personalized Search Engine Based on Web-Snippet Hierarchical Clustering. Proceedings of WWW14, pp. 801-810.

Frakes W. y Baeza-Yates R.
1992 Information Retrieval: Data Structures and Algorithms. Prentice Hall.

Google
2006 <http://www.google.com> (Mayo de 2006)

Gulli A. y Signorini A.
2005 The Indexable Web is More than 11.5 billion pages. Poster Proceedings of the 14th International Conference on World Wide Web, pp. 902-903, Chiba, Japan.

Halkidi M., Batistakis Y. y Vazirgiannis M.
2001 On clustering Validation Techniques. Journal of Intelligent Information Systems, 17(2-3):107-145.

Hearst M. A. y Pedersen J. O.
1996 Reexamining the Cluster Hypothesis: Scatter/Gather on Retrieval Results. Proceedings of the 19th Annual International ACM/SIGIR Conference, Zurich.

Hu W. y Chen Y.
2001 An Overview of World Wide Web Search Technologies. Proceedings of 5th World Multi-conference on System, Cybernetics and Informatics (SCI2001).

Ixquick
2006 <http://www.ixquick.com> (Mayo de 2006)

Jain A. y Dubes R.
1988 Algorithms for Clustering Data. Prentice Hall, Englewood Cliffs, NJ.

Jain A. K., Murty M. N. y Flynn P. J.
1999 Data clustering: a review. ACM Computing Surveys, 31(3):264-323.

Jansen B. J. y Spink A.
2005 How are We Searching the World Wide Web? A Comparison of Nine Search Engine Transaction Logs. Information Processing and Management, 42(1):248-263.

Jizba R, ©
2000 Creighton University. Measuring Search Effectiveness. <http://www.hsl.creighton.edu/hsl/Searching/Recall-Precision.html> (página consultada en Mayo del 2006)

- Klink S., Hust A., Junker M. y Dengel A.
2002 Improving Document Retrieval by Automatic Query Expansion Using Collaborative Learning of Term-Based Concepts. In Proceedings of the 5th International Workshop on Document Analysis Systems, 23:376-387.
- Lang N. C.
2003 A tolerance rough set approach to clustering web search results. Master thesis, Warsaw University.
- LookSmart
2006 <http://search.looksmart.com/> (Mayo de 2006)
- Maarek Y. S., Fagin R., Ben-Shaul I. Z. y Pelleg D.
2000 Ephemeral Document Clustering for Web Applications. Technical Report RJ 10186, IBM Research.
- Mamma
2006 <http://www.mamma.com> (Mayo de 2006)
- Mandl T.
2005 Recent Developments in the Evaluation of Information Retrieval Systems: Moving toward Diversity and Practical Applications. Information Science, Universität Hildesheim (*mimeo*).
- MSN
2006 <http://search.msn.com/> (Mayo de 2006)
- Netskills, Quality Internet Training.
Search Engines and Other Animals. University of Newcastle. <http://www.netskills.ac.uk/> (Mayo de 2006)
- Pirolli P., Schank P., Hearst M. y Diehl C.
1996 Scatter/Gather Browsing Communicates the Topic Structure of a Very Large Text Collection. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 213-220.
- Popescul A. y Ungar H.
2000 Automatic Labeling of Document Clusters (*mimeo*).
- Porter M. F.
1980 An Algorithm for Suffix Stripping. Program, 14:130-137.
- Radev D. R., Jing H. y Budzikowska M.
2000 Centroid-Based Summarization of Multiple Documents: Sentence Extraction, Utility-Based Evaluation, and User Studies. ANLP/NAACL Workshop on Summarization ANLP/NACC.
- Rijsbergen C. J. van.
1979 Information Retrieval. Butterworths, London, second edition.
- Salton G.
1989 Automatic Text Processing: the Transformation, Analysis and Retrieval of Information by computer. Addison-Wesley Publishing.
- Sanderson M. y Croft B.
1999 Deriving Concept Hierarchies from Text. In Proceedings of the 22nd ACM SIGIR Conference, pp. 206-213.
- Search Engine Showdown
2006 <http://www.searchengineshowdown.com/dir/> (Mayo de 2006)

- Shyu M., Chen S., Chen M. y Rubin S. H.
2004 Affinity-Based Similarity Measure for Web Document Clustering. Proceedings of the 2004 IEEE International Conference on Information Reuse and Integration, November 8-10 pp. 247-252, Las Vegas, Nevada USA.
- Steinbach M., Karypis G. y Kumar V.
2000 A Comparison of Document Clustering Techniques. In Text Mining Workshop (ACM KDD'00).
- Strehl A., Ghosh J. y Mooney R.
2000 Impact of Similarity Measures on Web-Page Clustering. In AAAI Workshop on AI for Web Search, pp. 58-64.
- Tan P., Steinbach M. y Kumar V.
2005 Introduction to Data Mining. Addison-Wesley.
- Tonella P., Ricca F., Pianta E., Girardi C., Di Lucca G., Fasolino A. R. y Tramontana P.
2003a Evaluation Methods for Web Application Clustering. WSE, pp. 33, 5th International Workshop on Web Site Evolution.
- Tonella P., Ricca F., Pianta E., y Girardi C.
2003b Using Keyword Extraction for Web Site Clustering. WSE, pp. 41-48, 5th International Workshop on Web Site Evolution.
- Treeratpituk P. y Callan J.
2006 Automatically Labeling Hierarchical Clusters. Proceedings of the Sixth National Conference on Digital Government Research, pp. 167-176. San Diego, Ca.
- Ukkonen E.
1995 On-line Construction of Suffix Tree. Algorithmica, 14(3):249-260.
- Vivísimo
2006 <http://vivisimo.com/> (Mayo de 2006)
- Wang Y.
2005 Incorporating Semantic and Syntactic Information into Document Representation for Document Clustering. PhD Thesis, Mississippi State University.
- Weiss D.
2001 A Clustering Interface for Web Search Results in Polish and English. Master Thesis, Poznan University of Technology.
- Widdows D.
2004 Geometry and Meaning. Stanford, California:CSLI publications.
- Wroblewski M.
2003 A Hierarchical WWW Pages Clustering Algorithm based on the Vector Space Model. Master Thesis, Poznan University of Technology.
- Wu M. y Sonnenwald D. H.
1999 Reflections on Information Retrieval Evaluation, Proceedings of the 1999 EBTI, ECAI, SEER & PNC Joint Meeting. Academia Sinica, Taipei, pp. 63-81.
- Yahoo
2006 <http://www.yahoo.com> (Mayo de 2006)

Yang Y. y Pedersen J. O.

1997 A Comparative Study on Feature Selection in Text Categorization. In Proceedings of ICML-97, 14th International Conference on Machine Learning, pp. 412-420.

Zamir O.

1999 Clustering Web Documents: A Phrase-Based Method for Grouping Search Engine Results. Ph.D. Thesis, University of Washington.

Zamir O. y Etzioni O.

1998 Web Clustering: A Feasibility Demonstration. Proceedings of the 19th International ACM SIGIR Conference on Research and Development of Information Retrieval, pp. 46-54.

1999 Grouper: A Dynamic Clustering Interface to Web Search Results. Proceedings of WWW8, Toronto, Canada.

Zeng H. J., He Q. C., Chen Z., Ma W. y Ma J.

2004 Learning to Cluster Web Search Results. Proceedings of SIGIR'04, pp. 210-217.

Zhang D. y Dong Y.

2004 Semantic, Hierarchical, Online Clustering of Web Search Results. APWEB 2004, pp. 69-78.