TESIS DE MAGISTER

EN INGENIERIA DE SOFTWARE

CATEGORIZACION AUTOMATICA DE DOCUMENTOS CON MAPAS AUTO-ORGANIZADOS DE KOHONEN

Autor: Lic. Daniel Goldenberg

Directores de Tesis:

M. Ing. Hernán Merlino

M. Ing. Enrique Fernández

2007

Dedicado

A mi esposa e hija, quienes supieron entender mi dedicación a este proyecto, apoyándome en todo momento.

Resumen

La categorización de documentos es la tarea de separar documentos en grupos.

En una época como la actual en que la obtención de información adecuada y en tiempos mínimos se hace indispensable en cualquier área de negocios, es importante mejorar la técnica de categorización de modo que los resultados obtenidos al buscar documentos sean satisfactorios tanto por la calidad del mismo como por el tiempo de respuesta.

En el presente trabajo de tesis se presenta un buscador de documentos de texto por palabra clave que devuelve los resultados categorizados, utilizando para esto un mapa auto-organizado de Kohonen.

Abstract

Document clustering is the task of separate documents in groups.

In a time like the actual time when it is indispensable to obtain the adequate information in minimal time in every business area, it is important to improve the technique of clustering in order to obtain satisfactory results taking into account the quality and the response time, when searching documents.

In the following thesis, a tool for searching text documents that return clustered results by using Kohonen self-organized maps is presented.

Agradecimientos

A mis tutores a lo largo de mi magíster: M. Ing Edgardo Claverie y M. Ing. Bibiana Rossi, por todo el apoyo y el aliento que me brindaron.

Al equipo docente, por la flexibilidad que mostró en momentos particulares de mi vida para poder proseguir con el magíster.

A mis tutores en la tesis, el Dr. Ramón García Martínez y la M. Ing. Paola Britos, por su dedicación, comentarios y apoyo desde el mismo momento de la elección del tema hasta la finalización de la misma.

A todos los colaboradores desinteresados en el desarrollo de la presente tesis: Esp. Ing. Hernán Merlino, Esp. Ing. Enrique Fernández y M. Ing. Carlos Rivero Bianchi y su equipo.

ÍNDICE

CAPÍTULO 1

INTRODUCCIÓN

La *Categorización de Documentos* (Document Clustering) puede definirse como la tarea de separar documentos en grupos. El criterio de agrupamiento se basa en las similitudes existentes entre ellos [Kaufmann *et al.*, 1990]. En la bibliografía consultada, los términos clasificación ("classification"), y categorización o agrupamiento ("categorization" ó "clustering"), son utilizados con distinto significado [Lewis, 1991; Yang, 1997; Maarek *et. al.*, 2000; Clerking *et. al.*, 2001]. El concepto de *clasificación* de documentos refiere al problema de encontrar para cada documento la clase a la que pertenece, asumiendo que las clases están predefinidas y que se tienen documentos preclasificados para utilizar como ejemplos. En la presente tesis, se estudia la *categorización* o *agrupamiento* de documentos, entendiéndose por esto el proceso de encontrar grupos dentro de una colección de documentos basándose en las similitudes existentes entre ellos, sin un conocimiento *a priori* de sus características.

Un criterio de agrupamiento utilizado es dividir los documentos en una jerarquía de temas. Un ejemplo de esto último son las categorías que presenta el buscador Yahoo® [Rüger *et. al.*, 2000]. Un documento en particular se podría encontrar, por ejemplo, dentro de "Tecnología → Informática → Internet → Buscadores". El problema que presenta esta técnica es la dificultad de encontrar la categoría que mejor describa a un documento [Hearst et. al., 1996]. Los documentos no tratan un sólo tema, y aunque lo hicieran, el enfoque con el que el tema es tratado puede hacer que el documento encuadre en otra categoría. Esto hace que la categorización de documentos sea una tarea compleja y subjetiva, ya que dos personas podrían asignar el mismo documento a categorías diferentes, cada una aplicando un criterio válido [Macskassy *et.al.*, 2001].

Las siguientes secciones de este capítulo dan una introducción al contenido, objetivo y estructura de esta tesis. La sección 1.1 presenta el problema de la categorización automática de documentos, describiendo su utilidad y las causas que han despertado el interés por resolverlo. En la sección 1.2 se muestra la imposibilidad de encontrar la solución al problema de categorización automática mediante una búsqueda exhaustiva, lo que ha llevado a la creación de algoritmos que encuentran soluciones aproximadas al problema. La sección 1.3 describe en forma breve qué son los mapas auto-organizados de Kohonen, y por qué pueden aplicarse al problema en estudio. La sección 1.4 presenta el objetivo de la tesis, y la sección 1.5, la forma en la que se estructura su contenido en los capítulos que la componen.

1.1 CATEGORIZACIÓN AUTOMÁTICA DE DOCUMENTOS

La tarea de encontrar grupos de documentos con características comunes no sólo es compleja sino que además consume tiempo. Un bibliotecario que tratara de clasificar un documento tendría que leerlo para comprender su significado para luego asignarle una categoría utilizando su experiencia y sentido común. El costo y el tiempo asociados a la categorización de documentos ha llevado a la investigación de técnicas que permitan automatizar la tarea [Rüger *et. al.*, 2000]. Debido al incremento en los volúmenes de información disponibles en forma electrónica y a la necesidad cada vez

mayor de encontrar la información buscada en un tiempo mínimo, éstas técnicas han estado recibiendo creciente atención [Hearst et al, 1996; Zamir et al, 1999; Strehl *et al.*, 2000; Maarek *et al.*, 2000].

1.1.1 La Hipótesis del Agrupamiento

La categorización automática de documentos se comenzó a investigar dentro de la rama de "Recuperación de Información" (en inglés , "Information Retrieval"), que es la rama de la informática que investiga la búsqueda eficiente de información relativa a un tema en particular en grandes volúmenes de documentación [ISO, 1993]. En su forma más simple, las consultas están dirigidas a determinar qué documentos poseen determinadas palabras en su contenido. Por ejemplo, la consulta "buscador internet" podría tener por resultado todos los documentos que contengan las palabras "buscador" e "internet" como parte de su contenido. El objetivo de las técnicas de recupero de información es poder resolver consultas en forma eficaz y eficiente. La eficiencia viene dada por la rapidez con la cual se resuelve la consulta. El criterio para evaluar la eficacia se basa en la relevancia de los resultados encontrados [Raghavan *et. al.*, 1989]; en el ejemplo del párrafo anterior, una búsqueda eficaz debería retornar todos los documentos que trataran sobre la búsqueda de información en internet, aún si la palabra "buscador" no estuviera en el contenido de los mismos.

En el contexto del recupero de información, Van Rijsbergen [Van Rijsbergen, 1979] formula la denominada "Hipótesis del Agrupamiento" (en inglés, "Cluster Hypothesis"); básicamente, la hipótesis del agrupamiento sostiene que "los documentos fuertemente asociados tienden a ser relevantes para la misma consulta", lo cual ha sido verificado experimentalmente [Cutting *et. al.*, 1992; Schütze *et. al.*,1997; Zamir *et. al.*, 1999]; basándose en dicha hipótesis, la categorización automática tiene en cuenta el contenido de los documentos para agruparlos, ya que documentos similares contendrán palabras (términos) similares.

1.1.2 Aplicaciones

La categorización automática de documentos se investiga dentro del campo del recupero de información como una herramienta capaz de mejorar la calidad de las soluciones ofrecidas. Sus aplicaciones más importantes son:

Mejorar el rendimiento de los motores de búsqueda de información mediante la categorización previa de todos los documentos disponibles [Van Rijsbergen, 1979; Dunlop et al, 1991; Faloutsos et al, 1995; Zamir *et. al.*, 1999; Rüger *et. al.*, 2000]. Antes de comenzar a resolver las consultas, el conjunto de documentos es separado en grupos. A cada grupo de documentos se le asigna un "representante de grupo"; luego, al resolver una consulta, no se examinan todos los documentos, sino que se busca el "representante de grupo" que mejor responda a la consulta. Por la hipótesis del agrupamiento, el grupo encontrado estará compuesto de los documentos más relevantes para esa búsqueda.

Facilitar la revisión de resultados por parte del usuario final, agrupando los resultados luego de realizar la búsqueda [Croft, 1978; Cutting *et al.*, 1992; Allen *et al.*, 1993; Leousky et al, 1996; Maarek *et al.*, 2000]. Cuando se realizan búsquedas sobre un gran volumen de documentos, la cantidad de resultados puede ser muy grande. Si la lista se presenta sin ningún tipo de procesamiento previo, el usuario del sistema se verá obligado a revisar todos los documentos descartando aquellos que no son relevantes para él. Si los resultados se agrupan, la hipótesis del agrupamiento indica que los documentos del mismo grupo serán relevantes para una subconsulta más

específica que la original. De esta forma, los documentos quedarán separados en grupos temáticos (los grupos son una división de los documentos originales, y cada grupo responde a una subconsulta más específica dentro del tema buscado) [Rüger *et. al.*, 2000]. Así, con sólo revisar uno o dos documentos de cada grupo, el usuario podrá determinar cuál es el subtema al que responden todos los documentos del grupo, pudiendo descartar rápidamente los documentos irrelevantes para su búsqueda. La presente tesis tiene en cuenta esta aplicación en particular.

1.2 NATURALEZA COMBINATORIA DEL PROBLEMA DE CATEGORIZACIÓN

Una de las características principales del problema de la categorización de documentos es su naturaleza combinatoria [Duran et al, 1974; Pentakalos et al., 1996]. La teoría combinatoria [Liu, 1968] indica que S(n,K), la cantidad de maneras de agrupar *n* objetos en *K* grupos, está dada por: $S(n,K) = \frac{1}{K!}\sum_{i=0}^{K}(-1)^i \binom{K}{i}(K-i)^n$.

Utilizando esta fórmula puede calcularse, por ejemplo: S (25,5) = 2 x 10^{16}, o, dicho en otras palabras, que hay más de dos mil millones de millones de formas de agrupar 25 objetos en 5 grupos. Los 25 objetos del ejemplo, representan un problema de dimensiones demasiado pequeñas para ser de utilidad. En la práctica, se querría aplicar la categorización automática, por lo menos, a varios cientos de documentos. Sin embargo, haciendo algunos cálculos sobre este ejemplo de dimensiones pequeñas, puede comprenderse la complejidad del problema. Supóngase un algoritmo que intente probar todas las posibles categorizaciones para encontrar la mejor de ellas, asignándoles puntajes de acuerdo a un criterio predefinido. Suponiendo que el algoritmo consiga calcular el puntaje de 100 mil soluciones por segundo (el algoritmo tendría que ser muy rápido, pero supóngase que lo es), para probar las 2 x 10^{16} posibilidades, el algoritmo tomaría más de 6 mil años en dar el resultado.

Un algoritmo que evalúe cada una de las posibles categorizaciones no es aplicable en forma práctica a la categorización automática de documentos, por lo que se han desarrollado algoritmos que hacen uso de heurísticas [Cutting *et al.*, 1992; Zamir *et. al.*, 1999; Jain *et.al.*, 1999; Estivill-Castro, 2000; Zhao *et.al.*, 2001] para explorar una parte de estas posibles categorizaciones buscando una solución de calidad aceptable.

1.3 MAPAS AUTO-ORGANIZADOS DE KOHONEN

Los mapas auto-organizados de Kohonen [Kohonen, 1982; Kohonen, 1995; Lagus, 2001] constituyen un método de proyección no lineal que mapea un espacio de datos multidimensional en un mapa usualmente bidimensional en forma ordenada. Los nodos del mapa son asociados a los llamados vectores de referencia que actúan como modelos locales de los datos más cercanos y, más ampliamente, de las regiones vecinas del mapa. Gracias a las propiedades del algoritmo de los mapas auto-organizados de Kohonen, posiciones cercanas del mapa contienen datos similares, permitiendo una visualización intuitiva del espacio de datos. Más aún, los vectores de referencia dividen los datos en subconjuntos de datos similares, efectuando una categorización de los datos.

1.4 OBJETIVO DE LA TESIS

En este contexto, el objetivo de esta tesis es estudiar cómo pueden aplicarse los mapas auto-organizados de Kohonen al problema de encontrar en forma automática la mejor categorización de documentos resultantes de una búsqueda por palabra clave. Se estudiará de qué forma las características de los mapas auto-organizados de Kohonen pueden utilizarse para diseñar un algoritmo que supere el rendimiento de los algoritmos que actualmente se aplican al problema, y se evaluarán los resultados de acuerdo a la calidad de las soluciones obtenidas.

1.5 ESTRUCTURA DE LA TESIS

La tesis se divide en seis capítulos y un apéndice.

- El capítulo 2 describe el estado actual de los campos de estudio relacionados con esta tesis. La sección 2.1 define formalmente el problema de la categorización automática de documentos. La categorización automática de documentos se vincula con las ciencias de la “Recuperación de Información” y de la “Minería de Datos”. Del campo de la Recuperación de Información toma los conceptos y métodos utilizados para el procesamiento de documentos. Estas técnicas son las que permiten transformar la información no estructurada que contienen los documentos (llamados “documentos de texto libre”, o en inglés “free text documents”), a estructuras de datos manejables mediante algoritmos computacionales. Las secciones 2.2 y 2.3 describen estas técnicas. Del campo de la Minería de Datos toma las técnicas que se ocupan de los problemas de categorización de objetos. Los algoritmos utilizados para la categorización automática de documentos son adaptaciones de los que se utilizan para el problema más general de categorizar objetos de cualquier tipo. La sección 2.4 muestra la ubicación de la categorización automática de documentos dentro de este área, y detalla los algoritmos utilizados actualmente para categorizar documentos en forma automática. En las secciones 2.6 y 2.7 se describen dos algoritmos presentados recientemente, que superan a los métodos clásicos, y contra los cuales se comparará la solución propuesta en esta tesis. Las secciones 2.8 y 2.9 presentan los principios básicos de los mapas auto-organizados de Kohonen y de la búsqueda de documentos de texto por palabra clave, que conforman la base del trabajo presentado en esta tesis.
- El capítulo 3 plantea las cuestiones que esta tesis apunta a responder, y analiza el trabajo previo que existe acerca de la aplicación de los mapas auto-organizados de Kohonen a la categorización automática de documentos. En la sección 3.1 se exponen las limitaciones que presentan los algoritmos actuales, relacionados con la forma en la que exploran el espacio de posibles soluciones. La sección 3.2 describe los trabajos previos que aplicaron los mapas auto-organizados de Kohonen a la categorización automática.

- En el capítulo 4 se presenta la solución propuesta en esta tesis, que apunta a responder a las cuestiones planteadas en el capítulo 3. En la sección 4.1 se describe la Planificación de Sistemas de Información de la metodología utilizada: Métrica V3. En la sección 4.2 se describe el Desarrollo de Sistemas de Información de Métrica V3. En la sección 4.3 se describe la Gestión de Configuración de Métrica V3.

- El capítulo 5 describe la comparación de los resultados de las pruebas que se realizaron para evaluar la efectividad de la solución propuesta en el capítulo 4.

La sección 5.1 describe los conjuntos de datos utilizados. Los mismos fueron extraídos de una colección reconocida como un estándar dentro de la comunidad de investigadores dedicados a la categorización automática de documentos. En las secciones 5.2 y 5.3 se detalla la metodología seguida en la experimentación. Se enumeran las variables que intervienen en los experimentos y los distintos tipos de experimentos realizados. La sección 5.4 presenta los resultados de la experimentación. La presentación se hace en forma de gráficos y para cada variable se incluye además el resultado del test estadístico realizado sobre los resultados. En la sección 5.5 se analizan brevemente los resultados, que luego se tratan con más detalle en el capítulo 6.

- El capítulo 6 presenta las conclusiones extraídas a partir de la investigación realizada y los resultados experimentales obtenidos. Los resultados experimentales obtenidos confirman la tesis que los mapas auto-organizados de Kohonen son una poderosa herramienta para la resolución de problemas de categorización de documentos de texto.

- El apéndice 1 detalla el análisis estadístico realizado sobre los resultados que se exponen en el capítulo 5, y que soportan las afirmaciones realizadas en la sección 5.4 (“Resultados”) de ese capítulo.

CAPÍTULO 2

ESTADO DEL ARTE

Este capítulo describe el estado actual de los campos de estudio relacionados con esta tesis. La sección 2.1 define formalmente el problema de la categorización automática de documentos.

La categorización automática de documentos se vincula con las ciencias de la "Recuperación de Información" y de la "Minería de Datos". Del campo de la Recuperación de Información toma los conceptos y métodos utilizados para el procesamiento de documentos. Estas técnicas son las que permiten transformar la información no estructurada que contienen los documentos (llamados "documentos de texto libre", o en inglés "free text documents"), a estructuras de datos manejables mediante algoritmos computacionales. Las secciones 2.2 y 2.3 describen estas técnicas.

La minería de datos es una tarea que apunta a identificar patrones en los datos que sean válidos, nuevos, potencialmente útiles y entendibles (Fayyad et al., 1995). Del campo de la Minería de Datos toma las técnicas que se ocupan de los problemas de categorización de objetos. Los algoritmos utilizados para la categorización automática de documentos son adaptaciones de los que se utilizan para el problema más general de categorizar objetos de cualquier tipo. Las secciones 2.4 y 2.5 muestran la ubicación de la categorización automática de documentos dentro de esta área, y detallan los algoritmos utilizados actualmente para categorizar documentos en forma automática.

En las secciones 2.6 y 2.7 se describen dos algoritmos presentados recientemente, que superan a los métodos clásicos, y contra los cuales se comparará la solución propuesta en esta tesis. Las secciones 2.8 y 2.9 presentan los principios básicos de los mapas auto-organizados de Kohonen y de la búsqueda de documentos de texto por palabra clave, que conforman la base del trabajo presentado en esta tesis.

2.1 CATEGORIZACIÓN DE OBJETOS

La categorización de documentos es un tipo de problema perteneciente a la familia de problemas asociados a encontrar agrupamientos entre objetos de cualquier tipo. Si bien la categorización de documentos tiene características particulares que surgen de las propiedades de los documentos como objetos a agrupar, los principios generales coinciden con los que se aplican para categorizar cualquier otro tipo de elementos. Los algoritmos para la categorización automática de documentos son los mismos que se utilizan para agrupar otros tipos de objetos, o adaptaciones de éstos [Qin He, 1996; Cole, 1998; Fasulo, 1999; Jain *et.al.*, 1999].

Una definición del problema del agrupamiento de documentos (que es aplicable al agrupamiento de cualquier tipo de elementos), puede enunciarse de la siguiente manera [Zhao *et.al.*, 2001]:

Dado un conjunto S, de N documentos, se quiere encontrar la partición $S_1, S_2, ..., S_k$, tal que cada uno de los N documentos se encuentre sólo en un grupo S_i, y que cada documento sea más similar a los documentos de su mismo grupo que a los documentos asignados a los otros grupos.

2.2 REPRESENTACIÓN VECTORIAL

Para poder definir medidas de semejanza entre los objetos a agrupar, éstos se representan mediante vectores $v = (a_1 , a_2 , ... , a_m)$, donde cada componente del vector es el valor de un atributo del objeto. De esta forma, cada uno de los objetos a agrupar es un punto en un Espacio Euclideano de *m* dimensiones, R^m. Los investigadores dedicados a las ramas de la Recuperación de Información adoptaron tempranamente una representación vectorial para el manejo de documentos [Van Rijsbergen, 1979]; en este modelo, cada documento es considerado un vector *d* en el espacio de términos (el conjunto de palabras que aparecen en por lo menos uno de los documentos de la colección).

Una forma aceptada [Van Rijsbergen, 1979; Faloutsos *et.al.*, 1995; Jones *et.al.*, 1995] de representar los documentos es asignar a cada atributo del vector la presencia o ausencia del término correspondiente. De esta manera, para *m* términos el vector consistiría en *m* atributos binarios que pueden tomar valores de cero o uno, según los términos que aparezcan o no en el documento. Otra manera es representar a cada documento por su vector de frecuencia de términos [Steinbach *et.al.*, 2000]: $d_{tf} = (tf_1 , tf_{2 , ... , } tf_m)$, donde tf_i es la frecuencia del término número *i* en el documento (cantidad de veces que aparece esa palabra en el documento). De esta forma, se considera que los documentos quedan caracterizados por la cantidad de veces que aparece cada término dentro del mismo.

Un refinamiento de este modelo consiste en multiplicar a la frecuencia de cada término por su "frecuencia documental inversa" (en inglés, "inverse document frequency"). Ésta técnica, denominada "tf-idf", se basa en la premisa que si un término aparece en gran parte de los documentos, es poco discriminante, y por lo tanto, debe restársele importancia [Faloutsos *et.al.*, 1995; Zhao *et.al.*, 2001]. Así, la representación del documento resultaría: $d_{tf\text{-}idf} = (tf_1 \log(N / df_1) , tf_2 \log(N / df_2) , ... , tf_m \log(N / df_m))$, donde *N* es la cantidad total de documentos de la colección y df_i es la cantidad de documentos que contienen al término *i*.

La normalización de los vectores (|| *d* || = 1) asegura que los documentos se evalúen por la composición de su contenido, sin tener en cuenta su tamaño, ya que en los documentos más extensos, las frecuencias de aparición de los términos serán mayores.

2.2.1 Definición del "centroide" de un grupo

La definición de "centroide" de un grupo de elementos representados vectorialmente es utilizada en el contexto de los algoritmos de agrupamiento. Dado un grupo de elementos *S*, que contiene *h* elementos s_i, se define a su centroide C_s como el promedio de los vectores que componen el grupo:

$$C_s = \frac{\sum_{i=1}^{h} s_i}{h}$$ (Fórmula 2.1)

Cada componente del vector centroide es el promedio del valor de esa componente para los miembros del grupo [Steinbach *et.al.*, 2000; Zhao *et.al.*, 2001]; su propiedad

más importante es que la distancia promedio desde un punto cualquiera del espacio hasta cada elemento del grupo es igual a la distancia entre ese punto y el centroide del grupo.

2.2.2 Reducción de la dimensionalidad del espacio de términos

Es evidente que en una colección de documentos aparecerán cientos o miles de palabras distintas. Como la dimensión del espacio vectorial de representación está dada por la cantidad de palabras diferentes, cada vector contendrá cientos o miles de componentes (gran parte de ellas con valor igual a cero) [Zervas *et al.*, 2000]. Esto es conocido como la "maldición de la dimensionalidad" (en inglés "the curse of dimensionality") [Yang *et al.*, 1997]. Hay dos técnicas que se utilizan para reducir la dimensionalidad del espacio de términos: la reducción de palabras a su raíz, y la remoción de los términos poco discriminantes.

2.2.2.1 Reducción de palabras a su raíz

Para reducir el número de términos distintos con los que se trabaja, el primer paso es reducir todas las palabras a su raíz (en inglés "steeming"). Por ejemplo, las palabras "medicina", "médico" y "medicinal" se reducen a la forma "medic". Esta técnica es llamada "remoción de sufijos" (en inglés "suffix stripping"). Si bien es posible que de esta manera se lleven a la misma raíz palabras que en realidad tienen significados distintos, en general las palabras que tienen la misma raíz refieren al mismo concepto, y la pérdida de precisión es compensada por la reducción de la dimensionalidad del espacio [Van Rijsbergen, 1979].

Las aplicaciones de agrupamiento de documentos usan técnicas de remoción de sufijos [Krovetz, 1993; Zamir *et.al.*, 1998; Steinbach *et.al.*, 2000], existiendo consenso en cuanto a la conveniencia de su aplicación. La técnica de remoción de sufijos más utilizada es la llamada "regla de Porter" [Porter, 1980]; esta técnica se basa en un algoritmo que aplica una serie de reglas que determinan si las últimas sílabas de una palabra deben ser removidas.

2.2.2.2 Remoción de términos poco discriminantes

Otro método para reducir la dimensión del espacio de términos (que es complementario del anterior), es el de descartar los términos que se consideran poco discriminantes [Yang *et al.*, 1997]. Un término es poco discriminante si el hecho de saber que ese término se encuentra en un documento nos dice poco (o nada), acerca del posible contenido o tema del documento. El ejemplo más simple de dichos términos son las preposiciones, artículos y toda otra palabra auxiliar que ayude a dar forma a las oraciones. Éstas palabras se consideran poco discriminantes en cualquier colección documental con la cual se trabaje, por lo que se utiliza una lista de palabras irrelevantes (en inglés, "stop list") y se descartan todas las palabras que aparecen en esa lista [Van Rijsbergen, 1979; Strehl *et.al.*, 2000].

Existen términos que pueden ser discriminantes dependiendo del conjunto de documentos con los que se trabaje. Para una colección de documentos determinada, entre sus términos habrá algunos más útiles que otros para la tarea de categorización. Por ejemplo, si en una serie de documentos sobre física cuántica, el término "electrón" aparece en el 98% de los documentos, es evidente que la presencia o ausencia de ese término no es relevante a los efectos de agrupar los mismos. De la misma forma,

si dentro de ese grupo de documentos, la palabra “sol” aparece solamente en uno de ellos, ese término tampoco será importante para agruparlos.

En [Yang *et al.*, 1997] se exponen diversas técnicas orientadas a detectar, en una colección de documentos, qué términos son irrelevantes para la tarea de categorización. Algunas de ellas requieren un pequeño grupo de documentos previamente categorizados, para ser utilizados como un conjunto de entrenamiento, por lo que a veces su aplicación puede no ser posible. A continuación se describe brevemente cada una de las técnicas analizadas.

a) Umbral de frecuencia documental

Es el método más sencillo. Se calculan las frecuencias con que aparece cada término en la colección documental y los términos que no superan cierto umbral mínimo son descartados. Se asume que las palabras muy poco frecuentes no contienen información sobre la categoría a la que pertenece el documento, o que son términos “ruidosos” (que pueden llegar a confundir al algoritmo de agrupamiento).

b) Ganancia de información, Información mutua, Estadística X^2

Teniendo un conjunto de documentos ya agrupados, para cada término se calcula (en base a fórmulas probabilísticas) qué tan bueno es para predecir la categoría a la que pertenece el documento. El puntaje más alto es obtenido por aquellos términos presentes en todos los documentos de una categoría y que no aparecen en ningún documento de las demás categorías. Los términos con puntaje más bajo (que son aquellos que aparecen por igual en documentos de todas las categorías) son descartados. Los tres métodos difieren en la forma de calcular los puntajes y normalizar los resultados.

c) Fuerza del término

Usando un grupo de documentos de entrenamiento, se toman aquellos pares de documentos cuya semejanza excede un determinado valor (parámetro del método). Para cada término, se cuenta la cantidad de veces que el mismo aparece en ambos documentos de cada par. En base a eso, se calcula la probabilidad de que el término esté en el segundo documento del par, sabiendo que está en el primero. Se asignan puntajes más altos cuanto mayor sea esa probabilidad, asumiendo que el término es descriptivo de la categoría de esos documentos.

El análisis comparativo [Yang *et al.*, 1997], usando cada uno de los métodos de remoción de términos para diferentes algoritmos de agrupamiento, llega a la conclusión de que las técnicas de Umbral de frecuencia, Ganancia de información y Estadística X^2 obtienen resultados similares, notando que el método de Umbral de frecuencia no requiere documentos previamente clasificados, lo que amplía el rango de situaciones en las que puede ser aplicado.

2.3 MEDIDAS DE SEMEJANZA

En la definición del problema de agrupamiento, se dijo: “...que cada documento sea más similar a los documentos de su mismo grupo, que a los documentos asignados a los otros grupos...” (los términos “similar”, “semejante” y “cercano” se utilizan indistintamente para referirse a éste concepto). Para poder evaluar esta condición, es

necesario definir una medida cuantitativa de la similitud existente entre dos documentos.

En el modelo de representación más sencillo, en el que los atributos del vector son valores binarios, y definiendo $|v|$ como la cantidad de atributos de *v* que toman el valor 1, las medidas más comunes son:

- Coeficiente de Jaccard

$$\frac{|d_1 \cap d_2|}{|d_1 \cup d_2|}$$ (Fórmula 2.2)

- Coeficiente del coseno

$$\frac{|d_1 \cap d_2|}{\sqrt{|d_1|} * \sqrt{|d_2|}}$$ (Fórmula 2.3)

Ambas medidas definen el concepto de semejanza de los documentos por la cantidad de términos en común que contienen en relación al tamaño de los documentos.

En el modelo de representación más utilizado [Qin He, 1996; Maarek *et.al.*, 2000; Steinbach *et.al.*, 2000], en el cual se calculan los vectores de frecuencia o de frecuencia inversa, y siendo $||v||$ la longitud (norma) del vector *v*, las medidas más comunes son [Cole, 1998; Strehl *et.al.*, 2000; Zhao *et.al.*, 2001]:

Coeficiente del coseno extendido

$$\cos(d_1, d_2) = \frac{d_1 \bullet d_2}{\|d_1\| * \|d_2\|}$$ (Fórmula 2.4)

Es una extensión del correspondiente del modelo binario para el caso de atributos con valores reales. Esta medida tiene la propiedad de no depender del tamaño de los documentos, ya que $\mathbf{\cos(d_1,d_2) = \cos(a.d_1,d_2)}$ para a > 0. Sin embargo, los documentos se normalizan para que tengan longitud unitaria, ya que entonces,

$$\cos(d_1, d_2) = d_1 \bullet d_2$$ (Fórmula 2.5)

Y la semejanza entre los documentos se puede calcular como el producto vectorial entre ellos.

La similitud queda comprendida en el intervalo [0,1]. Para un documento cualquiera, el vector que lo representa es un punto en el espacio. Si se traza la recta que definen ese punto y el eje de coordenadas, todo documento que se encuentre sobre la recta tiene similitud 1 con el documento que la define. Si se trazan hiperconos concéntricos

cuyo eje sea esa recta, la semejanza irá decreciendo a medida que se agranda el ángulo del hipercono, y todos los documentos situados en la pared de cada hipercono tienen la misma similitud con el documento que define el eje. La semejanza igual a cero se alcanza cuando el hipercono se convierte en el hiperplano perpendicular al eje que define el documento [Strehl *et.al.*, 2000].

- Coeficiente de Jaccard extendido

$$jac(d_1, d_2) = \frac{d_1 \bullet d_2}{\|d_1\|^2 + \|d_2\|^2 - (d_1 \bullet d_2)}$$ (Fórmula 2.6)

Es una extensión del coeficiente de Jaccard del modelo binario para el caso de atributos con valores reales. Los valores posibles de similitud se encuentran en el rango [0,1]. Esta medida tiene propiedades intermedias entre el coeficiente del coseno extendido y la distancia Euclideana, que se detalla a continuación.

- Distancia Euclideana

$$dist\_euc(d_1, d_2) = \|d_1 - d_2\|$$ (Fórmula 2.7)

Es la fórmula tradicional para calcular el tamaño del segmento que une dos puntos. La semejanza de dos documentos queda definida en forma inversa, ya que los documentos más similares serán los que estén a menor distancia. La fórmula comúnmente utilizada es:

$$euc(d_1, d_2) = \frac{1}{1 - \|d_1 - d_2\|}$$ (Fórmula 2.8)

Los posibles valores de similitud están en el rango [0,1], pero un documento tiene semejanza igual a 1 sólo consigo mismo. Para un documento cualquiera, el vector que lo representa es un punto en el espacio. Si se trazan hiperesferas concéntricas alrededor del punto, todos los documentos ubicados en la superficie de una hiperesfera tienen el mismo valor de similitud con el documento que define el centro. La semejanza decrece a medida que aumenta el radio de las hiperesferas.

En el campo de la categorización automática de documentos, la medida más utilizada es el coeficiente del coseno extendido [Cutting *et.al.*, 1992; Qin He, 1996; Steinbach *et.al.*, 2000; Zhao *et.al.*, 2001]. Se ha realizado [Strehl *et.al.*, 2000] un análisis completo de las medidas expuestas anteriormente, comparando el rendimiento de distintos algoritmos de agrupamiento utilizando cada una de las medidas de similitud, llegando a la conclusión de que los coeficientes del coseno y Jaccard extendidos son más apropiados que la distancia euclideana para espacios de gran dimensionalidad y con datos dispersos, como es el caso del agrupamiento de documentos.

2.4 MÉTODOS PARA CATEGORIZAR DOCUMENTOS

Las formas de clasificación de objetos, tales como asignar clases predeterminadas a cada elemento ó agruparlos en forma significativa, son susceptibles de dividirse según el esquema de la figura 2.1 [Qin He, 1996]:

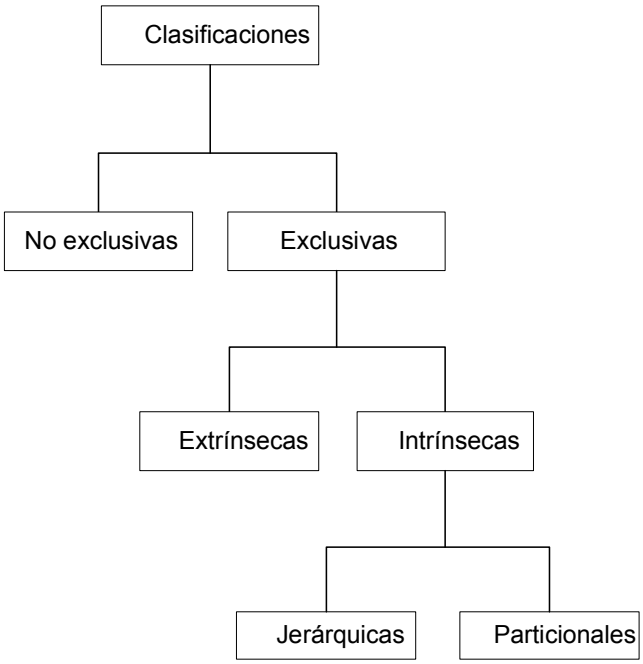


Figura 2.1. División de las formas de clasificar objetos

- **No exclusivas**: Un mismo objeto puede pertenecer a varias categorías, clases o grupos.

- **Exclusivas**: Cada objeto pertenece solamente a una categoría, clase o grupo.

- **Extrínsecas (supervisadas)**: Las clases a las que pertenecen los objetos están predefinidas, y se conocen ejemplos de cada una, ó algunos de los objetos ya están clasificados y son utilizados por el algoritmo para aprender a clasificar a los demás.

- **Intrínsecas (no supervisadas)**: La clasificación se realiza en base a las características propias de los objetos, sin conocimiento previo sobre las clases a las que pertenecen.

- **Jerárquicas**: Los métodos jerárquicos consiguen la categorización final mediante la separación (métodos divisivos) o la unión (métodos aglomerativos) de grupos de documentos. Así, estos métodos generan una estructura en forma de árbol en la que cada nivel representa una posible categorización de los documentos [Willet, 1998]

- **Particionales (no jerárquicas)**: Los métodos no jerárquicos, también llamados particionales, o de optimización llegan a una única categorización que optimiza un criterio predefinido o función objetivo, sin producir una serie de grupos anidados [Everitt, 1993].

La categorización automática de documentos se encuentra en la categoría “intrínseca”, ya que los criterios de agrupamiento se basan en la información contenida en los mismos para determinar sus similitudes

2.5 MÉTODOS DE CATEGORIZACIÓN INTRÍNSECOS

2.5.1 Métodos Jerárquicos

Los algoritmos jerárquicos se caracterizan por generar una estructura de árbol (llamada “dendograma”), en la que cada nivel es un agrupamiento posible de los objetos de la colección [Jain *et.al.*, 1999]. Cada vértice (nodo) del árbol es un grupo de elementos. La raíz del árbol (primer nivel) se compone de un sólo grupo que contiene todos los elementos. Cada hoja del último nivel del árbol es un grupo compuesto por un sólo elemento (hay tantas hojas como objetos tenga la colección). En los niveles intermedios, cada nodo del nivel *n* es dividido para formar sus hijos del nivel *n* + 1.

Las figuras 2.2 y 2.3 ilustran estos conceptos mediante un ejemplo simple.

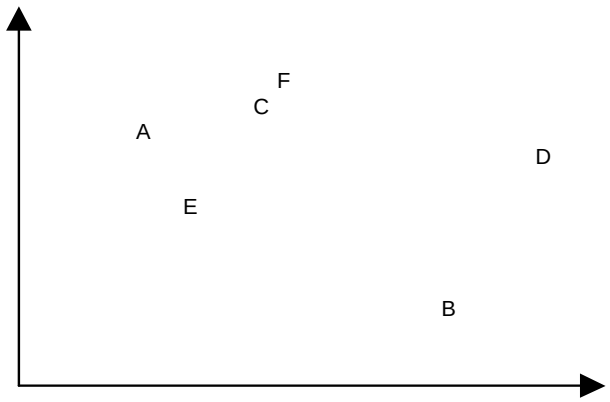


Figura 2.2. Posible colección de objetos en un espacio de 2 dimensiones

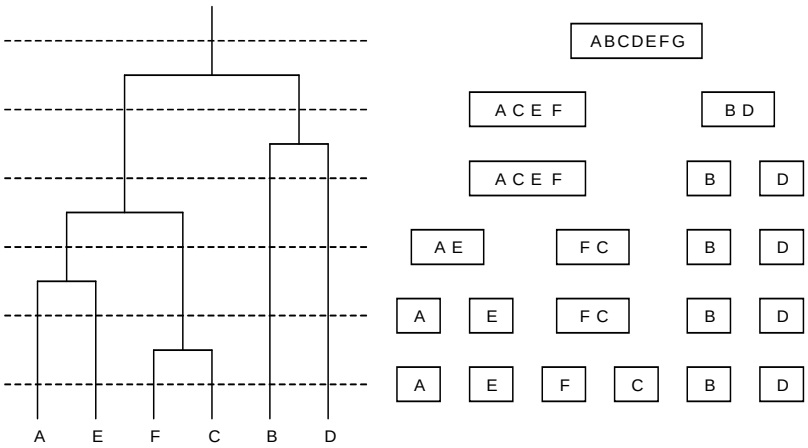


Figura 2.3. Posible dendograma para la colección de la figura 2.2 y agrupamientos que representa cada nivel.

Los algoritmos de agrupamiento jerárquicos fueron uno de los primeros enfoques para los problemas de categorización de documentos, y todavía se siguen utilizando debido a la forma simple e intuitiva en la que trabajan [Dash *et.al.*, 2001]. De acuerdo a la metodología que aplican para obtener el dendograma, los algoritmos jerárquicos pueden dividirse en *aglomerativos* y *divisivos* [Han et al., 2001].

Los métodos aglomerativos parten de las hojas del árbol, ubicando a cada elemento en su propio grupo, y en cada paso buscan los dos grupos más cercanos para juntarlos. Los divisivos, por su parte, hacen el camino inverso. Comenzando en la raíz, en cada paso seleccionan un grupo para dividirlo en dos, buscando que el agrupamiento resultante sea el mejor de acuerdo a un criterio predeterminado. El análisis necesario para pasar de un nivel a otro (decidir qué grupo dividir o cuales juntar) es más sencillo para los métodos aglomerativos [Dash *et.al.*, 2001], y esto hace que éstos sean más utilizados que los divisivos [Fasulo, 1999; Steinbach *et.al.*, 2000]. En adelante, cuando se hable de métodos jerárquicos, se hará referencia únicamente a los algoritmos de Agrupamiento Jerárquico Aglomerativo (HAC, su sigla en inglés).

Las distintas variantes de algoritmos jerárquicos aglomerativos difieren únicamente en la manera de determinar la semejanza entre los grupos al seleccionar los dos grupos más cercanos [Qin He, 1996; Willet, 1998; Cole, 1998; Jain *et.al.*, 1999; Fasulo 1999]. Debe notarse la diferencia entre medidas de semejanza entre *documentos* y medidas de semejanza entre *grupos*. La similitud de dos grupos se calcula en base a los valores de semejanza existentes entre sus documentos, pero la forma de hacer este cálculo no es única. Dada una medida de semejanza entre *documentos*, que puede considerarse la misma para todos, los distintos algoritmos jerárquicos aglomerativos se distinguen por la medida de semejanza entre *grupos* que utiliza cada uno.

La figura 2.4 presenta un espacio bidimensional en el cual se han colocado 4 grupos de objetos. Los objetos de un mismo grupo se han representado mediante el mismo símbolo (x, *, # ó +).

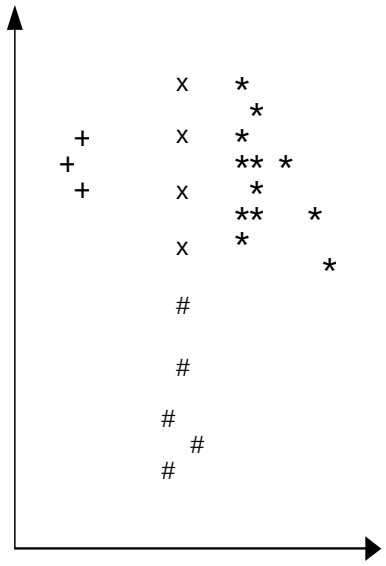


Figura 2.4. Cuatro grupos de objetos en un espacio de dos dimensiones

Las figuras 2.5, 2.6 y 2.7 muestran cómo los métodos pueden diferir entre ellos al seleccionar cuáles son los grupos más semejantes. En las figuras se utiliza la distancia Euclideana como medida de semejanza entre los documentos. La disposición presentada de los grupos se ha elegido especialmente para provocar las diferencias, si los grupos están suficientemente separados y son compactos todos los métodos coinciden en la selección de los grupos más cercanos.

2.5.1.1 Enlace simple ("single link")

El método de enlace simple, también llamado "del vecino cercano" (en inglés "nearest neighbour"), calcula la semejanza entre dos grupos como la semejanza entre los dos elementos más cercanos de ambos (ver figura 2.5). Este método es eficaz cuando los grupos tienen formas irregulares, pero es muy sensible a la existencia de elementos dispersos que no forman parte de ningún grupo definido, llevando a la creación de grupos alargados, compuestos de objetos disímiles [Cole, 1998; Karypis *et.al.*, 1999]. Éste efecto recibe el nombre de "encadenamiento".

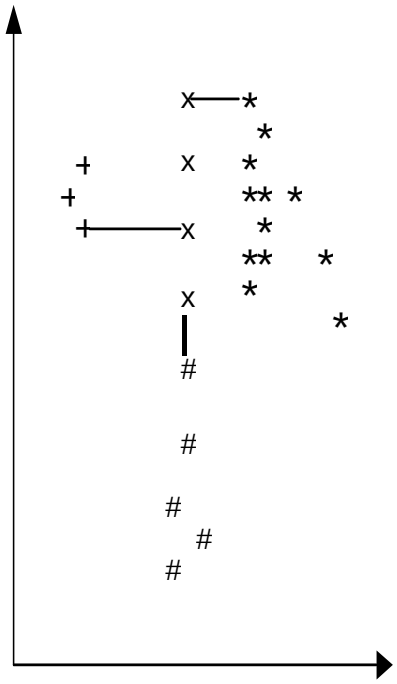


Figura 2.5. Distancias al grupo de las " x " según el método de enlace simple. La distancia entre los grupos más cercanos está remarcada.

2.5.1.2 Enlace completo ("complete link")

En el extremo opuesto del método de enlace simple se encuentra el método de enlace completo, que calcula la semejanza entre dos grupos usando la semejanza de los dos elementos más lejanos (ver figura 2.6). De esta manera, el método no sufre del efecto de "encadenamiento", y encuentra con eficacia grupos pequeños y compactos. Sin embargo, cuando los grupos no están bien definidos, puede llevar a la creación de grupos sin significado.

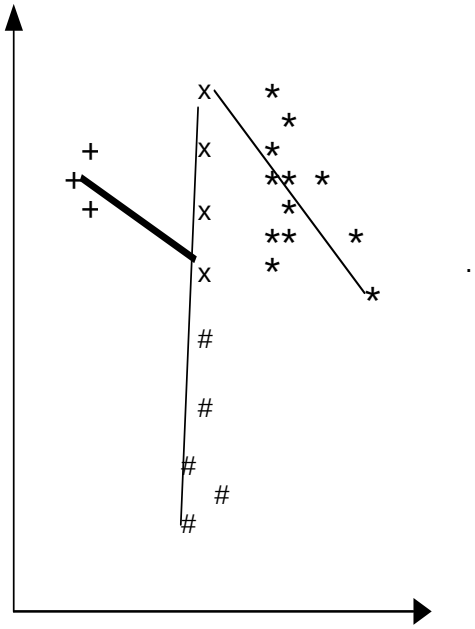


Figura 2.6. Distancias al grupo de " x " según el método de enlace completo. La distancia entre los grupos más cercanos está remarcada

2.5.1.3 Enlace promedio ("average link")

A mitad de camino entre los dos métodos anteriores, el algoritmo de enlace promedio define a la semejanza entre dos grupos como el promedio de las semejanzas de cada miembro de uno con cada miembro del otro (ver figura 2.7). Al tomar propiedades de los métodos de enlace simple y completo, éste algoritmo obtiene resultados aceptables en un rango de situaciones más amplio [Cole, 1998].

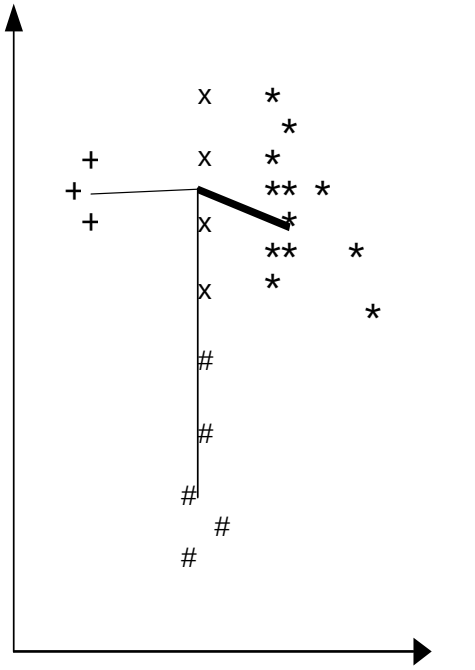


Figura 2.7. Distancias al grupo de " x " según el método de enlace promedio. La distancia entre los grupos más cercanos está remarcada.

2.5.1.4 Método de Ward

Este método define la "suma de errores cuadrados" como la suma del cuadrado de la distancia de cada objeto al centroide de su grupo. Así, para un grupo de elementos *S*, compuesto por *h* elementos s_i y cuyo centroide es C_s:

$$W_s = \sum_{i=1}^{h} \| s_i - C_s \|^2 \quad \text{(Fórmula 2.9)}$$

Y, para un agrupamiento de *k* grupos:

$$W = \sum_{i=1}^{k} W_i \quad \text{(Fórmula 2.10)}$$

Al comenzar el algoritmo, la suma de errores cuadrados vale cero, ya que cada elemento forma su propio grupo y coincide con el centroide. En cada paso, el método evalúa cada una de las posibles uniones y elige aquella que produce el menor incremento del error. Este método tiende a producir grupos de tamaños iguales, y su rendimiento es comparable al del método de enlace promedio [Cole, 1998].

2.5.1.5 Resumen de características

Entre las características de éstos métodos pueden destacarse las siguientes:

a) Su forma de trabajo es simple e intuitiva: El enfoque utilizado por estos métodos es semejante al que utilizaría una persona para realizar la tarea del agrupamiento, especialmente los aglomerativos (comenzar juntando los documentos más similares entre si, y luego buscar similitudes entre los grupos).

b) Su resultado es una serie de agrupamientos anidados: Esto facilita la revisión de los resultados por parte del usuario, que puede recorrer la estructura de árbol para ver agrupamientos con diferentes niveles de detalle [Maarek *et.al.*, 2000].

c) Son deterministas: Al aplicar dos veces un algoritmo jerárquico a una colección de documentos, las dos veces seguirá el mismo camino hacia la solución. Hay algunos agrupamientos que el algoritmo nunca considerará, sin importar la cantidad de veces que se lo ejecute [Steinbach *et.al.*,2000].

d) No revisan las decisiones que toman en los pasos anteriores: Una vez que dos documentos se han asignado al mismo grupo (o se han colocado en distintos grupos, en los divisivos), ningún paso posterior los volverá a separar (o juntar), por lo que una mala asignación en los primeros pasos no puede corregirse [Cole, 1998].

e) Requieren grandes tiempos de cómputo: La forma de buscar en cada paso los grupos a unir (o dividir, en los divisivos), hacen que las implementaciones conocidas de estos algoritmos tengan tiempos de ejecución del orden de n^2 (enlace simple) ó n^3 (enlace completo) [Zamir *et.al.*, 1998].

2.5.2 Métodos particionales

Los métodos de optimización o particionales, a diferencia de los jerárquicos, no van generando distintos niveles de agrupamiento de los objetos, sino que trabajan en un sólo nivel, en el que se refina (optimiza), un agrupamiento [Everitt, 1993]. Si bien los distintos niveles de agrupamiento generados por los algoritmos jerárquicos son más apropiados para la presentación de los resultados al usuario, las técnicas de optimización se están comenzando a utilizar con más frecuencia en aplicaciones de categorización automática de documentos debido a requieren considerablemente menos recursos [Zhao *et.al.*, 2001]. Estos métodos asumen que el valor de *k* (la cantidad de grupos), está definida de antemano [Qin He, 1996].

La estructura general de éstos métodos se compone de los siguientes pasos [Han *et.al.*, 2001]:

- Seleccionar *k* puntos representantes (cada punto representa un grupo de la solución).
- Asignar cada elemento al grupo del representante más cercano, de forma de optimizar un determinado criterio.
- Actualizar los *k* puntos representantes de acuerdo a la composición de cada grupo.
- Volver al punto 2)

Este ciclo se repite hasta que no sea posible mejorar el criterio de optimización.

2.5.2.1 Selección inicial de los representantes

El método más frecuentemente utilizado para obtener los *k* puntos representantes iniciales es eligiéndolos al azar [Bradley et al., 1998]. Esta técnica es la más rápida y simple, pero también la más riesgosa, ya que los puntos elegidos pueden ser una mala representación de la colección de objetos. Cuando se utiliza esta técnica, se ejecuta varias veces el algoritmo de agrupamiento para distintas selecciones aleatorias, tomando el mejor resultado y descartando el resto [Steinbach *et.al.*, 2000].

El resto de las técnicas para la selección inicial utilizan algún algoritmo de agrupamiento (generalmente jerárquico) para obtener la selección inicial [Bradley et al., 1998]. Lógicamente, no se realiza un agrupamiento de todos los objetos de la colección ya que el objetivo no es llegar a la solución mediante un algoritmo jerárquico, sino solamente obtener los puntos iniciales para luego usar un algoritmo de optimización (más rápido); las técnicas “Buckshot” y “Fractionation” [Cutting *et.al.*, 1992] son un ejemplo de esto último.

2.5.2.2 Criterios de optimización

Los algoritmos particionales buscan optimizar el valor de un criterio de optimización. Estos criterios deben ser funciones que den una medida cuantitativa de la calidad de un agrupamiento. En [Zhao *et.al.*, 2001] se analizan los criterios de optimización más frecuentemente utilizados en la categorización automática de documentos, que se detallan a continuación.

Criterios internos

Los criterios internos dan una medida de la cohesión interna de los grupos. Para cada grupo se calcula un valor en base a los objetos que lo componen (sin tener en cuenta elementos de otros grupos), y luego se suman los valores de cohesión de cada uno.

a) Maximización de la suma de similitudes promedio: Para cada grupo se calcula el promedio de las similitudes que existen entre cada par de documentos que lo componen. Por ejemplo, para el grupo *S*, que tiene n_s elementos:

$$sim\_prom_s = \frac{1}{n_s^2} \sum_{\substack{d \in s \\ d' \in s}} sim(d, d') \quad \text{(Fórmula 2.11)}$$

El valor total para el criterio se obtiene sumando las similitudes promedio de cada grupo multiplicadas por su cantidad de elementos.

$$sim\_prom = \sum_{i=1}^{k} n_i * sim\_prom_i \quad \text{(Fórmula 2.12)}$$

Este criterio toma valores más altos cuando los elementos de cada grupo son más similares entre sí.

b) Maximización de la suma de las similitudes con el centroide: Para cada grupo se calcula la suma de las similitudes que existen entre cada elemento y el centroide. Por ejemplo, para el grupo *S*:

$$sim\_cent_s = \sum_{d \in s} sim(d, c_s) \quad \text{(Fórmula 2.13)}$$

El valor total para el criterio se obtiene sumando las similitudes promedio de cada grupo.

$$sim\_cent = \sum_{i=1}^{k} sim\_cent_i \quad \text{(Fórmula 2.14)}$$

Los valores más altos se alcanzan cuando cada objeto se encuentra cerca del centro de su grupo.

c) Minimización de la suma de errores cuadrados: Este criterio es el mismo que utiliza el método de Ward, que se analiza en la sección 2.5.1, dedicada a los métodos jerárquicos.

Criterios externos

Los criterios externos tienen en cuenta la separación que existe entre los distintos grupos. Se considera que un agrupamiento es mejor que otro cuando sus grupos están más separados del centro de la colección.

a) Minimización de la similitud de los centroides con centroide de la colección: Este criterio calcula la similitud existente entre el centroide de cada grupo y el centro de la colección, y luego suma los valores multiplicados por el tamaño de cada grupo.

$$ext\_sim = \sum_{i=1}^{k} sim(C_i, C)$$ (Fórmula 2.15)

b) Maximización de la distancia de los centroides al centroide de la colección: En lugar de minimizar las similitudes, este criterio intenta maximizar las distancias.

$$ext\_dist = \sum_{i=1}^{k} \|C_i - C\|$$ (Fórmula 2.16)

Evaluación de los criterios

En [Zhao *et.al.*, 2001] se evalúa cada uno de los criterios detallados, aplicados al agrupamiento de colecciones de documentos. El criterio interno de maximización de la suma de similitudes con el centroide (que es el que se utiliza más comúnmente en la bibliografía), obtiene los mejores resultados al aplicar cada uno de los criterios en forma individual. El trabajo propone una combinación de éste criterio con el criterio externo de minimización de la similitud de los centroides con el centro de la colección, que mejora el rendimiento del algoritmo de agrupamiento, produciendo grupos de tamaños más balanceados.

2.5.2.3 Algoritmos de optimización

Existen variantes de algoritmos de optimización en la literatura [Rasmussen, 1992; Qin He, 1996; Jain *et.al.*, 1999; Han *et.al.*, 2001] que implementan la estructura básica de cuatro pasos descripta anteriormente. Estos algoritmos son similares entre sí, por lo que se describirá únicamente el algoritmo “k-means”, que, además de ser el referente más típico en la bibliografía, es el que más frecuentemente se encuentra aplicado al campo de categorización automática de documentos [Steinbach *et.al.*, 2000].

Algoritmo “K-means”

Este algoritmo, presentado originalmente por [McQueen, 1967], utiliza a los centroides de cada grupo como sus puntos representantes. Partiendo de una selección inicial de *k* centroides (que pueden ser *k* elementos de la colección seleccionados al azar, o los que se obtengan mediante la aplicación de alguna técnica de inicialización), cada uno de los elementos de la colección se asigna al grupo con el centroide más cercano.

A continuación, se calcula el centroide de cada uno de los grupos resultantes. En los primeros pasos se obtienen las mayores diferencias entre los centroides originales y los calculados luego de las reasignaciones. Los puntos de la colección vuelven a asignarse al grupo del centroide más cercano, y estos pasos se repiten hasta que los *k*

centroides no cambian luego de una iteración (esto es equivalente a decir que el valor de la función utilizada como criterio de optimización no varía). El algoritmo “k-means” encuentra una categorización que representa un óptimo local del criterio elegido [Bradley *et.al.*, 1998].

Las figuras 2.8, 2.9, 2.10 y 2.11 ilustran la forma de trabajo del algoritmo. En ellas puede verse cómo una iteración del algoritmo refina el agrupamiento. Los objetos de la colección están representados mediante signos “+”, y los centroides de cada grupo con los signos “**X**”. En la figura 2.8 se muestran los objetos de la colección y los centroides que el algoritmo ha encontrado hasta el paso N. En la figura 2.9, cada objeto de la colección se ha asignado al grupo con el centroide más cercano. Los nuevos centroides, calculados a partir de la composición de los grupos, se grafican en la figura 2.10. En la figura 2.11 puede verse la situación inicial para el paso N+1. En éste paso, el algoritmo encontrará los 3 grupos claramente definidos que existen en la colección. La disposición de los objetos se ha elegido especialmente para que la mejora en el agrupamiento sea evidente.

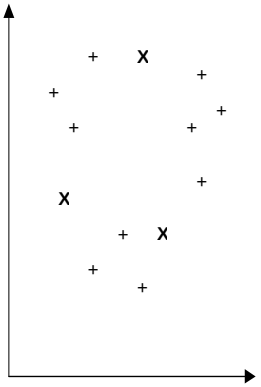


Figura 2.8. Objetos en una colección y los tres centroides del paso N

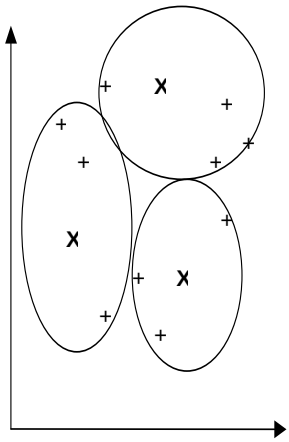


Figura 2.9. Los objetos de la colección se asignan al grupo del centroide más cercano

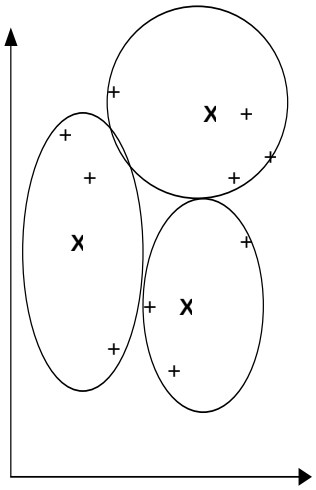


Figura 2.10. Se calculan los centroides para el paso N + 1

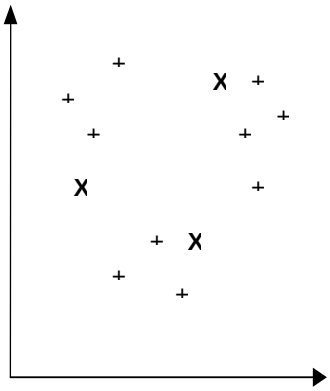


Figura 2.11. Situación inicial para el paso N+1

2.5.2.4 Resumen de características

Entre las características de éstos métodos pueden destacarse las siguientes:

a) Pueden ser no deterministas: Partiendo del mismo agrupamiento inicial, los métodos llegarán siempre a la misma solución. Sin embargo, los métodos para la selección inicial son no deterministas. El algoritmo evaluará diferentes agrupamientos cada vez que se lo ejecute, y (si los grupos no están claramente separados) podrá llegar a soluciones distintas [Steinbach *et.al.*, 2000].

b) Pueden corregir errores cometidos en pasos anteriores: En cada paso del algoritmo los objetos de la colección se asignan al grupo más apropiado según el criterio de optimización. De esta manera, el algoritmo va refinando el agrupamiento en cada iteración [Qin He, 1996].

c) Pueden implementarse en forma eficiente: Las restricciones de recursos son la causa principal por la que se utilizan este tipo de métodos. Estos algoritmos pueden implementarse de forma que sus tiempos de ejecución sean del orden de *n* [Han *et.al.*, 2001].

2.6 MAPAS AUTO-ORGANIZADOS DE KOHONEN

Existen evidencias que demuestran que hay neuronas en el cerebro que se organizan en muchas zonas de forma tal que las informaciones captadas del entrono a través de los órganos sensoriales se representan internamente en forma de mapas bidimensionales [Hilera González et al, 1995].

Aunque en gran medida esta organización neuronal está predeterminada genéticamente, es probable que parte de ella se origine mediante el aprendizaje. Esto sugiere que el cerebro podría poseer la capacidad de formar mapas topológicos de las informaciones recibidas del exterior. De hecho, esta teoría podría explicar su poder de operar con elementos semánticos. Algunas áreas del cerebro podrían crear y ordenar neuronas especializadas o grupos con características de alto nivel y sus combinaciones. Se trataría de construir mapas espaciales para atributos y características.

A partir de estas ideas, Teuvo Kohonen presentó en 1982 [Kohonen, 1982] un sistema con un comportamiento semejante. Se trataba de un modelo de red neuronal con capacidad para formar mapas de características de manera similar a como ocurre en el cerebro. El objetivo de Kohonen era demostrar que un estímulo externo (información de entrada) por sí solo, suponiendo una estructura propia y una descripción funcional del comportamiento de la red, era suficiente para forzar la formación de los mapas.

Este modelo tiene dos variantes: LVQ (Learning Vector Quantization) y SOM (Self-Organizing Map) o en español, mapa auto-organizado. Ambas se basan en el principio de formación de mapas topológicos para establecer características comunes entre las informaciones (vectores) de entrada a la red, pero difieren en las dimensiones de éstos. En LVQ son de una dimensión mientras que en el mapa auto-organizado son de dos dimensiones y también de tres.

2.6.1 Arquitectura

El mapa auto-organizado trata de establecer una correspondencia entre los datos de entrada y un espacio bidimensional de salida, creando mapas topológicos de dos dimensiones, de forma tal que ante datos de entrada con características comunes se deben activar neuronas situadas en zonas próximas de la capa de salida. Por esta razón, la representación habitual de esta red suele ser la mostrada en la figura 2.22, donde las M neuronas de salida se disponen en forma bidimensional para representar los mapas de características.

La interacción lateral entre las neuronas de la capa de salida sigue existiendo aunque ahora hay que entender la distancia como una zona bidimensional que existe alrededor cada neurona. Esta zona puede tomar la forma de cualquier polígono regular centrado en dicha neurona.

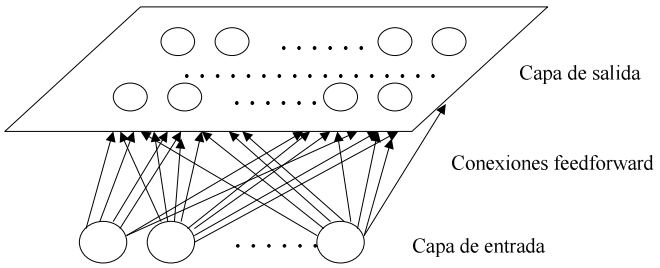


Figura 2.22 Arquitectura de mapa auto-organizado

2.6.2 Funcionamiento

El funcionamiento del mapa auto-organizado es relativamente simple. Cuando se presenta a la entrada una información $E_{\underline{k}} = \left(e_1^{(k)},...,e_N^{(k)}\right)$ cada una de las M neuronas de la capa de salida la recibe a través de las conexiones feedforward con pesos $w_{\underline{ji}}$. Estas neuronas también reciben las correspondientes entradas por medio de las conexiones laterales con el resto de las neuronas de salida y cuya influencia dependerá de la distancia a la que se encuentren.

Así, la salida generada por una neurona de salida j ante un vector de entrada $E_{\underline{k}}$ sería:

$$s_{\underline{j}}(t+1) = \sum_{i=1}^{n} w_{\underline{ji}} e_{\underline{i}}^{(k)} + \sum_{p=1}^{M} Int_{\underline{pj}} s_{\underline{p}}(t) \quad \text{(Fórmula 2.17)}$$

donde $Int_{\underline{pj}}$ es una función del tipo sombrero mexicano que representa la influencia lateral de la neurona p sobre la neurona j. La función de activación de las neuronas de salida (f) será del tipo continuo, lineal o sigmoidal, ya que el mapa auto-organizado trabaja con valores reales.

Se trata de una red de tipo competitivo ya que al presentar una entrada $E_{\underline{k}}$ la el mapa evoluciona hasta una situación estable en la que se activa una neurona de salida, la vencedora. Por ello, la formulación matemática de su funcionamiento puede simplificarse mediante la siguiente expresión que representa cuál de las M neuronas se activaría al introducir dicha información $E_{\underline{k}}$:

$$s_{\underline{j}} \begin{cases} 1 \; si \; MIN\left\| E_{\underline{k}} - w_{\underline{j}} \right\| = MIN\sqrt{\sum_{i=1}^{N}\left(e_{\underline{i}}^{(k)} - w_{\underline{ij}}\right)^2} \\ 0 \; en \; otros \; casos \end{cases} \quad \text{(Fórmula 2.18)}$$

Donde $\left\| E_{\underline{k}} - w_{\underline{j}} \right\|$ es una medida, por ejemplo la distancia euclideana, de la diferencia entre el vector de entrada $E_{\underline{k}} = \left(e_1^{(k)},...,e_N^{(k)}\right)$ y el vector de los pesos $W_{\underline{j}} = \left(w_{j1},..., w_{jN}\right)$ de las conexiones entre cada una de las neuronas de entrada y la neurona de salida j. En estos pesos se registran los datos almacenados en la red durante el proceso de aprendizaje. En la fase de funcionamiento se pretende encontrar

el dato aprendido más parecido al de entrada para averiguar qué neurona se activará y sobre todo en qué zona del espacio bidimensional de salida se encuentra.

Lo que hace el mapa auto-organizado es realizar una tarea de clasificación ya que la neurona de salida activada ante una entrada representa la clase a la que pertenece dicha información de entrada. Además, como ante otra entrada parecida se activa la misma neurona de salida u otra cercana, debido a la semejanza entre las clases, se garantiza que las neuronas topológicamente próximas sean sensibles a entradas físicamente similares. Por esta causa, el mapa auto-organizado es especialmente útil para establecer relaciones, desconocidas previamente, entre conjuntos de datos.

2.6.3 Aprendizaje

El aprendizaje es de tipo off-line, por lo que se distingue una etapa de aprendizaje y otra de funcionamiento. En la etapa de aprendizaje se fijan los valores de los pesos de las conexiones (feedforward) entre la capa de entrada y la de salida.

El aprendizaje es no supervisado y competitivo. Las neuronas de la capa de salida compiten por activarse y sólo una de ellas permanece activa ante una determinada información de entrada. Los pesos de las conexiones se ajustan en función de la neurona vencedora.

Durante la etapa de entrenamiento se presenta al mapa auto-organizado un conjunto de informaciones de entrada (vectores de entrenamiento) para que ésta establezca, en función de la semejanza entre los datos, las diferentes categorías (una por neurona de salida) que servirán durante la fase de funcionamiento para realizar clasificaciones de nuevos datos que se presenten a la red. Los valores finales de los pesos de las conexiones entre cada neurona de la capa de salida con la de entrada se corresponderán con los valores de los componentes del vector de aprendizaje que consigue activar a la neurona correspondiente. En el caso de existir más patrones de entrenamiento que neuronas de salida, más de uno deberá asociarse con la misma neurona, es decir, pertenecerán a la misma clase. En tal caso, los pesos se obtienen como un promedio de dichos patrones.

En el modelo, el aprendizaje no concluye después de presentarle una vez todos los patrones de entrada, sino que habrá que repetir el proceso varias veces para refinar el mapa topológico de salida, de tal forma que cuantas más veces se presenten los datos, tanto más se reducirán las zonas de neuronas que se deben activar ante entradas parecidas, consiguiendo que el mapa auto-organizado pueda realizar una clasificación más selectiva.

El algoritmo de aprendizaje utilizado para establecer los valores de los pesos de las conexiones entre las N neuronas de entrada y las M neuronas de salida es el siguiente:

1. Se inicializan los pesos $w_{\underline{ji}}$ con valores aleatorios pequeños y se fija la zona inicial de vecindad entre las neuronas de salida.

2. Se presenta al mapa auto-organizado una información de entrada (la que debe aprender) en forma de vector $E_{\underline{k}} = \left(e_1^{(k)},...,e_N^{(k)}\right)$, cuyas componentes $e_i^{(k)}$ serán valores continuos.

3. Puesto que se trata de un aprendizaje competitivo, se determina la neurona vencedora de la capa de salida. Esta será aquella j cuyo vector de pesos $W_{\underline{j}}$ (vector cuyas componentes son los valores de los pesos de las conexiones entre esa neurona y cada una de las neuronas de la capa de entrada) sea el más parecido a la información de entrada $E_{\underline{k}}$ (patrón o vector de entrada). Para ello, se calculan las distancias o diferencias entre ambos vectores, considerando una por una todas las neuronas de salida. Suele utilizarse la distancia euclideana o la siguiente expresión, que es similar a aquella pero eliminando la raíz cuadrada:

$$d_{\underline{j}} = \sum_{i=1}^{N} \left(e_{\underline{i}}^{(k)} - w_{\underline{ji}}\right)^2 \qquad 1 \le j \le M \quad \text{(Fórmula 2.19)}$$

Siendo:

$e_{\underline{i}}^{(k)}$: Componente i-ésimo del vector k-ésimo de entrada.

$w_{\underline{ji}}$: Peso de la conexión entre la neurona i de la capa de entrada y la neurona j de la capa de salida.

4. Una vez localizada la neurona vencedora, $\underline{j}^*$, se actualizan los pesos de las conexiones entre las neuronas de entrada y dicha neurona, así como los de las conexiones entre las neuronas de entrada y las neuronas vecinas de la vencedora. En realidad, lo que se consigue con esto es asociar la información de entrada con una cierta zona de la capa de salida.

$$w_{\underline{ji}}(t+1) = w_{\underline{ji}}(t) + \alpha(t)\left[e_{\underline{i}}^{(k)} - w_{\underline{j^*i}}(t)\right] \quad para\ j \in Zona_{i^*}(t) \quad \text{(Fórmula 2.20)}$$

$Zona_{j^*}(t)$ es la zona de vecindad alrededor de la neurona vencedora $\underline{j}^*$ en la que se encuentran las neuronas cuyos pesos son actualizados. El tamaño de esta zona se puede reducir en cada iteración del proceso de ajuste de los pesos, con lo que el conjunto de las neuronas que pueden considerarse vecinas es cada vez menor (figura 2.23). Sin embargo, en la práctica es habitual considerar una zona fija en todo el proceso de entrenamiento de la red.

El término $\alpha(t)$ es un parámetro de ganancia o coeficiente de aprendizaje, con un valor entre 0 y 1, decrece con el número de iteraciones (t) del proceso de entrenamiento. De tal forma que cuando se ha presentado un gran número de veces todo el juego de patrones de aprendizaje $(500 \le t \le 10000)$ su valor es prácticamente nulo, con lo que la modificación de pesos es insignificante. Suele utilizarse alguna de las siguientes expresiones:

$\alpha(t) = \frac{1}{t}$ (Fórmula 2.21)

$\alpha(t) = \alpha_1 - \frac{1}{\alpha_2}$ (Fórmula 2.22)

Siendo α_1 un valor de 0.1 o 0.2 y α_2 un valor próximo al número total de iteraciones del aprendizaje. Suele tomarse un valor α_2=10000.

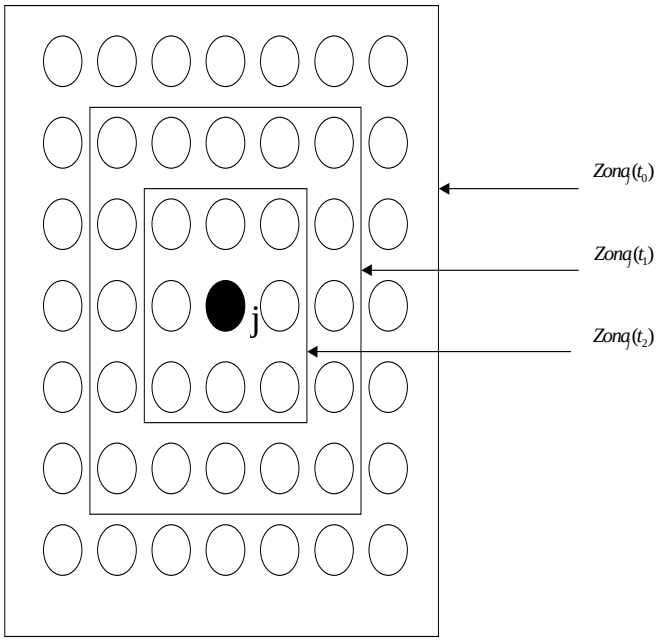


Figura 2-23. Posible evolución de la zona de vecindad

5. Se repite el proceso volviendo a presentar todo el juego de patrones de aprendizaje $\underline{E}_1, \underline{E}_2, \ldots$ un mínimo de 500 veces ($(t \geq 500)$.

Más detalles acerca de la selección de parámetros, variantes del mapa y miles de ejemplos de aplicación pueden ser encontrados en [Kohonen, 1995]. También pueden encontrarse detalles acerca del uso de los mapas auto-organizados en análisis y exploración de datos en [Kaski, 1997].

2.6.4 Aplicaciones

El modelo de Kohonen es uno de los más útiles en computación neuronal, a pesar de sus limitaciones en cuanto a la duración del proceso de aprendizaje y a la imposibilidad de aprender nuevos datos sin tener que volver a repetir completamente el proceso de aprendizaje con todos los patrones. Esta utilidad se debe a su capacidad para establecer clases o categorías de datos sin supervisión.

Se pueden destacar aplicaciones relacionadas con el reconocimiento de patrones (voz, texto, imágenes, señales, etc.), codificación de datos, compresión de imágenes, resolución de problemas de optimización, como el del Viajante, análisis de imágenes y monitoreo de procesos. [Honkela, 1997; Kohonen, 1995; Hilera González et al, 1995]

2.7 BÚSQUEDA DE DOCUMENTOS POR PALABRA CLAVE

Uno de los métodos más antiguos para buscar textos de acuerdo a una consulta es indexar todas las palabras que aparecen en la colección de documentos [Honkela, 1997]. La consulta, que es una lista de palabras clave, es convertida en una lista de índices. La lista es comparada con la lista de índices de cada documento para encontrar los documentos que contengan la lista de palabras clave. La búsqueda puede ser acotada mediante expresiones de lógica booleana. Sin embargo existen tres problemas básicos que dificultan la aplicación de la lógica booleana a la recuperación de texto [Salton, 1989]:

Las dos medidas Standard de de la efectividad y precisión de la recuperación son la precisión y el llamado (del inglés, recall). La efectividad es la relación entre el número de documentos relevantes recuperados y el número total de documentos recuperados. El llamado es la relación entre el número total de documentos recuperados y el número total de documentos relevantes conocidos. Ambas medidas son sensitivas a pequeños cambios en la formulación de la consulta. En consultas booleanas no hay una forma simple de controlar el tamaño de la salida y la salida no es clasificada por relevancia.

Considerando los resultados de la consulta no se sabe qué no fue encontrado, especialmente si la colección de documentos no es conocida.

Si el dominio de la consulta no es bien conocido, es difícil formular la consulta.

Un problema básico en el manejo de documentos de texto es que se usan diferentes palabras y frases para expresar objetos similares. Los lenguajes naturales se usan para la comunicación entre los seres humanos con variaciones de acuerdo al conocimiento, formas de expresarse, etc. Así, si la recuperación de la información se basa sólo en las palabras clave tal como éstas aparecen el texto, algunos documentos

interesantes podrían no ser descubiertos, o se podrían encontrar documentos que no están relacionados con lo que se está buscando. Los métodos tradicionales de recuperación de la información tienen posibilidades limitadas de abordar este fenómeno. Claramente, los sistemas de recuperación de información deberían encontrar beneficios si tienen en cuenta el contexto de las palabras y derivan relaciones de las palabras basándose en información contextual.

CAPÍTULO 3

DESCRIPCIÓN DEL PROBLEMA

En este capítulo se plantean las cuestiones que esta tesis apunta a responder, y se analiza el trabajo previo que existe acerca de la aplicación de los mapas auto-organizados de Kohonen a la categorización automática de documentos. En la sección 3.1 se exponen las limitaciones que presentan los algoritmos actuales, relacionados con la forma en la que exploran el espacio de posibles soluciones. La sección 3.2 describe los trabajos previos que aplicaron los mapas auto-organizados de Kohonen a la categorización automática.

3.1 LA BÚSQUEDA EN EL ESPACIO DE SOLUCIONES

Ante la imposibilidad de evaluar todas las posibles soluciones, los algoritmos de categorización automática exploran sólo una parte del espacio de búsqueda (solamente evalúan algunas de las posibles categorizaciones).

La diferencia fundamental entre los distintos algoritmos consiste en:

- La elección del punto inicial de la búsqueda.
- Las reglas para pasar de un punto del espacio de búsqueda a otro.

3.1.1 Problemas de los algoritmos actuales

3.1.1.1 Algoritmos jerárquicos aglomerativos

Si bien se acepta que los métodos jerárquicos aglomerativos obtienen resultados de buena calidad [Cutting *et.al.*, 1992; Cole, 1998; Dash *et.al.*, 2001], los tiempos de cómputo que requieren son inaceptables para las aplicaciones prácticas, ya que son del orden de N^2, donde N es la cantidad de documentos a agrupar [Zamir *et.al.*, 1998; Dash *et.al.*, 2001].

3.1.1.2 Algoritmo K-Means

El algoritmo K-Means es mucho más eficiente (los tiempos de cómputo requeridos son lineales) con la cantidad documentos a agrupar [Han *et.al.*, 2001]), pero es dependiente de la selección inicial de centroides [Bradley et al, 1998]. Sus resultados pueden ser bastante pobres y suelen variar mucho si se aplica varias veces a la misma colección de documentos, ya que si la selección de centroides al azar es mala, la solución encontrada también lo será.

3.1.1.3 Algoritmo Bisecting K-Means

Aunque este algoritmo ha demostrado obtener mejores soluciones que el algoritmo k-means y los métodos jerárquicos aglomerativos [Steinbach *et.al.*, 2000], en tiempos de cómputo lineales con la cantidad de documentos a agrupar, la forma en que explora el

espacio de búsqueda es claramente sub-óptima. Al momento de dividir un grupo, el algoritmo “Bisecting K-Means” realiza 5 divisiones diferentes del grupo (aplicando 5 veces el algoritmo K-Means con k=2 a los elementos del grupo), y luego elige una de ellas (la que más hace crecer el criterio de optimización), descartando las demás divisiones que creó.

El siguiente análisis muestra por qué esta forma de trabajo no es eficiente:

Cuando el grupo es fácilmente divisible (hay 2 subgrupos claramente definidos), las 5 corridas del algoritmo K-Means encontrarán la misma división, o divisiones con muy pocas variaciones. En este caso, el algoritmo estaría repitiendo 5 veces el mismo trabajo.

Cuando no hay dos subgrupos claramente definidos, las 5 corridas del algoritmo K-Means encontrarán divisiones diferentes. En este caso, el problema es que el algoritmo Bisecting K-Means no tiene ningún mecanismo para aprovechar el trabajo realizado al armar las divisiones que son descartadas. Es muy posible que en alguna de ellas haya encontrado un grupo excelente y uno muy malo que en promedio hagan que descarte esa división, y no tiene ningún mecanismo que le permita quedarse con la parte buena y descartar sólo lo malo.

3.1.1.4 Algoritmos Genéticos

Se ha demostrado [Yolis, 2003] que los algoritmos genéticos obtienen mejores resultados que los algoritmos K-Means y Bisecting K-Means, tanto en tiempo de respuesta como en calidad. De todos modos, esa mejoría no es suficiente para lo vertiginosas que son las operaciones diarias.

Estos problemas de los algoritmos actuales dan lugar a la siguiente cuestión:

Cuestión 1: ¿Pueden ayudar los mapas auto-organizados de Kohonen a encontrar soluciones de mejor calidad que las ofrecidas por los algoritmos K-Means, Bisecting K-Means y Genéticos?

3.2 APLICACIÓN DE LOS MAPAS AUTO-ORGANIZADOS DE KOHONEN A LA MINERÍA DE TEXTOS

La aplicación de los mapas auto-organizados de Kohonen ya ha sido extensamente estudiada [Kaski, 1997]. Nos ocuparemos ahora de algunos aspectos generales de la minería de textos.

En minería de textos y recuperación de información las necesidades del usuario pueden variar desde buscar un conocimiento específico hasta un deseo de familiarizarse con un área de estudio [Honkela, 1997]. El nivel de conocimiento de la persona determina que tan bien esa persona seleccionará las palabras y frases en una búsqueda específica. Cuando se evalúan sistemas de recuperación de información, las necesidades del usuario se consideran, por lo general, limitadamente. Sería de utilidad a la hora de la evaluación, conocer qué tanto el usuario conoce del dominio de la consulta y de la colección de documentos y cuáles son sus necesidades de información.

Las colecciones de documentos pueden variar en muchos aspectos. Algunos de los más relevantes son: el tamaño de la colección, el tamaño de los documentos y posible información adicional disponible como clasificación de los documentos o índices.

Los servicios básicos que un sistema de recuperación de información y minería de datos debería ofrecer son: búsqueda de documentos mediante palabras clave, exploración de la colección de documentos mediante la organización de los documentos en alguna forma lógica y filtrado. La organización de los documentos está usualmente basada en algún esquema jerárquico de clasificación, donde cada documento es asignado a una o más clases. Otro tipo de organización puede ser provisto mediante la introducción de asociaciones entre documentos. El filtrado se refiere al descarte de documentos no interesantes de una colección.

3.2.1 Requerimientos de los sistemas de recuperación de la información y minería de textos

Un sistema exitoso debería ser capaz de abordar los problemas más importantes en el área previamente discutidos. Además debería ser posible usar el sistema en diferentes formas para satisfacer las distintas necesidades. En resumen, el sistema debería [Honkela, 1997]:

- Soportar todas la funciones básicas, búsqueda, exploración y filtrado.
- Requerir la menor intervención humana posible para permitir el procesamiento de grandes colecciones de documentos.
- En búsqueda, proveer los resultados en orden de relevancia y ofrecer formas de explotar lo que quedó fuera del resultado.
- Soportar la exploración mediante vistas de la colección de documentos.
- Proveer resultados correctos aún si los documentos están pobremente escritos.
- Ser computacionalmente posible.
- Tener en cuenta en forma general los aspectos inherentes al lenguaje natural.

3.2.2 Aplicaciones basadas en mapas auto-organizados de Kohonen

Para producir mapas que muestren relaciones de semejanza entre los contenidos de documentos se debe idear un método para codificar los documentos. Tal vez el método más directo es codificar los documentos basándose en las palabras que ellos contienen. En un estudio prematuro, se formó un pequeño mapa de documentos científicos basado en las palabras contenidas en los títulos [Lin et al., 1991; Lin, 1992]. Más tarde también Lin extendió el método al texto completo de los documentos [Lin, 1997]. Cuando se usa el modelo de espacio de vectores, el procedimiento general para convertir documentos en vectores puede ser resumido en tres pasos [Honkela, 1997; modificado de Lin,1997]:

- Identificar un vocabulario de la colección de documentos (basado en las palabras contenidas en los títulos, en los resúmenes (abstract) o en los textos completos)

- Borrar algunas palabras comunes usando una lista. Remover también los términos que más se repiten

- Indexar la colección de documentos. Los componentes del vector de documentos pueden ser:

- Dígitos binarios. El valor depende de la aparición o no del término en el documento

- Pesos basados en la frecuencia de ocurrencia del término en el documento

- Pesos basados en la frecuencia del término y la frecuencia inversa del documento. La frecuencia inversa del documento es la inversa del número de documentos en que el término aparece. [Salton et al., 1975; Salton, 1989].

Scholtes desarrolló, basándose en mapas auto-organizados de Kohonen, un filtro y un mapa para recuperación de la información [Scholtes, 1993]. Scholtes también utilizó los caracteres n-gramas (secuencia de n caracteres). Esta aproximación también fue seguida en [Hyötyniemi, 1996]. Merkl utilizó los mapas auto-organizados de Kohonen para agrupar descripciones de componentes de bibliotecas de programas basándose en principios similares [Merkl, 1993; 1994; 1995a; 1995b; 1997; Merkl et al., 1994]. Rozmus y Zavrel también crearon mapas de documentos [Rozmus, 1995; Zavrel, 1995; 1996]. Kohonen y algunos de sus alumnos presentaron entre 1997 y 1998 un método llamado WEBSOM que organiza una colección de documentos en un mapa que da una imagen de la colección así como un mecanismo para recorrerla. [Honkela, 1997, Kaski et al., 1998].

Ninguna de las aplicaciones ha abordado la posibilidad de la búsqueda por palabra clave, lo cual nos lleva a las siguientes cuestiones:

Cuestión 2: ¿Pueden ayudar los mapas auto-organizados de Kohonen a categorizar los resultados de una búsqueda por palabra clave?

Cuestión 3: ¿Se pueden mejorar los resultados a medida que se le proporciona al mapa auto-organizado de Kohonen mayor entrenamiento?

CAPÍTULO 4

METODOLOGÍA DE DESARROLLO Y SOLUCIÓN PROPUESTA

Para el desarrollo se han seguido los lineamientos dados por la metodología Métrica V3. En este capítulo se presenta la solución propuesta en esta tesis, que apunta a responder a las cuestiones planteadas en el capítulo 3.

En la sección 4.1 se describe la Planificación de Sistemas de Información de Métrica V3.

En la sección 4.2 se describe el Desarrollo de Sistemas de Información de Métrica V3 que está compuesta por:

- Sección 4.2.1 Estudio de Viabilidad del Sistema
- Sección 4.2.2 Análisis del Sistema de Información
- Sección 4.2.3 Diseño del Sistema de Información
- Sección 4.2.4 Construcción del Sistema de Información

En la sección 4.3 se describe la Gestión de Configuración.

4.1 PLANIFICACIÓN DE SISTEMAS DE INFORMACIÓN

4.1.1 Inicio del plan de sistemas de información

4.1.1.1 Análisis de la Necesidad del PSI

El objetivo del presente PSI es la creación de un sistema que permita la búsqueda de documentos de texto por palabra clave y su posterior categorización automática utilizando mapas auto-organizados de Kohonen.

El sistema propuesto ha de cumplir con las siguientes funcionalidades:

- Búsqueda de documentos de texto por palabra clave.
- Categorización automática de resultados mediante mapas auto-organizados de Kohonen.

4.1.1.2 Identificación del Alcance del PSI

El presente sistema abarca a todo usuario que tenga información almacenada en documentos de texto y necesite realizar búsquedas frecuentes.

Objetivos estratégicos del presente sistema:

- Reducir el tiempo de respuesta en búsquedas de documentos de texto en comparación con el tiempo de respuesta obtenido con los sistemas de búsqueda basados en los algoritmos K-Means, Bisecting K-Means y Genéticos.
- Mejorar la calidad del resultado de las búsquedas en comparación con la calidad de los resultados de las búsquedas utilizando sistemas de búsqueda basados en los algoritmos K-Means, Bisecting K-Means y Genéticos.

La tabla 4.1 resume las características de los objetivos estratégicos mencionados.

Objetivo	Factores de Éxito	Componentes del Factor de Éxito
Menor tiempo de respuesta	Reducción de la carga de trabajo	Reasignación de recursos Optimización de recursos Reducción de costos Distribución de procesos de búsqueda de documentos de texto Automatización de los procesos de búsqueda de documentos de texto
Mejor calidad	Seguridad y precisión en la obtención de resultados	Optimización de recursos Mejora de resultados

Tabla 4.1. Objetivos estratégicos del presente sistema

4.1.1.3 Determinación de Responsables

El encargado del proyecto de sistemas (tesis) es el Lic Daniel Goldenberg,
El encargado de dar la aceptación son el Dr. Ramón García Martínez y la M. Ing. Paola Britos.

4.1.2 Definición y organización del PSI

4.1.2.1 Especificación del Ámbito y Alcance

El objetivo general del presente sistema es mejorar el proceso de búsqueda de documentos de texto, tanto en el tiempo que se requiere para encontrar el documento adecuado como en la calidad de los resultados.

4.1.2.2 Definición del Plan de Trabajo

La tabla 4.2 define el plan de trabajo.

Etapa	Producto	Duración
Planificación de Sistemas de Información – PSI	Documento de aceptación de sistemas	8 días
Análisis de Requerimientos de Sistemas – ARS	Documento de aceptación de requerimientos	12 días
Especificación Funcional del Sistema - EFS	Documento de aceptación de especificación funcional	3 días
Diseño Técnico del Sistema - DTS	Documento de aceptación de diseño técnico	10 días
Desarrollo de Componentes del Sistema - DCS	Documento de aceptación de pruebas de unidad del sistema.	20 días
Desarrollo de Procedimientos de Usuario – DPU	Documento de aceptación de procedimientos del usuario	5 días
Pruebas, Implantación y Aceptación - PIA	Documento de aceptación del sistema	4 días
Total		62 días

Tabla 4.2. Definición del Plan de Trabajo

4.1.3 Estudio de la información relevante

4.1.3.1 Selección y Análisis de Antecedentes

Se relevarán herramientas similares que existen en el mercado y se estudiará la arquitectura de los mapas auto-organizados de Kohonen. La investigación se hará mediante búsquedas en internet, consultas bibliográficas y consultas de documentación científica y académica.

4.1.3.2 Valoración de Antecedentes

La propia arquitectura de los mapas auto-organizados de Kohonen va a permitir la construcción de un producto estable y confiable. El relevamiento de herramientas similares permitirá comparar resultados.

4.1.4 Identificación de requisitos

4.1.4.1 Catalogación de Requisitos

- Debe ser posible ejecutar el sistema en plataforma Windows.

4.1.5 Estudio de los sistemas de información actuales

4.1.5.1 Alcance y Objetivos del Estudio de los Sistemas de Información Actuales

El objetivo de estudio de los sistemas de información actuales es la búsqueda de documentos de texto por palabra clave.

4.1.5.2 Análisis de los Sistemas de Información Actuales

Existen actualmente sistemas que realizan búsquedas de documento de texto devolviendo el resultado en forma de lista de documentos donde normalmente es tedioso encontrar el o los documentos que son de real interés. Algunos ejemplos de estos sistemas son sistemas basados en los algoritmos K-Means, Bisecting K-Means y Genéticos.

4.1.5.3 Valoración de los Sistemas de Información Actuales

El principal problema que presentan los sistemas actuales es que la forma en que los resultados son devueltos (lista de documentos) hace que normalmente la búsqueda para encontrar documentos de interés resulte tediosa y consuma mucho tiempo. Esto se ha constatado con sistemas de búsqueda basados en los algoritmos K-Means, Bisecting K-Means y Genéticos. Esta situación hace que en muchos casos extremos el usuario abandone la búsqueda sin resultados satisfactorios.

4.1.6 Diseño del modelo de sistemas de información

4.1.6.1 Diagnóstico de la Situación Actual

Se intentará que los usuarios de sistemas de búsqueda de documentos de texto basados en los algoritmos K-Means, Bisecting K-Means y Genéticos, remplacen su sistema actual por el nuevo sistema cuyo principal objetivo es proveer resultados en donde los documentos integrantes aparezcan automáticamente categorizados siendo muy fácil y sencillo para el usuario encontrar documentos de interés si los hay.

4.1.6.2 Definición del Modelo de Sistemas de Información

El proceso comienza con el ingreso de la palabra clave por parte del usuario. A continuación se realiza la búsqueda y finalmente se muestran los resultados ya categorizados.

La representación del flujo se encuentra en la figura 4.1:

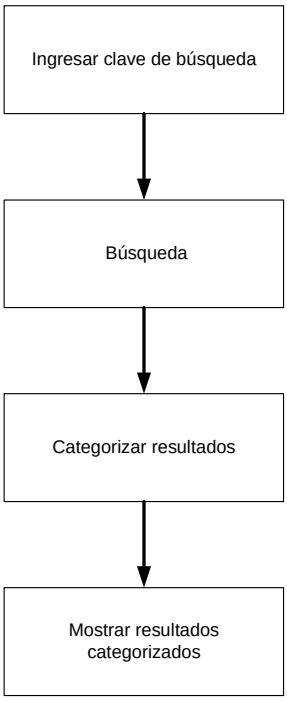


Figura 4.1 Flujo de información

4.1.7 Definición de la arquitectura tecnológica

4.1.7.1 Identificación de las Necesidades de Infraestructura Tecnológica

Las alternativas tecnológicas están limitadas por el requerimiento de que el sistema debe ser ejecutado en Windows.

4.1.7.2 Selección de la Arquitectura Tecnológica

Lenguajes de programación:

- Java: La selección del lenguaje se ha definido por su facilidad para ser ejecutado en cualquier plataforma.
- Jython: Al haber seleccionado Java como lenguaje principal de programación, la elección inmediata es Jython por ser un dialecto de Java.
- ANSI C: El mapa auto-organizado de Kohonen utilizado está desarrollado en ANSI C
- Visual Basic 6: Para la interfaz de usuario.

4.1.8 Definición del plan de acción

4.1.8.1 Definición de Proyectos a Realizar

El proyecto se divide en cinco subproyectos:

- Definición de los programas necesarios para la preparación de los datos. Esto puede visualizarse en 4.2.1.1.2, 4.2.2.3.1, 4.2.2.4, 4.2.2.5, 4.2.3.1 y 4.2.3.3.

- Definición de los programas necesarios para la búsqueda de los documentos por palabra clave. Esto puede visualizarse en 4.2.1.1.2, 4.2.2.3.1, 4.2.2.4, 4.2.2.5, 4.2.3.1 y 4.2.3.3.

- Definición de los programas necesarios para la categorización de los documentos. Esto puede visualizarse en 4.2.1.1.2, 4.2.2.3.1, 4.2.2.4, 4.2.2.5, 4.2.3.1 y 4.2.3.3.

- Definición de los programas necesarios para la interfaz de usuario de entrada. Esto puede visualizarse en 4.2.1.1.2, 4.2.2.3.1, 4.2.2.4, 4.2.2.5, 4.2.2.6, 4.2.3.1 y 4.2.3.3.

- Definición de los programas necesarios para la interfaz de usuario de salida. Esto puede visualizarse en 4.2.1.1.2, 4.2.2.3.1, 4.2.2.4, 4.2.2.5, 4.2.2.6, 4.2.3.1 y 4.2.3.3.

4.2 DESARROLLO DE SISTEMAS DE INFORMACIÓN

4.2.1 Estudio de Viabilidad del Sistema

4.2.1.1 Establecimiento del alcance del sistema

4.2.1.1.1 Estudio de la solicitud

Descripción General del Sistema

El objetivo del presente plan de sistemas de información es la creación de un sistema que permita la búsqueda de documentos de texto por palabra clave y su posterior categorización automática utilizando mapas auto-organizados de Kohonen. Con este sistema se busca mejorar el proceso de búsqueda de documentos de texto, tanto en el tiempo que se requiere para encontrar el documento adecuado como en la calidad de los resultados.

Catálogo de Objetivos del EVS

- Determinar la factibilidad de construcción del sistema.

- Estimar el tiempo de realización.

- Analizar la posibilidad de dividir al sistema en dos o más subsistemas.

Catálogo de Requisitos

La tabla 4.3 muestra el catálogo de requisitos.

Catálogo de requisitos			
Tipo de Requisito	Descripción	Prioridad	Estado
Funcional	El sistema debe poder ser ejecutado sobre plataforma Windows	Alta	Aprobado

Tabla 4.3. Catálogo de requisitos

4.2.1.1.2 Identificación del alcance del sistema

Descripción General del Sistema

Contexto del Sistema

El sistema puede ser dividido en cinco subsistemas:

Subsistema de Preparación de los Datos: Este subsistema es el encargado de realizar la preparación de los archivos de texto para que puedan ser interpretados por el mapa auto-organizado de Kohonen. Las especificaciones generales del subsistema son:

- Debe ser posible ejecutar el sistema en plataforma Windows.
- Los archivos de entrada deben ser de texto y en formato txt.

Subsistema de Búsqueda de documentos por Palabra Clave: Este subsistema es el encargado de realizar la búsqueda de los documentos que contengan la palabra clave ingresada por el usuario. Las especificaciones generales del subsistema son:

- Debe ser posible ejecutar el sistema en plataforma Windows.

Subsistema de Categorización de Documentos: Este subsistema es el encargado de realizar la categorización de los documentos hallados por el subsistema de Búsqueda por Palabra Clave. Las especificaciones generales del subsistema son:

- Debe ser posible ejecutar el sistema en plataforma Windows.
- Los documentos deben haber sido procesados por los subsistemas de Preparación de los Datos y de Búsqueda por Palabra Clave.

Subsistema de Interfaz de Usuario de Entrada: Este subsistema es el encargado de solicitar al usuario la palabra clave por la que desea realizar la búsqueda. Las especificaciones generales del subsistema son:

- Debe ser posible ejecutar el sistema en plataforma Windows.

Subsistema de Interfaz de Usuario de Salida: Este subsistema es el encargado de mostrar los resultados finales. Las especificaciones generales del subsistema son:

- Debe ser posible ejecutar el sistema en plataforma Windows.
- La salida debe ser en forma de mapa con navegador.

4.2.1.1.3 Especificación del alcance del estudio de viabilidad

Catálogo de Objetivos del EVS

Objetivos del Estudio de la Situación Actual

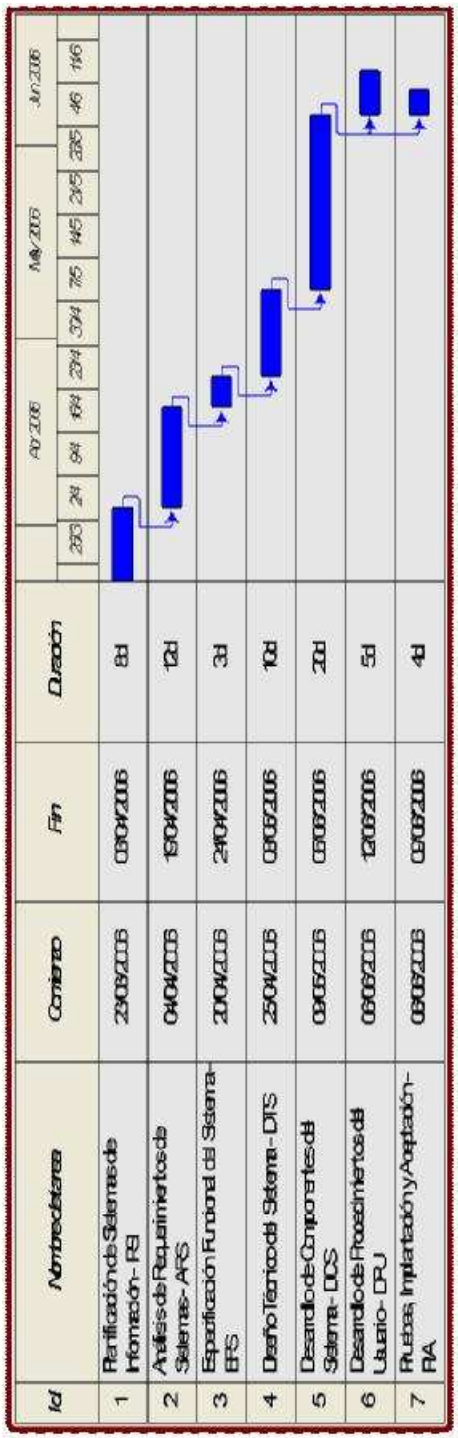
Existen herramientas similares en el mercado las que se evaluarán y adicionalmente se estudiará la arquitectura de los mapas auto-organizados de Kohonen.
Pueden verse algunas de las herramientas evaluadas en 3.2.2. Como allí se menciona, estas herramientas no incluyen la opción de poder buscar por palabra clave.

Plan de Trabajo

La tabla 4.4 muestra el plan de trabajo y en la figura 4.2 se ve el diagrama de Gantt asociado.

Etapa	**Producto**	**Duración**
Planificación de Sistemas de Información – PSI	Documento de aceptación de sistemas	8 días
Análisis de Requerimientos de Sistemas – ARS	Documento de aceptación de requerimientos	12 días
Especificación Funcional del Sistema – EFS	Documento de aceptación de especificación funcional	3 días
Diseño Técnico del Sistema - DTS	Documento de aceptación de diseño técnico	10 días
Desarrollo de Componentes del Sistema - DCS	Documento de aceptación de pruebas de unidad del sistema.	20 días
Desarrollo de Procedimientos de Usuario – DPU	Documento de aceptación de procedimientos del usuario	5 días
Pruebas, Implantación y Aceptación - PIA	Documento de aceptación del sistema	4 días
Total		62 días

Tabla 4.4. Plan de Trabajo



Id	Nombre de tarea	Comienzo	Fin	Duración
1	Ratificación de Sistemas de Información - RSI	23/03/2006	08/04/2006	8d
2	Análisis de Requerimientos de Sistemas - ARS	04/04/2006	19/04/2006	12d
3	Especificación Funcional del Sistema - EFS	20/04/2006	24/04/2006	3d
4	Diseño Técnico del Sistema - DTS	25/04/2006	08/05/2006	10d
5	Desarrollo de Componentes del Sistema - DCS	09/05/2006	06/06/2006	20d
6	Desarrollo de Procedimientos del Usuario - DPU	06/06/2006	12/06/2006	5d
7	Pruebas, Implantación y Aceptación - PIA	06/06/2006	09/06/2006	4d

Figura 4.2. Diagrama de Gantt

4.2.1.2 Estudio de la situación actual

4.2.1.2.1 Valoración del Estudio de la Situación Actual

Descripción de la Situación Actual

Actualmente se consume mucho tiempo y esfuerzo en encontrar documentos de interés, sin llegar a encontrarlos en muchos casos. Esto se debe a que los sistemas de búsqueda muestran sus resultados en forma de lista de documentos.

4.2.1.2.2 Realización del Diagnóstico de la Situación Actual

El hecho de tener que buscar un documento en una lista muchas veces extensa ocasiona que no siempre se pueda encontrar el documento deseado o que este se encuentre luego de un tedioso y extenso trabajo de investigación.

4.2.1.3 Definición de requisitos del sistema

4.2.1.3.1 Identificación de las Directrices Técnicas y de Gestión

Se realizará el desarrollo de acuerdo a las fases y técnicas sugeridas por la metodología Métrica V3.

4.2.1.3.2 Identificación de Requisitos

Se ha identificado que el sistema deberá cumplir con los siguientes requisitos:

- Debe ser posible ejecutar el sistema en plataforma Windows.
- La salida debe ser en forma de mapa con navegador.
- Los archivos de entrada para el mapa auto-organizado de Kohonen deben ser de texto y en formato txt.

4.2.1.3.3 Catalogación de Requisitos

La tabla 4.5 muestra el catálogo de requisitos.

Catálogo de requisitos			
Tipo de Requisito	Descripción	Prioridad	Estado
Funcional	El sistema debe poder ser ejecutado sobre plataforma Windows	Alta	Aprobado
	Los archivos de entrada para el mapa auto-organizado de Kohonen deben ser de texto y en formato txt	Alta	Aprobado

Catálogo de requisitos			
Tipo de Requisito	Descripción	Prioridad	Estado
	La salida debe ser en forma de mapa con navegador	Media	Aprobado

Tabla 4.5. Catálogo de requisitos

Nota: Luego de la investigación inicial realizada sobre el mapa auto-organizado de Kohonen se agregó el requsito relativo al formato de los archivos de entrada debido a la restricción propia del mismo.

4.2.1.4 Estudio de alternativas de solución

4.2.1.4.1 Preselección de Alternativas de Solución

Descomposición Inicial del Sistema en Subsistemas

El presente sistema se puede descomponer en cinco subsistemas:

- Subsistema de preparación de los datos.
- Subsistema de búsqueda de documentos por palabra clave.
- Subsistema de categorización de los documentos.
- Subsistema de interfaz de usuario de entrada.
- Subsistema de interfaz de usuario de salida.

Alternativas de Solución a Estudiar

Para el subsistema de categorización de los documentos, específicamente para el mapa auto-organizado de Kohonen que se encargará de la categorización, existen tres alternativas:

- Utilización de fuentes libres.
- Comprar algún producto del mercado.
- Generar un desarrollo propio.

Para satisfacer la primera opción se hizo una búsqueda intensiva en los principales sitios de fuentes libres habiéndose encontrado una versión del mapa auto-organizado de Kohonen que el mismo Teuvo Kohonen ha dejado disponible para uso académico y de investigación. Esta versión del mapa auto-organizado de Kohonen cumple con las especificaciones solicitadas en los requisitos. Con esto se han descartado las otras opciones.

4.2.1.4.2 Descripción de las Alternativas de Solución

Alternativa Fuentes Libres

Descripción del Producto

Esta versión del mapa auto-organizado de Kohonen ha sido desarrollada por su mismo creador, lo cual ya está dando una seguridad de estar utilizando una versión original y que sin dudas funciona correctamente.

El mismo ha sido desarrollado en ANSI C, lenguaje que el tesista conoce en caso de que sea necesario recurrir a detalles de programación del mapa.

Incluye documentación que explica con detalles acerca de su funcionamiento por lo que no se debería tener que invertir mucho tiempo en comprenderlo.

Cumple con el requisito de poder ser ejecutado en Windows ya que incluye una versión para DOS que puede ejecutarse en Windows sin inconvenientes. Además provee una versión para Unix que se podría utilizar en caso de querer ampliar el sistema a esa plataforma.

4.2.1.5 Valoración de las alternativas

4.2.1.5.1 Estudio de la inversión

No es necesario adquirir licencias de software. Además el desarrollo será realizado por el propio equipo de desarrollo. Esto implica que el impacto en términos de costos es mínimo.

Se está pasando de un método de búsqueda tedioso y que en ocasiones no arroja resultados satisfactorios a un método de búsqueda amigable y será mucho más fácil hallar los resultados. Esto está hablando de los beneficios que se obtendrían con este sistema.

Estimación de Costo-Beneficio

Costo Desarrollo Anualizado

La tabla 4.6 muestra el resultado del análisis costo-beneficio.

Ítem	Valor	Observaciones
Adquisición de hardware y software	0$	
Gastos de Mantenimiento de hardware y software	0$	
Gastos de Comunicación	0$	
Gastos de Desarrollo	21080$	496 hs * 42.5$
Gatos de Mantenimiento	0$	
Gastos de Consultaría	0$	
Gastos de Formación	2000$	20 hs * 100$

Ítem	Valor	Observaciones
Gastos de Material	5000$	Varios
Costos Financieros	0$	

Tabla 4.6. Costo-Beneficio

Beneficios del desarrollo anualizado

La tabla 4.7 muestra el beneficio esperado.

Ítem	Valor	Observaciones
Reducción del tiempo de respuesta	58600$	Ahorro de 2 horas promedio en el tiempo de respuesta. Cantidad de búsquedas realizadas por mes 100, valor anualizado. Son estimaciones
Ahorro de adquisición y mantenimiento de hardware y software	0$	
Ahorro de material de todo tipo	0$	
Beneficios Financieros	0$	
Otros beneficios tangibles	0$	
Beneficios intangibles	0$	

Tabla 4.7. Beneficio esperado

En función del análisis del tiempo de retorno de la inversión y el valor actual seria factible la realización del proyecto.

4.2.2 Análisis del Sistema de Información

4.2.2.1 Definición del sistema

4.2.2.1.1 Determinación del Alcance del Sistema

Catálogo de Requisitos:

Funcionales

- El sistema debe poder ser ejecutado sobre plataforma Windows.
- Los archivos de entrada para el mapa auto-organizado de Kohonen deben ser de texto y en formato txt.

Glosario

- Fuentes libres: programas que se pueden bajar de Internet, y su distribución es gratuita y se cuenta con el código fuente de los mismos.

4.2.2.1.2 Identificación del Entorno Tecnológico

Descripción General del Entorno Tecnológico del Sistema:

El principal escollo tecnológico a afrontar es que el sistema debe poder ser ejecutado en plataformas Windows. Por esto se selecciona el lenguaje de programación Java que permite ser ejecutado en Windows. El lenguaje Java, además de satisfacer esta condición es conocido por el tesista.

Con respecto a tiempos de respuesta y requerimientos de memoria mínimos y máximos no se han hecho especificaciones. Lo que se debe tener en cuenta es que generalmente se ejecutara en computadoras personales de usuarios, que en muchos casos podrán contar con cantidades de memoria reducidas, 256 Mb.

4.2.2.2 Establecimiento de requisitos

4.2.2.2.1 Obtención de Requisitos

Catálogo de Requisitos:

- El sistema debe poder ser ejecutado sobre plataforma Windows.
- Los archivos de entrada para el mapa auto-organizado de Kohonen deben ser de texto y en formato txt.
- La salida debe ser en forma de mapa con navegador.

Modelo de Casos de Uso:

La figura 4.2 muestra el esquema del modelo de caso de uso relevante para el presente sistema.

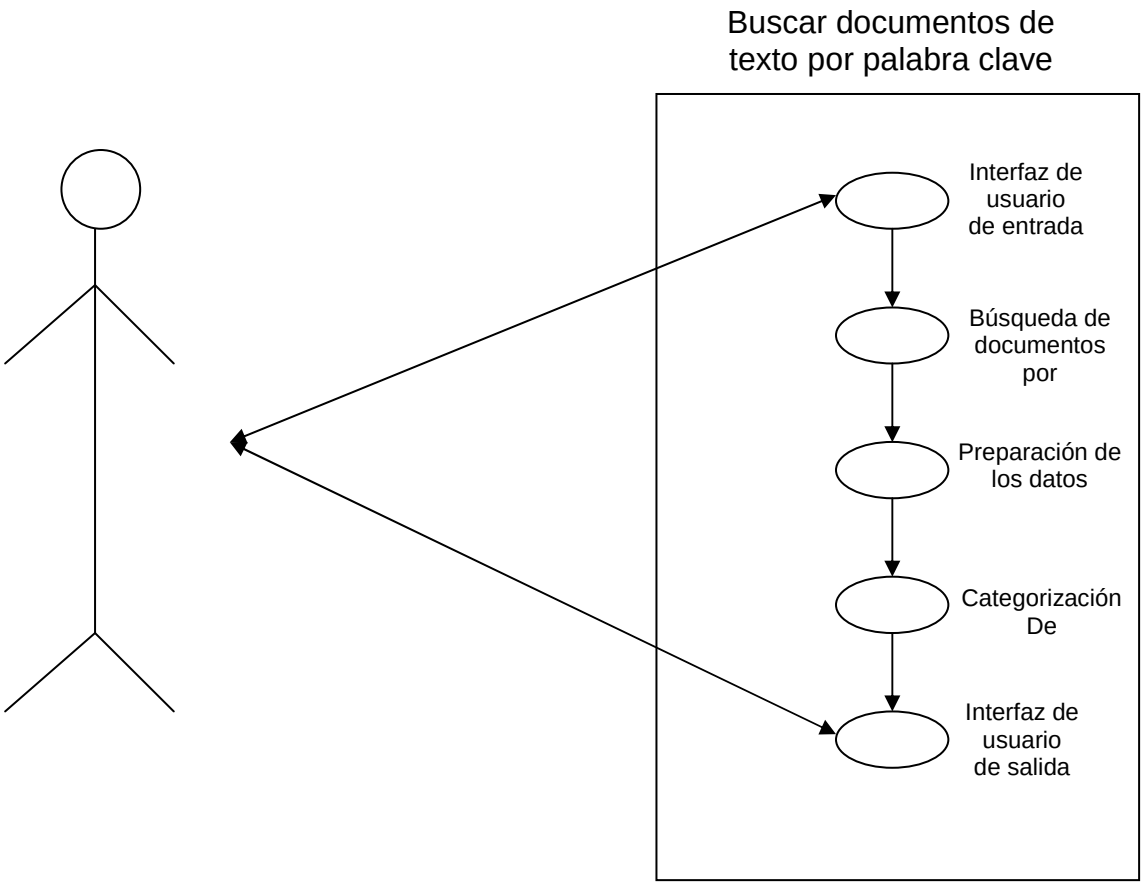


Figura 4.2. Modelo de caso de uso Buscar documentos de texto por palabra clave

4.2.2.2.2 Especificación de Casos de Uso

La tabla 4.8 muestra la especificación del caso de uso definido para el presente sistema:

Caso de Uso	Buscar documentos de texto por palabra clave
Resumen	Buscar documentos de texto que contengan la palabra clave dada por el usuario y devolver el resultado en forma categorizada
Prioridad	Esencial
Frecuencia de Uso	Siempre
Actores Directos	Usuarios
Inversionistas	***Usuarios***: quieren búsquedas rápidas y confiables
Prerrequisitos	No hay

Postcondiciones	Devolver los documentos encontrados en forma categorizada
Escenario Principal de Éxito	El usuario ingresa una palabra clave El usuario obtiene los documentos hallados en forma categorizada y puede encontrar fácilmente el documento de interés
Escenario de Extensiones Alternativas	No hay
Lista de tecnología y variaciones de datos	Sistema operativo con el cual debe correr: Windows
Notas y Preguntas	No hay

Tabla 4.8. Caso de Uso

Caso de Uso Búsqueda de documentos de texto por palabra clave

Un usuario necesita buscar documentos de texto por una palabra clave. Ingresa la palabra clave para la realización de la búsqueda. El sistema devuelve los documentos encontrados y el usuario debe buscar entre los documentos devueltos cuál es el que realmente es de su interés.

4.2.2.3 Identificación de subsistemas de análisis

4.2.2.3.1 Determinación de Subsistemas de Análisis

El sistema se divide en cinco subsistemas:

- Subsistema de preparación de los datos.
- Subsistema de búsqueda de los documentos por palabra clave.
- Subsistema de categorización de los documentos.
- Subsistema de interfaz de usuario de entrada.
- Subsistema de interfaz de usuario de salida.

4.2.2.3.2 Integración de Subsistemas de Análisis

El usuario interactuará con el sistema sólo a través del subsistema de interfaz de usuario. El resto de los subsistemas interactuarán entre sí a medida que lo necesiten.

4.2.2.4 Análisis de los casos de uso

4.2.2.4.1 Identificación de Clases Asociadas a un Caso de Uso

Del estudio del caso de uso se han detectado las siguientes actividades comunes a todas las búsquedas:

- Interfaz de usuario de entrada: Todas las búsquedas se inician con el ingreso de la palabra clave a ser buscada.
- Búsqueda de documentos por la palabra clave: Para encontrar qué documentos contienen la palabra clave
- Preparación de los datos: Esta preparación es necesaria para que los documentos encontrados puedan ser comprendidos por el mapa auto-organizado de Kohonen.
- Categorización de documentos: Para que puedan ser correctamente mostrados en la salida.
- Interfaz de usuario de salida: Es el paso final de toda búsqueda, en el que se muestra el mapa al usuario y se le permite navegar por él.

Se detallan a continuación los nombres tentativos de las clases para su implementación y sus principales características:

- ingresapalabraclave: Esta clase se encarga de tomar la palabra clave ingresada por el usuario a ser buscada en los documentos.

Características principales:

- Debe poder convertir a texto cualquier cadena de caracteres ingresada.
- Da servicio a la clase buscadocumentos.

- buscadocumentos: Esta clase se encarga de buscar los documentos que contengan la palabra clave ingresada por el usuario.

Características principales:

- Debe ser capaz de buscar en grandes conjuntos de documentos.
- Recibe servicio de la clase ingresapalabraclave.
- Da servicio a la clase transformadocumentos.

- transformadocumentos: Esta clase se encarga de transformar los documentos encontrados para que puedan ser procesados por el mapa auto-organizado de Kohonen.

Características principales:

- Recibe servicio de la clase buscadocumentos.
- Da servicio a la clase categorizadocumentos.

- categorizadocumentos: Esta clase se encarga de categorizar los documentos transformados utilizando el mapa auto-organizado de Kohonen.

Características principales:

- Debe ser capaz de operar en grandes conjuntos de documentos.
- Recibe servicio de la clase transformadocumentos.
- Da servicio a la clase muestramapa.

- muestramapa: Esta clase se encarga de mostrar el mapa resultante de la búsqueda y navegarlo de acuerdo a lo indicado por el usuario.

Características principales:

- Debe permitir la navegación a través del mapa.
- Debe permitir editar el documento original (sin transformar) cuando el usuario lo solicita.
- Recibe servicio de la clase categorizadocumentos.

4.2.2.4.2 Descripción de la Interacción de Objetos

La figura 4.3 ilustra la interacción de los objetos pertenecientes al presente sistema mediante una matriz de interacción.

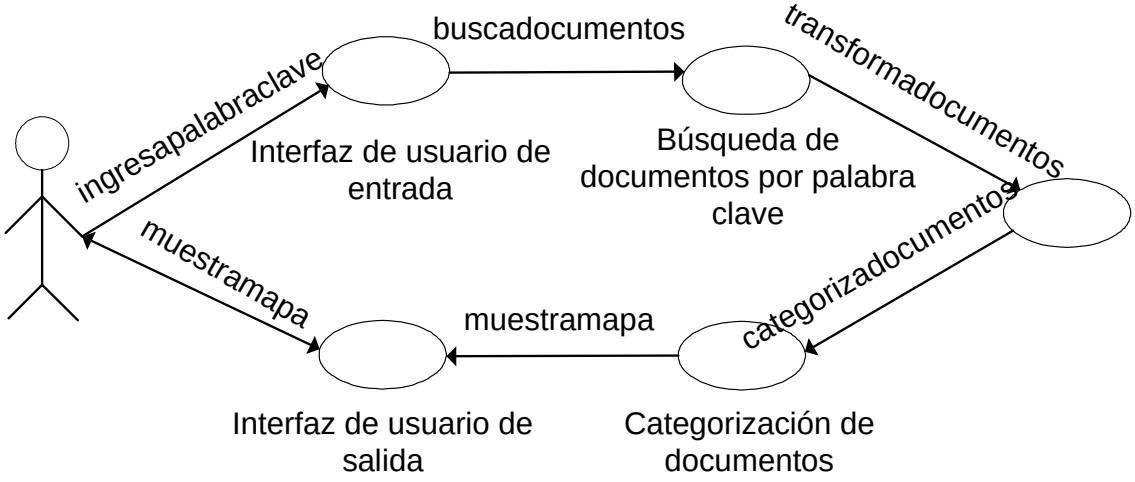


Figura 4.3. Interacción para el caso de uso Buscar documentos de texto por palabra clave

4.2.2.5 Análisis de clases

4.2.2.5.1 Identificación de Responsabilidades y Atributos

Identificación de atributos candidatos

Ingresapalabraclave: (ver la especificación en 4.2.2.4.1)

- Leerpalabra
- transmitirpalabra

buscadocumentos: (ver la especificación en 4.2.2.4.1)

- leerdocumentos
- buscarpalabra
- seleccionardocumentos

transformadocumentos: (ver la especificación en 4.2.2.4.1)

- leedocumentos
- creadocumentos
- transmitedocumentos

categorizadocumentos: (ver la especificación en 4.2.2.4.1)

- leedocumentos
- asignaacategorias
- transmitecategorizacion

muestramapa: (ver la especificación en 4.2.2.4.1)

- leecategorización
- muestravista
- muestraarchivo
- recibeorden

4.2.2.6 Definición de interfaces de usuario

4.2.2.6.1 Especificación de Principios Generales de la Interfaz

La interfaz de entrada consta de una ventana donde el usuario ingresa la palabra clave a ser buscada en los documentos.
La interfaz de salida es en forma de mapa con navegador que permite al usuario encontrar fácilmente un documento.

4.2.2.7 Análisis de consistencia y especificación de requisitos

4.2.2.7.1 Elaboración de la Especificación de Requisitos Software (ERS)

La tabla 4.9 muestra la especificación de requisitos de software.

Requisitos de sistema	El sistema debe poder ser ejecutado sobre plataforma Windows
	Los archivos de entrada para el mapa auto-organizado de Kohonen deben ser de texto y en formato txt.
Plataforma de uso	Windows
Lenguaje de Programación	Java versión 1.5
	Jython
	ANSI C
	Visual Basic 6
Módulos del Sistema	Preparador de datos
	Buscador de documentos por palabra clave
	Categorizador de documentos
	Interfaz de usuario de entrada
	Interfaz de usuario de salida

Tabla 4.9. Especificación de Requisitos de Software

4.2.2.8 Especificación del plan de pruebas

4.2.2.8.1 Definición del Alcance de las Pruebas

Las pruebas consistirán en búsquedas de documentos de texto por palabra clave en un conjunto de documentos predefinido. Se utilizará un conjunto de datos conocido y se verificará que los documentos obtenidos en cada búsqueda sean los correctos y estén ubicados en la categoría correcta.

4.2.2.8.2 Definición de Requisitos del Entorno de Pruebas

Las pruebas se realizarán sobre el mismo equipo utilizado para el desarrollo. Ver sección 4.2.2.1.2.

4.2.3 Diseño del Sistema de Información

4.2.3.1 Definición de la arquitectura del sistema

4.2.3.1.1 Definición de Niveles de Arquitectura

Los cinco subsistemas citados en 4.2.1.1.2, 4.2.1.4.1 y 4.2.2.3.1 se encontrarán en computadoras personales de usuarios.

4.2.3.1.2 Identificación de Requisitos de Diseño y Construcción

Subsistema de Preparación de los Datos:

Este subsistema se puede encontrar en una computadora personal de usuario.

Como lenguaje de programación se ha seleccionado Jython, que es un dialecto de Java. Jython se enrola dentro de los lenguajes de Scripting, lo que le da gran flexibilidad, cumple con los requisitos de poder ser ejecutado sobre plataformas Windows y permite integrar las librerías de JAVA, con lo cual se logra gran poder de procesamiento.

Los archivos de entrada deben ser de texto y en formato txt.

Funciones del subsistema:

- Lectura de los archivos de texto hallados por el subsistema de búsqueda por palabra clave.
- Formateo de los archivos.
- Envío de los archivos formateados al subsistema de categorización de documentos.

Subsistema de Búsqueda por Palabra Clave:

Este subsistema se puede encontrar en una computadora personal de usuario.

Como lenguaje de programación se ha seleccionado Jython, que es un dialecto de Java. Jython se enrola dentro de los lenguajes de Scripting, lo que le da gran flexibilidad, cumple con los requisitos de poder ser ejecutado sobre plataformas Windows y permite integrar las librerías de JAVA, con lo cual se logra gran poder de procesamiento.

La entrada debe ser un texto

Los archivos de salida deben ser de texto y en formato txt.

Funciones del subsistema:

- Lectura de la palabra clave ingresada por el usuario.
- Búsqueda de los archivos que contengan la palabra clave.
- Envío de los archivos encontrados al subsistema de preparación de los datos.

Subsistema de Categorización de Documentos:

Este subsistema se puede encontrar en una computadora personal de usuario. Se ha seleccionado el Mapa auto-organizado de Kohonen el algoritmo ha sido desarollado en ANSI C y el mismo es una Fuente Libre cuyo autor es Teuvo Kohonen (quien lo ha desarrollado). Además de brindar la confianza de haber sido desarrollado por su mismo creador, cumple con los requisitos de poder ser ejecutado sobre plataformas Windows. La entrada son los archivos formateados por el subsistema de preparación de los datos.

La salida es un archivo plano que muestra los archivos categorizados.

La función del subsistema es categorizar los documentos.

Subsistema de Interfaz de Usuario de Entrada:

Este subsistema se puede encontrar en una computadora personal de usuario.

Como lenguaje de programación se ha seleccionado Visual Basic 6.

Las funciones del subsistema son:

- Solicitar al usuario el ingreso de una palabra clave.
- Enviar la palabra clave al subsistema de búsqueda de documentos por palabra clave.

Subsistema de Interfaz de Usuario de Salida:

Este subsistema se puede encontrar en una computadora personal de usuario.

Como lenguaje de programación se ha seleccionado Visual Basic 6, ya que es el que mejor se adapta a la idea de obtener un mapa con navegador.

Las funciones del subsistema son:

- Tomar la salida del subsistema de categorización de documentos.
- Mostrar la salida en forma de mapa con navegador.

4.2.3.1.3 Identificación de Subsistemas de Diseño

- Subsistema de preparación de los datos.
- Subsistema de búsqueda de los documentos por palabra clave.
- Subsistema de categorización de los documentos.
- Subsistema de interfaz de usuario de entrada.
- Subsistema de interfaz de usuario de salida.

4.2.3.1.4 Especificación del Entorno Tecnológico

Se ha establecido el siguiente entorno para los cinco subsistemas:

- Hardware: Computadora personal con placa principal con arquitectura PCI/ISA con AGP y bus de 100 MHZ y 64 bits para acceso a memoria – Procesador Intel Pentium III – 800 MHZ o superior compatible - 256 Mb de RAM – Disco Rígido con capacidad disponible de 10 MB para el Sistema –
- Software: Sistema Operativo Microsoft Windows.

4.2.3.2 Diseño de casos de uso reales

4.2.3.2.1 Diseño de la Realización de los Casos de Uso

El diagrama de secuencia relacionado al item (caso de uso definido en el sistema) se muestra en la figura 4.3.

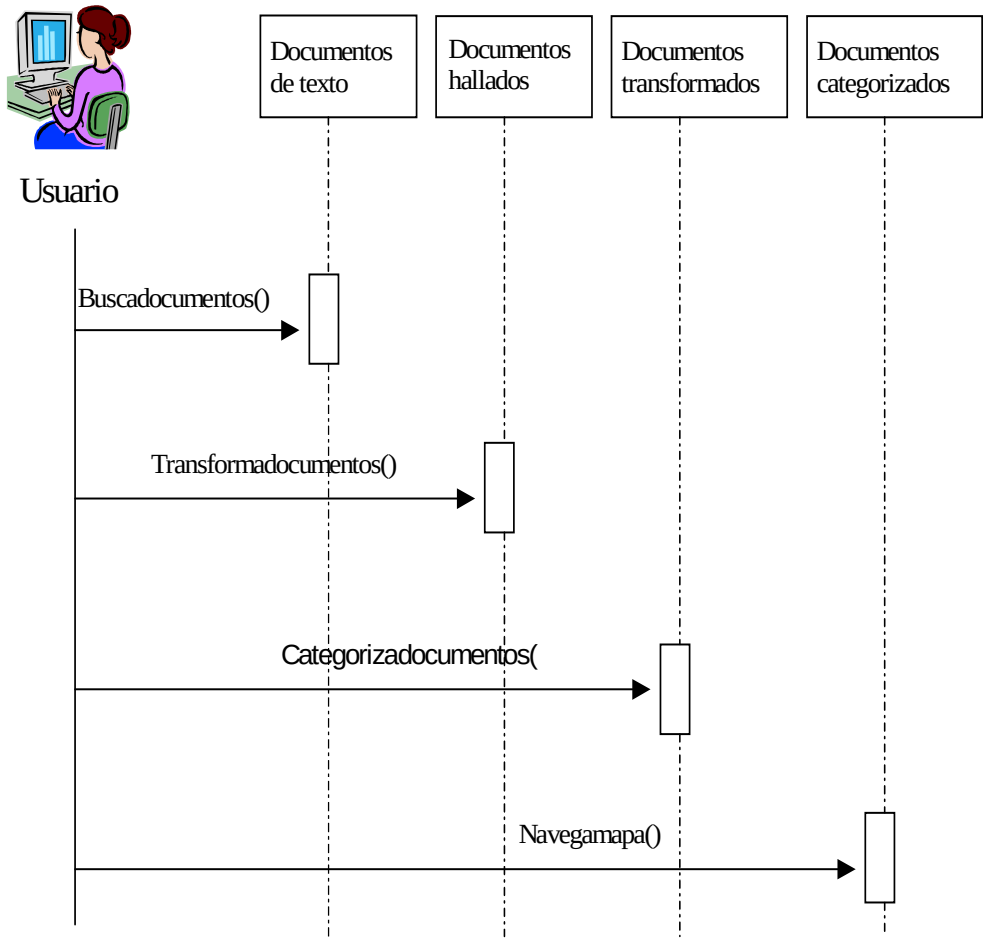


Figura4.3. Diagrama de secuencia del proceso de búsqueda de documentos de texto por palabra clave

4.2.3.2.2 Revisión de la Interfaz de Usuario

Se detalla a continuación la interfaz de usuario:

- ingresapalabraclave: Interactúa con la clase buscadocumentos. Corresponde al subsistema de interfaz de usuario de entrada.
- muestramapa: Interactúa con la clase categorizadocumentos. Corresponde al subsistema de interfaz de usuario de salida.

4.2.3.3 Diseño de clases

4.2.3.3.1 Identificación de Atributos de las Clases

Se detallan a continuación los atributos propuestos para el esquema de clases:

ingresapalabraclave: (ver la especificación en 4.2.2.4.1)

- Leerpalabra
- Transmitirpalabra

buscadocumentos: (ver la especificación en 4.2.2.4.1)

- leerdocumentos
- buscarpalabra
- seleccionardocumentos

transformadocumentos: (ver la especificación en 4.2.2.4.1)

- leedocumentos
- creadocumentos
- transmitedocumentos

categorizadocumentos: (ver la especificación en 4.2.2.4.1)

- leedocumentos
- asignaacategorias
- transmitecategorizacion

muestramapa: (ver la especificación en 4.2.2.4.1)

- leecategorización
- muestravista
- muestraarchivo
- recibeorden

4.2.3.4 Especificación técnica del plan de pruebas

4.2.3.4.1 Especificación del Entorno de Pruebas

Se realizarán pruebas mediante búsquedas de documentos de texto en un conjunto de documentos preparado para tal fin. Se definen para las pruebas dos categorías de documentos: documentos de informática y documentos de zoología. Se preparan tres conjuntos de documentos para las pruebas: un conjunto de documentos que corresponden a la primera categoría, un conjunto de documentos que corresponden a la segunda categoría y un conjunto de documentos que no corresponden a ninguna de las dos categorías definidas para las pruebas.

La tabla 4.10 muestra la definición de los casos de prueba

Identificación del caso de prueba	Descripción	Resultado esperado
PR001	Búsqueda de la palabra “objetos”	El sistema debe hallar la palabra “objetos” en los documentos “Doc correcto 1 - Informatica.txt” y “Doc correcto 5 - Informatica.txt” e incluir ambos documentos en la categoría “informática”
PR002	Búsqueda de la palabra “ratón”	El sistema debe hallar la palabra “ratón” en los documentos “Doc correcto 7 - Zoologia.txt” y “Doc incorrecto 1.txt”. Además debe incluir el primer documento en la categoría “Zoología” e indicar que el segundo documento no pertenece a ninguna de las categorías definidas.
PR003	Búsqueda de la palabra “maestro”	El sistema debe hallar la palabra “maestro” en los documentos “Doc correcto 7 - Informatica.txt”, “Doc incorrecto 5.txt” y “Doc incorrecto 6.txt”. Además debe incluir el primer documento en la categoría “Informática” e indicar que los otros dos documentos no pertenecen a ninguna de las categorías definidas.
PR004	Búsqueda de la palabra “España”	El sistema debe hallar la palabra “España” en los documentos “Doc correcto 11 - Zoologia.txt” y “Doc correcto 7 - Informatica.txt”. Además debe incluir el primer documento en la categoría “Zoología” y el segundo en la categoría “Informática”.

PR005	Búsqueda de la palabra “silla”	El sistema debe informar que no se han encontrado documentos que satisfagan la búsqueda realizada.

Tabla 4.10. Definición de los casos de prueba

4.2.3.4.2 Especificación Técnica de Niveles de Prueba

La revisión técnica consistirá de:

Calidad de la información obtenida. Se verificará que los documentos encontrados en cada búsqueda sean los correctos y que cada documento se muestre en la categoría correcta.

4.2.3.5 Establecimiento de requisitos de implantación

4.2.3.5.1 Especificación de Requisitos de Documentación de Usuario

Se entregarán a los usuarios manuales de uso y operaciones.

4.2.4 Construcción del Sistema de Información

4.2.4.1 Ejecución de las pruebas del sistema

4.2.4.1.1 Preparación del Entorno de las Pruebas del Sistema

Las pruebas del sistema se realizarán en una computadora personal en poder del equipo de desarrollo. Ver los detalles en la sección 4.2.3.1.4.

Se necesitará también contar con el conjunto de documentos a utilizar, los diferentes textos a buscar y los resultados esperados en cada caso.

4.2.4.5.2 Realización de las Pruebas del Sistema

La tabla 4.11 muestra los resultados obtenidos al probar los casos de prueba detallados en la tabla 4.10:

Identificación del caso de prueba	Resultado obtenido
PR001	Idéntico al resultado esperado
PR002	Idéntico al resultado esperado
PR003	Idéntico al resultado esperado
PR004	Idéntico al resultado esperado
PR005	Idéntico al resultado esperado

Tabla 4.11. Resultados de las pruebas

Nota: No se detallan aquí las pruebas con errores detectados.

4.2.4.5.3 Evaluación del Resultado de las Pruebas del Sistema

Los resultados detallados en la sección anterior demuestran el correcto funcionamiento del sistema.

4.3 GESTIÓN DE CONFIGURACIÓN

4.3.1 ESTUDIO DE VIABILIDAD DEL SISTEMA

4.3.1.1 DEFINICIÓN DE LOS REQUISITOS DE GESTIÓN DE CONFIGURACIÓN

4.3.1.1.1 Definición de los Requisitos de Gestión de Configuración

Se establece la siguiente lista de requisitos de Gestión de Configuración:

- Crear y mantener un plan de Gestión de Configuración para el presente sistema.
- Asegurar que todos los cambios al presente sistema se realicen en base al plan de Gestión de Configuración establecido.
- Asegurar que no se realicen cambios no autorizados al presente sistema.
- Auditar al presente sistema.

4.3.1.2 ESTABLECIMIENTO DEL PLAN DE GESTIÓN DE LA CONFIGURACIÓN

4.3.1.2.1 Definición del Plan de Gestión de la Configuración

Los elementos de configuración del presente sistema son el código fuente y la documentación asociada. Estos serán almacenados en un medio electrónico del proyecto con acceso limitado al tesista, Directores de tesis y colaboradores.

Los documentos se numerarán de acuerdo a la versión actual. En caso de requerir cambios a los mismos, estos se incorporarán a una nueva versión del documento con el número siguiente. La versión inicial es la número 1.

4.3.1.2.2 Especificación del Entorno Tecnológico para la Gestión de Configuración

El entorno tecnológico para la Gestión de Configuración del presente sistema consistirá del propio entorno tecnológico del sistema sumándole a ello una o más unidades grabadores de CD o DVD y los correspondientes medios magnéticos.

4.3.2 ANÁLISIS, DISEÑO, CONSTRUCCIÓN E IMPLANTACION Y ACEPTACIÓN DEL SISTEMA DE INFORMACIÓN

4.3.2.1 IDENTIFICACIÓN Y REGISTRO DE PRODUCTOS

4.3.2.1.1 Identificación y Registro de los Productos de los Procesos en el Sistema de Gestión de la Configuración

Se muestra en la tabla 4.12 la identificación y registro de la primera versión de los documentos pertenecientes al presente sistema.

Documento	Versión	Estado	Propósito	Motivo del cambio
Planificación	1	Finalizado	Planificación general del sistema	Primera versión
Viabilidad	1	Finalizado	Estudio de Viabilidad del sistema	Primera versión
Análisis	1	Finalizado	Modelo de Análisis del sistema	Primera versión
Diseño	1	Finalizado	Modelo de Diseño del sistema	Primera versión
Construcción	1	Finalizado	Construcción del sistema	Primera versión
Implementación	1	Finalizado	Implementación del Sistema	Primera versión

Tabla 4.12. Ejemplo de registración de los elementos de configuración

4.3.2.2 IDENTIFICACIÓN Y REGISTRO DEL PRODUCTO GLOBAL

4.3.2.2.1 Registro en el Sistema de Gestión de la Configuración del Producto Global de Proceso

Se muestra en la tabla 4.13 la identificación y registro del producto global correspondiente al presente sistema.

Documento	Versión	Estado	Propósito	Motivo del cambio
Categorización Automática de Documentos	1	Finalizado	Sistema de categorización automática de documentos	Primera versión

Tabla 4.13. Ejemplo de registración del producto global

CAPÍTULO 5

EXPERIMENTACIÓN

En este capítulo se comparan los resultados de pruebas que se realizaron para evaluar la efectividad de la solución propuesta en el capítulo 4. La sección 5.1 describe los conjuntos de datos utilizados. Los mismos fueron extraídos de la colección "Reuters 21578" reconocida como un estándar dentro de la comunidad de investigadores dedicados a la categorización automática de documentos. En las secciones 5.2 y 5.3 se detalla la metodología seguida en la experimentación. Se enumeran las variables que intervienen en los experimentos y los distintos tipos de experimentos realizados. La sección 5.4 presenta los resultados de la experimentación. La presentación se hace en forma de gráficos y para cada variable se incluye además el resultado del test estadístico realizado sobre los resultados. En la sección 5.5 se analizan brevemente los resultados, que luego se tratan con más detalle en el capítulo 6.

5.1 CONJUNTO DE DATOS UTILIZADO

Para la realización de los experimentos, se utilizaron datos de la Colección de Prueba para Categorización de Documentos "Reuters 21578" [Lewis, 1997]. Ésta colección se ha convertido en un estándar "de facto" dentro del dominio de la categorización automática de documentos y es utilizada por numerosos autores de la materia [Joachims, 1998; Steinbach *et.al.*, 2000; Zhao *et.al.*, 2001]. La búsqueda de "Reuters 21578" en el índice de recursos ResearchIndex/Citeseer [Citeseer] (un sitio que se dedica a la publicación en línea de bibliografía especializada en las ciencias de la computación), arroja alrededor de 160 documentos que la mencionan o utilizan. Si se repite la búsqueda en el motor de búsqueda Google [Google], los resultados traen cerca de 19.000 páginas web, entre las cuales se encuentran páginas de productos comerciales que realizan categorización de documentos y que utilizan esta colección para sus pruebas.

Los documentos en la colección son noticias reales que aparecieron en cables de la agencia Reuters durante 1987. Los documentos fueron recopilados y categorizados manualmente por personal de la agencia y de la compañía Carnegie Group, Inc., en 1987. En 1990, la agencia entregó los documentos al Laboratorio de Recupero de Información (Information Retrieval Laboratory) de la Universidad de Massachusetts. Desde 1991 hasta 1996, la colección se distribuyó bajo la denominación "Reuters 22173". En 1996, durante la conferencia ACM SIGIR (una de las reuniones más importantes en el campo de recupero de información y categorización de documentos), un grupo de investigadores realizó un trabajo sobre esta colección con el objetivo de que los resultados de distintos trabajos que utilizaran la colección fueran más fáciles de comparar entre sí [ACM SIGIR, 1996]. El resultado fue la distribución 21578, que es la que actualmente se utiliza en los trabajos sobre categorización automática de documentos para asegurar una metodología de prueba uniforme. La colección se compone de 21.578 documentos (cantidad que le da nombre a la misma), distribuidos en 22 archivos. Cada documento tiene 5 campos de categorización distintos: Valor Bursátil, Organización, Persona, Lugar y Tema. En cada campo, el documento puede tener un sólo valor, varios, o ninguno.

Los criterios utilizados en esta tesis para medir la calidad de un agrupamiento requieren que cada documento tenga solamente una categoría asignada, por lo que para los experimentos se decidió utilizar dos subconjuntos extraídos de esta colección previamente utilizados por otros investigadores [Zhao *et.al.*, 2001]. El procedimiento seguido para extraer los subconjuntos fue tomar solamente los documentos que tenían sólo un valor en el campo “Tema”, luego dividir los posibles valores del campo en dos subconjuntos y asignar a cada documento al subconjunto correspondiente. Esto dio lugar a dos conjuntos de datos (re0 y re1), de 1504 y 1657 documentos respectivamente.

Las principales razones para usar estos documentos para las pruebas experimentales fueron:

La colección “Reuters 21578” es un estándar para la prueba de algoritmos de categorización, por lo que los resultados obtenidos tendrán mucha más validez que si los experimentos se realizaran sobre un conjunto de datos recopilado sin seguir una metodología estándar.

En los subconjuntos re0 y re1 extraídos de esta colección [Zhao *et.al.*, 2001] ya se han filtrado aquellos documentos sin clasificar, o con más de una clasificación (que complican innecesariamente la evaluación de los resultados) y son más fáciles de manipular que la colección completa.

Los documentos de los subconjuntos re0 y re1 ya se han preprocesado utilizando técnicas estándar. Primero, se han removido de cada documento las palabras comunes que no sirven para el proceso de categorización, utilizando una “stop list”, y el resto de las palabras fueron llevadas a su raíz utilizando el algoritmo de Porter [Porter, 1980].

5.2 VARIABLES A OBSERVAR

Una vez que se obtienen los agrupamientos de documentos que arroja cada algoritmo como salida, debe medirse cuantitativamente la calidad de cada uno para poder compararlos.

5.2.1 Variables independientes

Los siguientes son los parámetros que pueden variarse con cada muestra de datos y cada corrida de los algoritmos:

- Cantidad de documentos (N): es la cantidad de documentos que contiene la muestra, y que se van a categorizar.

- Cantidad de grupos (K): es la cantidad de grupos que va a formar cada algoritmo con los documentos de la muestra.

5.2.2 Variables dependientes

A continuación se detallan las diversas medidas que se utilizaron para evaluar los agrupamientos.

5.2.2.1 Similitud promedio

Esta es una medida "interna", ya que puede evaluarse sin conocimientos externos (no hace falta conocer la categorización "real" de los documentos). En el capítulo 2 ("Estado del arte"), sección 2.3, se describen las distintas formas de medir la similitud entre dos documentos. Tal como se explica en esa sección, el uso del coeficiente del coseno extendido es uno de los más difundidos, y es el que se utilizará para medir la similitud entre dos documentos dados.

La similitud promedio (*sim_prom*) de un grupo es, justamente, la que surge de promediar la similitud existente entre todos los documentos (*d*) del grupo tomados de a pares. Por ejemplo, para el grupo *j*, que tiene n_j elementos, utilizando el coeficiente del coseno extendido como medida de similitud,

$$sim\_prom_j = \sum_{\substack{i=1 \\ j=1}}^{n_j} \frac{d_i \bullet d_j}{\|d_i\| * \|d_j\|} \quad \text{(Fórmula 5.1)}$$

La similitud promedio de todo el agrupamiento se obtiene sumando la similitud promedio de cada grupo multiplicada por la cantidad de elementos del grupo, y dividiendo el total por la cantidad total de elementos de la muestra,

$$sim\_prom = \frac{\sum_{j=1}^{k} n_j * sim\_prom_j}{N} \quad \text{(Fórmula 5.2)}$$

De esta manera se pondera la similitud promedio de cada grupo asignándole un peso acorde con la cantidad de elementos que contiene. Esta ponderación permite que un grupo grande con baja similitud promedio impacte negativamente en la evaluación del agrupamiento y no sea compensado tan fácilmente por un grupo pequeño. Si no se realizara esta ponderación, una muestra de 1.000 documentos que se dividiera en un grupo de 900 elementos con muy baja similitud promedio y 10 grupos compactos de 10 elementos cada uno, podría sumar mejor similitud promedio que un agrupamiento balanceado de 11 grupos con buena similitud promedio en cada uno.

Esta medida da cuenta de cuánto tienen en común los elementos de cada grupo. Cuanto más homogéneo sea cada grupo, mayor valor tendrá este parámetro. Valores más grandes de similitud promedio indican una mejor calidad del agrupamiento.

5.2.2.2 Entropía

La entropía es una medida "externa" (para calcularla es necesario conocer a qué categoría "real" pertenece cada documento). El concepto de entropía tiene su origen en el campo de la física (más específicamente, en el área de la termodinámica), y es utilizada como medida del grado de desorden de un sistema [Carter, 2000]. En 1948, Shannon [Shannon, 1948] incorpora el término a la teoría de la información como una función que mide la cantidad de información generada por un proceso.

Un ejemplo simple permitirá entender qué es lo que mide la entropía y cómo puede utilizarse para medir la calidad de un agrupamiento dado. Supóngase que se tiene un grupo con 1000 documentos de 8 categorías distintas. Una persona debe tomar cada documento, y escribir un código (en forma binaria) para describir la categoría a la cual pertenece. La forma más simple de hacerlo sería escribir, para cada documento, el

número de la categoría (de 0 a 7), en base 2. Para esto, necesitaría 3 bits por cada documento (en base 2 se requieren 3 bits para codificar los números 0 a 7).

Supóngase ahora que los documentos del grupo no pertenecen a cada categoría en cantidades iguales, sino que el 42% de los documentos son de la primer categoría, el 40% son de la segunda categoría, y el 18% restante pertenece a las otras 6 categorías en cantidades iguales. La persona encargada de anotar la categoría de cada documento podría desarrollar una codificación más eficiente (la forma simple, descrita anteriormente le insumía 3 bits por cada uno). Por ejemplo, podría adoptar el código mostrado en la tabla 5.1.

Puede calcularse fácilmente, que, al utilizar solamente 2 bits en el 82% de los casos, y 4 bits en el 18% de los casos restantes, estará utilizando en promedio 2,36 bits por cada documento. Aplicada a una situación como la descrita anteriormente, la entropía mide justamente la mínima cantidad de bits promedio para codificar la categoría de cada documento del grupo.

CATEGORÍA	CÓDIGO	PROBABILIDAD
0	00	42%
1	01	40%
2	1000	3%
3	1001	3%
4	1010	3%
5	1011	3%
6	1100	3%
7	1101	3%

Tabla 5.1. Ejemplo de codificación de categorías

Si se tienen *h* categorías y p_i es la probabilidad de que un documento sea de la categoría *i*, la formula para calcular la entropía en una situación como ésta es, $H = -\sum_{i=0}^{h} p_i * \log(p_i)$. La máxima entropía se obtiene cuando los documentos pertenecen en cantidades iguales a las categorías (en ese caso, H = 3, que es el caso de la codificación en base 2 del número de categoría). En el caso que se describe en la tabla 5.1, el valor de H es aproximadamente 1,96, lo que indica que la codificación elegida no es la óptima. Cuanto más desparejas sean las probabilidades, el valor de la entropía irá disminuyendo. Si todos los documentos del grupo fueran de una sola categoría, el sistema estará totalmente ordenado, y la entropía será igual a 0 (no hará falta escribir de qué categoría es cada documento).

Para medir la calidad de un agrupamiento utilizando la entropía, primero se calcula entropía de cada grupo. Para cada categoría "real" *i*, se calcula la probabilidad p_{ij} de que un miembro del grupo *j* sea de la categoría *i*, $p_{ij} = n_{ij} / n_j$, donde n_{ij} es la cantidad de documentos de la categoría *i* que se encuentran en el grupo *j* y n_j es la cantidad de documentos del grupo *j*. La entropía de cada grupo se calcula usando la fórmula:

$H_j = -\sum_i p_{ij} * \log(p_{ij})$, donde *i* recorre todas las categorías "reales" de los documentos.

La entropía total del agrupamiento se calcula luego ponderando la entropía de cada grupo de acuerdo a su tamaño,

$$H = \frac{\sum_{j=1}^{k} n_j * H_j}{N}$$ (Fórmula 5.3)

La entropía tendrá un valor máximo cuando los documentos de cada categoría estén distribuidos uniformemente entre los grupos, y un valor igual a 0 cuando cada categoría tenga sus documentos en un sólo grupo. Valores más chicos de entropía indican una mejor calidad del agrupamiento.

5.3 REALIZACIÓN DE LOS EXPERIMENTOS

Los algoritmos que se compararon fueron:

- Bisecting K-Means con refinamiento (capítulo 2, sección 2.6).
- Algoritmo Genético (capítulo 2, sección 2.7).
- Algoritmo Genético con refinamiento (capítulo 2, sección 2.7.9.11).
- Mapas Auto-organizados de Kohonen (capítulo 2, sección 2.8)

5.3.1 Metodología utilizada

Se realizaron dos tipos de experimentos. El primero consistió en tomar muestras de datos con una cantidad de documentos fija, y realizar corridas de los algoritmos variando la cantidad de grupos a formar. La segunda clase de experimentos consistió en tomar muestras de datos con diferentes cantidades de documentos, y realizar corridas de los algoritmos formando una cantidad fija de grupos. El análisis estadístico de los resultados (aplicando el test de Wilcoxon, que se detalla en el apéndice 1) se realizó sobre los resultados del primer tipo de experimentos, ya que sus valores son más fácilmente comparables por provenir de corridas sobre muestras con la misma cantidad de documentos. El segundo tipo de experimentos se utilizó para comprobar y validar las conclusiones que se extrajeron del análisis de los resultados del primer tipo de experimentos.

5.3.1.1 Experimentos variando la cantidad de grupos

La forma más simple de realizar los experimentos hubiera sido tomar cada subconjunto (re0 y re1) como una muestra de datos, aplicar cada uno de los algoritmos a evaluar, y comparar los resultados. Sin embargo, se hubieran obtenido solamente dos conjuntos de resultados, lo que hubiera hecho muy difícil un análisis estadístico de los mismos. El rendimiento de los algoritmos se hubiera comparado

basándose solamente en la categorización de dos muestras, que podrían haber sido casos favorables para alguno de ellos, quitándole validez a las conclusiones.

Con el objetivo de obtener resultados que fueran válidos desde un punto de vista estadístico, tratando de minimizar las probabilidades de que un caso particular favorable a uno u otro algoritmo influyera en los resultados en forma excesiva, se prepararon las muestras de datos de la siguiente manera: de cada subconjunto de datos (re0 y re1) se extrajeron 10 muestras de 500 documentos al azar, y se realizaron 5 corridas de cada algoritmo sobre cada muestra. De esta manera, los resultados particulares que no reflejan el comportamiento estadístico de los algoritmos tienen menos posibilidades de distorsionar los resultados.

Para cada una de las muestras de datos, se obtuvieron agrupamientos de 5, 10, 15 y 20 grupos con cada algoritmo y se calcularon las medidas de evaluación para cada uno. El procedimiento se repitió 5 veces, y se promediaron las medidas de evaluación de las 5 iteraciones. Esto dio como resultado 20 tablas (una para cada muestra de datos), con las medidas de evaluación de cada algoritmo para los agrupamientos de 5, 10, 15 y 20 grupos. Promediando los valores de las 20 tablas, se construyó una tabla de promedios generales, con el valor promedio de cada medida de evaluación para cada algoritmo, para los agrupamientos de 5, 10, 15 y 20 grupos.

Para la confección de los gráficos, se utilizó la tabla de promedios generales. Para la realización del test de Wilcoxon, se utilizaron los valores cada una de las 20 tablas, ya que cada una de ellas se corresponde a una muestra de datos.

5.3.1.2 Experimentos variando la cantidad de documentos

Se tomaron muestras al azar de 400, 700, 1000 y 1300 documentos del primer conjunto de datos (re0). Para cada tamaño se tomaron 3 muestras de datos, y se realizaron 5 corridas de cada algoritmo, formando agrupamientos de 10 grupos. Para la confección de los gráficos se promediaron los valores para las 5 corridas de cada algoritmo sobre cada muestra de datos.

5.3.2 Entrenamiento del mapa auto-organizado de Kohonen

Se realizarán pruebas con dos niveles de entrenamiento (bajo y alto) para verificar que a mayor entrenamiento del mapa, mejores son los resultados obtenidos. A efectos de la presente experimentación se define el nivel bajo de entrenamiento a un entrenamiento de cien iteraciones y nivel alto, a uno de mil iteraciones. Ver sección 2.6.3.

5.4 RESULTADOS

5.4.1 Experimentos variando la cantidad de grupos

En esta sección se presentan los resultados que se obtuvieron en los experimentos realizados variando la cantidad de grupos. Para seguir el comportamiento de cada una de las variables independientes al ir variando la cantidad de grupos, se confeccionaron gráficos con los promedios generales para cada algoritmo. En cada gráfico, el eje “x” representa la cantidad de grupos en que se fueron dividiendo las muestras de datos con los algoritmos. El eje “y” representa la variable independiente que se mide en cada gráfico. El título del gráfico lleva el nombre de la variable independiente. Los valores que se grafican son los promedios generales para cada algoritmo.

Además, se utilizó el test de Wilcoxon sobre cada una de las variables para obtener un análisis estadístico de los resultados.

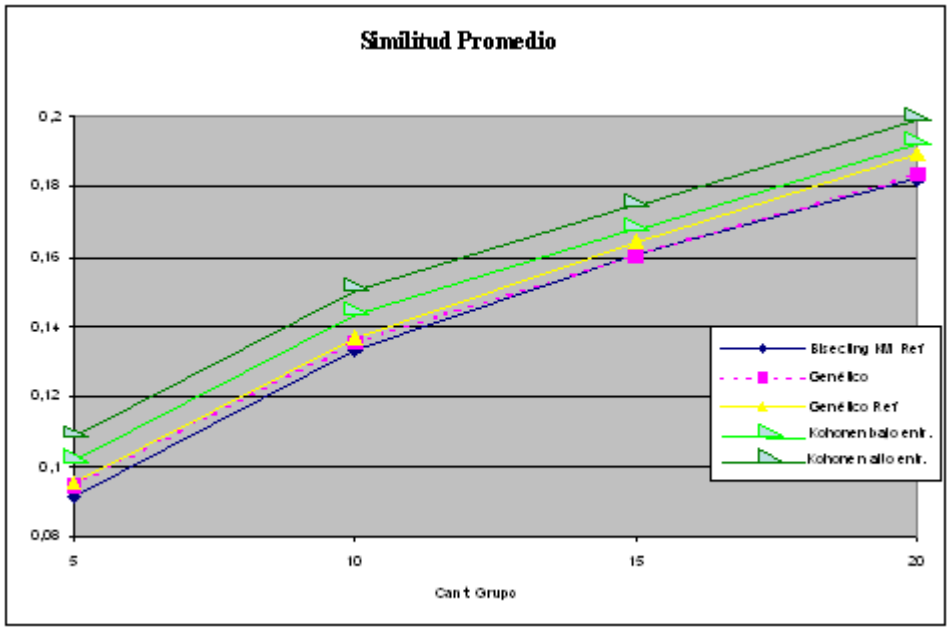


Figura 5.1. Similitud promedio de los agrupamientos encontrados en función de la cantidad de grupos.

La figura 5.1 muestra la similitud promedio de los agrupamientos encontrados en función de la cantidad de grupos armados. La curva de resultados del algoritmo “Genético” se encuentra ligeramente por encima de la curva del algoritmo “Bisecting K-Means con refinamiento”, mientras que la curva para el algoritmo “Genético con refinamiento” supera claramente a la del algoritmo “Bisecting K-Means con refinamiento”. A su vez, la curva del mapa auto-organizado de Kohonen con bajo nivel de entrenamiento se encuentra por encima de la curva del algoritmo “Genético con refinamiento”, mientras que la curva del mapa auto-organizado de Kohonen con alto nivel de entrenamiento se encuentra por encima de la curva del mapa auto-organizado de Kohonen con bajo nivel de entrenamiento. Esta apreciación es consistente con los resultados que arroja el test de Wilcoxon para esta variable (que se detalla en el apéndice 1). Tomando en cuenta los resultados de las 20 muestras, el test permite afirmar con un margen de confianza del 95% que el promedio de valores de similitud promedio para los algoritmos “Bisecting K-Means con refinamiento”, “Genético” y “Genético con refinamiento” es inferior al del mapa auto-organizado de Kohonen, ya sea con bajo o con alto nivel de entrenamiento.

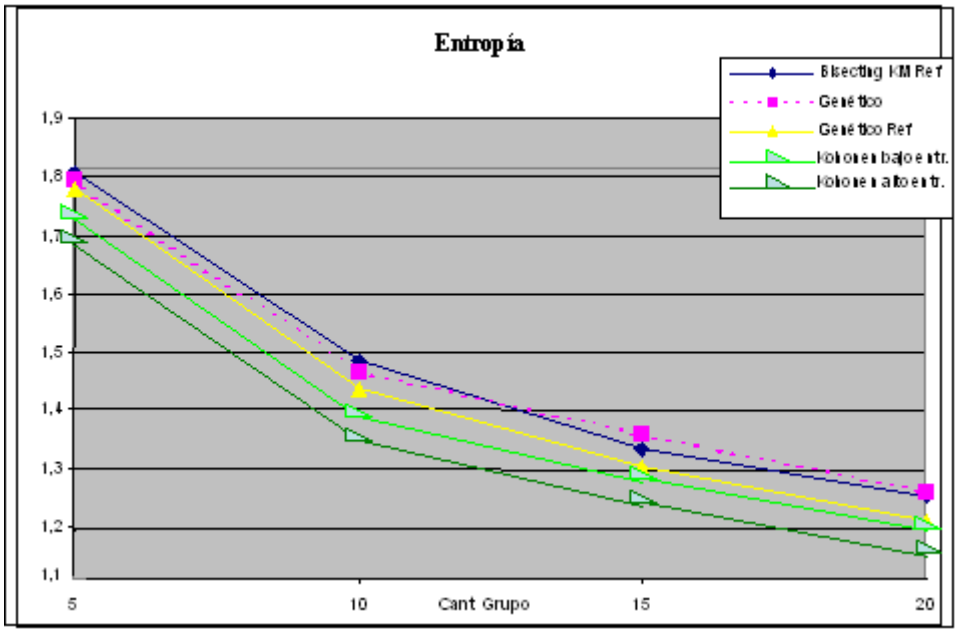


Figura 5.2. Entropía de los agrupamientos encontrados en función de la cantidad de grupos.

La figura 5.2 muestra la entropía de los agrupamientos encontrados en función de la cantidad de grupos armados. Las curvas de resultados de los algoritmos “Genético” y “Bisecting K-Means con refinamiento” son bastante similares, y tienen diferencias en uno y otro sentido que no permiten afirmar que una sea mejor que la otra, mientras que la curva para el algoritmo “Genético con refinamiento” está claramente por debajo de ambas. A su vez, la curva de resultados de los mapas auto-organizados de Kohonen con bajo nivel de entrenamiento está por debajo de la curva del algoritmo “Genético con refinamiento” mientras que la curva de los mapas auto-organizados de Kohonen con alto nivel de entrenamiento está por debajo de la curva de los mapas auto-organizados de Kohonen con alto nivel de entrenamiento. Esta apreciación es consistente con los resultados que arroja el test de Wilcoxon para esta variable. Tomando en cuenta los resultados de las 20 muestras, el test no puede distinguir si los valores de entropía son diferentes para los algoritmos “Genético” y “Bisecting K-Means con refinamiento”, mientras que permite afirmar con un grado de confianza de 95% que los valores para los mapas auto-organizados de Kohonen con alto y bajo entrenamiento son menores que para los tres algoritmos.

5.4.2 Experimentos variando la cantidad de documentos

Se graficó la evolución de las variables independientes para cada algoritmo al variar la cantidad de documentos a agrupar con el fin de comprobar si su comportamiento seguía las tendencias encontradas durante la realización del primer tipo de experimento. En cada gráfico, el eje “x” representa la cantidad de documentos de las muestras que se agruparon. El eje “y” representa los valores de la variable independiente que se está graficando. Cada gráfico lleva como título el nombre de la variable independiente que se grafica.

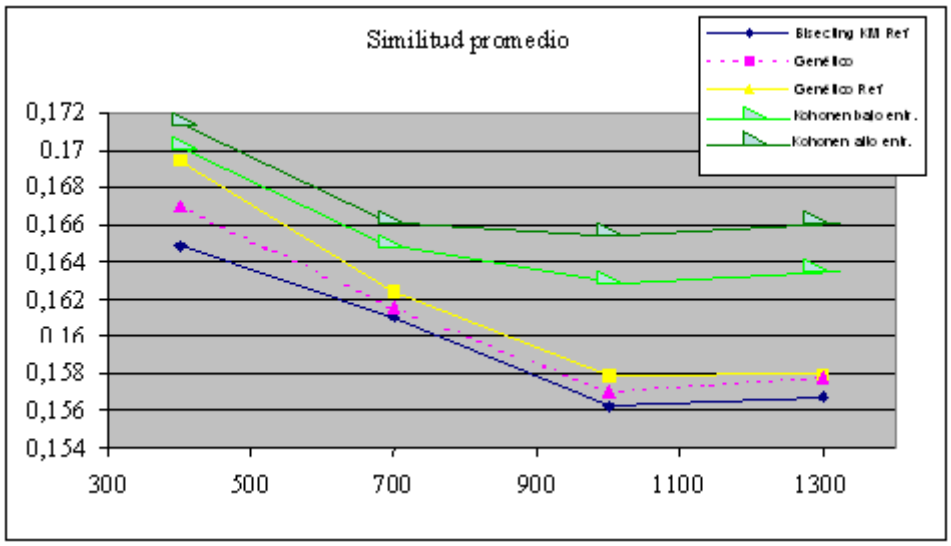


Figura 5.3. Similitud promedio de los agrupamientos encontrados en función de la cantidad de documentos.

La figura 5.3 muestra la evolución de la similitud promedio de los agrupamientos encontrados al variar la cantidad de documentos. Puede observarse que se confirma la tendencia encontrada. La curva de similitud promedio para el algoritmo “Bisecting K-Means con refinamiento” se encuentra por debajo de las curvas para los algoritmos “Genético” y “Genético con refinamiento” que, a su vez, se encuentran por debajo de las curvas para los mapas auto-organizados de Kohonen con bajo y alto nivel de entrenamiento..

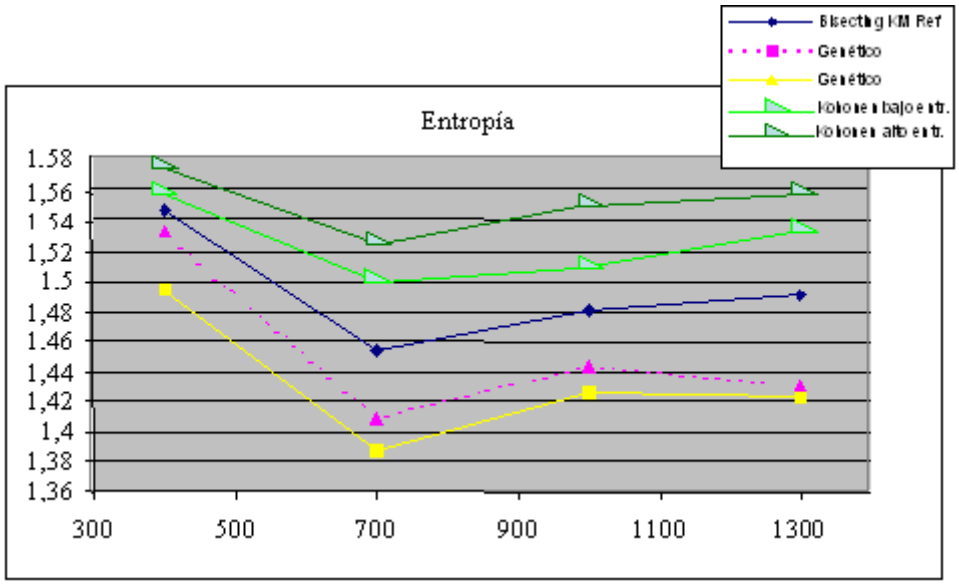


Figura 5.4. Entropía de los agrupamientos encontrados en función de la cantidad de documentos.

La evolución de la entropía con la cantidad de grupos (figura 5.4) muestra el mismo comportamiento observado en el primer tipo de experimento, aunque en este gráfico la entropía para los agrupamientos encontrados por el algoritmo “Genético” es menor que para el algoritmo “Bisecting K-Means con refinamiento”, aunque esto puede

deberse a particularidades de las muestras utilizadas, ya que el análisis estadístico realizado con los resultados del primer tipo de experimento no permitían afirmar esto.

5.5 ANÁLISIS DE LOS RESULTADOS

A partir de los gráficos y el análisis estadístico de los resultados, puede responderse en base a los resultados experimentales la cuestión 1 planteada en el capítulo 3 (“Descripción del problema”).

Cuestión 1: ¿Pueden ayudar los mapas auto-organizados de Kohonen a encontrar soluciones de mejor calidad?

Los resultados muestran que las soluciones encontradas por los mapas auto-organizados de Kohonen son de mayor calidad que las halladas con los algoritmos “Genético” y “Genético con refinamiento” y “Bisecting K-Means con refinamiento” tomados como referencia, para las medidas de calidad definidas.

CAPÍTULO 6

CONCLUSIONES

En esta tesis se propone una adaptación de un mapa auto-organizado de Kohonen al problema de la categorización automática de documentos, que incluye además la búsqueda de documentos por palabra clave. Los resultados experimentales obtenidos confirman la tesis que los mapas auto-organizados de Kohonen son una poderosa herramienta para la resolución de problemas en los cuales el espacio de soluciones es amplio y la función de optimización es compleja.

Se ha encontrado también que la calidad de la solución obtenida por los mapas auto-organizados de Kohonen depende, entre otras cosas, del entrenamiento que el mapa recibe. El mapa propuesto logra resultados efectivos porque se lo ha entrenado en un nivel tal que éstos se han conseguido.

6.1 RESPUESTA A LAS CUESTIONES PLANTEADAS

Cuestión 1: ¿Pueden ayudar los mapas auto-organizados de Kohonen a encontrar soluciones de mejor calidad?

Las soluciones halladas por el mapa auto-organizado de Kohonen propuesto son de mayor calidad (de acuerdo a las medidas definidas) que las de los algoritmos “Genético y Genético con refinamiento” y “Bisecting K-Means con refinamiento”, según lo muestran los resultados expuestos en el capítulo 5. El algoritmo “Bisecting K-Means” no puede llegar a ninguna solución que le obligue a pasar por un punto en el que disminuya el criterio de optimización (siempre elige la división que más hace crecer el criterio de optimización). Esto hace que haya regiones del espacio de búsqueda a las que le pueda resultar difícil llegar.

Por otra parte, el algoritmo “Genético” es capaz de generar soluciones de menor calidad como parte del proceso, y extraer características positivas de ellas. De esta manera, el algoritmo genético nunca se encuentra restringido a una región del espacio de búsqueda. Los elementos de azar que intervienen en la cruza y la mutación pueden llegar a generar agrupamientos en cualquier punto del espacio de búsqueda.

Finalmente, los mapas auto-organizados de Kohonen generan soluciones de mejor calidad que los algoritmos anteriores y su calidad aumenta en la medida en que aumenta el nivel de entrenamiento.

Cuestión 2: ¿Pueden ayudar los mapas auto-organizados de Kohonen a categorizar los resultados de una búsqueda por palabra clave?

Los resultados muestran que los mapas auto-organizados de Kohonen conforman una herramienta adecuada para categorizar los resultados de una búsqueda por palabra clave.

Cuestión 3: ¿Se pueden conseguir mejores resultados proporcionando al mapa auto-organizado de Kohonen mayor entrenamiento?

Los resultados muestran que a mayor nivel de entrenamiento, de mejor calidad serán los resultados obtenidos.

6.2 LÍNEAS FUTURAS DE INVESTIGACIÓN

Se propone como posibilidad para futuras investigaciones la posibilidad de extender el presente sistema de modo que pueda ser ejecutado en plataformas Unix. Esto es en principio posible de realizar debido a que el mapa auto-organizado de Kohonen utilizado puede también ser ejecutado en Unix.

APÉNDICE 1

ANÁLISIS ESTADÍSTICO DE LOS RESULTADOS

Este apéndice detalla el análisis estadístico hecho sobre los resultados que se exponen en el capítulo 5, y que soportan las afirmaciones realizadas en la sección 5.4 ("Resultados") de ese capítulo.

Las nociones teóricas de probabilidad y estadística fueron extraídas principalmente de [Canavos, 1984], pero son las mismas que pueden encontrarse en cualquier texto relativo al tema.

A1.1 PRUEBA DE HIPÓTESIS ESTADÍSTICAS

En todas las ramas de la ciencia, cuando un investigador hace una afirmación con respecto a un fenómeno (que puede estar basada en su intuición, o en algún desarrollo teórico que parece demostrarla), debe luego probar la misma mediante la realización de experimentos. La experimentación consiste en armar un ambiente de prueba en el que ocurra el fenómeno (o buscarlo en el ambiente real), y tomar mediciones de las variables involucradas. Luego, se realizan análisis estadísticos de los resultados para determinar si los mismos confirman la afirmación realizada.

Una hipótesis estadística es una afirmación con respecto a una característica desconocida de una población de interés. La esencia de probar una hipótesis estadística es el decidir si la afirmación se encuentra apoyada por la evidencia experimental que se obtiene a través de una muestra aleatoria.

Supongamos, por ejemplo, que los fabricantes de tubos de luz marca ACME están teniendo algunos problemas en el mercado debido a algunos casos de mala calidad de sus productos. Para recuperar su prestigio, hacen la siguiente afirmación: "El promedio de vida útil de los tubos de luz marca ACME es de 500 horas". Y encargan a una firma independiente que haga una serie de experimentos para contrastar esta afirmación con esta otra: "El promedio de vida útil de los tubos de luz marca ACME es menor a 500 horas".

A la afirmación "promedio = 500" se la llama *hipótesis nula*, y se escribe como:

H_0: promedio = 500

A la afirmación "promedio < 500" se la llama *hipótesis alternativa*, y se escribe como:

H_1: promedio < 500

La hipótesis nula debe considerarse verdadera a menos que exista suficiente evidencia en su contra. Es decir, se rechazará la afirmación de que la vida útil promedio es de 500 horas sólo si la evidencia experimental se encuentra muy en contra de ésta afirmación. En caso contrario, no se podrá rechazar la afirmación basándose en la evidencia experimental. Debe notarse que no poder rechazar la afirmación no es lo mismo que aceptarla. El caso es análogo al de un juicio, donde hay un sospechoso acusado de un crimen: si la evidencia es suficiente, se lo declarará

culpable. De lo contrario, se dirá que la evidencia no alcanza para demostrar su culpabilidad.

Existen entonces dos posibles decisiones con respecto a la hipótesis nula: rechazarla ó no poder rechazarla. A su vez, la hipótesis nula puede ser verdadera o falsa. Esto deja cuatro posibles escenarios, que se ven en la figura A1.1.

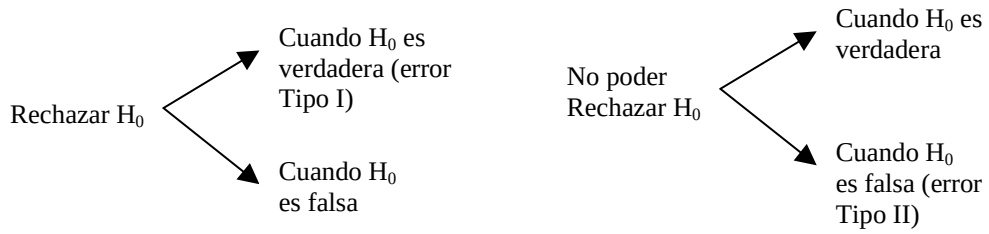


Figura A1.1. Escenarios posibles en la toma de decisiones

Se denomina α a la probabilidad de cometer un error de tipo I, y β a la probabilidad de cometer un error de tipo II. Los valores de α y β son interdependientes. Para cada experimento, al disminuir uno de ellos, aumenta el otro. En general, el error de tipo I se considera más grave que el de tipo II (volviendo a la analogía con el juicio, se prefiere dejar ir a un culpable y no condenar a un inocente), por lo que el procedimiento seguido habitualmente consiste en fijar un valor pequeño para α (por ejemplo, 5%), y luego tratar de minimizar β lo más que se pueda.

A1.2 EL TEST DE WILCOXON PARA LA COMPARACIÓN DE MEDIAS DE MUESTRAS APAREADAS

A1.2.1 Introducción

Existen numerosos métodos para la prueba de hipótesis estadísticas. El hecho de que cada uno de ellos pueda aplicarse a una situación en particular depende de varias condiciones, por ejemplo:

- De la cantidad de mediciones realizadas
- De los valores a analizar (si son valores en un intervalo numérico, si son categorías cualitativas, si son del tipo SI/NO, etc.)
- De la dependencia que exista o no entre las mediciones

Los experimentos realizados para comparar los algoritmos de categorización automática de documentos, son un caso que se denomina “de muestras apareadas”, en el cual se miden variables numéricas. Para estos casos, el test de Wilcoxon es el más apropiado [Canavos, 1984].

El término “muestras apareadas” se refiere a que las mediciones realizadas no son independientes, sino que son tomadas de a pares (la muestra de datos 1 es categorizada con los algoritmos 1 y 2). Esto hace que lo que deba analizarse sean las

diferencias que existen en cada par de valores, que es precisamente lo que hace el test de Wilcoxon.

A1.2.2 Descripción del test

Los experimentos para comparar dos algoritmos de categorización, con respecto a una medida de calidad, se realizan de la siguiente forma:

- Se toman N muestras de datos
- Se categorizan las muestras de datos con los tres algoritmos y el mapa auto-organizado de Kohonen con bajo y con alto nivel de entrenamiento
- Se mide una variable numérica que indica la calidad del agrupamiento

Luego de la realización de los experimentos, se confecciona una tabla, similar a la que se presenta a continuacion (La tabla A1.1 es un ejemplo para 4 muestras y dos algoritmos):

Muestra	Algoritmo 1	Algoritmo 2	Diferencia (1 -2)	Ranking	Ranking con signo
1	0,79	0,80	-0,01	1	-1
2	0,46	0,51	-0,05	4	-4
3	0,91	0,87	0,04	3	3
4	0,23	0,25	-0,02	2	-2

Tabla A1.1. Ejemplo de toma de valores en los experimentos

Como se ve, los valores pueden tener grandes variaciones de muestra a muestra, pero lo que importa es la diferencia entre los valores de cada algoritmo para cada muestra, ya que eso es lo que indicará el mejor o peor rendimiento de cada uno.

La hipótesis nula que es puesta a prueba es que el promedio de los valores es igual para los dos algoritmos (es decir, que las calidades de los agrupamientos producidos es equivalente). Se plantean dos hipótesis alternativas, una de ellas afirma que el promedio de los valores es mayor para el algoritmo 1, y la otra que el promedio de los valores es mayor para el algoritmo 2.

La metodología del test es la siguiente:

- Se calculan las diferencias de los valores para cada muestra
- Luego, se asigna a cada diferencia un valor en un ranking (de menor a mayor) en base a su valor absoluto
- Por último, a cada valor del ranking se le asigna el signo de la diferencia que le dio origen.

Se denomina T+ a la suma de los valores positivos y T- a la suma de los negativos.

Si no hubiera diferencias entre los algoritmos, es de esperar que T+ resulte igual a T- (en valor absoluto).

Para muestras lo suficientemente grandes (N > 15), la variable T+ puede aproximarse por medio de una distribución normal con media E(T+) y varianza Var(T+), donde

$$E(T+) = \frac{N(N+1)}{4}$$ (Fórmula A1.1)

$$Var(T+) = \frac{N(N+1)(2N+1)}{24}$$ (Fórmula A1.2)

Luego, si se define la transformación

$$zT+ = \frac{(T+) - E(T+)}{\sqrt{Var(T+)}}$$ (Fórmula A1.3)

La variable zT+ tiene una distribución normal estándar (media igual a 0 y varianza igual a 1).

El valor del parámetro α determina los límites mínimo y máximo para el valor observado de zT+, más allá de los cuales se rechaza la hipótesis nula.

Si el valor de zT+ es superior al límite máximo, se aceptará la hipótesis alternativa de que el promedio de valores para el algoritmo 1 es mayor que para el algoritmo 2.

Si el valor de zT+ es inferior al límite mínimo, se aceptará la hipótesis alternativa de que el promedio de valores para el algoritmo 2 es mayor que para el algoritmo 1.

A1.3 APLICACIÓN DEL TEST A LOS RESULTADOS

Tal como se describe en el capítulo 5, sección 5.3.1 (“Metodología Utilizada”), se utilizaron 20 muestras de datos. Cada una de ellas se procesó con cada algoritmo y cada mapa auto-organizado de Kohonen para producir agrupamientos de 5, 10, 15 y 20 grupos.

Esto da un total de 20 * 4 = 80 mediciones realizadas, por lo que N = 80.

El valor de α utilizado es de 5%, α = 0,05. Esto quiere decir que, en los casos en que rechacemos la hipótesis nula y aceptemos alguna hipótesis alternativa, el test nos dará un 95% de confianza.

Teniendo N y α, quedan definidos los límites mínimo y máximo para zT+, que son respectivamente -1,645 y 1,645.

Para cada variable se comparó a los algoritmos “Bisecting K-Means con refinamiento”, “Genético” y “Genético con refinamiento” con los mapas auto-organizados de Kohonen con bajo y con alto nivel de entrenamiento..

A1.3.1 Similitud promedio

Debe recordarse que valores más grandes para la similitud promedio indican una mejor calidad del agrupamiento.

A1.3.1.1 “Bisecting K-Means con refinamiento” contra mapa auto-organizado de Kohonen con bajo nivel de entrenamiento

Valor de zT+ observado: -10,765

Consecuencia: Debe rechazarse la hipótesis nula y aceptarse la hipótesis alternativa de que los valores para el mapa auto-organizado de Kohonen con bajo nivel de entrenamiento son mayores que para el algoritmo “Bisecting K-Means con refinamiento”.

A1.3.1.2 “Bisecting K-Means con refinamiento” contra mapa auto-organizado de Kohonen con alto nivel de entrenamiento

Valor de zT+ observado: -12,747

Consecuencia: Debe rechazarse la hipótesis nula y aceptarse la hipótesis alternativa de que los valores para el mapa auto-organizado de Kohonen con alto nivel de entrenamiento son mayores que para el algoritmo “Bisecting K-Means con refinamiento”.

A1.3.1.3 “Genético” contra mapa auto-organizado de Kohonen con bajo nivel de entrenamiento

Valor de zT+ observado: -10,298

Consecuencia: Debe rechazarse la hipótesis nula y aceptarse la hipótesis alternativa de que los valores para el mapa auto-organizado de Kohonen con bajo nivel de entrenamiento son mayores que para el algoritmo “Genético”.

A1.3.1.4 “Genético” contra mapa auto-organizado de Kohonen con alto nivel de entrenamiento

Valor de zT+ observado: -12,128

Consecuencia: Debe rechazarse la hipótesis nula y aceptarse la hipótesis alternativa de que los valores para el mapa auto-organizado de Kohonen con alto nivel de entrenamiento son mayores que para el algoritmo “Genético”.

A1.3.1.5 "Genético con refinamiento" contra mapa auto-organizado de Kohonen con bajo nivel de entrenamiento

Valor de zT+ observado: -4,829

Consecuencia: Debe rechazarse la hipótesis nula y aceptarse la hipótesis alternativa de que los valores para el mapa auto-organizado de Kohonen con bajo nivel de entrenamiento son mayores que para el algoritmo "Genético con refinamiento".

A1.3.1.6 "Genético con refinamiento" contra mapa auto-organizado de Kohonen con alto nivel de entrenamiento

Valor de zT+ observado: -6,845

Consecuencia: Debe rechazarse la hipótesis nula y aceptarse la hipótesis alternativa de que los valores para el mapa auto-organizado de Kohonen con alto nivel de entrenamiento son mayores que para el algoritmo "Genético con refinamiento".

A1.3.1.7 Mapa auto-organizado de Kohonen con bajo nivel de entrenamiento contra mapa auto-organizado de Kohonen con alto nivel de entrenamiento

Valor de zT+ observado: -3,773

Consecuencia: Debe rechazarse la hipótesis nula y aceptarse la hipótesis alternativa de que los valores para el mapa auto-organizado de Kohonen con alto nivel de entrenamiento son mayores que para el mapa auto-organizado de Kohonen con bajo nivel de entrenamiento.

A1.3.2 Entropía

Debe recordarse que valores más chicos para la entropía indican una mejor calidad del agrupamiento.

A1.3.2.1 "Bisecting K-Means con refinamiento" contra mapa auto-organizado de Kohonen con bajo nivel de entrenamiento

Valor de zT+ observado: -7,849

Consecuencia: Debe rechazarse la hipótesis nula y aceptarse la hipótesis alternativa de que los valores para el mapa auto-organizado de Kohonen con bajo nivel de entrenamiento son menores que para el algoritmo "Bisecting K-Means con refinamiento".

A1.3.2.2 “Bisecting K-Means con refinamiento” contra mapa auto-organizado de Kohonen con alto nivel de entrenamiento

Valor de zT+ observado: -10,655

Consecuencia: Debe rechazarse la hipótesis nula y aceptarse la hipótesis alternativa de que los valores para el mapa auto-organizado de Kohonen con alto nivel de entrenamiento son menores que para el algoritmo “Bisecting K-Means con refinamiento”.

A1.3.2.3 “Genético” contra mapa auto-organizado de Kohonen con bajo nivel de entrenamiento

Valor de zT+ observado: -7,734

Consecuencia: Debe rechazarse la hipótesis nula y aceptarse la hipótesis alternativa de que los valores para el mapa auto-organizado de Kohonen con bajo nivel de entrenamiento son menores que para el algoritmo “Genético”.

A1.3.2.4 “Genético” contra mapa auto-organizado de Kohonen con alto nivel de entrenamiento

Valor de zT+ observado: -10,574

Consecuencia: Debe rechazarse la hipótesis nula y aceptarse la hipótesis alternativa de que los valores para el mapa auto-organizado de Kohonen con alto nivel de entrenamiento son menores que para el algoritmo “Genético”.

A1.3.2.5 “Genético con refinamiento” contra mapa auto-organizado de Kohonen con bajo nivel de entrenamiento

Valor de zT+ observado: -5,276

Consecuencia: Debe rechazarse la hipótesis nula y aceptarse la hipótesis alternativa de que los valores para el mapa auto-organizado de Kohonen con bajo nivel de entrenamiento son menores que para el algoritmo “Genético con refinamiento”.

A1.3.2.6 “Genético con refinamiento” contra mapa auto-organizado de Kohonen con alto nivel de entrenamiento

Valor de zT+ observado: -7,612

Consecuencia: Debe rechazarse la hipótesis nula y aceptarse la hipótesis alternativa de que los valores para el mapa auto-organizado de Kohonen con alto nivel de entrenamiento son menores que para el algoritmo “Genético con refinamiento”.

A1.3.1.7 Mapa auto-organizado de Kohonen con bajo nivel de entrenamiento contra mapa auto-organizado de Kohonen con alto nivel de entrenamiento

Valor de zT+ observado: -4,169

Consecuencia: Debe rechazarse la hipótesis nula y aceptarse la hipótesis alternativa de que los valores para el mapa auto-organizado de Kohonen con alto nivel de entrenamiento son menores que para el mapa auto-organizado de Kohonen con bajo nivel de entrenamiento.

BIBLIOGRAFÍA

ACM SIGIR. (1996). Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval, ACM Press, New York, USA.

Allen, R. B., Obry, P. y Littman, M. (1993). *An interface for navigating clustered document sets returned by queries*, Proceedings of the ACM Conference on Organizational Computing Systems.

Bezdeck, J. C., Boggavaparu, S., Hall, L. O. y Bensaid, A. (1994). *Genetic algorithm guided clustering*, in Proc. of the First IEEE Conference on Evolutionary Computation, 3440.

Bradley, P.S. y Fayyad, U.M. (1998). *Refining initial points for k-means clustering*. In J. Shavlik, editor, Proceedings of the Fifteenth International Conference on Machine Learning (ICML '98), pages 91--99, San Francisco, CA, 1998. Morgan Kaufmann

Canavos, G.C. (1984). Probabilidad y estadística, Aplicaciones y Métodos. McGraw-Hill.

Carter, T. (2000). *An introduction to information theory and entropy*. Complex Systems Summer School.

Citeseer (Research Index). The NEC Research Institute Digital Library. Sitio dedicado a la difusión de literatura científica. http://citeseer.ist.psu.edu/.

Clerking, P., Cunningham, P., Hayes, C. (2001). *Ontology Discovery for the Semantic Web Using Hierarchical Clustering*. Department of Computer Science, Trinity College, Dublin.

Cole, Rowena M. (1998). *Clustering with Genetic Algorithms*. Thesis for the degree of Master of Science, Department of Computer Science, University of Western Australia.

Croft, W. B. (1978). *Organizing and searching large files of documents*, Ph.D. Thesis, University of Cambridge.

Cutting, D. R., Karger, D. R., Pedersen, J. O. y Tukey, J. W. (1992). *Scatter/Gather: A cluster-based approach to browsing large document collections*, Proceedings of the 15th International ACM SIGIR Conference on Research and Development in Information Retrieval, páginas 318-29.

Dash, M., y Liu, H. (2001). *Efficient Hierarchical Clustering Algorithms Using Partially Overlapping Partitions*. Pacific-Asia Conference on Knowledge Discovery and Data Mining, páginas 495-506.

Dunlop, M.D. y Van Rijsbergen, C. J. (1991). *Hypermedia and free text retrieval*. RIA091 Conference, Barcelona.

Duran, B. S. y Odell, P. L. (1974). *Cluster Analysis: A survey.* Berlin. Springer-Verlag.

Estivill-Castro, V. (2000). *Hybrid Genetic Algorithms Are Better for Spatial Clustering.* Pacific Rim International Conference on Artificial Intelligence, pages 424-434.

Estivill-Castro, V. y Murray, A. (1997). *Spatial Clustering for Data Mining with Genetic Algorithms.* FIT, Technical Report, 97-10.

Everitt, Brian S. (1993). *Cluster analysis.* Halsted Press, 3ra edición.

Falkenauer, Emanuel. (1999). Evolutionary Algorithms: Applying Genetic Algorithms to Real-World Problems. Springer, New York, Pag 65--88.

Fayyad, Piatetsky-Shapiro, Smyth amd (eds.), U. (1995). Advances in Knowledge Discovery and Data Mining. AAAI/MIT Press.

Faloutsos, Christos y Oard, Douglas W. (1995). *A survey of Information Retrieval and Filtering Methods*. Technical Report CS-TR3514, Dept. of Computer Science, Univ. of Maryland.

Fasulo, Daniel. (1999). *An analysis of recent work on clustering algorithms*. Technical Report #01-03-12, Dept. of Computer Science & Engineering, University of Washington.

Goldberg, David E. (1989). *Genetic Algorithms - in Search, Optimization and Machine Learning*. Addison-Wesley Publishing Company, Inc.

Google. Motor de búsqueda de páginas en internet. http://www.google.com/.

Han, J., Kamber, M. y Tung, A.K.H. (2001). *Spatial clustering methods in data mining: A survey*. Geographic Data Mining and Knowledge Discovery, H. Miller and J. Han, editors, Taylor and Francis.

Hearst, Marti A. y Pedersen, Jan O. (1996). *Reexaminig the Cluster Hypothesis: Scatter/Gather on Retrieval Results*. Proceedings of ACM SIGIR '96, Zurich.

Hilera González J. R. y Martínez Hernando V. J. (1995). Redes Neuronales Artificiales. Fundamentos, modelos y aplicaciones. RA-MA. Madrid.

Holland, J. H. (1975). *Adaptation in natural and artificial systems*. Ann Arbor: The University of Michigan Press.

Honkela T. (1997). Self-Organizing Maps in Natural Language Processing. Helsinki University of Technology.

Honkela T. (1997): Learning to understand - general aspects of using Self-Organizing Maps in natural language processing. Proceedings of the CASYS'97, Computing Anticipatory Systems, Liège, Belgium, August, 1997, in press.

Honkela Timo (1997). Emerging categories and adaptive prototypes: Self-organizing maps for cognitive linguistics. Extended abstract, accepted to be presented in the International Cognitive Linguistics Conference, Amsterdam, July 14-19, 1997.

Honkela Timo and Vepsäläinen Ari M. (1991). Interpreting Imprecise Expressions: Experiments with Kohonen's Self-Organizing Maps and Associative Memory. Artificial Neural Networks, T. Kohonen and K. Mäkisara (eds.), vol. I, 897-902.

Honkela Timo, Pulkki Ville, and Kohonen Teuvo (1995). Contextual relations of words in Grimm tales analyzed by self-organizing map. In F. Fogelman-Soulie and P. Gallinari (eds.) ICANN-95, Proceedings of International Conference on Artificial Neural Networks, vol. 2, pp. 3-7. EC2 et Cie, Paris.

Honkela Timo, Kaski Samuel, Lagus Krista, and Kohonen Teuvo. Newsgroup Exploration with WEBSOM Method and Browsing Interface. Report A32, Helsinki University of Technology, Laboratory of Computer and Information Science, January, 1996.

Hyötyniemi, H. (1996). Text document classification with self-organizing maps. In Alander, J., Honkela, T. and Jakobsson, M. editors. Proceedings of Finnish Artificial Intelligence Conference – Genes, Nets and Symbols. Pages 64-72. Finnish Artificial Intelligence Society.

ISO/IEC 2382-1:1993 *Information technology* -- *Vocabulary* --. Part 1: Fundamental terms

Jain, A. K., Murty, M.N., y Flinn, P.J. (1999). *Data Clustering: A review*. ACM Computing Surveys, Vol. 31, Nro 3, Septiembre 1999.

Joachims, T. (1998). *Text Categorization with Support Vector Machines: Learning with Many Relevant Features*. Proceedings of ECML-98, 10th European Conference on Machine Learning, Springer Verlag, Heidelberg, DE.

Jones, G., Robertson, A.M., Santimetvirul, C. y Willet, P. (1995). *Non-hierarchical document clustering using a genetic algorithm*. Information Research, an electronic journal, Vol 1, No 1, April, 1995.

Karipys, G., Han, E.H., Kumar, V. (1999). *CHAMELEON: A hierarchical clustering algorithm using dynamic modeling*, IEEE Computer: Special Issue on Data Analysis and Mining, 32(8), 68-75.

Kaski Samuel (1997). Data Exploration Using Self-Organizing Maps. Dr.Tech thesis. Helsinki University of Technology, Espoo, Finland, Acta Polytechnica Scandinavica, no. 82.

Kaski Samuel, Honkela Timo, Lagus Krista, and Kohonen Teuvo (1996). Creating an order in digital libraries with self-organizing maps. Proceedings of WCNN'96, World Congress on Neural Networks, Lawrence Erlbaum and INNS Press, Mahwah, NJ, pp. 814-817.

Kaski Samuel, Honkela Timo, Lagus Krista, and Kohonen Teuvo (1998). WEBSOM – Self-organizing maps of document collections. Neurocomputing 21 (1998) 101-117. Helsinki University of Technology. Neural Networks Research Centre.

Kaufmann, Leonard y Rousseeuw, Peter J. (1990). *Finding Groups in data: An introduction to Cluster Analysis*, John Wiley & Sons, Inc., NY.

Kohonen T. (1982). Self-organized formation of topologically correct feature maps. Biological Cybernetics.

Kohonen T. (1995). Self-Organizing Maps. Springer-Verlag.

Kohonen Teuvo, Kaski Samuel, Lagus Krista, and Honkela Timo (1996). Very large two-level SOM for the browsing of newsgroups. Proceedings of ICANN'96, International Conference on Artificial Neural Networks.

Krovetz, R. (1993). *Viewing morphology as an inference process*. In Proceedings of ACM-SIGIR93, pages 191—203

Lagus, Krista (2001). Text retrieval using self-organized document maps. Neural Networks Research Center, Helsinki University of Technology.

Lagus Krista, Honkela Timo, Kaski Samuel, and Kohonen Teuvo (1996). Self-organizing maps of document collections: a new approach to interactive exploration. E. Simoudis, J. Han, and U. Fayyad (eds.), Proceedings of the

Second International Conference on Knowledge Discovery & Data Mining, AAAI Press, Menlo Park, CA, pp. 238-243.

Leousky, A. V. 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.

Lewis, D. (1991). *Evaluating text categorization*, Proceedings of the Speech and Natural Language Workshop, Asilomar, Morgan.

Lewis, D. (1997). *Reuters-21578 text categorization test collection*, http://www.daviddlewis.com/resources/testcollections/reuters21578/ ó http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html.

Lin, X. (1992). Visualization for the document space. In Proceedings of Visualization '92, pages 274-281. Los Alamitos, CA, USA. Center for Comput. Legal Res., Pace Univ., White Plains, NY, USA. IEEE Comput. Soc. Press.

Lin, X. (1997). Map displays for information retrieval. Journal of the American Society for Information Science. 48:40-54.

Lin, X., Soergel, D. and Marchionini, G. (1991) A self organizing semantic map for information retrieval. In Proceedings of 14th. Ann. International ACM/SIGIR Conference on Research & Development in Information Retrieval, pages 262-269.

Liu, G.L. (1968). Introduction to combinatorial mathematics, McGraw Hill.

Maarek, Yoelle S., Fagin, Ronald, Ben-Shaul, Israel Z. y Pelleg, Dan. (2000). *Ephemeral Document Clustering for Web Applications*. IBM Research Report RJ 10186.

Macskassy, S.A., Banerjee, A., Davison, B.D., Hirsh, H. (2001). *Human performance on clustering web pages*. Technical Report DCS-TR-355, Department of computer Science, Rutgers, State University of New Jersey.

McQueen, J. (1967). *Some methods for classification and analysis of multivariate observations*, 5-th Berkeley Symposium on mathematics, Statistics and Probability, 1, S. 281-298.

Merkl, D. (1993). Structuring software for reuse - the case of self organizing maps. In Proceedings of IJCNN-93-Nagoya, International Joint Conference on Neural Networks, volume III. pages 2468-2471. Piscataway, NJ. IEEE Service Center.

Merkl, D. (1994). Self-Organization of Software Libraries: An Artificial Neural Network Approach. PhD thesis, Institut für Angewandte Informatik und Informationssysteme. Universität Wien.

Merkl, D. (1995a). Content-based document classification with highly compressed input data. In Fogelman-Soulié, F. and Gallinari, P. editors. Proceedings of ICANN ’95. International Conference on Artificial Neural Networks, volume II, pages 239-244. Nanterre, France. EC2.

Merkl, D. (1995b) Content-based software classification by self-organization. In Proceedings of ICNN ’95. IEEE International Conference on Neural Networks, volume II, pages 1086-1091. Piscataway, NJ. IEEE Service Center.

Merkl, D. (1997). Lessons learned in text document classification. In Proceedings of WSOM ’97. Workshop on Self-Organizing Maps, Espoo, Finland. June 4-6. pages 316-321. Helsinki University of Technology, Neural Networks Research Centre, Espoo, Finland.

Merkl, D., Tjoa, A. M. and Kappel, G. (1994). Application of self-organizing feature maps with lateral inhibition to structure a library of reusable software components. In Proceedings of ICNN ’94. International Conference on Neural Networks. Pages 3905-3908. Piscataway, NJ. IEEE Service Center.

Miller, B.L., Goldberg, D.E. (1995). Genetic algorithms, Selection Schemes and the Varying Effects of Noise, IlliGAL report No. 95009.

Pentakalos, O., Menascé, D. y Yesha, Y. (1996). *Automated Clustering-Based Workload Characterization*. 5th NASA Goddard Mass Storage Systems and Technologies Conference.

Porter, M. F., (1980). *An Algorithm for Suffix Stripping*, Program, vol.14, no. 3, 130-137, 1980

Qin He, (1996). *A review of clustering algorithms as applied in IR*, UIUCLIS-1996/6+IGR, University of Illinois at Urbana-Champaign.

Raghavan, V., Bollmann, P., y Jung, G. (1989). *A critical investigation of recall and precision as measures of retrieval system performance*. ACM Transactions on Information Systems, 7(3):205--229.

Rasmussen, E. (1992). *Clustering Algorithms*, W. B. Frakes and R. Baeza-Yates, editors, Information Retrieval, Páginas 419-442. Prentice Hall, Eaglewood Cliffs, N. J.

Rozmus, J. (1995). Information retrieval by self-organizing maps. In Williams, M. editor. 16th National Online Meeting Proceedings - 1995, pages 349-354. Medford, NJ, USA. Smart Syst., USA, Learned Inf.

Rüger, S. M. R. y Gauch, S. E. (2000). *Feature reduction for document clustering and classification*. Technical report, Computing Department, Imperial College, London, UK.

Salton, G. (1989). Automatic Text Processing. Addison-Wesley, Reading, MA.

Salton, G., Wong, A. and Yang, C. S. (1975). A vector space model for automatic indexing. Communications of the ACM. 18(11):613-620.

Scholtes Jan C. (1993). Neural Networks in Natural Language Processing and Information Retrieval. PhD thesis, University of Amsterdam, Amsterdam.

Shannon, C.E. (1948) *A mathematical theory of communication*, Bell System Technical Journal, vol. 27, pp. 379-423 and 623-656, July and October.

Schütze, Hinrich y Silverstein Craig (1997) *Projections for Efficient Document Clustering*, in Proceedings of ACM/SIGIR'97, pp.74-81.

Steinbach, M., Karypis, G., y Kumar, V. (2000). *A comparison of Document Clustering Techniques*. Technical Report #00-034. University of Minnesota. In KDD Workshop on Text Mining.

Strehl, A., Ghosh, J. y Mooney, R. (2000). *Impact of Similarity Measures on Web-page Clustering*. AAAI-2000: Workshop of Artificial Inteligence for Web Search.

Van Rijsbergen, C. J. (1979). *Information Retrieval*, Butterworths, London, 2da edición.

Willet, P. (1998). *Recent trends in hierarchical document clustering: a critical review*. Information Processing and Management. 24:577-97.

Wilf, H.S. (1986). *Algorithms and Complexity*, Prentice Hall.

Yang, Y. (1997). *An evaluation of statistical approaches to text categorization*. School of Computer Science, Carnegie Mellon University, CMU-CS-97-127.

Yang, Y. y Pedersen J., (1997). *A comparative study on feature selection in text categorization*. Proc. of the 14th International Conference on Machine Learning ICML97, páginas 412 - 420.

Yolis, Eugenio (2003). Algoritmos genéticos aplicados a la categorización automática de documentos. Tesis de grado en Ingeniería Informática, Facultad de Ingeniería, Universidad de Buenos Aires.

Zavrel, J. (1995). Neural information retrieval - an experimental study of clustering and browsing of document collections with neural networks. Master s thesis University of Amsterdam, Amsterdam, Netherlands.

Zavrel, J. (1996). Neural navigation interfaces for information retrieval: are they more than an appealing idea? Artificial Intelligence Review. 10(5-6):477-504.

Zamir, Oren y Etzioni, Oren. (1998). *Web Document Clustering: A feasibility demonstration*. Proceedings of ACM/SIGIR'98.

Zamir, Oren y Etzioni, Oren. (1999). *Grouper: A Dynamic Clustering Interface to Web Search Results*. Proceedings of the Eighth International World Wide Web Conference, Computer Networks and ISDN Systems.

Zervas, Giorgio y Rüger, Stefan. (2000).*The curse of dimensionality and document clustering*. Dept. of Computing, Imperial College, England.

Zhao, Y. y Karypis, G., (2001). *Criterion Functions for Document Clustering*. Technical Report #01-40, Department of Computer Science, University of Minnesota.