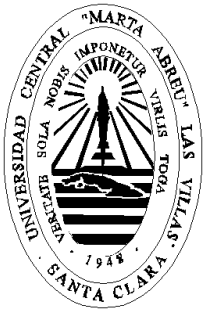
Universidad Central “Marta Abreu” de Las Villas

Facultad de Matemática, Física y Computación



Trabajo para optar por Título de Master en Ciencia de la Computación

Modelo para el agrupamiento de documentos afines y su ulterior resumen a través de la representación espacio vectorial de un corpus textual

Autora

Lic. Leticia Arco García

Tutor

Dr. Rafael Esteban Bello Pérez

Consultante

Lic. Manuel Llanes Abeijón

2005

A mis padres y abuelos

Agradezco a todas las personas que han contribuido al desarrollo de este trabajo:

A mis padres, abuelos y familia por todo el apoyo que me han brindado, especialmente a mi mamá por todos sus consejos, dedicación y paciencia para soportarme.

Al Dr. Rafael Bello, la Dra. María Matilde García, el Lic. Manuel Llanes, el Dr. Ricardo Grau y la Dra. Iliana Gutiérrez por el apoyo, las ideas, la revisión de este trabajo y el tiempo dedicado.

A Yoisy, Juan y Libernys por todo su entusiasmo en el desarrollo de esta investigación.

A Isis y a Yailé por estas siempre conmigo en los momentos difíciles.

A todos mis colegas y amigos por sus sugerencias y colaboración.

Resumen

El objetivo general de la investigación consiste en desarrollar un modelo conceptual y un procedimiento metodológico, soportado en el software CorpusMiner, que ofrece a los investigadores y desarrolladores en el campo de la minería de textos una herramienta que posibilita el agrupamiento de textos para la obtención de resúmenes extractos a partir de grupos homogéneos de documentos afines, con un enfoque de integración.

En el contenido del trabajo se expone el marco teórico-referencial de la investigación, enfatizando en las técnicas más empleadas en la actualidad para la representación de corpus textuales, el agrupamiento de documentos y la obtención de resúmenes extractos. Se desarrolla un modelo conceptual flexible que justifica la concepción y posterior aplicación de cada una de las etapas del procedimiento general propuesto: representación del corpus textual, agrupamiento de documentos, extracción de palabras claves de grupos textuales y obtención del resumen extracto de cada grupo de documentos.

Finalmente, se muestra la viabilidad del modelo desarrollado a partir de su aplicación en tres casos de estudio utilizando la herramienta CorpusMiner que lo soporta. Se evaluaron, utilizando pruebas estadísticas no paramétricas, la etapa de agrupamiento así como la fase de reducción de dimensionalidad dentro de la etapa de representación textual. Se demostró de esta forma la hipótesis de investigación planteada.

Abstract

Índice

Introducción

En la Revolución de la Información que vivimos ha surgido una tarea difícil – los humanos no estamos diseñados para procesar cantidades masivas de información y encontrar asuntos de interés. La computadora primero encontró su uso en acelerar el funcionamiento de grandes cálculos numéricos eficientemente. Ahora es necesario que las computadoras resuelvan otra incompetencia humana, analizar grandes volúmenes de datos para encontrar elementos interesantes.

La proliferación de información disponible en el World Wide Web, intranets corporativas y bases de datos, cables de noticias electrónicas, y otros medios de comunicación es arrolladora. La creación y diseminación de información es soportada por un número creciente de herramientas, sin embargo, mientras que la cantidad de información disponible está continuamente creciendo, nuestra habilidad de procesarla y asimilarla permanece constante [DIX97] [LAN01]. Es decir, que en la medida en que los usuarios finales cuenten con herramientas que permitan procesar la información, tendrán en su poder la posibilidad de extraer el conocimiento requerido para futuras tomas de decisiones.

El conocimiento es un recurso estratégico para el desarrollo económico, científico y social contemporáneo. Las tecnologías, métodos y herramientas asociadas con los procesos de adquisición, generación, gestión y transmisión del conocimiento se han desarrollado notablemente en los últimos años. Es imprescindible el empleo de técnicas y herramientas que le den sentido y utilidad a la información existente, ya que ésta es un elemento básico principal [FEB02]. Las organizaciones, investigadores y usuarios enfrentan actualmente la necesidad de forma imperiosa de contar con aplicaciones que puedan analizar y evaluar la relevancia de la información.

El procesar automáticamente grandes cantidades de datos para encontrar conocimiento útil es el objetivo principal del área de Descubrimiento de Conocimiento en Bases de Datos o KDD (*Knowledge Discovery from Data base*) y podemos definirlo como un proceso no trivial de identificar patrones válidos, novedosos, potencialmente útiles y en última instancia comprensibles a partir de datos, o como la extracción no trivial de información implícita, desconocida, y potencialmente útil de los datos [LEZ02].

La minería de datos (*Data Mining*) es una fase del KDD que integra los métodos de aprendizaje y estadísticas para obtener hipótesis de patrones y modelos. Ésta surge como las mejores herramientas para realizar exploraciones más profundas y extraer información nueva, útil y no trivial que se encuentra oculta en grandes volúmenes de datos estructurados [LEZ02].

La limitante que existe es que las técnicas de minería de datos procesan información estructurada, y sin embargo, aproximadamente un 80% de la información está almacenada en forma textual no estructurada, de ahí que se desarrollen actualmente técnicas de minería de textos (*Text Mining*), que

permiten encontrar patrones interesantes y útiles en un corpus de información textual no estructurada [DUR01].

Para lograr sus propósitos, la minería de textos necesita combinar varias técnicas, de ahí que sea un campo multidisciplinario que incluye la recuperación de información, el análisis de textos, la extracción de información, el agrupamiento, el resumen, la categorización, la clasificación, la visualización, la tecnología de bases de datos, el aprendizaje automático y la minería de datos [DIX97] [TAN99].

La lengua se ha convertido en un elemento clave de la llamada Sociedad de la Información (SI). Sin embargo, la tecnología asociada al procesamiento automático de textos, no se ha desarrollado al nivel suficiente para proporcionar la infraestructura que soporte a la SI. Además, si queremos crecer en las redes de comunicación, no sólo es importante incrementar los contenidos, sino también es indispensable crear herramientas y recursos capaces de brindar servicios a partir de esos textos almacenados [TAL04].

Hasta ahora, las organizaciones están siendo incapaces de usar eficientemente grandes cantidades de datos textuales no estructurados (*free text*) que existen en varias fuentes, tales como la Web. La minería de textos es una tecnología poderosa que facilita la explotación efectiva de tales datos. Como un resultado, todos los textos no estructurados pueden ser automáticamente analizados y se pueden extraer los conceptos más importantes, etiquetar los documentos con términos claves, resumirlos y hacerlos disponibles en una forma normalizada y explotable [MAL04].

La minería de textos (*text mining*), también conocida como minería de datos textuales o descubrimiento de conocimiento desde bases textuales, pretende algo similar a la minería de datos: identificar relaciones y modelos en la información no cuantitativa. Es decir, proveer una visión selectiva y perfeccionada de la información contenida en documentos, sacar consecuencias para la acción y detectar patrones no triviales e información sobre el conocimiento almacenado en las mismas [OBE01].

La minería de textos es mucho más compleja que la minería de datos porque tiene que trabajar con datos textuales que son inherentemente no estructurados y borrosos. La minería de textos es un campo multidisciplinario que incluye recuperación de información, análisis de textos, extracción de información, agrupamiento, resumen, categorización, clasificación, visualización, tecnología de bases de datos, aprendizaje automático y minería de datos [TAN99] [DIX97].

Aunque la concepción actual de la minería de textos posee un carácter estratégico, aún no ha tenido un desarrollo paralelo a las posibilidades que se brindan de almacenamiento de la información no estructurada disponible. Si verdaderamente se quiere avanzar hacia el desarrollo con ese enfoque

estratégico será necesario en primera instancia desarrollar herramientas que automáticamente asistan a los usuarios en la organización y manipulación de ese exceso de información. De esta forma, por ejemplo, poder filtrar información relevante o interesante desde información no relevante acorde a intereses especificados por los usuarios. La información puede ser resumida y visualmente presentada [LAN01]. Estas razones por sí solas, determinarían la necesidad de enfatizar en el desarrollo de modelos que permitan realizar un estudio integrado de las técnicas de minería de textos que posibiliten el desarrollo futuro de aplicaciones para usuarios finales en esta área, así como herramientas flexibles que los soporten.

En la actualidad puede constatarse que se han desarrollado métodos y técnicas para cada una de las áreas de la minería de textos de forma aislada y no existen herramientas que las integren para el desarrollo de investigaciones en este campo, aunque sí para usuarios finales. Por tal motivo, se requiere de un conjunto de procedimientos, instrumentos y métodos que permitan, al menos, la integración de un conjunto de técnicas de la minería de textos. Por lo anteriormente expuesto, se deriva el **problema científico** a resolver que se manifiesta en ausencia de herramientas con un fundamento metodológico que permitan el desarrollo de investigaciones que integren la representación, agrupamiento y obtención de resúmenes extractos de corpus textuales, como es el caso abordado en esta investigación.

Para contribuir a la solución del problema científico antes plateado, se formuló la **hipótesis general de investigación** siguiente:

Con la concepción e implementación de un modelo que permite la selección de formas de representación de los textos, combinación de métodos de agrupamiento, determinación de medidas para el cálculo de la similitud entre documentos, la selección de diferentes formas de extracción de palabras claves y la obtención de resúmenes extractos, es posible agrupar documentos textuales para obtener resúmenes extractos de los grupos homogéneos de documentos afines, contribuyendo al desarrollo de investigaciones en estas áreas de la minería de textos.

Esta hipótesis quedará validada si se comprueba que:

1. El procedimiento general del modelo desarrollado se caracteriza tanto en su concepción como en su implementación, por poseer las cualidades que hacen factible su aplicación en el estudio práctico de los corpus textuales seleccionados, a partir de su pertinencia, consistencia lógica y parsimonia, así como por poseer la necesaria flexibilidad y generalidad que permita extender su empleo en corpus textuales con otras características.
2. La aplicación de la herramienta propuesta para un conjunto de corpus textuales permite:

a. Identificar las mejores variantes de técnicas de selección de rasgos, de cálculo de la similitud entre vectores, de algoritmos de agrupamiento y de formas de extracción de las palabras claves, que tributan a la obtención de un resumen extracto de mejor calidad.

b. Trazar políticas que permitan el desarrollo de herramientas para usuarios finales a partir de las mejores variantes identificadas.

En conformidad con la hipótesis de investigación identificada, el **objetivo general** de la investigación consiste en desarrollar un modelo conceptual y un procedimiento metodológico, soportado en un software, que ofrezca a los investigadores y desarrolladores en el campo de la minería de textos de una herramienta que posibilite el agrupamiento de documentos para la obtención de resúmenes extractos a partir de grupos homogéneos de documentos afines, con un enfoque de integración.

Este objetivo general fue desglosado en los **objetivos específicos** siguientes:

1. Construir el marco teórico-referencial de la investigación derivado de la consulta de la literatura nacional e internacional actualizada, y otras fuentes de referencia sobre la temática objeto de estudio.
2. Realizar un análisis crítico sobre el estado actual de las técnicas para la representación, agrupamiento y obtención de resúmenes extractos de corpus textuales, enfatizando en los algoritmos que permitan la integración de estas tres áreas de la minería de textos.
3. Diseñar y proponer sobre bases científicas, un modelo conceptual que explique y fundamente el procedimiento metodológico general para agrupar documentos y obtener resúmenes extractos a partir de grupos homogéneos de documentos afines, con un enfoque de integración en investigaciones de minería de textos.
4. Desarrollar el procedimiento metodológico general y los procedimientos específicos asociados, que permitan agrupar documentos para obtener resúmenes extractos a partir de grupos homogéneos de documentos afines, de forma tal que posibilite la identificación de las mejores variantes de las técnicas de agrupamiento y representación textual seleccionadas y las políticas a seguir.
5. Validar el modelo conceptual y el procedimiento propuesto a partir de corpus textuales representativos del universo investigado, como vía de comprobación y factibilidad de la investigación realizada, a partir de los resultados obtenidos por un software que soporte el modelo.

La **novedad científica** principal que aporta esta investigación, radica en la concepción y definición de un modelo conceptual que explica la problemática objeto de estudio y a la vez conduce a su solución, así como de un procedimiento general que, soportado por procedimientos específicos, con sus correspondientes métodos y herramientas analíticas conjugadas armónicamente, permitan agrupar documentos para obtener resúmenes extractos a partir de grupos homogéneos de documentos afines, con un enfoque integrador.

El **valor teórico** de la investigación está directamente vinculado con su novedad científica.

El **valor metodológico** se manifiesta a través del desarrollo del modelo y los procedimientos propuestos, estructurados en un método general que permite su aplicación a corpus textuales en otros idiomas.

El **valor práctico** se relaciona con la aplicación del procedimiento general soportado en el software para el procesamiento de corpus textuales, lo que implica la proyección de las políticas y estrategias de selección y combinación de técnicas que posibiliten un mejor agrupamiento y por tanto la obtención de resúmenes de mayor calidad.

El **valor social** de la investigación radica en su dimensión interna, en la contribución al desarrollo de investigaciones en el área de la minería de textos, y en su dimensión externa, en la implementación de herramientas que permitan a usuarios finales, organizaciones, empresas, e investigadores la extracción de conocimiento a partir de grandes volúmenes de textos.

Para el desarrollo de esta investigación se utilizaron métodos y técnicas de análisis y síntesis, análisis inductivo, de medición y comparativo, herramientas matemáticas, procesamiento computacional de los resultados, sin excluir el análisis lógico, la analogía, la reflexión y otros procesos mentales que también son inherentes a toda actividad de investigación científica.

Para la presentación de esta investigación, esta Tesis de Maestría se estructuró de la forma siguiente. Una Introducción, donde en lo esencial se caracteriza la situación problémica y se fundamenta el problema científico a resolver, así como la estrategia general seguida para su solución como problema científico. Un Capítulo 1, que contiene el marco teórico-referencial que sustentó la investigación originaria. Un Capítulo 2, en el que se resume y explica todo el modelo metodológico desarrollado. Un Capítulo 3, donde se describe el software que soporta el modelo y su evaluación, donde se muestran los casos de aplicación que evidencian la factibilidad y utilidad del empleo del modelo y el procedimiento desarrollado como vía para demostrar la hipótesis de investigación planteada. Un cuerpo de Conclusiones y Recomendaciones derivadas de la investigación realizada, la Bibliografía consultada y un grupo de Anexos como complemento de los resultados expuestos.

Capítulo 1 Acerca de la representación, agrupamiento y resumen de corpus textuales

Existen varias áreas que conforman la minería de textos. Entre ellas, la recuperación y extracción de información, el análisis de textos, resumen, agrupamiento, categorización y clasificación de documentos. A continuación describiremos brevemente cada una de ellas.

Recuperación de información: Se encarga de recuperar documentos que puedan ser considerados relevantes para la tarea a realizar. Típicamente los usuarios del sistema pueden especificar conjuntos de documentos, pero el sistema debe ser capaz de filtrarlos y dejar fuera aquellos irrelevantes [DIX97] [FRA92].

Extracción de la información: Es el proceso de filtrar la información a partir de documentos ya seleccionados y de las especificaciones de los usuarios [DIX97] [FRA92] [FRA03].

Análisis de textos: Involucra todas las técnicas relacionadas con el análisis léxico, sintáctico y semántico de los textos [JAC02].

Resumen: Es el proceso de extraer conocimiento a partir de una fuente de información y presentar el contenido más importante al usuario en una forma condensada y sensitiva para las necesidades de la aplicación o del usuario [JAC02] [BER04].

Agrupamiento: Consiste en encontrar grupos de documentos que están relacionados por tópicos similares y extraer las palabras claves que son consideradas en esa clasificación [BER04].

Categorización: Es el proceso de clasificar los documentos por sus contenidos [JAC02] [BER04].

Clasificación: Se usa como un término más amplio que la categorización, para incluir cualquier asignación de documentos a clases, no necesariamente basados en el contenido [JAC02] [BER04].

En este capítulo abordaremos en detalles dos áreas muy importantes dentro de la minería de textos: el agrupamiento y el resumen de corpus textuales. Pero, como un elemento importante para el desarrollo de las mismas, trataremos técnicas de análisis y representación de colecciones de documentos.

1.1 Representación de los textos

Una de las diferencias de la minería de textos, respecto a la minería de datos es que la minería de textos trabaja con datos textuales no estructurados. Esta característica hace que sea necesario representar los textos adecuadamente para poder realizar un procesamiento posterior.

El objetivo de representar textos es transformar un documento textual a un formato que sea adecuado como entrada para la aplicación de algoritmos (e.g. aprendizaje automático, agrupamiento y clasificación) que permitan hacer minería de textos [LEW92]. Nosotros usamos la representación *Vector Space Model* (VSM) [SAL75] para representar documentos porque es ampliamente reconocida como una representación efectiva para documentos en la comunidad de minería de textos, especialmente en las áreas de recuperación de información, agrupamiento y clasificación. Esta representación, además, puede ajustarse muy bien para imitar otros modelos de indexado bien conocidos tales como índices binarios o probabilísticos. En la representación VSM, cada documento es identificado como un vector de rasgos en un espacio en el cual cada dimensión corresponde a términos indexados distintos. Un vector documento dado tiene, en cada componente, un valor numérico para indicar su importancia. Este valor es comúnmente determinado como una función de cuán frecuente aparece el término correspondiente en el documento particular y cuánto este aparece en total en la colección de documentos. Variando esta función, podemos producir diferentes pesos de los términos que pueden ser interpretados como diferentes modelos de indexado. La representación resultante del texto es equivalente a la representación atributo-valor [JOA97].

Definición de *Vector Space Model* (VSM) [LAN01]

Sea $d \in D$ un documento textual. La representación de d es el vector documento $d=\rho(d)=(w_1,\ldots,w_m)^T \in R=\mathbf{R}^m{}_+$, donde cada dimensión corresponde a un término en la colección de documentos y w_i denota el peso del i-ésimo término. El conjunto de esos m términos indexados, $V=\{t_1,\ldots,t_m\}$, es referido como el vocabulario.

La representación del vector ignora la secuencia en la cual los términos indexados aparecen en un documento. Note, sin embargo, que debemos definir qué es considerado como términos indexados. Este asunto será tratado en la sección 1.2.2. Existen varios pasos que permiten la transformación de una colección de documentos original a la representación de cada documento en vectores de rasgos donde las palabras son usadas como términos indexados. Estos pasos son los siguientes [LAN01]:

1. Transformación del corpus
2. Extracción de términos
3. Reducción de dimensionalidad
4. Normalización y pesado de la representación

En el primer paso se transforma cualquier tipo de documentos en una secuencia de tokens de palabras, desde la cual una secuencia de índices de términos posibles es creada para cada documento durante el paso de extracción de términos. Los términos extraídos pueden ser usados, potencialmente, como el vocabulario del corpus. Un problema es que este número es

extremadamente grande, por tanto, es necesario reducir el vocabulario en el paso de reducción de dimensionalidad. Finalmente, el paso de normalización y pesado de la representación evalúa los pesos para todos los términos de cada documento dado.

1.1.1 Transformación del corpus

El objetivo de la transformación del corpus es convertir los ficheros de entrada en una secuencia de ítems lingüísticos, los cuales son referidos como tokens de palabras. En el paso subsecuente de extracción de términos, estos tokens serán usados para generar rasgos significativos, llamados también índices de términos. Hay dos pasos en el proceso de transformación del corpus.

Primero, los componentes textuales desde diferentes formatos deben ser reconocidos. Por ejemplo, los ficheros pueden ser correos electrónicos, artículos de listas de discusión, postscript, o documentos HTML. Sobre la base de los componentes extraídos del texto, éste debe ser dividido en una secuencia de tokens[1]. [LAN01].

Segundo, la secuencia resultante de tokens puede ser posteriormente transformada dependiendo de la aplicación. Como una regla, las letras son convertidas o todas a minúscula o todas a mayúscula, y son eliminadas las marcas de puntuación al final de los tokens. Además, los tokens que contienen caracteres no alfanuméricos son también omitidos. Todos los dígitos pueden ser convertidos a un dígito predeterminado, convertidos a las palabras que los describen o ser eliminados. Otro enfoque al transformar los textos, es identificar o marcar como tokens individuales aquellas entidades como nombres de personas, localidades, organizaciones y productos. Al transformar el corpus también podemos sustituir las contracciones, por ejemplo, "don't" por "do not", así como las abreviaturas por el nombre completo que ellas representan.

1.1.2 Extracción de términos

En la etapa de extracción de términos se parte de una secuencia de tokens, obtenida a partir del paso de transformación del corpus y se produce una secuencia de términos indexados basados en esos tokens. La forma en que esa secuencia de términos indexados es usada depende del procesamiento posterior que se le vaya a realizar al documento y si el vocabulario ha sido construido o no.

Si, por un lado, el vocabulario ya ha sido creado, cualquier índice de términos extraído que no esté en el vocabulario será omitido durante la extracción de términos. Para cada documento d_j bajo consideración, el vector de frecuencia de términos $d_{tf}=(tf_d(t_1),\ldots, tf_d(t_m))^T$ es creado, donde $tf_d(t)$

[1] Los tokens son cadenas de caracteres delimitadas por espacios en blanco (e.g. espacios, cambios de líneas, tabs)

denota el número de veces que el término $t \in V$ aparece en el documento d. Por otro lado, cuando el vocabulario no ha sido todavía establecido, las salidas del paso de extracción de términos para todos los documentos de entrenamiento son combinadas para generar un conjunto de términos indexados $V'=\{t'_n,\ldots,t'_{m'}\}$, el cual puede potencialmente constituir el vocabulario V.

Acorde a la definición de VSM, las dimensiones del vector corresponden a distinguir los términos indexados en la colección de documentos bajo consideración. Pero, exactamente: ¿qué consideramos como un término indexado? Un reto fundamental en el procesamiento del lenguaje natural es que la información o el significado que se da a conocer por el lenguaje, siempre depende del contexto. Las palabras pueden tener diferentes significados. Además, los lenguajes no pueden ser siempre analizados literalmente. Dos frases iguales pueden tener diferentes significados en dependencia del contexto en el cual ellas fueron usadas. Estos ejemplos muestran que tratar de representar documentos del lenguaje natural por el significado de un conjunto de índices de términos es un reto. Observe en el anexo 4 diferentes enfoques lingüísticos que intentan capturar, o ignorar, una cierta extensión del significado con respecto al contexto.

En [BLA92] se comenta que puede ser razonable asumir la representación de textos más compleja teniendo en cuenta que niveles más altos de análisis textual deben tender a obtener herramientas más efectivas. Pero, mientras más compleja sea la definición de términos indexados, más compleja será la representación de los textos, y la dimensionalidad de los rasgos crecerá correspondientemente. Por tanto, escoger un nivel adecuado del análisis de textos sobre el cual basar la definición de los términos es siempre un equilibrio entre la expresividad semántica y la complejidad de la representación. En la mayoría de las aplicaciones de la minería de textos, las definiciones simples de los términos son dominantes. Típicamente, se enfoca el análisis de textos a partir de los dos primeros niveles, y a su vez, alguna información sintáctica puede ser también incorporada.

Existen tres definiciones de términos ampliamente usadas y que básicamente explotan el plano estadístico de los textos. El primer enfoque depende de n-gramas de las letras, y corresponde al análisis de textos en el nivel de grafema. Existen también dos enfoques, llamados palabras simples y frases de palabras, que son los mejores para realizar un análisis léxico de los textos. Note, sin embargo, que ellos posteriormente puedan también utilizar alguna información sintáctica.

n-gramas de las letras: Las subsecuencias de palabras, solapadas, de n letras contiguas son conocidas como *n*-gramas[2], donde *n* es un número positivo (e.g. trigramas si *n*=3, o tetragramas si *n*=4) [CAV94]. Por ejemplo, la palabra "token" consiste de los trigramas "tok", "oke", y "ken", es

[2] Es importante aclarar que a veces el término n-grama es usado para referirse a secuencias de palabras de longitud n. En este trabajo nos referiremos a esos términos de múltiples palabras como frases.

también posible incluir "+to" y "en+". La ventaja de los *n*-gramas es que el conjunto de posibles términos es fijado y conocido por adelantado, son independientes del lenguaje, son robustos para los tratar los errores y variaciones morfológicas y ortográficas. Son fáciles de calcular, sin embargo, la representación resultante es difícil de analizar por los humanos [TAU00].

Palabras: En muchas áreas de la minería de textos se identifican las palabras simples como rasgos y se obtienen buenos resultados [SAL88]. Un enfoque común es usar cada token como una palabra para describir los textos. La representación VSM no considera la secuencia en la que las palabras aparecen en un documento, es por eso, que es también referida como el modelo bolsa de palabras (*bag-of-words model*) [LEW94]. Obviamente, esa definición de términos es independiente del lenguaje y computacionalmente muy eficiente. Sin embargo, una desventaja es que cada inflexión de una palabra es un posible rasgo y el número de rasgos posibles puede ser innecesariamente grande. De aquí que sea necesario aplicar técnicas de reducción de dimensionalidad (ver sección 1.1.3.2).

Frases: Las frases son combinaciones de tokens. El uso de frases es justificado, especialmente en Inglés, para observar muchas expresiones que son términos multi-palabras tales como "data mining" o "information filtering". Típicamente, sólo frases específicas del dominio son consideradas, porque de otra forma, el número de posibles términos podría crecer drásticamente. Términos multi-palabras pueden ser identificados como secuencias de palabras que frecuentemente ocurren [COH96], detectados a partir de la aplicación del procesamiento del lenguaje natural (llamado sintáctico de frases) [LEW92], o al proveer manualmente un conjunto de frases para un dominio particular [SAH98]. Obviamente, el uso de frases tiende a ser dependiente del dominio. Existen diferentes opiniones respecto al uso de las frases: algunos investigadores reportan una mejora en la calidad de los resultados cuando las usan, mientras que otros no logran obtener mejores resultados en comparación con usar palabras simples. Sahami [SAH98] dice que parte de esa discrepancia puede estar dada por la expresividad de los modelos usados.

1.1.3 Reducción de la dimensionalidad

La mayor parte del paso de reducción de dimensionalidad es aplicado solamente en el caso que se necesite construir el vocabulario. El conjunto de posibles términos indexados $V'=\{t'_1,\ldots,t'_{m'}\}$ resultante desde el paso de extracción de términos original es usualmente muy grande. El objetivo de la reducción de dimensionalidad es reducir el número de rasgos que son finalmente usados para representar los documentos. Por ejemplo, para un futuro agrupamiento o clasificación de los documentos, ese conjunto de rasgos resultante debe ser aún discriminante entre las diferentes clases. Como resultado, obtenemos un conjunto menor de términos indexados, el vocabulario $V=\{t_1,\ldots,t_m\}$, donde $m \leq m'$ denota el número de términos indexados que permanecen.

Controlar la dimensionalidad del espacio del vector es esencial por dos razones. La complejidad de muchos algoritmos de aprendizaje, agrupamiento o clasificación, dependen no solamente del número de ejemplos de entrenamiento, sino crucialmente del número de rasgos. Así, reducir el número de términos indexados puede ser necesario para hacer esos algoritmos tratables [LAN01]. Además, aunque una mayor cantidad de rasgos se puede asumir como más información, existen rasgos que son irrelevantes y provocan la obtención de peores resultados. En la literatura se refieren al problema de tener muchos rasgos como "*curse of dimensionality*" [DUD73]. En muchos casos, eliminar los rasgos menos informativos puede realmente aumentar la eficiencia del procesamiento a realizar.

Usamos el término reducción de dimensionalidad para abarcar cualquier técnica que su objetivo sea controlar la dimensionalidad del vector. Esto incluye técnicas de selección de rasgos y técnicas basadas en la reparametrización [LAN01]. A continuación describiremos cada una de ellas.

1.1.3.1 Selección de rasgos

Las técnicas de selección de rasgos para la reducción de dimensionalidad toman como entrada un conjunto de rasgos y la salida es un subconjunto de esos rasgos, los cuales son relevantes para, por ejemplo, discriminar entre clases de documentos [DAS97]. Idealmente pudiéramos pensar en encontrar el mejor subconjunto dentro de todos los subconjuntos posibles que nos permitan los mejores resultados en el procesamiento posterior. Así, la selección de rasgos puede ser vista como un problema de optimización donde el espacio de búsqueda corresponde al conjunto potencia de todos los m' rasgos, i.e. donde hay $2^{m'}$ subconjuntos candidatos a analizar. Obviamente, esto requiere una búsqueda exhaustiva intratable teniendo en cuenta el número de rasgos que es usualmente muy grande en dominios textuales. Por tal motivo, la selección de rasgos puede ser guiada por heurísticas [SCH96].

Un enfoque es tratar cada rasgo independientemente y evaluarlo con una puntuación que nos permita decidir cuando incluirlo en el vocabulario y cuando excluirlo. El vocabulario final es establecido seleccionando todos aquellos rasgos que su puntuación sea superior o inferior a un umbral predeterminado o seleccionando los m mejores rasgos, i.e. los m rasgos con mayor o menor puntuación acorde a la magnitud de la puntuación. La determinación de un umbral apropiado o un número m de rasgos, es aún un tema abierto.

Comenzaremos con un enfoque lingüístico ampliamente conocido como eliminación de palabras de parada (*stop word elimination*) [YAN97] [MLA98]. Además, comentaremos varias medidas numéricas (típicamente basadas en la frecuencia con que los términos ocurren en los documentos) frecuentemente usadas para evaluar la calidad de los términos con respecto a su habilidad para discriminar entre clases, útiles en problemas de clasificación y agrupamiento. Observe en el anexo 1

un resumen de los principales elementos de frecuencias de términos y número de documentos usados para la evaluación de las medidas numéricas de calidad de términos.

Nótese que $\bar{n}(t)$ es definido como:

$$\bar{n}(t)=n-n(t) \tag{1.1}$$

Además, se observan las siguientes relaciones entre las frecuencias de los términos:

$$tf = \sum_{i=1}^{m'} tf(t_i) \tag{1.2}$$

$$tf(t) = \sum_{j=1}^{n} tf_{d_j}(t) \tag{1.3}$$

donde *m*' denota el número de rasgos potenciales en el vocabulario *V*' antes de cualquier paso de reducción de dimensionalidad particular.

Eliminación de palabras de parada. Cuando analizamos un lenguaje, podemos frecuentemente observar que hay muchas palabras, llamadas palabras de parada (*stop words*), que pueden ocurrir en todos los documentos sin ofrecer información alguna sobre el contenido de los mismos (e.g. artículos, preposiciones, conjunciones y pronombres) [SAL83] [RIJ79]. Las palabras de parada tienen una alta frecuencia de aparición y poco poder discriminante, por tanto, es razonable eliminar esas palabras desde el conjunto de posibles términos indexados *V*', y como resultado, reducir la dimensionalidad del vector asociado [RIJ79] [SAH98].

Hay dos formas de obtener una lista de palabras de parada. Una forma común es establecer manualmente la lista de palabras de parada (*stop list*). Muchas de estas listas, para texto Inglés, pueden ser encontradas en Internet[3] y en la literatura [SAH98] (ver anexo 2). Un segundo enfoque para la eliminación de palabras de parada es construir la lista de palabras de parada automáticamente basándonos en la colección de documentos bajo consideración. Para hacerlo así, un enfoque común es considerar como palabras de parada los *i* términos más frecuentes o todos los términos con un umbral de frecuencia superior a un umbral dado. El umbral puede ser definido manualmente con anterioridad o puede calcularse basándonos en el histograma de frecuencias de la distribución de los términos. Este enfoque elimina las palabras de parada que son específicas para la colección de documentos bajo consideración y es, además, específica al dominio. Siguiendo esta idea, no solamente palabras comúnmente aceptadas como palabras de parada son eliminadas, sino también otras como sustantivos, verbos y adjetivos pueden ser eliminadas [LAN01].

[3] ftp://ftp.cs-cornell.edu/pub/smart/english.stop

Umbral de frecuencia de términos y Ley de Zpif. Una heurística de selección muy simple es eliminar todos los términos cuyas frecuencias son o superiores a un umbral predefinido o inferiores a un umbral[4] predefinido. A partir de observaciones hechas por Luhn, el énfasis es tomado como un indicador de significación. Por tanto, la frecuencia de ocurrencias de términos es una medida apropiada de la significación de los términos [LAN01]. Así, términos que raramente aparecen en una colección de documentos tendrán poco poder discriminante y pueden ser eliminados [RIJ79]. En contraste, términos con frecuencia de aparición alta se asumen que son comunes y que tampoco tienen poder discriminante[5].

Sahami en [SAH98], a partir de estudios de la Ley de Zipf [ZIP49], concluye que podemos reducir la dimensionalidad de los rasgos originales en un 50 %, si eliminamos los términos que tienen o muy alta o muy baja frecuencia de aparición, a partir de un cálculo adecuado del umbral.

Umbral de frecuencia de documentos. Teniendo en cuenta que $n(t)$ es el número de documentos en los cuales el término t aparece al menos una vez, una heurística simple de selección es excluir todos los términos desde el vocabulario cuya frecuencia de documentos es menor que algún umbral, ya que términos que ocurren en sólo muy pocos documentos improbablemente llevan información que permita distinguir los grupos textuales y tienden a ser ruidosos [YAN97]. Además, usar la ocurrencia de términos infrecuentes no es confiable estadísticamente. Al eliminar estos términos se mantiene el poder discriminante y se mejora la efectividad del agrupamiento y clasificación textual.

Frecuencia inversa de documento y TFIDF. En contraposición con lo anterior, términos que aparecen en una gran porción de la colección de documentos pueden ser no discriminantes. La importancia de los términos se asume inversamente proporcional al número de documentos en los cuales el término particular aparece. Una medida posible para esto es la frecuencia inversa del documento para el término t, la cual está típicamente definida como

$$idf(t) = \log \frac{n}{n(t)} \tag{1.4}$$

tal que aquellos términos con valores altos son preferidos. Después de eliminar las palabras de parada, la importancia de un término se incrementa con su frecuencia de uso. Combinando estas

[4] El cálculo del umbral de términos de baja frecuencia se justifica a partir de la Ley de Zipf sobre la frecuencia de la ocurrencia de las palabras en una colección de documentos [ZIP49].
[5] Eliminar esos términos corresponde a eliminar las palabras de parada donde la lista de palabras de parada es automáticamente construida desde la colección de documentos.

ideas se formuló la medida frecuencia del término / frecuencia inversa de documentos (*term frequency* / *inverse document frequency* (tfidf))

$$\text{tfidf}(t)=tf(t)\cdot idf(t) \tag{1.5}$$

la cual asigna valores altos a los términos que son considerados más importantes. También, una combinación similar de frecuencia de términos y frecuencia inversa de documentos es usualmente usada para asignar pesos a los términos [SAL88].

Razón de señal a ruido (*Signal-to-Noise Ratio*). Tomando en consideración la teoría de la información, la razón de señal a ruido de un término particular mide el poder discriminante que transmite ese término, por tanto los términos con grandes valores son preferidos. La evaluación del ruido del entorno se basa en la entropía [SAL83]. Ésta es una variable aleatoria discreta X sobre c valores diferentes con probabilidades p_i, i=1,…,c, definida por

$$\text{Entropía}(X) = -\sum_{i=1}^{c} p_i \log p_i \tag{1.6}$$

Así, la entropía puede ser evaluada como la cantidad de información que esperamos recibir sobre el promedio cuando observamos una variable aleatoria particular [SAL83]. La entropía es cero cuando una salida de la variable aleatoria ocurre con certeza, i.e. p_i=1 para un i arbitrario y p_j=0 para $j\neq i$. Mientras más uniforme sea una distribución, mayor es su entropía. Así, la entropía alcanza su máximo valor log c si todas las salidas de X son igualmente probables.

Cuando un término t está concentrado en sólo pocos documentos, se puede calcular el Ruido(t) [SAL83], como la entropía de la distribución de probabilidad del término t entre los documentos:

$$\text{Ruido}(t) = -\sum_{j=1}^{n} \text{P}(d_j,t)\log \text{P}(d_j,t), \quad \text{donde} \quad \text{P}(d_j,t) = \frac{tf_{d_j}(t)}{tf(t)} \tag{1.7}$$

es decir, la probabilidad que un documento d_j y un término t co-ocurran. El rango de la función ruido es [0,log n]. El ruido es cero cuando el término t aparece sólo en un documento y toma su valor máximo log n si el término t ocurre con la misma frecuencia en todos los documentos.

La razón señal a ruido se expresa como la diferencia de esos logaritmos SNR(t)=log tf(t) – Ruido(t).

En [NUR01] se utiliza la entropía, según Lochbaum y Streeter en 1989, como una medida para el cálculo de la importancia de las palabras:

$$\text{Entropía}(t) = 1 + \frac{1}{\ln(n)} \sum_{i=1}^{n} p_i(t) \cdot \ln(p_i(t)) \quad \text{donde } p_i(t) = \frac{tf_{d_i}(t)}{\sum_{j=1}^{n} tf_{d_j}(t)} \tag{1.8}$$

Empíricamente se consideran relevantes las palabras que tienen una alta entropía, dentro de aquellas que tienen una alta frecuencia de aparición (i.e. preferimos seleccionar aquellas palabras que tienen una entropía alta desde un conjunto de palabras que son igualmente frecuentes).

Calidad de términos. En [BER04] se muestran dos medidas que son utilizadas para medir la calidad de los términos y por tanto permiten la reducción de la dimensionalidad a partir de la selección de aquellos términos relevantes.

$$q_0(t) = \sum_{j=1}^{n} \left(tf_{d_j}(t)\right)^2 - \frac{1}{n}\left[\sum_{j=1}^{n} tf_{d_j}(t)\right]^2 \tag{1.9}$$

$$q_1(t) = \sum_{j=1}^{n_1} \left(tf_{d_j}(t)\right)^2 - \frac{1}{n_1}\left[\sum_{j=1}^{n_1} tf_{d_j}(t)\right]^2 \tag{1.10}$$

Nótese que la segunda expresión constituye una variante de la primera donde n_1 es el número de documentos en los cuales t ocurre al menos una vez

Skewness y Kurtosis. Skewness y kurtosis son medidas estadísticas que indican una distorsión de una distribución y pueden ser utilizadas, entre otras muchas aplicaciones, para conocer la parcialidad de los términos. La parcialidad de un término t se define como [FUK99]:

$$P(t)=w_1 \cdot \text{Skewness}(t)+w_2 \cdot \text{Kurtosis}(t)$$

donde w_1 y w_2 son pesos positivos para Skewness y Kurtosis, definidas a continuación.

$$\text{Skewness}(t) = \frac{1}{n} \sum_{i=1}^{n} \frac{\left(tf_{d_i}(t) - \frac{tf(t)}{n}\right)^3}{s^3} \tag{1.11}$$

$$\text{Kurtosis}(t) = \frac{1}{n} \sum_{i=1}^{n} \frac{\left(tf_{d_i}(t) - \frac{tf(t)}{n}\right)^4}{s^4} - 3 \tag{1.12}$$

donde s es la desviación estándar de la ocurrencia del término t en la colección de documentos. Valores altos de Skewness(t) y Kurtosis(t) indican que el término t es más general en el corpus de textos y viceversa.

1.1.3.2 Reparametrización

Las técnicas de reparametrización para la reducción de dimensionalidad toman como entrada un conjunto de rasgos derivados de la colección de documentos y construyen un nuevo conjunto de los rasgos que contiene menos rasgos pero formados a partir de combinaciones y transformaciones de los rasgos existentes, manteniendo y en algunos casos mejorando el poder discriminante. Los rasgos

resultantes pueden ser artificiales y pueden no corresponder con los términos indexados que fueron encontrados en los documentos [LAN01]. Comenzaremos examinando tres técnicas lingüísticas de transformación: homogeneidad ortográfica (*spelling*), *stemming* y el uso de tesauros.

Homogeneidad ortográfica (*spelling*). La homogeneidad ortográfica consiste en convertir todas las palabras del léxico de un idioma a un estándar. Por ejemplo, en el idioma Inglés se pueden convertir las palabras del léxico norteamericano al británico o viceversa (e.g. sustituir "*capitalize*" por "*capitalise*" y "*colour*" por "*color*").

***Stemming*.** Existen diferentes variantes morfológicas de una palabra, cada inflexión de una palabra es un rasgo potencial. *Stemming* es un proceso seudo-lingüístico que reduce las palabras a su forma raíz [YAN97] (e.g. reducir las palabras "classifier", "classified" y "classifying" a su raíz "classify"). Consecuentemente, la dimensionalidad del espacio de rasgos puede ser reducida haciendo corresponder las palabras morfológicamente similares con la palabra raíz asociada. Ejemplos de algoritmos que realizan *stemming* aparecen reportados en [POR80] y [FRA92]. Obviamente, *stemming* es dependiente del lenguaje, aunque no es dependiente del dominio. Para idiomas como el Inglés es relativamente sencillo especificar heurísticas que permitan realizar el *stemming*, sin embargo en lenguajes como el Alemán o el Francés, esto puede ser mucho más difícil [SAM98]. Es por eso, que una forma de realizar el *stemming* puede ser consultando con listas de palabras que tengan todas las variaciones morfológicas.

Uso de tesauros. El objetivo del uso de un tesauro en el contexto de reducción de dimensionalidad es considerar como un único término aquellos que sean sinónimos, i.e. palabras con significados iguales o similares. Generalmente, un tesauro es una colección de palabras que son agrupadas por analogía de su significado y ordenadas alfabéticamente. Estas relaciones pueden ser usadas como grupos de palabras en clases semánticas. La dimensión del espacio puede ser entonces reducido haciendo corresponder todos los términos indexados a las clases que ellos pertenecen. Obviamente, un tesauro es dependiente del dominio y del lenguaje [LAN01] [SAL83].

Análisis de latencia semántico (*Latent Semantic Analysis*). Esta es una técnica estadística para descubrir automáticamente una estructura semántica, i.e. encontrar asociaciones semánticas entre los términos en una colección de documentos y se basa en la descomposición en valores singulares. Explotando esas asociaciones puede resolver el problema de sinónimos y reducir la dimensionalidad del espacio de rasgos [LAN01].

Combinación de métodos

Hemos discutido varias técnicas que permiten controlar la dimensión del vocabulario a aquellos elementos que corresponden a las dimensiones del espacio del vector en las cuales los documentos

son representados. En lugar de aplicar justamente una de estas técnicas, en muchas aplicaciones, algunas de ellas son usadas en combinación [NIG00].

1.1.4 Normalización y pesado de la matriz

Dadas las estadísticas de frecuencias de términos de todos los documentos de entrenamiento, podemos generar un vector pesado $\mathbf{d}=(w(d,t_1),\ldots, w(d,t_m))^T$ para cualquier documento d basado en el vector de frecuencia de términos $\mathbf{d}_{tf}=(tf_d(t_1),\ldots, tf_d(t_m))^T$ [6], Cada peso $w(d,t)$ expresa la importancia del término t en el documento d con respecto a su frecuencia en todos los documentos.

Existen varias formas de pesar los términos indexados. Salton y Buckley [SAL88] identifican tres factores para determinar el peso de los componentes al calcular la importancia de los términos, los llaman factores local, global y de normalización. Así, el peso del término se define como

$$w(d,t)=\frac{w_{local}(d,t)w_{global}(t)}{w_{normalización}(d)} \tag{1.13}$$

Componente local: El factor de peso local $w_{local}(d,t)$ refleja la importancia del término t en un documento particular d. Típicamente, este factor es obtenido aplicando una función de transformación f a la frecuencia del término t cuando aparece en el documento d

$$w_{local}(d,t)=\begin{cases} f(tf_d(t)) & \text{if } tf_d(t)>0 \\ 0 & \text{en otro caso} \end{cases} \tag{1.14}$$

Ejemplos de funciones de transformación pueden ser la función identidad, la función de indicador binario y normalizar la frecuencia de los términos a través de la división por la mayor frecuencia de términos $tf_d(t_{\max})$ en el documento d.

Componente global: El factor de peso global $w_{global}(t)$ tiene en cuenta la importancia del término t en la colección de documentos. Una medida que usualmente se utiliza aquí es la frecuencia inversa de documentos.

Componente de normalización: Normalizar el vector de pesos de un documento d permite abstraernos de la variedad de longitudes de documentos. El factor de normalización se representa como $w_{normalización}(d)$. Típicamente, los vectores pesados son normalizados siguiendo la normalización coseno. En un contexto probabilístico, estos son también comúnmente normalizados por la suma de los componentes del vector pesado de un documento.

[6] Este vector se obtiene comúnmente del paso de extracción de términos.

Observe en el anexo 3 que existen muchas formas de pesar una matriz. La mayoría de las formas de pesado se basa en alguna variación de la fórmula TF-IDF. La idea de una expresión TF-IDF es que el peso de los términos deba reflejar la importancia relativa de un término en un documento con respecto a los otros términos en el documento. Una de estas variantes permite calcular el peso del *i*-ésimo término *t* en el documento *d* según las siguientes expresiones [MAN00] [BER04]:

$$w(d,t) = tf_d(t)\left(1 + \log_2(\frac{n}{n(t)})\right) \tag{1.15}$$

Una modificación del cálculo anterior de TF-IDF se muestra en la siguiente expresión, teniéndose en cuenta la cantidad de veces que ocurren en un documento aquellos términos que más aparecen:

$$w(d,t) = \frac{tf_d(t)}{\max\limits_k tf_d(k)}\left(1 + \log_2(\frac{n}{n(t)})\right) \tag{1.16}$$

donde $\max\limits_k tf_d(k)$ representa el número de ocurrencias que tiene la palabra que más aparece en *d*.

Por último, otras de las formas reportadas para calcular TF-IDF es la expresión que aparece a continuación [BER04]:

$$w(d,t) = \begin{cases} (1 + tf_d(t_i))\log_2(\frac{n}{n(t_i)}), & si\ tf_d(t_i) \geq 1 \\ 0 \qquad\qquad\qquad\qquad , & si\ tf_d(t_i) = 0 \end{cases} \tag{1.17}$$

Con el objetivo de que los pesos obtenidos sean valores en el intervalo [0,1] y teniendo en cuenta la componente de normalización antes descrita, en [SEB99] se propone calcular TF-IDF y normalizar los valores según coseno:

$$w(d,t) = \frac{\text{tfidf}(d,t)}{\sqrt{\sum_{j=1}^{m}\left(\text{tfidf}(d,t_j)\right)^2}} \tag{1.18}$$

1.2 Resumen extracto

Un problema de la minería de textos, en la actualidad, es encontrar documentos relevantes sobre la información que necesitamos, pero realmente la situación es incluso más compleja. Después de encontrar algunos documentos relevantes, el problema es encontrar el tiempo necesario para leerlos,

por tanto, resumirlos puede contribuir a realizar una revisión en tiempo de los aspectos relevantes de ellos.

Resumir automáticamente textos consiste en tomar una fuente de información, extraer contenido de ésta, representando los conceptos más importantes de cada documento y presentar el contenido más importante al usuario en una forma compacta y sensible para satisfacer las necesidades de la aplicación o del usuario [NAN00]. Es un proceso que toma un documento como entrada, ofreciendo como salida un documento más corto, llamado documento sustituto, que contiene el contenido más importante del inicial. La importancia puede ser considerada a partir de diferentes puntos de vista: fragmentos asociados a las palabras claves, requerimientos de usuarios y relevancia a partir de tópicos seleccionados, entre otros [JAC02].

Existen dos tipos principales de resúmenes: los resúmenes abstractos (*abstract*) y los resúmenes extractos (*summarization*). Sus aplicaciones y formas de obtención difieren sustancialmente. Un extracto es un resumen que es construido, sobre todo, escogiendo los fragmentos relevantes del texto, quizás con algunas pequeñas revisiones. Un abstracto es un resumen que describe el contenido de un documento sin presentar, necesariamente, algunas de las partes del contenido del texto original explícitamente. En ambos casos, podemos pensar en el resumen como la compresión o la compactación de un documento. Un extracto realiza la compresión descartando el material menos relevante, mientras que un abstracto realiza la compresión de un modo más sofisticado, e.g. suprimiendo detalles y sustituyendo datos específicos con generalidades. Obviamente, se pudieran mezclar estos dos modos de compresión para obtener un resumen de mayor calidad [JAC02].

Otra forma de clasificar los resúmenes es en genéricos (*generis summaries*) y teniendo en cuenta las preguntas de los usuarios (*query-relevant summaries*). Los primeros se generan teniendo en cuenta todo el contenido del documento y los segundos resumen el contenido relevante a partir del entorno de la consulta realizada por el usuario.

La tarea de resumir textos puede ser dividida en dos fases: (a) construcción de una representación del texto; (b) generación del resumen, el cual puede incluir extracción de oraciones y/o construcción de nuevas oraciones. La construcción automática de nuevas oraciones es una tarea extremadamente difícil, la mayoría de los sistemas generan un resumen extrayendo las oraciones relevantes del documento original [LAR00]. En este trabajo pretendemos obtener un resumen extractivo.

Un resumen se puede realizar a partir de un único documento o de múltiples documentos relacionados o no. Este es otro aspecto que permite la clasificación de los resúmenes. A continuación describiremos en qué consiste realizar un resumen extracto de un único documento (*single summarization*) a partir de la generación de textos pequeños y concisos que representan las ideas generales presentadas en un texto, y un resumen extracto de múltiples documentos (*multiple*

summarization) donde se extraen las ideas fundamentales a partir de múltiples documentos [LAR99].

1.2.1 Resumen extracto de un único texto

La forma más conocida para construir resúmenes extractos de documentos simples es teniendo un programa selector de fragmentos relevantes desde el documento y luego combinarlos en un extracto [JAC02].

En [MOE00], [JAC02] y [FUK03] se reportan varias técnicas de resumen extracto de un único documento. Algunas de estas técnicas son: resumen por selección de oraciones, resumen por selección de párrafos, resumen basado en discurso y resumen basado en co-referencia. A continuación describiremos cada una de estas estrategias y ejemplificaremos con sistemas reportados en la literatura.

Resumen por selección de oraciones. Teniendo en cuenta que la oración es frecuentemente (no siempre), seleccionada como la unidad para construir resúmenes, podemos partir de la selección de oraciones y reducir la generación del resumen a un problema de clasificación de las oraciones en relevantes o no para formar parte del extracto. Tener una clasificación permite incluir o excluir oraciones según la longitud deseada del resumen [JAC02]. Esta técnica tiene sus ventajas y desventajas en dependencia del texto a resumir y de la longitud del resumen deseado. Es ventajosa cuando deseamos resumir noticias o textos no extremadamente largos. Un problema es que los resúmenes resultantes son a menudo, desunidos y no se leen bien. Las oraciones, además, tienen obviamente sus ventajas y desventajas respecto a usar unidades mayores (e.g. párrafos) o unidades pequeñas (e.g. frases).

Resumen por selección de párrafos. Como mencionamos anteriormente, problemas al considerar las oraciones como unidad básica al resumir son la desunión y la falta de coherencia, es decir, no se leen bien los resúmenes. La utilización de componentes básicos más grandes puede contribuir a la obtención de un resumen más coherente. Así, un enfoque puede ser asumir en un texto un número de párrafos que se pueden considerar mejores o relevantes, y que pueden describir el contenido del texto en su totalidad. Esto es particularmente efectivo para ciertos tipos de documentos, donde uno de los primeros párrafos, típicamente el primero, proporciona una descripción coherente de lo que sigue. Este enfoque es ventajoso si el resumen requerido es relativamente largo, o si el material es tal que el aspecto principal de un documento es probablemente contenido en un párrafo simple. La selección de párrafos ha sido bien estudiada, aunque no tiene tanta popularidad como la selección de oraciones. [JAC02].

Resumen basado en discurso. Como hemos descrito en las secciones 1.2.1.1 y 1.2.1.2, resumir documentos teniendo en cuenta las oraciones o párrafos como unidades tiene sus ventajas y desventajas. La estrategia de resumen basado en discurso propone determinar inicialmente la forma típica de discurso del documento a resumir, es decir, modelar inicialmente la estructura del documento que será resumido [MOE00]. Para ello, es necesario inicialmente dividir el documento en unidades de discurso coherentes. Los bloques de textos obtenidos deben reflejar los subtópicos contenidos en el texto [YAR92]. Esta estrategia tiene gran utilidad porque los tipos diferentes de documentos tienen variaciones de su estructura, y por tanto, es muy útil identificar inicialmente la presencia o la ausencia de segmentos claves, para posteriormente realizar el extracto a partir de las unidades básicas detectadas. Un desventaja de esta estrategia es que generalmente funciona correctamente para un tipo particular de documentos y utilizándolo en un contexto específico.

Resumen basado en co-referencia. Las asociaciones que existen entre términos que tienen co-referencia[7] pueden ser usadas para clasificar y seleccionar oraciones a incorporar en el resumen. Estos métodos están más enfocados a generar el resumen teniendo en cuenta las preguntas de los usuarios y extrayendo el contenido relevante a partir del entorno de dichas consultas. Los métodos que siguen esta estrategia son capaces de generar resúmenes que son casi tan efectivos como el texto completo, ayudando al usuario a determinar la relevancia [BAL98]. La desventaja es que se requiere realizar un preprocesamiento más costoso del documento (e.g. etiquetarlo, trabajar con tesauros).

1.2.1.1 Ejemplos

En [FUK03] se reporta un sistema que permite el resumen de un documento siguiendo la estrategia de selección de oraciones. Se extraen aquellas con mayor peso según el cálculo TF/IDF, se organizan en dependencia de su posición en los párrafos y se eliminan las partes innecesarias en las oraciones para resolver anáforas. Las oraciones seleccionadas conforman el extracto.

Un ejemplo de la estrategia de resumen basado en el discurso es el sistema SALOMON [JAC02], que antes de obtener el resumen extracto es capaz de identificar las partes principales del documento (e.g. título, introducción) y esto contribuye a la obtención de un resumen de mayor calidad. El método *Rhetorical Structura Theory* es otro ejemplo de esta estrategia [MAR00]. La idea básica es descomponer el texto en dos tipos de unidades elementales: núcleo y satélite. El núcleo expresa algo esencial para el propósito del escritor, mientras que el satélite expresa algo menos esencial. En [LAR00a] también se presenta un algoritmo que sigue la estrategia basada en discurso teniendo en cuenta la importancia de los tópicos de un documento. Se utiliza el algoritmo *TextTiling* para

[7] Co-referencia es un fenómeno lingüístico donde dos o más expresiones pueden representar o indicar la misma entidad (e.g., 'Bill Gates' y 'The Chairman of Microsoft'. La co-referencia puede admitir la ambigüedad.

identificar los tópicos basado en la métrica TF-IDF, se calcula la relevancia de cada tópico en el documento y se extrae de cada tópico el número de oraciones proporcional a su relevancia.

En [IND01] se describe un programa para resumir un documento combinando las estrategias basadas en discurso y co-referencia. Éste inicialmente describe las unidades de discurso del texto y elimina la información extraña, posteriormente realiza una detección robusta de co-referencia. Otro ejemplo que combina varias estrategias es reportado en [LAR04]. Los autores proponen un sistema para resumir noticias y obtener una estructura argumentativa aproximada del texto original. Para alcanzar estos objetivos se usan varias técnicas y heurísticas, como descubrimiento de los conceptos principales en el texto, conectividad entre oraciones, presencia de nombres propios, anáforas, marcadores de discurso y una representación de árbol binario.

1.2.2 Resumen de múltiples textos

Si el resumen de un documento simple se dificulta, resumir múltiples documentos presenta aún más problemas. Sin embargo, el éxito en este empeño ofrecerá una utilidad real a muchos investigadores y "trabajadores de ciencia", por permitirles procesar colecciones enteras de documentos con menos esfuerzo que en la actualidad [JAC02].

En [STE00] se plantea que el resumen de documentos simples es sólo una de las tareas críticas necesarias para formar un resumen completo de múltiples documentos. Ejemplos de esto se mencionan a continuación:

- Identificar los temas de importancia en la colección de documentos.
- Seleccionar el resumen de un documento simple representativo por cada uno de los temas.
- Organizar los resúmenes representativos para formar el resumen final de múltiples documentos.

Actualmente, resumir múltiples documentos es un tema que se aborda mucho en las investigaciones sobre la obtención de resúmenes de documentos. Inicialmente, se pensó que se podían aplicar las técnicas usadas para resumir un único documento para obtener el resumen de múltiples documentos considerando la colección de todos los textos como un único documento. Aunque muchas de las técnicas usadas en el resumen de un único documento pueden ser usadas también en el resumen de múltiples documentos, existen al menos cuatro diferencias significativas [GOL99]:

1. El grado de redundancia en la información contenida en un grupo de documentos relacionados es mucho mayor que el grado de redundancia en un documento.
2. Un grupo de documentos puede contener una dimensión temporal (e.g. reportes noticiosos).

3. El tamaño del resumen con respecto al tamaño del documento será típicamente mucho menor para colecciones de documentos que para un único documento.
4. El problema de la co-referencia en los resúmenes presenta retos mayores al resumir múltiples documentos que al resumir un único documento.

En [JAD00] se describen varias formas de crear resúmenes por extracto de múltiples documentos:

Resumir secciones comunes de los documentos. Encontrar las partes importantes y relevantes que la colección de documentos tiene en común (su intersección) y usarla como un resumen.

Resumir secciones comunes y secciones únicas de los documentos. Encontrar las partes importantes y relevantes que la colección de documentos tiene en común y las partes relevantes que son únicas y usarlas como un resumen.

Resumir el documento centro (más representativo). Crear el resumen de un único documento a partir del documento centro o más representativo de la colección.

Resumir el documento centro (más representativo) más el resto de los documentos de la colección. Crear el resumen de un único documento a partir del documento centro o más representativo de la colección y agregar alguna representación del resto de los documentos (pasajes o extracción de palabras claves) para proveer un cubrimiento total de la colección de documentos.

Resumir el documento más reciente más el resto de los documentos de la colección. Crear el resumen del documento que tiene información más reciente y adicionar alguna representación del resto de los documentos para proveer un cubrimiento de la colección de documentos.

Resumir secciones comunes y secciones únicas de documentos teniendo en cuenta el factor tiempo. Encontrar las partes relevantes e importantes que la colección de documentos tiene en común y las partes relevantes que son únicas. Tener en cuenta la secuencia de tiempo de la información extraída en la generación del resumen.

1.2.2.1 Ejemplos

A partir de la definición de las diferentes formas de resumir múltiples documentos, se reportan en la literatura varios ejemplos de sistemas que permiten realizar esta tarea.

En [FUK03] se muestra un método para resumir un corpus textual donde el primer documento es resumido en una proporción requerida y los siguientes documentos son resumidos en esa proporción más un 10%. Las oraciones son la unidad básica y son colocadas según su orden original en la colección. En [SCH02] se describe un método similar donde se aplican nuevas estrategias para seleccionar oraciones interesantes e informativas, a partir de una medida de importancia de las mismas derivada del análisis del corpus textual.

MEAD es un software reportado en [RAD00] que genera resúmenes de múltiples documentos usando los centros de los grupos obtenidos a partir de la detección de tópicos en la colección. Por otra parte, en [MCK99] se identifican párrafos similares e intersección de frases similares en los párrafos para obtener un resumen conciso de múltiples documentos relacionados. En [LAR00b] se presenta un sistema para resumir noticias y obtener una estructura jerárquica del texto fuente.

En [WHI02] se presenta y evalúa un procedimiento aleatorio de búsqueda local de selección de oraciones a incluir en un resumen de múltiples documentos. Aquí, los autores focalizan su atención en la selección de bloques de oraciones adyacentes para construir el resumen de múltiples documentos. El reto es mejorar la inteligibilidad sin afectar excesivamente la información.

En [KEO99] se desarrolla un sistema de resumen de múltiples documentos para generar automáticamente un resumen conciso identificando y sintetizando semejanzas a través de un conjunto de documentos relacionados. Se presenta un enfoque único en la integración de técnicas estadísticas y análisis sintáctico para identificar párrafos similares, intersección de frases similares, y la generación de las expresiones que conformarán el resumen de la colección.

Se ha trabajado intensamente en la generación de resúmenes de noticias. Tal es el caso reportado en [JAM01], donde se definen los resúmenes temporales de actualidades dentro de un tópico de noticias, donde las historias son presentadas una por una y las oraciones de una historia deben ser clasificadas antes de que la siguiente historia pueda ser considerada.

Existen dos tipos de situaciones en las cuales los resúmenes de múltiples documentos pueden ser útiles: (1) cuando los documentos no están relacionados; (2) cuando los documentos están relacionados [GOL99]. Por tanto, es útil identificar inicialmente en una colección, si los documentos que la forman están relacionados o no, porque esto influye decisivamente en el resumen a obtener.

En la mayoría de los trabajos reportados en el área de la generación automática de múltiples documentos se tiene en cuenta un agrupamiento inicial del corpus, para después extraer las oraciones principales. De esta forma se verifica la homogeneidad del corpus antes de comenzar con la aplicación propiamente de las técnicas de generación de resúmenes de múltiples documentos. Resultados con estas características se reportan en [WHI02], [GOL00], [LAR00a] [MAN02] y [FUK99].

Por tal motivo, una tendencia al resumir múltiples documentos es verificar la homogeneidad del corpus agrupando los textos en función de sus semejanzas semánticas.

En esta investigación obtenemos el resumen de un corpus de documentos verificando previamente la homogeneidad del corpus textual. Por tanto, previo a la extracción de las principales oraciones agruparemos los documentos afines de la colección para verificar su homogeneidad. Es por eso que

a continuación definiremos qué es el agrupamiento de datos y describiremos las principales técnicas de agrupamiento, particularizando en el agrupamiento de documentos.

1.3 Técnicas de agrupamiento

El análisis de clusters o agrupamiento de datos es una actividad humana muy importante. Esta actividad usualmente forma las bases del aprendizaje y del conocimiento. El agrupamiento puede ser aplicado a disímiles campos. Por ejemplo, en los negocios para agrupar las necesidades comunes de consumidores, en la medicina para agrupar las enfermedades, y en la Biología, entre otros campos, para categorizar genes con funcionalidades similares.

La minería de textos no constituye una excepción respecto a la importancia de la aplicación de técnicas de análisis de clusters. El agrupamiento es una tarea importante en el correcto funcionamiento de muchos sistemas de minería de textos y de recuperación de información. Éste puede ser usado eficientemente para encontrar los vecinos más cercanos de un documento, para mejorar la calidad de sistemas de recuperación de información, en la organización y personalización de la información en motores de búsqueda, en la verificación de la homogeneidad de un corpus textual, en el resumen de colección de documentos y en la categorización de términos, entre otros.

En esta investigación abordaremos las principales técnicas de agrupamiento útiles en la verificación de la homogeneidad de una colección textual, así como su influencia en la calidad del resumen de la colección a obtener a partir de la formación de grupos de documentos afines.

El objetivo del agrupamiento de documentos es formar una colección de clusters (subconjuntos, grupos, clases) que cumplan las propiedades siguientes [HOP99]:

- La homogeneidad dentro de los clusters, i.e. los documentos que pertenecen al mismo cluster deben ser tan similares como se pueda.
- La heterogeneidad entre clusters, i.e. los documentos que pertenecen a clusters diferentes deben ser tan diferentes como se pueda.

Antes de mencionar los principales resultados en el agrupamiento de documentos, definiremos en qué consiste el agrupamiento de datos y describiremos las principales técnicas de agrupamiento que se reportan en la literatura.

1.3.1 ¿Qué es el agrupamiento de datos?

El agrupamiento es un proceso de división de un conjunto de datos u objetos en un conjunto de subclases significativas, llamadas clusters. Formalmente, dada una colección de n objetos descritos por un conjunto de p atributos, el objetivo del agrupamiento es derivar una división útil de los n

objetos en un número de clusters. Un cluster es una colección de objetos similares entre sí, basado en los valores de sus atributos, y puede ser tratado colectivamente como un grupo. El agrupamiento es útil para obtener una distribución interna del conjunto de datos [FUN02]. A continuación mencionaremos algunos requerimientos típicos del agrupamiento en la minería de datos [FUN02]:

Escalabilidad: Ser capaces de funcionar correctamente con grandes volúmenes de datos.

Alta dimensionalidad: Ser capaces de trabajar con espacios de gran dimensionalidad, descritos por un gran número de atributos.

Formas arbitrarias de los clusters: Ser capaces de asimilar diferentes formas de clusters en dependencia de las medidas utilizadas.

No ser sensitivos al orden de los datos de entrada: El orden de los objetos no puede influir en la calidad del agrupamiento a obtener.

Manipulación de datos ruidosos: Ser capaces de minimizar el impacto de los datos ruidosos.

Conocimiento previo del dominio: Ser capaces de reducir el número de parámetros iniciales que influyen en la calidad del agrupamiento y que deben ser suministrados por los usuarios, requiriéndose en conocimiento previo del dominio (e.g. especificar número de clusters a obtener).

1.3.2 Acerca de las principales técnicas de agrupamiento

Existen varias técnicas de agrupamiento, entre ellas, técnicas de agrupamiento incompleto o heurístico, técnicas de agrupamiento duro y determinista, técnicas de agrupamiento duro y con solapamiento, técnicas de agrupamiento probabilísticas, técnicas de agrupamiento borroso, técnicas de agrupamiento jerárquico, técnicas de agrupamiento basadas en funciones objetivos y técnicas de estimación de grupos [HOP99] [CHE98]. A continuación las describiremos brevemente:

Agrupamiento incompleto o heurístico. Se utilizan métodos geométricos y técnicas de representación o proyección. Los datos multidimensionales son analizados por la reducción de la dimensión usando *principal component analysis* para obtener una representación gráfica en dos o tres dimensiones. Los clusters son determinados posteriormente (e.g. utilizando métodos heurísticos basados en la visualización de los datos).

Agrupamiento duro y determinista. Se asigna cada dato exactamente a un cluster de modo que la partición de clusters defina una partición ordinaria del conjunto de los datos.

Agrupamiento duro y con solapamiento. Cada dato será asignado a un cluster al menos, o puede ser simultáneamente asignado a varios clusters.

Agrupamiento probabilístico. Para cada dato, se determina una probabilidad de distribución sobre los clusters que especifica la probabilidad con la cual un dato es asignado a un cluster[8].

Agrupamiento borroso. Son algoritmos de agrupamiento borroso puro donde los grados de pertenencia indican en qué medida un dato pertenece a los clusters. La suma de las pertenencias de cada dato a todos los clusters es igual a uno.

Agrupamiento jerárquico. Se agrupan los datos en un árbol de clusters. Un método es dividir los datos en varios pasos obteniendo clases con una granularidad cada vez más fina (*divisive*, construye el árbol *top-down*), y el otro, construye el árbol de una forma inversa, combina pequeñas clases en otras con una granularidad más gruesa (*agglomerative*, construye el árbol *bottom-up*).

Agrupamiento basado en funciones objetivos. Se basa en una función objetivo que asigna a cada posible partición un valor de calidad o error que tiene que ser optimizado, por tanto, conduce a un problema de optimización. La solución ideal es la partición cluster que obtenga la mejor evaluación.

Agrupamiento de estimación de clusters. Adopta una alternativa del algoritmo de optimización usado en la mayoría de métodos basados en función objetivo, pero usa ecuaciones heurísticas para construir particiones y estimar parámetros de cluster.

1.3.3 Diferentes medidas para comparar objetos

Al agrupar los objetos de un conjunto de datos, se requieren algunas medidas para cuantificar el grado de asociación entre ellos. Con este propósito, podemos utilizar una medida de distancia, o una medida de similitud o disimilitud. Algunos algoritmos de agrupamiento tienen un requerimiento teórico para el uso de una medida específica, pero lo más común es que el investigador seleccione qué medida utilizará con determinado método.

Una distancia ampliamente utilizada es la Euclidiana [GAB04]. Siguiendo esta distancia, podemos considerar que dos objetos son similares si la distancia entre ellos es cercana a cero y diferentes si la distancia entre ellos es cercana a uno. La distancia Euclidiana entre dos objetos O_i y O_j, descritos por k rasgos, es calculada a partir de la siguiente fórmula:

$$D(O_i, O_j) = \sqrt{\sum_{h=1}^{k} (O_{ih} - O_{jh})^2} \tag{1.19}$$

[8] Estas técnicas también son llamadas algoritmos de agrupamiento borrosos si las probabilidades son interpretadas como grados de pertenencia.

La métrica *Minkowski* es también ampliamente referenciada en la literatura [BER01]. Siguiendo esta métrica, se calcula la distancia entre dos objetos O_i y O_j, descritos por k rasgos y donde se cumple que $\gamma \geq 1$:

$$D(O_i, O_j) = \left(\sum_{h=1}^{k} \left| O_{ih} - O_{jh} \right|^{\gamma} \right)^{\frac{1}{\gamma}} \tag{1.20}$$

Cuando $\gamma = 1$, la forma de calcular la distancia entre los objetos se llama la métrica *Manhattan*. Si $\gamma = 2$, nos referimos a la distancia Euclidiana. Para los valores $\gamma \geq 2$, estamos en presencia de la métrica *Supermum* [BER01].

Otra forma de medir la distancia entre dos objetos O_i y O_j, descritos por k rasgos, es la métrica Canberra, calculada a partir de la siguiente fórmula [BER01]:

$$D(O_i, O_j) = \sum_{h=1}^{k} \frac{\left| \overrightarrow{O_{ih}} - \overrightarrow{O_{jh}} \right|}{\left(\left| \overrightarrow{O_{ih}} \right| + \left| \overrightarrow{O_{jh}} \right| \right)} \tag{1.21}$$

Otra forma muy utilizada de medir la distancia entre objetos es la distancia de *Hamming*. Tanto en su definición binaria a partir de la cantidad de atributos en que difieren dos objetos, como en sus variantes pesadas [DUC00]. En [KAR01] y en [DUC00] se presentan formas de medir la similitud entre objetos, considerando la medida como una relación asimétrica y teniendo en cuenta la probabilidad condicional de un objeto respecto al otro. Otro enfoque se muestra en [SAR00] [WIL97], donde obtienen la similitud entre dos objetos O_i y O_j calculando la correlación de *Pearson* entre ellos:

$$D(O_i, O_j) = \frac{\sum_{h=1}^{k} \left(O_{ih} - \overline{atributo_h} \right) \left(O_{jh} - \overline{atributo_h} \right)}{\sqrt{\sum_{h=1}^{k} \left(O_{ih} - \overline{atributo_h} \right)^2 \sum_{h=1}^{k} \left(O_{jh} - \overline{atributo_h} \right)^2}} \tag{1.22}$$

donde $\overline{atributo_h}$ es el valor promedio que toma el $atributo_h$ en el conjunto de datos.

Existe un gran número de medidas de similitud disponibles, sin embargo, existen pocos estudios comparativos de ellas y sus efectos en el agrupamiento. En la minería de textos, al comparar documentos con el objetivo de agruparlos, la determinación de la similitud entre documentos depende de la representación del documento y de los pesos que se le asignen al caracterizarlo. A partir de estudios realizados, se ha demostrado que al agrupar documentos, los coeficientes Dice,

Jaccard y Coseno, han reportado los mejores resultados [FRA92] [CRO02]. A continuación presentaremos tres formas de calcular la similitud *S* entre dos documentos D_i y D_j, con pesos asociados a los *k* términos que los describen.

El coeficiente Dice se define como sigue:

$$S(D_i, D_j) = \frac{2\sum_{h=1}^{k}\left(peso_{ih} \cdot peso_{jh}\right)}{\sum_{h=1}^{k} peso_{ih}^2 + \sum_{h=1}^{k} peso_{jh}^2} \tag{1.23}$$

Si el peso de los términos es binario, el coeficiente Dice se reduce a

$$S(D_i, D_j) = \frac{2C}{A+B} \tag{1.24}$$

donde *C* es el número de términos que D_i y D_j tienen en común, y *A* y *B* son el número de términos de D_i y D_j respectivamente.

Siguiendo la notación anterior, el coeficiente Jaccard se define como sigue:

$$S(D_i, D_j) = \frac{\sum_{h=1}^{k}\left(peso_{ih} \cdot peso_{jh}\right)}{\sum_{h=1}^{k} peso_{ih}^2 + \sum_{h=1}^{k} peso_{jh}^2 - \sum_{h=1}^{k}\left(peso_{ih} \cdot peso_{jh}\right)} \tag{1.25}$$

En [GAS01] se muestran variantes binaria y pesada de la medida de Jaccard para determinar la similitud entre palabras, no obstante, esta medida puede ser extendida para comparar otro tipo de contexto sintáctico.

La medida de Jaccard binaria se puede utilizar para comparar dos documentos teniendo en cuenta los términos que ellos comparten:

$$S(D_i, D_j) = \frac{C}{A+B} \tag{1.26}$$

donde *A* es el número de palabras que describen a D_i, *B* es el número de palabras que describen a D_j y *C* es el número de palabras que aparecen tanto en D_i como en D_j.

Una variante de la medida de Jaccard pesada es considerar un peso local (*PL*) y otro global (*PG*) para cada término que describe los documentos.

El peso global tiene en cuenta cómo muchos documentos diferentes se asocian con un término dado

$$PG(palabra_h) = 1 - \sum_{l=1}^{N} \frac{|p_{lh} \log p_{lh}|}{m_h} \tag{1.27}$$

donde N es el número total de documentos en la colección, m_k es el número de documentos en los cuales la $palabra_h$ ocurre y p_{lh} se calcula según la siguiente expresión:

$$p_{lh} = \frac{\text{frecuencia de la } palabra_h \text{ en } D_l}{\text{cantidad total de palabras en } D_l} \tag{1.28}$$

El peso local se calcula como:

$$PL(palabra_h, D_j) = \log(\text{frecuencia de la } palabra_h \text{ en } D_j) \tag{1.29}$$

El peso de un término es la multiplicación del peso global y el peso local:

$$peso(palabra_h, D_i) = PG(palabra_h) \cdot PL(palabra_h, D_i) \tag{1.30}$$

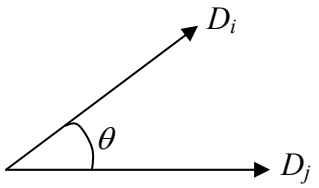
La similitud Jaccard entre dos atributos, siguiendo este cálculo de los pesos se plantea como:

$$S(D_i, D_j) = \frac{\sum_{h=1}^{k} \min(peso(palabra_h, D_i), peso(palabra_h, D_j))}{\sum_{h=1}^{k} \max(peso(palabra_h, D_i), peso(palabra_h, D_j))} \tag{1.31}$$

El coeficiente Coseno se define en la siguiente expresión [FRA92]:

$$S(D_i, D_j) = \frac{\sum_{h=1}^{k} (peso_{ih} \cdot peso_{jh})}{\sqrt{\sum_{h=1}^{k} peso_{ih}^2 \cdot \sum_{h=1}^{k} peso_{jh}^2}} \tag{1.32}$$

Este coeficiente es ampliamente utilizado al determinar la similitud entre documentos y se basa en el coseno del ángulo que existe entre ellos. Si el coseno del ángulo es cercano a uno, se consideran similares, y si es cercano a cero, los documentos son considerados diferentes. En la siguiente figura se muestra la gráfica de los vectores documentos, así como la expresión que describe el coseno del ángulo entre ellos.



$$\cos\theta = \frac{\overrightarrow{D_i} \cdot \overrightarrow{D_j}}{\|D_i\| \cdot \|D_j\|} \tag{1.33}$$

1.3.4 Principales algoritmos de agrupamiento en la minería de textos

A continuación haremos referencia a los principales algoritmos de agrupamiento que se han utilizado en la minería de textos, específicamente en el agrupamiento de documentos similares. Particularizaremos en aquellos que aplican técnicas duras y deterministas, las técnicas duras y con solapamiento, y las técnicas de agrupamiento borroso.

Generalmente en una colección de documentos, un documento no pertenece a un único grupo (e.g. agrupamiento de noticias, textos de divulgación científica, artículos científicos), por tanto el agrupamiento con solapamiento es muy útil, sin embargo, éste no da la posibilidad de conocer cuánto pertenece cada documento a cada grupo. En tal sentido, el agrupamiento borroso nos dará la posibilidad de agrupar los textos de un corpus de forma tal que conozcamos el grado de pertenencia de los textos a cada grupo. Una técnica de agrupamiento que tiene gran importancia en la minería de textos es el agrupamiento jerárquico, debido a que da la posibilidad de conocer cuán cercanos están los grupos obtenidos.

Un método muy utilizado es aplicar una red de Kohonen (*Self-Organizing Maps*) para agrupar documentos. En [NUR01] se utiliza esta red para agrupar documentos similares, donde no es necesario especificar el número de grupos que se desea obtener. Además se obtienen las palabras claves que describen cada grupo. Una deficiencia es que hay que definir el tamaño de la red manualmente.

Otro ejemplo de algoritmo de agrupamiento duro y determinista es Autoclass [LAR00]. Este, al igual que el anterior, no requiere especificar el número de grupos a obtener y además de obtener los grupos de textos, obtiene las palabras claves de cada grupo. Este algoritmo ha aportado mejores resultados que la red Kohonen en el agrupamiento de documentos.

Un clásico en el análisis de clusters es el algoritmo *k-means* [QUE67] [JAN97] [BER04], éste tiene la desventaja que requiere que el número de clusters a obtener sea especificado a priori, por tanto requiere un cierto conocimiento del dominio, ya que es sensible a cómo se hizo inicialmente la partición. El algoritmo *k-means* agrupa los documentos teniendo en cuenta la similitud que existe entre ellos. A partir de este algoritmo se han derivado varios como el *Batch k-means* e *Incremental*

k-means. El primero de estos tiene dos desventajas, una de ellas es que la calidad de la partición final depende de una buena selección de la partición inicial y además puede quedar atrapado en mínimos locales. El segundo, el *Incremental k-means*, resuelve las dos desventajas del *Batch k-means*, pero es más lento. Realizando mejoras a estos dos últimos algoritmos se obtuvo el algoritmo *Means* [BER04].

En [BAO00a] y [BAO00b] se muestra un nuevo método de agrupamiento de documentos que utiliza una extensión de la teoría clásica de los conjuntos aproximados (*Rough Set Theory*), llamada *Tolerance Rough Set* que permite formar clases de tolerancia de las palabras y utilizarlas para realizar el agrupamiento de documentos.

Los algoritmos mencionados hasta el momento son duros y deterministas. En la minería de textos de los algoritmos jerárquicos también su muy usados, mencionaremos uno reciente, el algoritmo *Principal Direction Divisive Partitioning* (PDDP) que no requiere que el número de clusters sea fijado inicialmente porque hace una subdivisión sucesiva de la colección inicial hasta detenerse cuando se cumpla cierto criterio de calidad [BOL98]. Este algoritmo no es basado en ninguna medida de distancia ni de similitud, y toma como ventaja lo dispersa que es la matriz de términos por documentos. Devuelve grupos de documentos, pero generalmente su salida es utilizada como entrada al algoritmo *Means* para mejorar su eficiencia y no tener que fijar manualmente el número inicial de clusters ni la partición inicial [BER04]. Otro algoritmo que usualmente se combina con el algoritmo *Means* es el algoritmo *Spherical Principal Directions Divisive Partitioning* (sPDDP), también jerárquico y que sigue la idea de [BOL98]. Este algoritmo no requiere que el número de clusters sea fijado inicialmente y mejora los resultados de PDDP. En [BAO00a] y [BAO00b] también se muestra una variante del algoritmo duro y determinista que utiliza el modelo *Tolerance Rough Set*, para realizar un agrupamiento jerárquico.

El algoritmo *Ando* realiza el agrupamiento con objetivos específicos, ya que su idea general es identificar clusters pequeños en contextos limitados. Este algoritmo tiene varias desventajas, una de ellas es que es poco escalable, y funciona utilizando *Latent Semantic Indexing* y los vectores bases no son siempre ortogonales, aspecto esencial al buscar los valores y vectores propios [BER04].

Al agrupar documentos es muy útil obtener simultáneamente los grupos de documentos y las palabras claves de cada grupo. Así, se dividir la colección de documentos en categorías más significativas y se genera automáticamente una descripción compacta de cada cluster en términos no sólo de los valores de los atributos, sino también de su relevancia.

Siguiendo la idea antes expuesta, el algoritmo *Simultaneous Clustering and Attribute Discrimination* (SCAD), es un algoritmo que utiliza una técnica dura y determinista logrando, simultáneamente, obtener los grupos de documentos similares y pesar los rasgos. SCAD tiene varias ventajas, una de

ellas es pesar los rasgos continuos, lo cual provee una representación de la relevancia de los rasgos más rica que la selección de los rasgos binaria. Además, SCAD aprende una representación de la relevancia de los rasgos diferente para cada cluster. Este algoritmo utiliza la distancia Euclidiana para calcular la similitud entre documentos. Para el caso especial de documentos textuales se conoce que la distancia Euclidiana no es apropiada, de forma general, esta medida no es buena en problemas de alta dimensionalidad. En [BER04] se muestra una extensión de este algoritmo que permite simultáneamente agrupar documentos textuales y dinámicamente pesar el conjunto de palabras claves. Este nuevo enfoque es llamado *Simultaneous Keyword Identification and Clustering of text documents* (SKWIC), y es computacional y conceptualmente simple. Este algoritmo es una extensión del *k-means* y funciona mejor que éste cuando no todos los rasgos son igualmente relevantes. SKWIC utiliza una medida de disimilitud basada en el coeficiente coseno para comparar los documentos, pero puede ser adaptado a otras medidas de disimilitud. La forma de pesar los términos puede ser modificada. Este algoritmo requiere que sea especificado el número inicial de clusters y devuelve la colección de clusters y la relevancia de las palabras por clusters.

Es muy importante al agrupar colecciones de textos utilizar algoritmos que no creen particiones, porque varios documentos de una colección pueden abordar más de un tema, por tanto, pertenecer a varios clusters. Los algoritmos que aplican técnicas duras y con solapamiento pueden resolver este problema. Ejemplos de ellos son el algoritmo *Star* [ASL98] y el algoritmo *Extended Star* [GIL03]. Ambos algoritmos tiene la ventaja que no requieren que el número de clusters sea especificado inicialmente. Sin embargo, el primero de ellos tiene la desventaja que el algoritmo depende del orden de los datos y esto puede provocar la construcción de clusters ilógicos. El algoritmo *Extended Star* supera esta desventaja, es decir, en la formación de los clusters no influye el orden de los datos.

Los algoritmos duros y con solapamiento ayudan a resolver el problema de la inclusión de un documento a más de un grupo, pero no se conoce en qué grado pertenecen los documentos a los grupos. Por tal motivo, es muy útil realizar el análisis de clusters aplicando algoritmos de agrupamiento borrosos. Un clásico de los algoritmos de agrupamiento siguiendo esta técnica es el *Fuzzy c-means* [BEZ73] [JAN97] [BER04], que reconoce nubes esféricas de puntos en un espacio *p*-dimensional, y asigna un grado de pertenencia de los documentos a los clusters. Siguiendo la idea del *Fuzzy c-means* se han desarrollado varios algoritmos, entre ellos el algoritmo *Relational Alternating Cluster Estimation* [RUN01] y el algoritmo *Simultaneous Soft Clustering and Term Weighting of Text Document* (Fuzzy SKWIC) [BER04]. Ambos algoritmos requieren que el número de clusters sea especificado inicialmente y además de devolver la colección de clusters, calculan simultáneamente la relevancia de las palabras en los grupos.

1.4 Conclusiones parciales

A partir de la consulta de la bibliografía internacional y nacional realizada, así como de otras fuentes referenciales se pueden extraer las conclusiones fundamentales siguientes:

- Cuatro elementos fundamentales centran la representación textual: la transformación del corpus, la extracción de términos, la reducción de dimensionalidad, y la normalización y el pesado de la representación. Así como la representación VSM como una de las más utilizadas en el análisis léxico de documentos. Esta representación será que la utilizaremos en el modelo propuesto en el capítulo 2. Además, a partir de las variantes presentadas, consideramos que el uso de las palabras simples como términos indexados es adecuado para nuestro problema por dos razones. Primero, esa representación de textos puede ser implementada eficientemente. Segundo, y la más importante, esta representación no se construye sobre un conocimiento del dominio o en dependencia de un lenguaje, por tal motivo, ella puede ser directamente aplicada a cualquier colección de documentos.
- Si bien existen numerosos técnicas para el agrupamiento de documentos, se seleccionan las técnicas duras y deterministas, borrosas, y duras y con solapamiento a través de los algoritmos SKWIC, *Fuzzy* SKWIC y *Extended Star*, respectivamente. La selección de los algoritmos SKWIC y *Fuzzy* SKWIC se debe a que simultáneamente logran agrupar y calcular la relevancia de las palabras por grupos, cuestión fundamental en nuestro trabajo ya que el objetivo del agrupamiento es obtener posteriormente resúmenes extractos de los grupos obtenidos, y por tanto, la relevancia de las palabras es fundamental. *Extended Star* fue seleccionado debido a que no requiere un conocimiento previo del dominio en la determinación de grupos a obtener.
- Resumir el documento más representativo y resumir el documento más representativo más el resto de los documentos de la colección permite obtener resúmenes más o menos concentrados en función de la técnica seleccionada.
- El análisis del estado del arte ha permitido conocer la situación actual en que se encuentra la representación, agrupamiento y resumen de corpus textuales como parte del paradigma de la minería de textos, manifestándose la necesidad de más estudios dirigidos a la integración de estas áreas para obtener procesamientos textuales más completos. De manera tal que se incursione en modelos y procedimientos que permitan la obtención de resúmenes extractos de corpus textuales a partir del agrupamiento previo de documentos afines. El análisis bibliográfico evidencia que existen muchos estudios acerca de la representación, agrupamiento y resumen de corpus textuales de manera aislada y no un modelo que conciba estas etapas de forma integral.
- Si bien existen numerosos estudios sobre técnicas y algoritmos que permiten el procesamiento de corpus textuales que marchan en paralelo con el desarrollo teórico, en el análisis del estado de

la práctica menos avances se han hecho patente en el área de la elaboración de modelos y procedimientos para el desarrollo de investigaciones en el campo de la minería de textos que persiguen dotar para las investigaciones en esta área de una herramienta flexible y extensible que permita transitar por varias etapas del procesamiento textual hasta obtener resúmenes extractos. Lo anterior constituye un problema aún no resuelto y evidencia que la falta de integración de estas técnicas limitan el desarrollo de investigaciones en este campo.

Capítulo 2 Modelo para el agrupamiento de documentos afines y su ulterior resumen a partir de corpus textuales

A partir de la revisión bibliográfica realizada acerca de la representación textual, el agrupamiento de documentos y el resumen de corpus textuales, en este capítulo se propone un modelo que permite realizar el agrupamiento de documentos afines para su ulterior resumen partir de una colección de documentos.

2.1 Modelo conceptual propuesto que permite el agrupamiento de textos para la obtención de resúmenes extractos a partir de corpus textuales

El procesamiento de documentos textuales debe tomar como punto de partida un modelo conceptual que lo fundamente y cuyo valor metodológico consiste en integrar un conjunto de conceptos objetivamente interrelacionados, que sirven de base para establecer procedimientos para la obtención de resúmenes extractos de corpus textuales. En conformidad con lo expuesto, en esta investigación se propone un modelo que, no sólo explica el problema científico formulado y al mismo tiempo muestra la solución conceptual a éste, sino que también está dirigido a ayudar y orientar a investigadores y desarrolladores en el campo de la minería de textos (observe la figura 2.1). Como es característico a todo modelo se le definen objetivos, principios, premisas, entradas, salidas, procedimientos y control.

El objetivo del modelo es dotar a los investigadores y desarrolladores en el campo de la minería de textos de una herramienta que posibilite el agrupamiento de textos para la obtención de resúmenes extractos a partir de grupos homogéneos de documentos afines.

Los principios en que se sustenta el modelo son:

Consistencia lógica. En función de la ejecución de sus pasos en la secuencia planteada en la correspondencia con la lógica de la ejecución de este tipo de estudio.

Flexibilidad. Por la potencialidad de aplicarse a otras áreas de la minería de textos con características no necesariamente idénticas a las seleccionadas dentro del universo de estudio y por la capacidad de actualización y reajuste en los diferentes procesos y procedimientos específicos.

Parsimonia. Referido a su cualidad de ser "simple" dentro de la complejidad inherente que presentan estos estudios.

Racionalidad. De acuerdo con la relación gasto - beneficio que se requiere para su implementación.

Perspectiva o generalidad. Dada por la posibilidad de su extensión como instrumento metodológico para ejecutar estos estudios en otros procesos dentro de la minería de textos.

Las premisas fundamentales del modelo se refieren a continuación:

- La colección de documentos en idioma Inglés debe estar almacenada en un fichero texto. El formato de este fichero requiere que se especifiquen delimitadores de documentos, párrafos y oraciones[9].
- Es necesaria la especificación de diccionarios para la lematización, homogeneidad ortográfica, contracciones, abreviaturas y palabras gramaticales.

El modelo conceptual reúne todos los elementos considerados relevantes e imprescindibles cuando se pretende por cada funcionalidad brindar varios métodos que la sustenten, teniendo en cuenta uno de los principios en los que se sustenta, su flexibilidad. Realizando las adecuaciones pertinentes el modelo puede aplicarse al procesamiento de textos en otros idiomas.

La entrada al modelo es una colección de documentos en idioma inglés y las salidas principales son la representación VSM del corpus, la relevancia de las palabras en el corpus, la colección de grupos homogéneos de documentos afines, las palabras claves por grupos de documentos y el resumen extracto de grupos textuales.

El modelo incluye el proceso de control para la evaluación de los resultados alcanzados en el agrupamiento a través de las medidas de calidad definidas.

Hasta aquí se ha intentado definir un marco conceptual para obtener un resumen extracto de un corpus textual partiendo de la verificación de la homogeneidad del mismo utilizando métodos de agrupamiento. Se reconoce que pudieran seguirse diferentes variantes con tales propósitos; sin embargo, se ha tratado de captar en el modelo propuesto aquellos elementos esenciales que cualquier investigador o desarrollador en el campo de la minería de textos debe tomar en consideración al pretender agrupar textos para obtener resúmenes extractos de un corpus textual. Como se hizo alusión con anterioridad, al modelo se le definen, además, procedimientos que posibilitan su implementación.

[9] La delimitación automática de oraciones no es un problema trivial, por eso se presupone que ellas están previamente delimitadas.

2.2 Concepción teórica-metodológica del procedimiento general para obtener el resumen extracto de corpus textuales

Como parte del modelo conceptual se desarrolla un procedimiento metodológico general que incluye varios procedimientos específicos, estructurados en cuatro etapas con sus fases correspondientes que en su conjunto resumen el contenido del modelo.

Las etapas del procedimiento general son (observe la figura 2.2):

1. Representación del corpus textual.
2. Agrupamiento de documentos.
3. Extracción de palabras claves de grupos textuales.
4. Obtención del resumen extracto de cada grupo de documentos.

2.2.1 Etapa 1. Representación del corpus textual

Según la revisión bibliográfica realizada en el capítulo 1, la representación del corpus incluye su transformación, la extracción de términos, la reducción de la dimensionalidad (selección de rasgos y reparametrización), y el pesado y normalización de la representación. Estos cuatro elementos necesarios en la representación, los hemos reflejado en el modelo a través de las fases siguientes:

1.1 Transformar el corpus de textos.

1.2 Obtener representación VSM del corpus.

1.3 Normalizar y pesar la matriz.

1.4 Calcular las medidas de calidad de términos.

1.5 Reducir la dimensionalidad de la representación.

2.2.1.1 Transformar el corpus de textos

La primera fase contempla la transformación de un corpus de textos escrito en idioma Inglés. Remitiéndonos al capítulo 1, la transformación está, a su vez, dividida en dos operaciones. La primera, reconocer los componentes textuales desde diferentes formatos. La segunda, transformar la secuencia resultante de tokens [LAN01]. Aunque el reconocimiento de componentes textuales debe comprender el manejo de textos desde diferentes formatos, en este modelo omitimos esta operación. Partimos de una colección en formato texto y así no es necesario incluir la operación que permita el

reconocimiento de formatos tales como postscripts, pdf, y HTML, entre otros. El formato de este fichero requiere que se especifiquen delimitadores de documentos, párrafos y oraciones[10].

En nuestro modelo la fase de transformar el corpus incluye la extracción de términos. Se considera un término como un conjunto de caracteres que pueden ser combinaciones de letras, apóstrofes (’), guiones (-) y puntos (.). Consideramos el uso de las palabras simples como términos indexados por dos razones. Primero, esa representación de textos puede ser implementada eficientemente. Segundo, y la más importante, esta representación no se construye sobre un conocimiento del dominio o en dependencia de un lenguaje, por tal motivo, ella puede ser directamente aplicada a cualquier colección de documentos. Además, en lenguajes como el Inglés, una definición compleja de los términos indexados no provoca una mejora significativa en los resultados. Para otros lenguajes, un análisis lingüístico profundo puede ser más beneficioso. En [LAN01] se concluye que el beneficio que podemos tomar del análisis lingüístico es fuertemente dependiente del lenguaje y del dominio.

A partir de una colección en formato texto se obtienen tokens a los cuales se les aplican transformaciones. Algunas de estas transformaciones son convertir todos los caracteres a mayúscula, la sustitución de las contracciones por sus expansiones, de las abreviaturas por sus formas completas y la eliminación de números y símbolos.

Ya conocemos que el modelo propuesto considera una representación VSM del corpus textual a partir de los tokens indexados. Teniendo en cuenta la matriz que queramos formar (e.g. términos – documentos, términos – oraciones), es necesario tener en cuenta la cantidad de veces que aparece cada término en cada documento o en cada oración respectivamente. Aunque parezca sencillo, en un corpus de sólo 1 MB puede haber un total aproximado de 150 000 palabras y alrededor de 15 000 palabras distintas. Por tanto, es necesario tener en cuenta qué tokens indexar. Es por eso que en esta fase de transformación se introducen técnicas de reducción de datos por reparametrización.

Esta reducción se logra realizando la lematización y verificando la homogeneidad ortográfica. Se etiquetan cada uno de los términos en gramaticales o no, permitiendo en la fase de generación de la representación VSM la incorporación o no de estos términos.

Cada una de las transformaciones antes mencionadas, así como etiquetar términos gramaticales se concibe en el modelo a partir del uso de diccionarios.

Uso de diccionarios para la etapa de transformación de corpus textuales

[10] La delimitación automática de oraciones no es un problema trivial, por eso se presupone que ellas están previamente delimitadas.

En el modelo propuesto se utilizan diccionarios para realizar la lematización, homogenizar ortografía, sustituir contracciones y abreviaturas, y para identificar las palabras gramaticales. A continuación describiremos como están confeccionados cada uno de estos diccionarios:

- Diccionario para la lematización: Cada palabra derivada está acompañada de la palabra raíz.

  Ejemplo:

PLAYED	PLAY
PLAYING	PLAY
PLAYS	PLAY

- Diccionario para homogeneidad ortográfica: Por cada palabra que no coincida en ambos idiomas aparece la palabra en Inglés británico y seguidamente su correspondiente en Inglés americano.

  Ejemplo:

COLOURFUL	COLORFUL
COLOUR	COLOR

- Diccionario de contracciones: Aparece por cada contracción el conjunto de palabras que la forman.

  Ejemplo:

I'LL	I WILL
I'M	I AM
I'VE	I HAVE

- Diccionario de abreviaturas: Aparece por cada abreviatura el conjunto de palabras que la forman.

  Ejemplo:

AT&T	AMERICAN TELEPHONE AND TELEGRAPH
U.K.	UNITED KINGDOM

- Diccionario de palabras gramaticales: Está compuesto por una lista de las palabras gramaticales en el idioma Inglés.

  Ejemplo:

  BUT
  BY
  CAN

2.2.1.2 Obtener representación VSM del corpus

Esta fase permite obtener una representación VSM del corpus textual a partir de los tokens indexados. Según se reporta en la literatura, la representación VSM generalmente contempla la formación de vectores documentos descritos a partir de la frecuencia de los términos. En el modelo que proponemos dicha matriz puede conformarse con términos indexados en sus filas y todos los documentos de la colección en sus columnas, donde las celdas representan la frecuencia de aparición de cada término en cada documento, o puede construirse considerando en las columnas cada una de las oraciones o párrafos que conforman los documentos del corpus, por tanto, cada celda corresponde a la frecuencia de aparición de los términos en la oración o párrafo, respectivamente. De esta forma, la representación espacio vectorial no sólo está enmarcada en el contexto de documentos, sino que contempla la oración o incluso el párrafo como otro contexto posible.

Es importante señalar que las transformaciones consideradas en el subepígrafe 2.2.1.1 pudieran hacerse después de generada la matriz como parte de un proceso de reducción de dimensionalidad, pero en el modelo sugerimos hacerlas primero, porque de lo contrario, la dimensión de la matriz podría ser muy grande. Esto se debe a que palabras como “playing”, “play”, “played” serían consideradas como diferentes, observe la tabla 2.1. Sin embargo, haciendo las transformaciones primero, todas las palabras que tienen el mismo lema serán consideradas en la matriz como una sola (e.g. play), observe la tabla 2.2. De esta forma reducimos el número de términos indexados en la matriz.

Término\ Documento	Doc-1	Doc-2	…	Doc-k
⋮	⋮	⋮	⋮	⋮
playing	n_1	n_2	…	n_k
played	m_1	m_2	…	m_k
play	o_1	o_2	…	o_k
⋮	⋮	⋮	⋮	⋮

Tabla 2.1 Matriz generada sin realizar previamente la lematización.

Termino\ Docuemnto	Doc-1	Doc-2	…	Doc-k
⋮	⋮	⋮	⋮	⋮
play	$n_1+o_1+m_1$	$n_2+o_1+m_1$	…	$n_k+o_k+m_k$
⋮	⋮	⋮	⋮	⋮

Tabla 2.2 Matriz generada realizando primeramente la lematización.

Al construir la representación incluimos otra variante de reducción de dimensionalidad por selección de rasgos, permitiendo o no la incorporación en la misma de las palabras gramaticales. Esto es posible debido a que en la fase de transformación cada una de las palabras se etiquetó como gramatical o no, consultando el diccionario de palabras gramaticales.

2.2.1.3 Normalizar y pesar la matriz

Trabajar directamente con las frecuencias absolutas de los términos en la representación VSM no arroja buenos resultados en el agrupamiento.

Respecto a la normalización, las posibles causas son, por ejemplo, que la frecuencia absoluta está condicionada al tamaño del documento. En un documento pequeño una palabra que existe *n* veces pude ser relevante, sin embargo, con la misma frecuencia en un documento grande puede que no tenga ningún valor. Documentos largos usualmente tienen más términos indexados distintos. Y con un número grande de términos indexados presentes en un documento, la posibilidad de encontrar términos que se hagan corresponder en otros documentos es mayor. Consecuentemente, documentos grandes tienen mayores posibilidades de ser similares a otros documentos si ellos no han sido normalizados. Es más eficiente normalizar los vectores pesados antes de involucrar estas operaciones en el cálculo de la similitud (e.g. al agrupar documentos). En nuestro modelo nos abstraemos de la variedad de longitudes de documentos normalizando a partir de la división de cada frecuencia absoluta de aparición de un término *t* en un documento *d*, por la suma de los componentes del vector pesado del documento como se muestra en la expresión 2.1.

$$w(d,t) = \frac{tf_d(t)}{\sum_{j=1}^{m} tf_d(t_j)} \tag{2.1}$$

Como comentamos en el subepígrafe 1.1.4, la variante más utilizada y que permite obtener mejores resultados en el procesamiento posterior de los corpus textuales es TF-IDF. Es por eso, y además porque los métodos de agrupamiento así lo requieren, que proponemos pesar la representación VSM a partir de la aplicación de las formas de calcular TF-IDF propuestas por [MAN00] y [BER04] (observe las expresiones 1.16 y 1.17).

2.2.1.4 Calcular las medidas de calidad de términos

Las medidas de calidad de los términos tienen un papel importante en la reducción de dimensionalidad por selección de rasgos en la representación del corpus de textos. Ellas, al brindar un valor de cuán bueno es cada término para representar el corpus de textos, permiten que sean

fácilmente utilizadas para escoger los mejores términos y eliminar los de peor valor. Además, para un especialista, la calidad de los términos puede ser valiosa para hacer un análisis de un corpus.

Esta fase incluye el cálculo de varias de las medidas mencionadas en el subepígrafe 1.1.3.1. Entre ellas están la entropía [NUR01], la calidad de términos en sus dos variantes [BER04] y las medidas estadísticas skewness y kurtosis [FUK99]. Observar las expresiones de la 1.8 a la 1.12.

Hemos incluido cuatro criterios adicionales: frecuencia en el corpus, rango en el corpus, frecuencia como palabra clave y rango de palabras, descritos a continuación.

Frecuencia en el corpus. Es la misma expresión definida en el subepígrafe 1.1.3.1 como *tf*(*t*) en la fórmula 1.3 y no es más que la frecuencia del término en todo el corpus

$$\mathrm{FC}(t) = tf(t) \tag{2.2}$$

Rango en el corpus. Coincide con la expresión *n*(*t*) definida en el anexo 1 y es la cantidad de documentos en que el término aparece al menos una vez

$$\mathrm{RC}(t) = n(t) \tag{2.3}$$

Frecuencia como palabra clave. Para el cálculo de esta medida primero se determina por documentos la media de las frecuencias de todos los términos, luego, para cada término se recorre cada documento y se suman aquellos valores de frecuencia que son mayores que la media del documento.

$$\mathrm{FP}(t) = \sum_{i=1}^{n} f_{d_i}(t) \quad \forall i \mid f_{d_i}(t) > \mathrm{media}(d_i), \text{ donde } \mathrm{media}(d_j) = \frac{\sum_{i=1}^{m} f_{d_j}(t_i)}{m} \tag{2.4}$$

donde *m* es la cantidad de términos del corpus.

Rango de palabras. Es muy similar al criterio anterior, pero en lugar de sumar la frecuencia en cada documento donde el valor de frecuencia es mayor que la media del documento, se suma la cantidad de documentos donde se cumple esta condición

$$\mathrm{RP}(t) = \sum_{i=1}^{n} i \quad \forall i \quad \mid f_{d_i}(t) > \mathrm{media}(d_i), \text{ donde } \mathrm{media}(d_j) = \frac{\sum_{i=1}^{m} f_{d_j}(t_i)}{m} \tag{2.5}$$

Finalmente, a diferencia de las medidas expuestas en el subepígrafe 1.1.3.1, el resultado de estas medidas permite clasificar los términos en tres clases posibles. La clase 1 indica los peores términos,

la clase 3 los mejores términos y la clase 2 considera aquellos términos con una calidad intermedia. Para obtener la clase correspondiente a cada término son necesarios dos pasos:

1. Calcular la medida para todos los términos.
2. Para cada término comparar su valor con la media. Si el valor de la medida está por encima de la media de todos los valores obtenidos se le asigna al término la clase 3, si el valor es 1 se le asigna la clase 1 y si no se le asigna clase 2. Observe la expresión 2.6.

$$\text{clase}(t) = \begin{cases} 1 & \text{si medida}(t) > \text{media} \\ 3 & \text{si medida}(t) = 1 \\ 2 & \text{en otro caso} \end{cases} \tag{2.6}$$

Nótese que la media es calculada a partir de los resultados de aplicar la medida a cada término.

2.2.1.5 Reducir la dimensionalidad de la representación

Aunque en las fases de transformación y obtención de la representación hemos incluido algunos elementos de reducción de dimensionalidad, aún estamos considerando muchos términos para someternos directamente a una etapa de agrupamiento. Es por eso, que el modelo propuesto incluye la posibilidad de reducir la dimensionalidad de la representación como una fase dentro de la etapa de representación textual.

El modelo propuesto considera en esta fase reductores de dimensionalidad basados en la selección de rasgos teniendo en cuenta las medidas de calidad de los términos descritas en el subepígrafe 2.2.1.4. Siguiendo estas ideas desarrollamos cuatro métodos diferentes para reducir la dimensionalidad de la representación VSM en nuestro modelo. A continuación cada uno de ellos es descrito.

Reducir basándonos en un umbral de calidad. Permite escoger aquellos términos cuyo valor de calidad supera un umbral. Es posible establecer dos umbrales: uno superior y otro inferior. Así, podamos tomar los elementos mayores que el umbral inferior, los menores que el umbral superior o la intersección de estos dos, es decir, los comprendidos entre estos dos umbrales.

Seleccionar los *n* mejores términos. Reduce el conjunto de términos a una cantidad *n* escogiendo o los de mayor o los de menor valor de calidad de términos. Permitir tanto los mayores como los menores valores se debe, a que en función de la semántica de las medidas, la calidad puede estar dada por los menores o por los mayores valores. Esta cantidad puede ser especificada en función de un número *n* de elementos o en un porciento *p* de la totalidad de términos que indique aquellos de mayor calidad.

Seleccionar los términos mayores que la media de calidad. Se determina la calidad promedio después de ser calculada la medida de calidad para todos los términos y se escogen aquellos términos cuyo valor excede la media. Teniendo en cuenta la semántica de la medida de calidad de términos seleccionada, es posible seleccionar o los términos cuyo valor de calidad es mayor que la media o los de calidad menor que la media.

Reducir por frecuencia de rango. Este es un reductor de dimensionalidad especial que se incorpora al modelo y utiliza los cuatro criterios mencionados en el subepígrafe 2.2.1.4: frecuencia en el corpus, rango en el corpus, frecuencia de palabras y rango de palabras. Su funcionamiento se basa en la selección de aquellos términos que, a partir de su identificación en las clases 1, 2 o 3 en función de los criterios de calidad definidos en las expresiones de la 2.2 a la 2.6, sean clasificados en clase 3 según los cuatro criterios. Los términos que poseen clase 3 según los cuatro criterios caracterizan el corpus de manera general pues son los que tienen valores por encima de la media de repeticiones en cada una de las dimensiones consideradas (frecuencia en el corpus absoluta y como palabra clave y rango en el corpus absoluto y como palabra clave).

Es importante destacar que las tres últimas fases descritas dentro de la etapa de representación textual no tienen un orden predeterminado de aplicación. No obstante, sugerimos primero normalizar y pesar la matriz, después calcular la calidad de los términos y aplicar técnicas de reducción de dimensionalidad, excepto para algunos criterios de calidad y reductores asociados a éstos que requieren su aplicación con los valores de frecuencia absoluta de aparición de los términos.

2.2.2 Etapa 2. Agrupamiento de documentos

Partiendo de una representación VSM de la colección de documentos, ya sea modificada o no por la aplicación de alguna técnica de normalización, pesado de la matriz, reducción de dimensionalidad o combinación de estas, es posible agrupar aquellos documentos [11] que sean similares por su contenido. En el modelo hemos considerado tres métodos de agrupamiento. Estos algoritmos requieren calcular la similitud que existe entre los documentos. Es por eso que las fases que componen esta etapa son:

2.1 Seleccionar el algoritmo de agrupamiento o combinación de algoritmos de agrupamiento a utilizar.

2.2 Calcular la similitud entre vectores.

[11] Nótese que generalmente nos referimos a agrupar documentos, sin embargo, en dependencia de la forma que haya sido construido la representación VSM, pudiera ser agrupar oraciones o párrafos similares.

2.2.2.1 Seleccionar el algoritmo de agrupamiento o combinación de algoritmos de agrupamiento a utilizar

A partir de la revisión bibliográfica realizada sobre los métodos de agrupamiento que son frecuentemente utilizados en la minería de textos, hemos decidido incluir en el modelo propuesto los algoritmos:

- *Simultaneous Keyword Identification and Clustering of Text Documents* (SKWIC) que realiza un análisis de clusters duro y determinista [BER04].
- *Simultaneous Keyword Identification and Fuzzy Clustering of Text Documents* (*Fuzzy* SKWIC) que utiliza una técnica de análisis de cluster borrosa [BER04].
- *Extended Star* que aplica una técnica de análisis de cluster dura y con solapamiento [GIL03].

Los tres algoritmos tienen en común que parten de una representación VSM del corpus textual y devuelven una colección de clusters. La estructura de los clusters varía en dependencia de los algoritmos de agrupamiento.

Los dos primeros nos brindan una colección de clusters, donde cada cluster tiene el vector centro de cluster, una lista de la relevancia de las palabras en el cluster y una lista de documentos con sus particularidades si es SKWIC o *Fuzzy* SKWIC. En el caso específico de SKWIC, al ser duro y determinista, cada cluster tiene la lista de los documentos que lo conforman y el algoritmo crea una partición. En el caso del algoritmo *Fuzzy* SKWIC, todos los documentos de la colección pertenecen a todos los clusters, pero con un grado de pertenencia asociado, por tanto la lista de documentos de cada cluster tiene por cada documento su grado de pertenencia al cluster. Existen varios algoritmos que se aplican al agrupamiento de documentos y que utilizan estas técnicas, sin embargo, hemos decidido incluir éstos al modelo porque ambos logran calcular la relevancia de las palabras a los clusters simultáneamente al proceso de agrupamiento de documentos y porque utilizan una variante del coeficiente Coseno para calcular la disimilitud entre documentos, medida que tiene muy buenos resultados en el agrupamiento de textos. Los documentos, por sus características, pueden pertenecer a más de un grupo, de ahí la utilidad de incorporar *Fuzzy* SKWIC, aunque SKWIC haya sido incorporado. Ambos algoritmos tienen la desventaja que requieren que el número de clusters a obtener sea especificado a priori, por tanto, es necesario tener conocimiento del dominio para poder definir ese valor y en la mayoría de las aplicaciones no contamos con el conocimiento suficiente. Otra desventaja de estos algoritmos es que en la primera iteración se seleccionan aleatoriamente los centros de clusters, por tanto, lograr que los centros queden estabilizados puede consumir varias iteraciones, además, debido a esa aleatoriedad, el algoritmo SKWIC puede devolver centros de clusters vacíos.

A partir de las desventajas mencionadas, decidimos incluirle al modelo un tercer algoritmo de agrupamiento, el *Extended Star* [GIL03]. Este algoritmo también retorna una colección de clusters, donde cada cluster tiene señalado el documento que es centro del cluster, así como una lista con los documentos de la colección que pertenecen a él. Una desventaja de este algoritmo es que no calcula, simultáneamente al proceso de agrupamiento, la relevancia de las palabras a los clusters. Una gran ventaja, es que no requiere un conocimiento previo del dominio, porque no es necesario fijar el número de clusters inicialmente. El algoritmo *Extended Star* tiene otra ventaja, y es que permite calcular la similitud entre vectores utilizando diferentes medidas. Este algoritmo se puede utilizar para el agrupamiento de documentos y los mejores resultados se esperan con la similitud Coseno. Independientemente de la calidad del agrupamiento que se pueda obtener con *Extended Star*, la idea de su incorporación en el modelo es utilizarlo como un paso previo a los dos algoritmos anteriormente mencionados. De esta forma, esta etapa se describe en la figura 2.3 donde a partir de la salida del método *Extended Star* se inicializa el número de clusters, así como sus centros, en los algoritmos SKWIC y *Fuzzy* SKWIC.

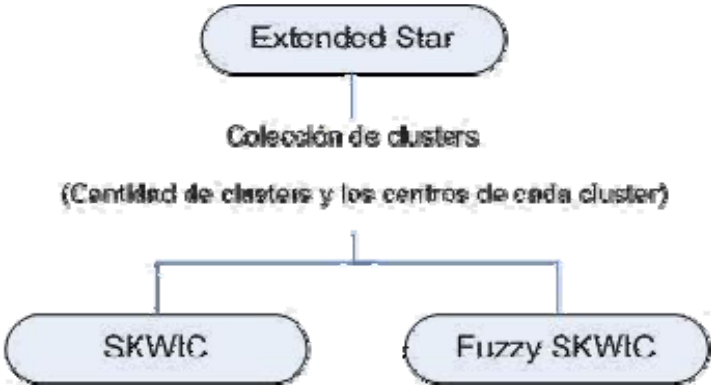


Figura 2.3 Combinación de los métodos de agrupamiento incorporados al modelo.

A partir de esta combinación, ya no es necesario tener un conocimiento previo del dominio para definir el número de clusters a obtener, ni generar aleatoriamente los centros de clusters. Por tanto, se superan las deficiencias de los algoritmos SKWIC y *Fuzzy* SKWIC, donde se espera obtener mejores agrupamientos en un menor número de iteraciones (i.e. converger de una forma más rápida). Además, al no generar los centros de clusters aleatoriamente en la primera iteración, esperamos que se reduzca la cantidad de clusters vacíos en SKWIC. A continuación describiremos estos tres algoritmos mencionados.

2.2.2.1.1 Algoritmo SKWIC

Una idea general del algoritmo *Simultaneous Keyword Identification and Clustering of Text Documents* (SKWIC) es [BER04]:

Entrada: Colección de documentos en su representación VSM.

Salida: Colección de clusters, donde cada cluster incluye el centro de cluster, la lista de documentos que pertenecen a él y la relevancia de términos en el cluster.

1. Fijar el número inicial de clusters.
2. Inicializar aleatoriamente los centros de clusters.
3. Inicializar las particiones y la relevancia de los términos.

REPETIR

a. Actualizar la relevancia de los términos por clusters (v_{ik}). (Ecuación 2.9)

b. Calcular las distancias de los documentos a los centros de clusters ($D^k_{wc_{ij}}$) (Ecuaciones 2.7 y 2.8).

c. Asignar cada documento a los cluster (en función de la menor distancia que exista entre un documento y todos los centros de clusters). Es decir, actualizar cada partición cluster χ_i (Expresión 2.11).

d. Recalcular los centros de cluster (c_{ik}) (Ecuación 2.12).

e. Recalcular los pesos δ_i por clusters (Ecuación 2.10).

HASTA (centros estabilizados)

En el algoritmo SKWIC se calcula la disimilitud entre cada documento x_j y los vectores centro de clusters c_i basándose en el cociente Coseno según la ecuación 2.8. Inicialmente se necesita calcular la distancia que hay entre dichos vectores para cada término k según la expresión 2.7.

$$D^k_{wc_{ij}} = \frac{1}{n} - (x_{jk} \cdot c_{ik}) \qquad (2.7)$$

$$\overline{D}^k_{wc_{ij}} = \sum_{k=1}^{n} v_{ik} D^k_{wc_{ij}} \qquad (2.8)$$

Nótese que n es el número total de términos en la colección de N documentos, c_{ik} es el k-ésimo componente del vector centro del i-ésimo cluster, y $V=[v_{ik}]$ es la relevancia del término k en el cluster i.

El cálculo de la relevancia de los términos por clusters se realiza a partir de la fórmula 2.9, donde el peso δ_i se calcula mediante la expresión 2.10:

$$v_{ik} = \frac{1}{n} + \frac{1}{2\delta i} \sum_{xj \in Xi} \left[\frac{1}{n} \sum D^{k}_{wcij} - D^{k}_{wcij} \right] \quad (2.9) \qquad \delta_i^{(t)} = K_\delta \frac{\sum_{x_j \in \chi_i} \sum_{k=1}^{n} v_{ik}^{(t-1)} (D^{k\,(t-1)}_{wc\,ij})}{\sum_{k=1}^{n} (v_{ik}^{(t-1)})^2} \quad (2.10)$$

K_δ es una constante y el subíndice (t-1) es usado sobre v_{ik} y c_{ik} para denotar sus valores en la iteración (t-1).

Al asignar un documento a un cluster, se toma en consideración la distancia que existe entre el documento a analizar y cada uno de los centros de clusters, asignándolo definitivamente al cluster más cercano como se muestra en la expresión 2.11:

$$X_i = \left\{ x_j \middle| \widetilde{D}_{wcij} \leq \widetilde{D}_{wckj} \forall k \neq i \right\} \quad (2.11)$$

Finalmente, es necesario recalcular los centros de clusters a partir de la siguiente expresión 2.12:

$$c_{ik} = \frac{\sum_{xj \in Xi} x_{jk}}{\sum_{xj \in Xi} 1} \quad (2.12)$$

Nótese que el algoritmo SKWIC crea una partición, como se puede observar en la figura 2.4.

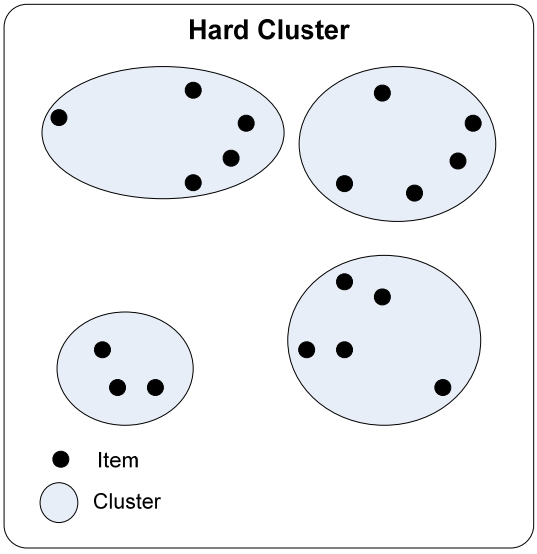


Figura 2.4 Partición creada por el algoritmo SKWIC.

Sin embargo, en los dominios textuales muchas veces un documento, por sus características, puede pertenecer a más de un grupo. Por tanto, en la sección 2.2.2.1.2 describiremos una versión borrosa de este método.

2.2.2.1.2 Algoritmo Fuzzy SKWIC

Una idea general del algoritmo *Simultaneous Keyword Identification and Fuzzy Clustering of Text Documents* (*Fuzzy* SKWIC) es [BER04]:

Entrada: Colección de documentos en su representación VSM.

Salida: Colección de clusters, donde cada cluster incluye el centro de cluster, la lista de todos los documentos y un valor de pertenencia de cada uno al cluster, además, la relevancia de términos en el cluster.

1. Fijar el número inicial de clusters
2. Inicializar aleatoriamente los centros de cluster
3. Inicializar los grados de pertenencia de los documentos a los clusters

REPETIR

   a. Calcular la relevancia de los términos (v_{ik}). (Ecuación 2.13)
   b. Calcular la distancia de los documentos a los centros de cluster ($D^{k}_{wc_{ij}}$) (Ecuaciones 2.7 y 2.8).
   c. Actualizar los grados de pertenencia de los documentos a los clusters (en función de la menor distancia que exista entre un documento y todos los centros de cluster) (u_{ij}) (Ecuación 2.15).
   d. Recalcular los centros de cluster (c_{ik}) (Ecuación 2.16).
   e. Calcular los pesos de la relevancia de los términos (δ_i) (Ecuación 2.14).

HASTA (centros estabilizados)

Esta variante, donde cada documento tiene un grado de pertenencia a cada cluster, muchas veces se ajusta más a las necesidades reales de problemas de minería de textos y por tanto es mucho más efectivo en el dominio textual, porque existen textos que por su contenido pertenecen a más de una categoría.

Los grados de pertenencia de los documentos a los clusters se inicializan asignando el valor $1/C$, donde *C* es la cantidad de clusters.

El cálculo de la relevancia de los términos por clusters se realiza a partir de la fórmula 2.13, donde el peso δ_i se calcula mediante la expresión 2.14:

$$v_{ik} = \frac{1}{n} + \frac{1}{2\delta_i} \sum_{j=1}^{N} \left(u_{ij}\right)^m \left[\frac{\widetilde{D}_{wc_{ij}}}{n} - D^k_{wc_{ij}} \right] \quad (2.13) \qquad \delta_i^{(t)} = K_\delta \frac{\sum_{j=1}^{N} \left(u_{ij}^{(t-1)}\right)^m \sum_{k=1}^{n} v_{ik}^{(t-1)} \left(D^{k^{(t-1)}}_{wc_{ij}} \right)}{\sum_{k=1}^{n} \left(v_{ik}^{(t-1)}\right)^2} \quad (2.14)$$

Para realizar el cálculo del grado de pertenencia de los documentos a los clusters se utiliza la fórmula 2.15:

$$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{\widetilde{D}_{wc_{ij}}}{\widetilde{D}_{wc_{ij}}} \right)^{\frac{1}{m-1}}} \quad (2.15)$$

Finalmente, es necesario recalcular los centros de clusters a partir de la siguiente expresión

$$c_{ik} = \begin{cases} 0, & \text{si } v_{ik} = 0 \\ \dfrac{\sum_{j=1}^{N} \left(u_{ij}\right)^m x_{jk}}{\sum_{j=1}^{N} \left(u_{ij}\right)^m}, & \text{si } v_{ik} > 0 \end{cases} \quad (2.16)$$

Nótese que en este algoritmo se realizan pasos similares al SKWIC pero en cada una de las expresiones se incorpora el grado de pertenencia de los documentos a los clusters.

2.2.2.1.3 Algoritmo Extended Star

El algoritmo *Extended Star* propuesto en [GIL03] es una mejora del *Star* reportado en [ASL00]. Por tanto, inicialmente comentaremos algunas ideas generales del algoritmo *Star*, así como sus desventajas para introducir posteriormente el *Extended Star*. A continuación definiremos algunos términos que son utilizados para describir los algoritmos.

Dos documentos se dicen β_0-semejantes si la semejanza entre ellos, dada por el valor de distancia entre sus vectores, es mayor o igual a β_0, donde β_0 es un parámetro definido por el usuario o calculado inicialmente a partir de la distancia entre los documentos.

Llamamos grafo β_0-semejante al grafo no dirigido cuyos vértices son los objetos a agrupar y existe una arista desde el vértice O_i al vértice O_j, si O_i es β_0-semejante a O_j.

El algoritmo *Star* forma un grafo β_0-semejante, donde los vértices son los documentos a agrupar. Este se basa en un cubrimiento *greedy* donde existen vértices satélites y vértices estrellas. Los vértices estrellas representan los documentos centros de clusters. Dos vértices estrellas nunca son adyacentes.

Alguna de las deficiencias que se mencionan del algoritmo *Star* es que los clusters obtenidos dependen del orden de los objetos. Otra desventaja del algoritmo *Star* es que se pueden formar clusters ilógicos, debido a que dos estrellas nunca pueden ser vecinas [GIL03]. Observe ejemplos de estas desventajas en el anexo 5.

En [GIL03] se propone el algoritmo *Extended Star* que supera las deficiencias del *Star*. Los dos cambios fundamentales respecto al algoritmo *Star* son:

- Utilizar el grado complementario de un documento ($CD(O)$) definido como el grado del documento sin incluir aquellos vecinos ya agrupados en algún cluster

$$CD(O) = \left| \frac{N(O)}{Clu} \right| \tag{2.17}$$

  donde $N(O)$ es el conjunto de documentos vecinos a O y Clu es el conjunto de documentos agrupados.

- Considerar un documento O como estrella si tiene al menos un vecino O' con menor o igual grado que O que satisface una de las condiciones siguientes:
  - O' no tiene vecinos estrellas.
  - El grado más alto de las estrellas que son los vecinos de O' no es mayor que el grado de O.

Una idea general del algoritmo *Extended Star* es [GIL03]:

Entrada: Colección de documentos en su representación VSM.

Salida: Colección de clusters, donde cada cluster incluye la estrella centro de cluster y la lista de todos los documentos que pertenecen a él.

1. Calcular todas las similitudes entre cada par de documentos para construir el grafo β_0-semejante.
2. Sea $N(O)$ los vecinos de cada documento O en el grafo β_0-semejante.
3. Para cada documento aislado O ($|N(O)| = 0$):
   a. Crear un cluster unitario $\{O\}$.
4. Sea L el conjunto de documentos no aislados.
5. Calcular el grado complementario de cada documento en L.
6. Mientras exista un documento no-agrupado:
   a. Sea M_0 el subconjunto de los documentos de L con máximo grado complementario.
   b. Sea M el subconjunto de los documentos de M_0 con máximo grado
   c. Para cada documento O en M
      i. Si O satisface la condición de ser estrella, entonces:

         Si $\{O\} \cup N(O)$ no existe:

            Crear un cluster $\{O\} \cup N(O)$
   d. Eliminar los documentos procesados de L (*)
   e. Actualizar el grado complementario de los objetos en L.

En el paso (*) se pueden eliminar de L los objetos en M o todos los objetos ya agrupados. Estas son versiones no restringidas y restringidas del algoritmo, respectivamente. En la versión restringida, solamente los objetos que no han sido agrupados, pueden ser estrella. Cada versión tiene ventajas y desventajas. La versión no restringida tiene más posibilidades de encontrar las mejores estrellas. La versión restringida es más rápida pero puede obtener clusters ilógicos [GIL03]. En nuestro modelo se concibe el algoritmo *Extended Star* eliminando los objetos de L considerando la versión no restringida.

La complejidad temporal del algoritmo *Extended Star* es de un O (n^2m), donde n es el número de objetos y m el número de rasgos [GIL03].

El algoritmo *Extended Star* [GIL03] crea clusters solapados y garantiza pares β_0-semejantes entre la estrella y sus vecinos. Los clusters se obtienen independientemente del orden de los datos. Si existen dos o más documentos con alta conectividad, este algoritmo selecciona como estrella a todos ellos.

Además, la selección de estrellas usando el grado complementario permite reducir el solapamiento entre clusters.

Cálculo del umbral β_0

Al describir los conceptos principales relacionados con los algoritmos *Star* y *Extended Star*, hacíamos referencia a que el valor β_0 o umbral de semejanza, es un parámetro definido por el usuario o calculado inicialmente a partir de la distancia entre los documentos.

El modelo propuesto considera tres variantes para el cálculo inicial del umbral β_0 y a continuación son descritas [GAR99]:

a) La primera variante calcula el umbral β_0 hallando la media de las distancias entre todos los pares de documentos posibles. Así se expresa en la fórmula 2.18:

$$\beta_0 = \frac{2}{n(n-1)} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} d(O_i, O_j) \tag{2.18}$$

b) La segunda variante halla la media de los valores máximos de las distancias entre cualquier par de documentos, según la expresión 2.19:

$$\beta_0 = \frac{1}{n} \sum_{i=1}^{n} \max_{\substack{j=1..n \\ i \neq j}} \{d(O_i, O_j)\} \tag{2.19}$$

c) La tercera variante toma la mínima de todas las distancias posibles entre pares de documentos, sin tener en cuenta las distancias que sean cero, así se muestra en la fórmula 2.20:

$$\beta_0 = \min_{\substack{i=1..n-1 \\ i \neq j}} \left\{ \min_{j=i+1..n} \{d(O_i, O_j)\} \right\} \tag{2.20}$$

La descripción de la notación utilizada es la siguiente: n es la cantidad de documentos de la colección y $d(O_i, O_j)$ es el valor de la distancia entre los vectores documento O_i y O_j.

2.2.2.2 Calcular la similitud entre vectores

El cálculo de la similitud entre documentos es una fase que está estrechamente relacionada con la selección del método de agrupamiento. La mayoría de los métodos de agrupamiento requieren el uso de medidas de similitud o disimilitud entre documentos. En los algoritmos de agrupamiento

incluidos en el modelo, las medidas de similitud entre los documentos influyen considerablemente en la calidad de los grupos a obtener.

Proponemos para el cálculo de la similitud entre vectores considerar la similitud Coseno [FRA92], y la distancia de Jaccard binaria y pesada [GAS01], por ser éstas las que se reportan en la literatura con mejores resultados (Observe las expresiones 1.32, 1.26 y 1.31 respectivamente). No obstante, en los algoritmos SKWIC y *Fuzzy* SKWIC se calcula la distancia entre los vectores documentos a partir de una variante del cociente Coseno solamente. En el algoritmo *Extended Star* es posible utilizar cualquier medida para comparar los vectores documentos.

2.2.3 Etapa 3. Extracción de palabras claves de grupos textuales

En el capítulo 1 se definió la selección de rasgos como un caso particular de la reducción de dimensionalidad, pero sólo la abordamos enmarcada en la representación textual. Esta reducción es muy útil para mejorar la eficiencia de procesos posteriores que se realizan sobre el corpus. Un ejemplo, es su utilidad en el proceso de agrupamiento de documentos. Sin embargo, existen otras etapas en el procesamiento de textos que requieren aplicar técnicas de selección de rasgos, generalmente aquellas que extraen conocimiento desde textos. Por ejemplo, si se desea obtener un extracto a partir de cada grupo obtenido, no es posible considerar todas las palabras que se obtuvieron en el proceso de reducción de dimensionalidad de la matriz de dispersión, sino que se hace necesario someter cada grupo a un nuevo proceso de reducción de dimensionalidad.

La tercera etapa del modelo incluye tres variantes de selección de rasgos para extraer las palabras claves de los clusters homogéneos de documentos afines a partir de los resultados del agrupamiento y útil para la etapa de obtención del resumen extracto de cada grupo. De esta forma se logran identificar aquellos términos que caracterizan cada grupo, a través de la:

- Selección de las palabras de mayor relevancia resultante de métodos de agrupamiento
- Selección de los términos con mayores valores de calidad en el grupo.
- Selección de los términos a partir de las reglas generadas por el algoritmo ID3.

2.2.3.1 Extracción de palabras claves a partir de su relevancia

Existen algoritmos de agrupamiento que junto a la colección de grupos de documentos devuelven la relevancia de cada término en cada grupo, tal es el caso de los algoritmos SKWIC y *Fuzzy* SKWIC.

Basándonos en la relevancia de los términos por grupos podemos escoger las palabras claves. El modelo que proponemos brinda dos formas de selección: las palabras que su relevancia sea mayor (o menor) que cierto umbral y las *n* palabras con mejor valor de relevancia. La primera, recorre por

grupos todos los términos y escoge aquellos que su valor de relevancia sea mayor (o menor) que un umbral especificado. La segunda, ordena según la relevancia todos los términos y selecciona los primeros *n* términos de la lista.

Nótese que esta forma de selección de palabras claves sólo es aplicable a resultados del agrupamiento con SKWIC y *Fuzzy* SKWIC.

2.2.3.2 Extracción de palabras claves según la calidad de términos

Como fue explicado, existen funciones que determinan la calidad de un término en una colección de documentos. En la primera etapa del modelo se mostró cómo utilizando esta calidad es posible reducir la dimensionalidad de la representación VSM de un corpus de textos eliminando las palabras de menor valor de calidad. En este caso queremos extrapolar esa forma de selección de palabras a cada grupo de documentos obtenido por un método de agrupamiento. Para ello, es necesario formar una representación VSM por cada grupo de documentos. Esta representación tiene los mismos términos que la matriz de la cual fue obtenida la colección de grupos de documentos, pero por cada grupo la representación tiene sólo aquellos documentos presentes en el grupo. Luego, por cada una de estas representaciones se calcula la calidad de todos los términos y se escogen aquellos donde este valor sea mayor (o menor) que un umbral o los *n* términos de mejor calidad.

Si el algoritmo de agrupamiento desarrolla una técnica dura, entonces, por cada grupo, se incluyen en las nuevas representaciones VSM todos los documentos que le pertenecen. Si el algoritmo de agrupamiento desarrolla una técnica borrosa, todos los documentos pertenecen a todos los grupos, por lo que existen dos posibilidades para seleccionar los documentos a considerar para la extracción de palabras claves. Una variante ordena descendentemente los documentos según el grado de pertenencia al cluster y selecciona los primeros *n* términos de la lista. Otra variante posible requiere la especificación de un umbral para considerar aquellos documentos que superan ese umbral de pertenencia.

2.2.3.3 Extracción de palabras claves utilizando el algoritmo ID3

En primer lugar, es necesario generar un árbol de decisión utilizando el algoritmo ID3 [QUI86] [MIT97]. A partir del árbol generado se obtienen las reglas que describen cada grupo de documentos en función del valor de los intervalos resultantes del proceso de discretización de la frecuencia de los términos. La extracción de palabras claves se realiza a partir del análisis de las reglas obtenidas.

El algoritmo ID3 realiza un aprendizaje supervisado, es decir, éste parte de ejemplos previamente clasificados. Podríamos pensar que no es posible aplicarlo a una colección de documentos, ya que los documentos originalmente no están etiquetados. Pero, partimos de un corpus textual previamente agrupado por documentos similares, donde a cada documento le asociamos él o los grupos de los

cuales forma parte, en dependencia si al algoritmo de agrupamiento aplicó una técnica dura determinista o con solapamiento.

En la tabla 2.3 se muestra la forma en que se distribuyen los elementos de entrada al algoritmo ID3 a partir de la colección de documentos previamente agrupada.

	Término 1	**Término 2**	**…**	**Término *m***	**Grupo**
Documento 1	$tf_{d_1}(t_1)$	$tf_{d_1}(t_2)$		$tf_{d_1}(t_m)$	Grupo$_{k1}$
Documento 2	$tf_{d_2}(t_1)$	$tf_{d_2}(t_2)$		$tf_{d_2}(t_m)$	Grupo$_{k2}$
…			…		…
Documento *n*	$tf_{d_n}(t_1)$	$tf_{d_n}(t_2)$		$tf_{d_n}(t_m)$	Grupo$_{kn}$

Tabla 2.3 Matriz de entrada al algoritmo ID3.

A partir de esta forma de entrada de los datos se aplica el algoritmo ID3, el cuál será brevemente descrito en la próxima sección.

2.2.3.3.1 Algoritmo ID3

La idea básica del algoritmo ID3 es aprender un árbol de decisión a partir de su construcción *top-down*, comenzando con la pregunta: ¿qué atributo debe ser seleccionado para la raíz del árbol? [MIT97].

El problema fundamental en el algoritmo ID3 es seleccionar cuál atributo colocar en cada nodo del árbol. Para ello, debemos seleccionar el atributo que es más útil para clasificar los ejemplos. Con este fin se define una función llamada ganancia de la información (*information gain*) que mide cuán bueno es un atributo dado para separar los ejemplos de entrenamiento acorde a su clasificación. ID3 usa esta ganancia de la información para seleccionar el atributo ganador entre los atributos candidatos en cada paso durante la formación del árbol de decisión [QUI86].

Para calcular la ganancia de la información, primero es necesario definir una medida comúnmente usada en teoría de la información, llamada entropía (*Entropy*), que caracteriza la (im)pureza de una colección arbitraria de ejemplos. Si el atributo objetivo puede tomar *c* diferentes valores, entonces la entropía de *S* relativa a las *c* posibles clasificaciones es definida por [MIT97]:

$$Entropy(S) = \sum_{i=1}^{n} - p_i \log_2 p_i \tag{2.21}$$

donde p_i es la proporción de *S* que pertenecen a la clase *i*. En el cálculo de la entropía, se define para el algoritmo ID3 que $0 \log_2 0$ es 0.

Dada la entropía como una medida de la impureza en una colección de ejemplos de entrenamiento, estamos en condiciones de mostrar la definición de la medida de ganancia de un atributo en la clasificación de los datos de entrenamiento, llamada ganancia de la información. Esta medida no es más que la reducción esperada de la entropía causada por los ejemplos acorde al atributo considerado. Más preciso, la ganancia de la información, $Gain(S, A)$ de un atributo A, relativo a una colección de ejemplos, está definida por la expresión 2.22:

$$Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{Sv}{S} Entropy(S_v) \qquad (2.22)$$

donde $Values(A)$ es el conjunto de todos los posibles valores para el atributo A, y S_v es el subconjunto de S para los cuales el atributo A tiene valor v (i.e. $S_v=\{s \in S | A(s)=v\}$). El primer término es justamente la entropía de la colección original S, y el segundo término es el valor esperado de entropía después que S es particionada usando el atributo A. La entropía esperada descrita por el segundo término es simplemente la suma de la entropía por cada subconjunto de S_v, pesada por la fracción de ejemplos S_v /S que pertenecen a S_v. $Gain(S,A)$ es entonces la reducción esperada de la entropía causada por conocer el valor del atributo A. Por otra parte, $Gain(S,A)$ es la información ofrecida acerca del valor objetivo de la función, dado el valor de algún atributo A. El valor de $Gain(S,A)$ es el número de bits salvados cuando codificamos el valor objetivo de un miembro arbitrario de S, por conocer el valor del atributo A.

Nótese que el algoritmo ID3 requiere que los valores de cada rasgo sean discretos. Por tanto, su aplicación a partir de la tabla 2.3 no es válida si antes no discretizamos los valores de frecuencia de las palabras. Es por eso que a continuación describiremos qué es la discretización y dos métodos clásicos que pueden ser aplicados en este proceso.

Algoritmos de discretización

Los valores discretos tienen importancia en el descubrimiento de conocimiento desde datos [HUS99]. Muchos estudios demuestran los beneficios de la discretización: las reglas con valores discretos son normalmente más cortas y entendibles, la discretización puede conducir al perfeccionamiento de una predicción efectiva, además varios de los algoritmos que aparecen en la literatura requieren de rasgos discretos (e.g., algoritmo ID3).

La discretización es el proceso de transformar atributos de dominio continuo a dominios discretos. Dado un dominio definido como un intervalo $[a,b]$ discretizar el atributo significa producir una partición $¶_A$. A partir de allí los valores de los atributos son etiquetas que representan cada elemento de la partición [RUI95].

Existen varios métodos de discretización, dos de los clásicos y más sencillos son:

Intervalos de igual tamaño. Este método, como su nombre lo indica, discretiza teniendo en cuenta el tamaño, es decir, la amplitud de las particiones, de forma tal que esta sea la misma para todos los intervalos. Para lograr esto, el tamaño del intervalo responde a un cálculo dado por la fórmula 2.23:

$$Amplitud = \frac{\max - \min}{K} \tag{2.23}$$

donde,

$\max$ y $\min$: son los valores máximo y mínimo del intervalo inicial [*a*,*b*],

K: es la cantidad de clases que se quieren formar con la discretización.

Culminado el proceso de discretización por este método se tienen *K* intervalos, todos de igual amplitud, sin importar cuántos elementos tiene cada uno de ellos.

Igual frecuencia por intervalo. Al discretizar por este método el objetivo es lograr que todas las particiones tengan la misma cantidad de elementos. Esto responde a la fórmula 2.24:

$$Frecuencia = \frac{N}{K} \tag{2.25}$$

donde,

N: es la cantidad de elementos del intervalo inicial [*a,b*],

K: es la cantidad de clases que se quieren formar con la discretización.

Cuando se haya hecho el proceso de discretización se tendrán *K* intervalos, todos con la misma cantidad de elementos, es decir, la misma frecuencia, sin importar la amplitud de ellos.

2.2.3.3.2 Generación de reglas que describen un corpus textual a partir del ID3

Cada camino en el árbol de decisión de la raíz a las hojas es una regla, donde el antecedente es una conjunción de todos los nodos internos del árbol que pertenecen al camino (con sus respectivos valores discretos asociados) y el consecuente es el nodo hoja (i.e. el grupo asociado). Por tanto, a partir del comportamiento de las frecuencias de las palabras en los grupos, podemos caracterizarlos.

En la construcción del árbol, los nodos que no son hojas representan términos y tienen un hijo por cada valor posible a tomar. Los nodos hojas son el grupo que clasifica los términos que se hallan en el camino de este nodo hoja a la raíz del árbol con sus correspondientes valores. Por ejemplo, si tenemos una colección de cuatro grupos de documentos, podríamos obtener un árbol como el mostrado en la figura 2.5.

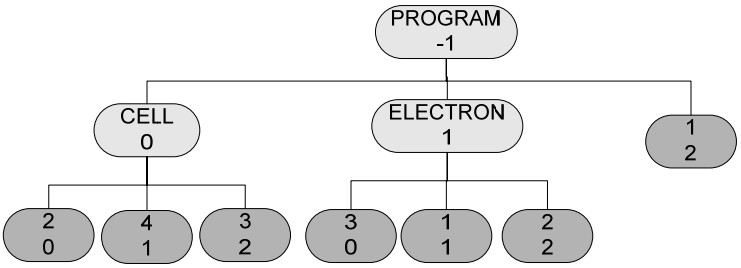


Figura 2.5 Ejemplo de un árbol de decisión de una colección de cuatro grupos de documentos.

Las reglas que describen la colección se obtienen a partir del árbol de decisión recorriéndolo de la raíz a cada hoja del árbol. Cada camino de la raíz a una hoja determina una regla.

Para el árbol de la figura 2.5 obtendríamos las reglas de la tabla 2.4.

Term: PROGRAM Value: 0 ^ Term: CELL Value: 0 -> cluster 2
Term: PROGRAM Value: 0 ^ Term: CELL Value: 1 -> cluster 4
Term: PROGRAM Value: 0 ^ Term: CELL Value: 2 -> cluster 3
Term: PROGRAM Value: 1 ^ Term: ELECTRON Value: 0 -> cluster 3
Term: PROGRAM Value: 1 ^ Term: ELECTRON Value: 1 -> cluster 1
Term: PROGRAM Value: 1 ^ Term: ELECTRON Value: 2 -> cluster 2
Term: PROGRAM Value: 2 -> cluster 1

Tabla 2.4 Reglas obtenidas a partir de un árbol de decisión.

2.2.3.3.3 Generación de las palabras claves a partir de las reglas obtenidas por el ID3

A partir de las reglas obtenidas es posible generar las palabras que logran discernir entre grupos. A cada grupo son asociados aquellos términos que formen parte de los antecedentes de las reglas que ellos son consecuentes y que su valor (i.e, intervalo asociado en el proceso de discretización) sea uno de los *n* mejores valores que puede alcanzar ese término, donde *n* es un valor de entrada al algoritmo. Por ejemplo, si seleccionamos un *n* igual a 1 para las reglas de la tabla 2.4, obtenemos las listas de palabras claves siguientes:

cluster 1
    PROGRAM 2
cluster 2
    ELECTRON 2
cluster 3
    CELL 2
cluster 4
    <empty>

Teniendo en cuenta el ejemplo anterior, cada atributo puede tomar tres valores distintos (0,1,2), por tanto, los *n* mejores valores para *n* igual a uno es solamente el valor 2. Se toman por cada regla aquellas palabras que tienen valor 2. Como en ninguna regla con consecuente grupo 4 se encontró una palabra con valor 2, la lista de palabras claves de este grupo esta vacía.

Si en lugar de escoger *n*=1 escogemos *n*=2, los *n* mejores serían los valores 2 y 1, por lo que se seleccionarían de las reglas de la tabla 2.4 las listas de palabras claves siguientes:

cluster 1
PROGRAM 2
ELECTRON 1
cluster 2
ELECTRON 2
PROGRAM 1
cluster 3
CELL 2
PROGRAM 1
cluster 4
CELL 1

Si no se especifica un *n*, se seleccionan todas las palabras de la regla. Por ejemplo, partiendo de las reglas de la tabla 2.4 obtenemos:

cluster 1
PROGRAM 2
ELECTRON 1
cluster 2
ELECTRON 2
PROGRAM 1
CELL 0
cluster 3
CELL 2
PROGRAM 1
ELECTRON 0
cluster 4
CELL 1
PROGRAM 0

Si el algoritmo que generó la colección de grupos de documentos generó también la relevancia de cada término por grupo, proponemos interceptar las listas de palabras claves que se obtienen con el ID3 con las listas de palabras que se obtienen al escoger por grupos los términos que su relevancia supera un umbral determinado. De esta forma, se obtienen por grupos aquellas palabras relevantes y que logran discernir entre ellos.

Nótese que esta forma de selección de palabras claves utilizando el algoritmo ID3 sólo es factible aplicarla cuando en la etapa de agrupamiento se utilizaron los algoritmos SKWIC o *Extended Star*.

Sin embargo, es factible incorporar a este modelo el algoritmo *Fuzzy* ID3 [WAN03]. Este algoritmo pudiera utilizarse para identificar términos en grupos borrosos como resultado del agrupamiento con *Fuzzy* SKWIC.

En la tabla 2.5 se muestra la forma en que se pudieran distribuir los elementos de entrada al algoritmo *Fuzzy* ID3 a partir de la colección de documentos previamente agrupada por *Fuzzy* SKWIC.

	Término 1	**Término 2**	**…**	**Término *m***	**Grupo**
Doc$_1$	$tf_{d_1}(t_1)$	$tf_{d_1}(t_2)$		$tf_{d_1}(t_m)$	(δ_{Grupo1} (Doc$_1$), …, δ_{Grupok}(Doc$_1$))
Doc$_2$	$tf_{d_2}(t_1)$	$tf_{d_2}(t_2)$		$tf_{d_2}(t_m)$	(δ_{Grupo1} (Doc$_2$), …, δ_{Grupok}(Doc$_2$))
…			…		…
Doc$_n$	$tf_{d_n}(t_1)$	$tf_{d_n}(t_2)$		$tf_{d_n}(t_m)$	(δ_{Grupo1} (Doc$_2$), …, δ_{Grupok}(Doc$_2$))

Tabla 2.5 Matriz de entrada al algoritmo Fuzzy ID3.

Los valores δ_{Grupo_i} (Doc$_j$) significan el grado de pertenencia que tiene el documento *j* en el grupo *i*.

Nótese que para aplicar este algoritmo es necesario obtener por cada rasgo (término) una variable lingüística, así como obtener los términos lingüísticos que la describen a partir de la especificación de las funciones de pertenencia.

2.2.4 Etapa 4. Obtención del resumen extracto de cada grupo de documentos

La última etapa permite obtener un resumen extracto por cada cluster de documentos de la colección mediante la extracción de las oraciones que contienen las palabras claves, ya sea del documento más representativo del cluster, o del documento más representativo del cluster y del resto de los documentos del cluster. Por tanto, es importante conocer cuál es el documento más representativo de un cluster. Es por eso que esta etapa está compuesta por dos fases:

4.1 Identificación del documento más representativo de cada grupo.

4.2 Selección de la técnica de obtención de resumen extracto a aplicar.

2.2.4.1 Identificación del documento más representativo de cada grupo

En dependencia del algoritmo de agrupamiento utilizado, así será la forma de seleccionar el documento más representativo de cada uno de los grupos obtenidos.

Los grupos formados por el método SKWICK tienen un vector centro de cluster que no necesariamente coincide con un vector documento, sino que es un documento ideal que se obtuvo a partir de las iteraciones realizadas por el algoritmo para determinada colección. Por tal motivo, para seleccionar el documento más representativo del grupo proponemos utilizar las medidas de cálculo

de distancias entre vectores documentos (descritas en el subepígrafe 1.3.3). De esta forma, al comparar cada documento del grupo con el vector centro de cluster, se selecciona como documento más representativo de dicho grupo aquel documento que menor distancia tenga con respecto al centro de cluster.

Al aplicar el algoritmo de agrupamiento de documentos *Fuzzy* SKWICK a un corpus, se obtiene como resultado la lista de todos los documentos y un valor de pertenencia de cada uno de ellos al cluster, por lo que se selecciona como documento más representativo de cada grupo aquel documento que mayor valor de pertenencia tenga a dicho grupo.

Por último, identificar el documento más representativo cuando los grupos son formados por el algoritmo *Extender Star* es mucho más sencillo que en los casos anteriores. Este algoritmo devuelve por cada cluster un elemento estrella que no es más que el vector centro del cluster y coincide con un vector real de la colección, por tanto, éste es el documento más representativo del cluster.

2.2.4.2 Selección de la técnica de obtención de resumen extracto a aplicar

En el modelo proponemos realizar un resumen extractivo de múltiples documentos, siguiendo algunas de las variantes descritas en el subepígrafe 1.2.2. Para ello, partimos de una colección de clusters que son formados a través de los distintos algoritmos de agrupamiento incorporados al modelo. Agrupar los documentos similares antes de obtener un resumen extracto permite verificar la homogeneidad del corpus, por tanto, los extractos se generarán a partir de grupos homogéneos de documentos afines.

Es posible seleccionar entre dos técnicas posibles de obtención de resúmenes: crear el resumen de un único documento a partir del documento más representativo de la colección, y crear el resumen de un único documento a partir del documento más representativo de la colección y agregar alguna representación del resto de los documentos de la colección para proveer un cubrimiento total de la misma, pero esto adaptado a cada cluster obtenido. El resumen se crea a partir de la extracción las oraciones que contienen las palabras claves, en función del contexto seleccionado por el tipo de resumen a obtener.

En el algoritmo *Fuzzy* SKWIC todos los documentos pertenecen a todos los clusters, por tanto para obtener un resumen del documento más representativo y el resto de los documentos, se seleccionan aquellos documentos que tengan un valor de pertenencia mayor que un umbral definido.

2.3 Conclusiones parciales

- El modelo conceptual para el agrupamiento de textos y la extracción de resúmenes extractos de corpus textuales, desarrollado en el marco de esta investigación, como una alternativa de solución metodológica al problema científico planteado, permite a los investigadores y desarrolladores agrupar textos y extraer resúmenes extractos a partir de grupos homogéneos de documentos afines de un corpus textual.
- Las características que presenta el procedimiento general del modelo conceptual desarrollado respecto a su carácter generalizador y visto desde dos ángulos, uno por considerar la integración entre varias etapas en el procesamiento textual, y el otro, por su carácter flexible, consistencia lógica que este posee que le confieren ventajas a los métodos establecidos.
- La fase reducir la dimensionalidad perteneciente a la etapa de representación textual a través del reductor de dimensionalidad formado por la combinación de los criterios: frecuencia en el corpus, rango en el corpus, frecuencia de palabras y rango de palabras, permite reducir considerablemente el número de términos a considerar en la etapa de agrupamiento de documentos y mejorar la eficiencia de ésta.
- La etapa extracción de palabras claves a través del conjunto de procedimientos desarrollados para ella permite extraer las palabras claves que caracterizan los grupos de documentos y a la vez logran discernir entre grupos, lo que facilita la obtención de resúmenes extractos de mayor calidad.
- La verificación de la homogeneidad del corpus textual a través de los algoritmos de agrupamiento, posibilita la aplicación en el modelo de las dos técnicas seleccionadas para la extracción de oraciones relevantes a partir de los grupos homogéneos de documentos afines y no de toda la colección. Esto permite obtener resúmenes extractos por cada tópico abordado en el corpus textual, facilitando de esta forma la obtención de resúmenes más coherentes.
- Tanto el modelo conceptual como el conjunto de procedimientos desarrollado en el marco de esta investigación, constituyen en principio, las bases para desarrollar una herramienta computacional que soporte éste, así como la evaluación de los resultados, y permita su aplicación práctica en investigaciones de la minería de textos.

Capítulo 3 Evaluación de los resultados obtenidos en la aplicación del modelo a través de CorpusMiner

En este capítulo se muestra la aplicación del modelo y su procedimiento general a través del software CorpusMiner que lo soporta, del cual se presenta el diseño. A partir de casos de estudio y utilizando las medidas de evaluación entropía, *F-Measure* (*Precision* y *Recall*) y *Overall Similarity* se evaluó la etapa del agrupamiento y la fase de reducción de dimensionalidad dentro de la etapa de representación textual. Por otra parte, los resultados correspondientes a las etapas de extracción de las palabras claves y de resúmenes extractos generados se realiza a partir de un corpus creado que contiene los requisitos que permiten evidenciar la calidad de las salidas.

3.1 Descripción general de la herramienta CorpusMiner

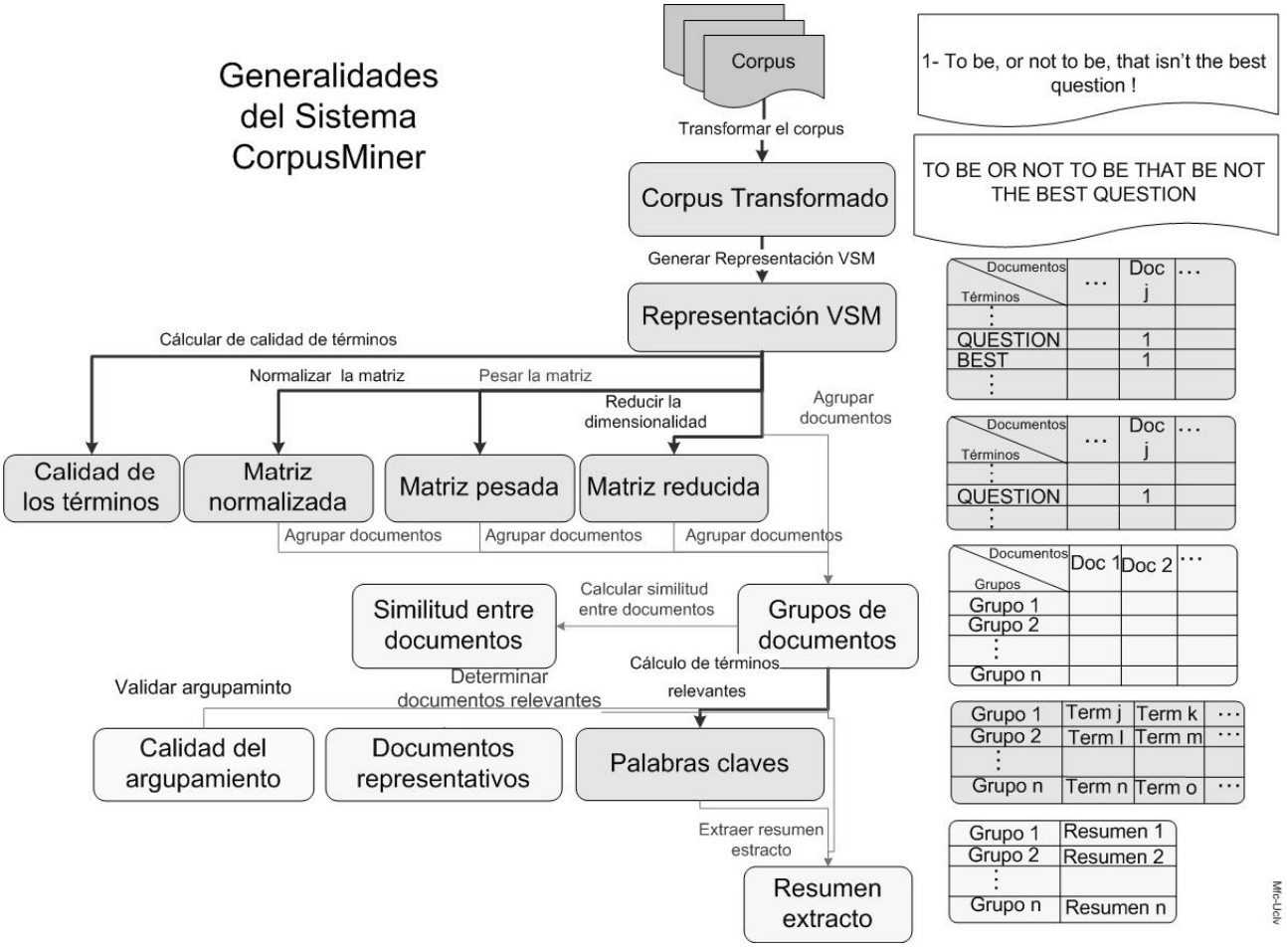


Figura 3.1 Diseño general de la herramienta CorpusMiner para el procesamiento textual.

La herramienta CorpusMiner se desarrolló en total correspondencia con el modelo propuesto en el capítulo 2. Por tanto, permite obtener un resumen extracto de un corpus textual partiendo de la verificación de la homogeneidad del mismo utilizando métodos de agrupamiento. CorpusMiner se desarrolló en Borland Delphi 7.0 y su ejecutable ocupa 900 KB [MED05] [VAL05].

Esta herramienta de procesamiento textual está compuesta por módulos que se corresponden con las etapas descritas en el modelo. Como se aprecia en la figura 3.1 es posible asociar los módulos que permiten transformar el corpus, generar la representación VSM, pesar y normalizar la matriz, calcular las medidas de calidad y reducir la dimensionalidad con la etapa 1 del modelo que permite la representación del corpus textual. Los módulos que permiten agrupar documentos y calcular la similitud entre vectores corresponden a la etapa 2 del modelo. En el módulo de agrupamiento el sistema parte de una matriz pesada, reducida, normalizada o combinaciones de éstas y obtiene grupos homogéneos de documentos afines. El cálculo de términos relevantes o extracción de palabras claves se hace corresponder con la etapa 3 del modelo. La entrada a este módulo es una colección de grupos de documentos y la salida es una lista de palabras claves por grupos, para algunos criterios de selección la entrada puede ser un grupo específico de la colección. Finalmente, el módulo de obtención del resumen extracto de cada grupo de documentos corresponde a la etapa 4 del modelo.

El diseño de CorpusMiner se dividió en tres capas fundamentales como se muestra en la figura 3.2. La primera capa o inferior es la capa del dominio, la segunda o intermedia es la capa controladora y la tercera o superior es la capa de interfaz de usuario.

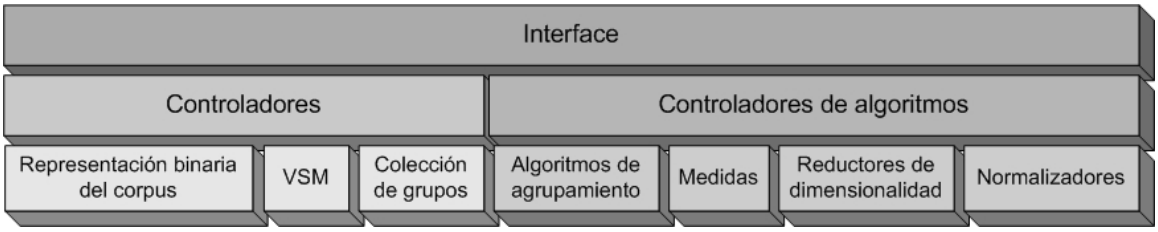


Figura 3.2 Diseño general del sistema CorpusMiner

En la capa inferior están las clases del dominio (i.e. las clases que representan elementos del dominio de aplicación). En esta capa, a su vez, se establecieron dos tipos de clases diferentes. En el primer tipo están aquellas clases que permiten la representación y manipulación de los datos (e.g., la representación VSM, la colección de clusters, las palabras claves de un cluster). Un segundo tipo incluye las clases correspondientes a los algoritmos que operan sobre estos datos (e.g., las medidas de calidad de términos, reductores de dimensionalidad, los algoritmos para normalizar y pesar la representación VSM). Por otra parte, la tercera capa es la encargada de la interfaz visual y posee todas las clases relacionadas con las formas visuales y la interacción con el usuario. La capa

intermedia es la que posee todas las clases controladoras y es la encargada de establecer la comunicación entre las clases de las dos capas mencionadas; ésta al igual que la primera capa, está dividida en dos tipos de clases fundamentales: las controladoras de las clases de datos y las controladoras de algoritmos. Observe en los anexos 6, 7 y 8 el diseño UML de las clases que permitieron la transformación del corpus, el cálculo de la calidad de términos y la extracción de palabras claves, respectivamente.

En el modelo se concibió que para una misma funcionalidad existan varios algoritmos capaces de desarrollarla. Por ejemplo, para calcular la calidad de un término existen siete variantes y aún pueden existir más. Por eso, en CorpusMiner por cada tipo de algoritmo se creó un controlador que sea capaz de encapsular a todos los miembros de ese tipo y que se encargue de la comunicación de éstos con la interfaz de usuario. Esta forma de encapsular trae consigo además, que sea muy fácil adicionar nuevos algoritmos al sistema. Observe en el anexo 9 el diseño general de los controladores en CorpusMiner, nótese como se integran al sistema los algoritmos de agrupamiento y de extracción de resúmenes extractos.

Como fue descrito en el modelo, para obtener el resumen extracto de un corpus textual es necesario transitar por cuatro etapas definidas. En correspondencia con éste, el funcionamiento general del sistema CorpusMiner, permite partir de una colección de documentos e ir obteniendo resultados intermedios hasta obtener el resultado final deseado. Cada resultado intermedio puede obtenerse a su vez de distintas formas a partir de diferentes algoritmos o a partir de un mismo algoritmo utilizando parámetros diferentes.

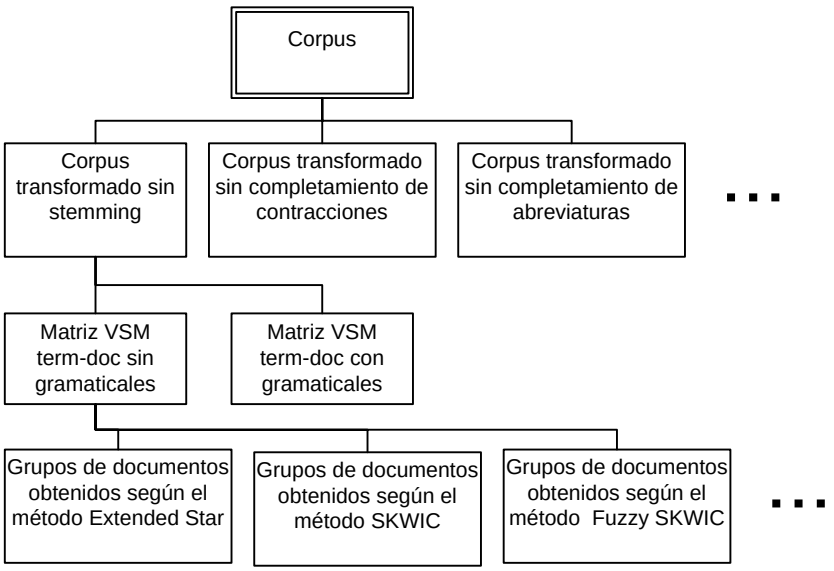


Figura 3.3 Ejemplo de la obtención de pasos intermedios.

Por citar un ejemplo, si nuestro objetivo es obtener grupos de documentos de un corpus de textos, debemos realizar varias etapas y sus respectivas hasta obtener el resultado final. Primeramente, es necesario transformar el corpus, pero ello se puede hacer de varias formas (i.e. realizando o no la lematización, aplicando o no la expansión de abreviaturas y contracciones, o realizar o no la homogenización ortográfica). Luego, partiendo del corpus transformado se requiere generar la representación VSM término–documento, pero aquí podemos incluir o no las palabras gramaticales. De igual forma el agrupamiento puede efectuarse utilizando diferentes algoritmos y cada uno de éstos requiere sus propios parámetros, los cuáles determinan los resultados. Observe la figura 3.3.

Esta idea sugiere una estructura de árbol donde cada nodo es el resultado de aplicarle un determinado algoritmo con ciertos parámetros a los datos que controla el nodo padre de éste. La raíz del árbol es la colección de corpus textuales. Observe la figura 3.4.

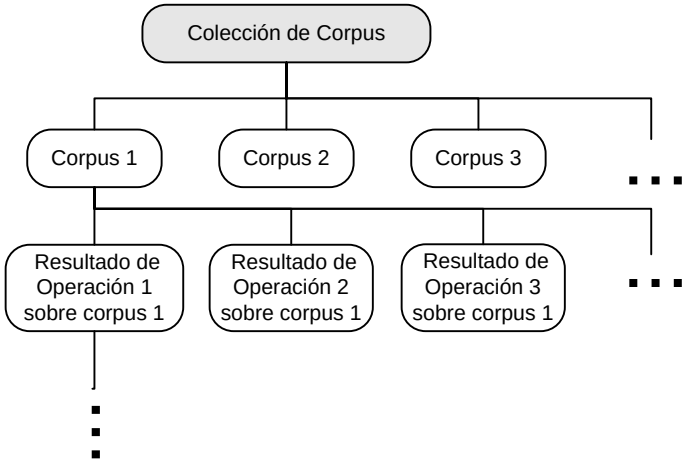


Figura 3.4 Estructura de árbol en CorpusMiner al aplicar los algoritmos.

Para lograr esta estructura se diseñó, por cada clase de representación de datos, una clase controladora que al mismo tiempo es un nodo de un árbol, en el cual sus hijos son los controladores de los objetos obtenidos al aplicar un cierto algoritmo a los datos que él controla. El diseño permite que cada nodo tenga todas las posibilidades relacionadas con un árbol (e.g. adicionar y eliminar un hijo). Además, como controlador tiene la posibilidad de salvar y mostrar los resultados (conexión con la capa de interfaz).

La interfaz gráfica de CorpusMiner está en total correspondencia con las facilidades que brinda el diseño y se encuentra dividida en tres partes fundamentales: menú principal, árbol de resultados y pantalla de resultados (observe figura 3.5).

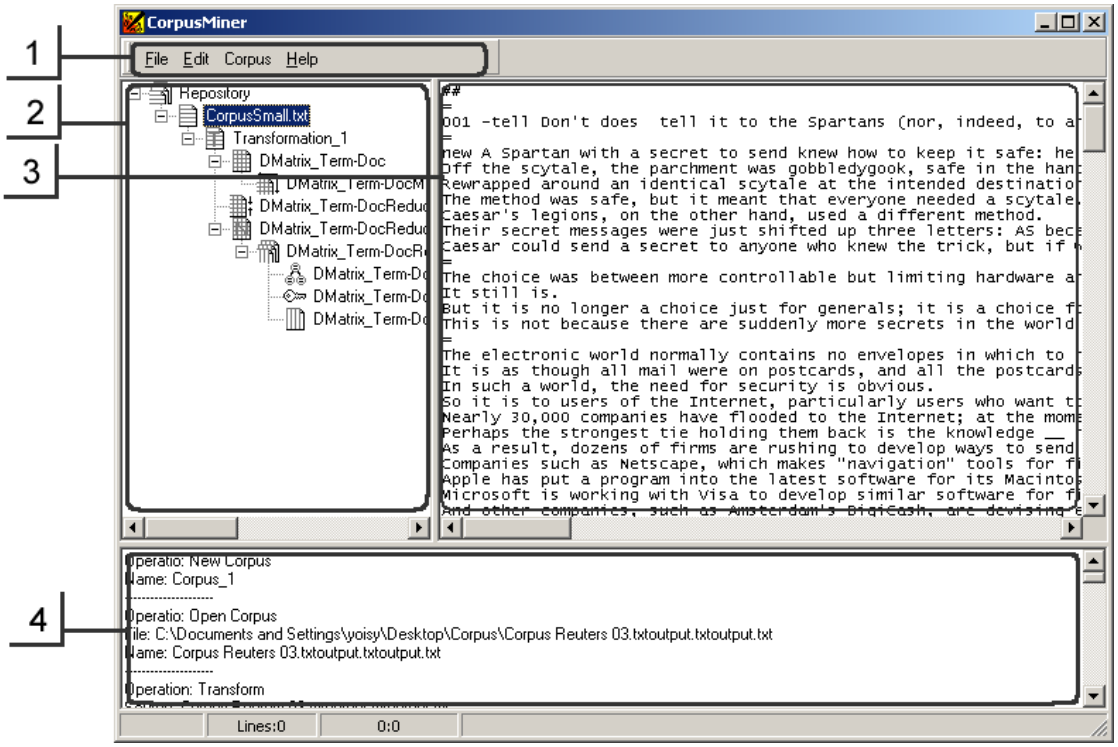


Figura 3.5 Interfaz gráfica de CorpusMiner.

1. Menú principal.
2. Árbol de resultados.
3. Pantalla de resultados.

En el árbol existen siempre un nodo activo y un nodo seleccionado, que puede coincidir con el activo. El nodo activo se diferencia del resto de los nodos mediante una marca y es además el que en cada momento esta siendo mostrado en la pantalla de resultados. El menú principal se corresponde a las operaciones que pueden ser efectuadas sobre los datos que representa el nodo activo. El menú contextual del árbol se refiere a las operaciones que pueden efectuarse sobre los datos que el nodo seleccionado referencia.

Para activar un nodo basta con dar doble clic sobre él e inmediatamente se muestra en la pantalla de resultados los datos que el nodo representa y el menú principal muestra las operaciones que pueden ser efectuadas sobre estos datos. Para seleccionar un nodo se utiliza un clic izquierdo o derecho y a continuación el menú contextual del árbol adquiere las opciones relativas a los datos que el nodo representa.

3.2 Métricas para evaluar la calidad del agrupamiento de documentos

Como se plantea en la definición del modelo, éste debe tener procedimientos de control. Uno de los procedimientos está asociado a la evaluación de la calidad de los resultados obtenidos en la etapa que permite el agrupamiento de documentos. El objetivo de este procedimiento es determinar cuáles son los métodos que reportan mejores resultados y poder proponer las políticas a seguir para la implementación de software dirigido a usuarios finales. Este procedimiento se basa en métricas que a partir de los resultados de la medición permiten evaluar el desempeño del agrupamiento, e incluso, no sólo éste, sino poder valorar a partir del agrupamiento como se comportaron otros indicadores seleccionados en etapas precedentes, por ejemplo, reductores de dimensionalidad o formas de calcular la similitud entre documentos.

Para evaluar el agrupamiento, dos métricas de calidad son ampliamente usadas [STE00]. Un tipo de métrica permite comparar diferentes conjuntos de grupos sin referenciar el conocimiento externo y es llamada métrica de calidad interna. En tal sentido, utilizaremos la métrica *overall similarity* basada en calcular la similitud de pares de documentos en un cluster. Existe otro tipo de métricas que permite evaluar cuán bueno es el agrupamiento mediante la comparación de los grupos producidos por la técnica de agrupamiento con las clases que se identifican en la colección. Este tipo de métrica es llamada métrica de calidad externa. Ejemplos de métricas externas son la entropía y *F-Measure*.

En la literatura se reportan diferentes métricas para validar la calidad del agrupamiento de documentos y los resultados obtenidos al aplicarlas a diferentes algoritmos de agrupamiento, puede variar sustancialmente dependiendo de cual métrica fue usada. Sin embargo, si un algoritmo de agrupamiento funciona mejor que otros para la mayoría de las métricas, entonces se puede decir que ciertamente ese algoritmo es el mejor para la situación que fue evaluada.

En CorpusMiner existe un módulo que permite evaluar los grupos generados por cada uno de los métodos implementados y sus combinaciones, a través de las métricas entropía, *F-Measure* (*precision y recall*) y *Overall Similarity* [STE00] [FOR03] [SEB99]. A continuación describiremos cada una de estas métricas mencionadas.

3.2.1 Entropía

En [STE00] usan la entropía como métrica de calidad de los clusters (con la advertencia que la mejor entropía es obtenida cuando cada cluster contiene exactamente un documento). Sea *CS* un resultado de agrupamiento. Para cada cluster es calculada primero la distribución de las clases (i.e. para un cluster *j* se calcula p_{ij}, la probabilidad que un miembro del clusters *j* pertenezca a la clase *i*)

$$p_{ij} = \frac{n_j^i}{n_j} \tag{3.1}$$

donde n_j^i es el número de documentos de la clase *i* que están asignados al cluster *j*. Entonces, usando esa distribución de la clase, la entropía de cada cluster *j* es calculada usando la fórmula estándar

$$E_j = -\sum_i p_{ij} \log(p_{ij}) \tag{3.2}$$

donde la sumatoria es aplicada sobre todas las clases. La entropía total para el conjunto de clusters *i* es calculada como la suma de las entropías de cada cluster y han sido pesadas por el tamaño de cada cluster

$$E_{CS} = \sum_{j=1}^{m} \frac{n_j * E_j}{n} \tag{3.3}$$

donde n_j es el tamaño del cluster *j*, *m* es el número de clusters y *n* es el números total de documentos.

3.2.2 F-Measure

La segunda métrica de calidad externa es *F-Measure* [STE00], es una métrica que combina las ideas de *precision* y *recall* de la recuperación de información [FRA92]. En [STE00] se trata cada cluster como si este fuera el resultado de una consulta y cada clase como si esta fuera el conjunto de documentos deseados para una consulta. Así, se calcula *precision* y *recall* de los clusters para cada clase dada. Más específicamente, para el cluster *j* y la clase *i*

$$precision(i, j) = n_{ij} / n_j \tag{3.4}$$

$$recall(i, j) = n_{ij} / n_i \tag{3.5}$$

donde n_{ij} es el número de miembros de la clase *i* en el cluster *j*, n_j es el número de miembros del cluster *j* y n_j es el número de miembros de la clase *i*.

Entonces, según [STE00] *F-Measure* del cluster *j* y la clase *i* es entonces dada por

$$F(i, j) = \frac{2 \cdot recall(i, j) \cdot precision(i, j)}{recall(i, j) + precision(i, j)} \tag{3.6}$$

Un valor de *F-Measure* es calculado para toda la colección calculando un promedio pesado de todos los valores de las métricas *F-Measure*, según la siguiente expresión

$$F = \sum_i \frac{n_i}{n} \max\{F - Measure(i, j)\} \tag{3.7}$$

Es bien conocido de la práctica diaria de la recuperación de información que los niveles más altos de *precision* generalmente se obtienen con valores bajos de *recall*. La siguiente expresión permite calcular el valor *F-Measure*, proporcionando un parámetro α ($0 \leq \alpha \leq 1$) que permite ponderar *precision* y *recall*.

$$F_\alpha(i, j) = \frac{1}{\alpha \frac{1}{precision(i, j)} + (1-\alpha)\frac{1}{recall(i, j)}} \tag{3.8}$$

En esta fórmula α puede verse como el grado relativo de importancia atribuida para *precision* y *recall*. Si α=1 entonces $F_\alpha(i,j)$ coincide con *precision*, si α=0 entonces $F_\alpha(i,j)$ coincide con *recall*. El valor α=0.5 es usado usualmente, así se considera igual importancia para *precision* y *recall*. En CorpusMiner utilizamos esta última variante del cálculo de $F_\alpha(i,j)$.

3.2.3 Overall Similarity

En la ausencia de información externa, tales como las etiquetas de clases, la cohesión de los clusters puede ser usada como una métrica de similitud de clusters [STE00]. Un método para calcular la cohesión de un cluster es usar la similitud pesada de la similitud interna del cluster,

$$OverallSimilarity = \frac{1}{|S|^2} \sum_{\substack{d \in S \\ d' \in S}} \text{distance}(d', d) \tag{3.9}$$

donde $|S|$ es el número de documentos que pertenecen al cluster a evaluar. En [STE00] utilizan el cociente Coseno para calcular la distancia entre los vectores, sin embargo, en CorpusMiner se permite calcular la distancia con cualquier métrica de similitud implementada.

3.3 Definición de los casos de estudio para la aplicación del procedimiento general del modelo a través de CorpusMiner

En este trabajo se definieron tres casos de estudio que permitirán aplicar el procedimiento general del modelo a través de CorpusMiner. El primer caso de estudio se concibió para evaluar los

resultados del agrupamiento, así como otros elementos que contribuyen al buen funcionamiento de éste, como son los reductores de dimensionalidad concebidos en la etapa de representación textual. El segundo caso de estudio fue diseñado para mostrar los resultados correspondientes a la etapa agrupamiento, pero en este caso, a partir de un corpus creado que contiene los requisitos que permiten evidenciar la calidad de las salidas. Por último, el tercer caso de estudio se diseñó para mostrar los resultados correspondientes a las dos últimas etapas del modelo, con un corpus también obtenido de noticias de la agencia Reuters.

Descripción del primer caso de estudio: Corpus textuales de la agencia de noticias Reuters

Todas las métricas utilizadas para evaluar la calidad del agrupamiento, excepto *Overall Similarity*, requieren que los corpus textuales hayan sido previamente etiquetados. Colecciones de documentos existen varias, pero que estén previamente etiquetas es menos probable.

Entre los corpus textuales publicados en Internet que se referencian en los artículos para mostrar la calidad de algoritmos en varias áreas de la minería de textos están:

- TDT 2 *Multilanguage text corpus* V 4.0[1]
- *A large benchmark data set for web document clustering*[2]
- 20-*newsgroups*[3]
- Colección de la agencia Reuters de noticias[4]

Sin embargo, procesar estas colecciones no es una tarea trivial debido a que los formatos son muy variados, los textos traen mucha información no útil y la longitud de los documentos a veces es muy pequeña, entre otras razones.

Por ejemplo, TDT2 sólo trae documentos etiquetados en dos clases: *news story* y *miscellaneous text*. El formato de los documentos es asequible ya que utilizan etiquetas de HTML por lo que es muy sencillo identificar los elementos que conforman el corpus. Sin embargo, consideramos que es una deficiencia para evaluar el agrupamiento que la colección sólo esté dividida en dos grupos. Por otra parte, *A large benchmark data set for web document clustering* tiene documentos clasificados en once categorías y éstas a su vez asociadas a cuatro temas, algo útil para evaluar algoritmos de agrupamiento. El problema de este corpus es que los textos tienen mucha información no útil y el formato en que se presentan los textos es muy difícil de preprocesar. En 20-*newsgroups* se clasifican los textos en veinte categorías, y éstas a su vez en seis temas. Presenta un formato sencillo, pero

[1] http://www.ldc.upenn.edu/Projects/TDT2
[2] http://www.pedal.reading.ac.uk/banksearchdataset
[3] http://www.ai.mit.edu/people/jrennie/20Newsgroups
[4] http://www.daviddlewis.com/resources/testcollections/reuters21578

como su nombre lo indica los documentos provienen de listas de discusión, por lo que hay mucha información no útil y los textos son extremadamente pequeños. Finalmente, la colección de noticias de la agencia Reuters tiene la información clasificada en cientotreinta y cinco tópicos, con un formato asequible y la longitud de los documentos es adecuada para la evaluación que se desea realizar. Una desventaja de esta colección es que no todos los documentos se encuentran etiquetados.

A partir del análisis de las colecciones revisadas, la colección de noticias de la agencia Reuters publicada por David D. Lewis cumple con los requerimientos de la evaluación. Debido a que una desventaja de la colección es que no todas las noticias están previamente clasificadas, se decidió extraer de la colección original un corpus textual que contiene 2802 noticias etiquetadas que ocupan 5.11 MB.

A partir de esta colección de la agencia Reuters de noticias, conformamos 10 corpus textuales con un tamaño promedio de 160 KB, con 88 documentos incluidos como promedio que abordan 32 tópicos aproximadamente. Nótese que el número elevado de tópicos se debe a que las noticias están multiclasificadas.

Descripción del segundo caso de estudio: Corpus de textos asociados a palabras

ABLE	BABIES	CALIFORNIA	…	…	…
ACCESS	BABY	CALLED		UNIVERSE	WEIGHT
ACTIVITY	BACTERIA	CANADA		UNIVERSITY	WELL
AFRICA	BASED	CANCER		USE	WHITE
AGE	BECOME	CAR		USED	WIRELESS
AGENCY	BEHAVIOUR	CARBON		USER	WOMAN
AGING	BEING	CARS	…	USERS	WOMEN
AIDS	BEST	CASE		USING	WORDS
AIR	BETTER	CASES		VACCINE	WORK
AIRCRAFT	BIG	CAUSE		VACCINES	WORKING
ALZHEIMER	BIOLOGICAL	CELL		VIDEO	WORKS
AMERICA	BIOLOGY	CELLS		VIRTUAL	WORLD
AMERICAN	BIOTECHNOLOGY	CENTER		VIRUS	YEAR
…	…	…	…	VIRUSES	YEARS

Tabla 3.1 Fragmento de las palabras más frecuentes seleccionadas

Este caso de estudio considera un corpus textual que se construyó intencionalmente por lingüistas expertos para mostrar los resultados de la etapa de agrupamiento de una forma clara.

La construcción de este corpus parte de una colección de documentos, de la cual se seleccionan las palabras que tienen una frecuencia de aparición alta. Por cada palabra altamente frecuente, se seleccionan de ese corpus las oraciones que la contienen, cada conjunto de oraciones asociado a

palabras frecuentes conformará un documento del corpus que se ha construido para este caso de estudio (observe la tabla 3.1).

El corpus construido está compuesto por 540 documentos (correspondientes a las 540 palabras más frecuentes del corpus original) y ocupa 29 MB. Observe en el anexo 10 fragmentos de documentos del corpus de textos asociados a palabras.

Descripción del tercer caso de estudio: Corpus textual de la agencia de noticias Reuters

Evaluar la calidad de la extracción de palabras claves por grupos y de resúmenes extractos es una tarea difícil y requiere tener en cuenta el criterio de los expertos. En este trabajo sólo pretendemos mostrar resultados de estas etapas, de tal forma que sin ser un especialista en la materia se pueda comprobar con facilidad la calidad de los mismos.

Por tal motivo, este caso de estudio está concebido para mostrar una aplicación real del modelo a través de CorpusMiner en la extracción de palabras claves y resúmenes extractos a partir de un corpus textual que tiene 1662 documentos y ocupa 3.81 MB, también obtenido del corpus de la agencia Reuters de noticias

3.4 Resultados de la evaluación del agrupamiento

La evaluación del agrupamiento se hizo para los dos primeros casos de estudio definidos. El primer caso fue evaluado estadísticamente, mientras que para el segundo se hace un análisis de los clusters obtenidos.

3.4.1 Resultados experimentales del agrupamiento para el caso de estudio corpus textuales de la agencia de noticias Reuters

Para validar el agrupamiento se aplicaron, al primer caso de estudio descrito, los métodos SKWIC, *Fuzzy* SKWIC y *Extended Star*, así como las combinaciones *Extended Star* – SKWIC y *Extended Star – Fuzzy* SKWIC. Las métricas utilizadas en la evaluación son: entropía, *F-Measure* (*Precision* y *Recall*) y *Overall Similarity*, descritas en el epígrafe 3.2.

Como ya se mencionó en el capítulo 2, el procedimiento general del modelo y CorpusMiner en consecuencia brindan diferentes variantes para una misma funcionalidad. Para el estudio comparativo de los algoritmos se consideró que los 10 corpus textuales, formados para el primer caso de estudio, fueron procesados siguiendo los mismos criterios. La etapa de representación consideró todas las opciones de transformación, las palabras gramaticales no fueron incluidas en la representación VSM y ésta se normalizó por la frecuencia de aparición de términos y se pesó según TF-IDF II. Se redujo la dimensionalidad a los mejores 1000 términos según calidad de términos 2.

La estructura de la matriz construida para analizar el agrupamiento se muestra en el anexo 11. Este análisis se realizó utilizando el paquete de análisis estadístico SPSS versión 11.0. A partir de estos datos se realizaron pruebas no paramétricas. Inicialmente se aplicó la prueba de Friedman para verificar si existen diferencias significativas de todos los algoritmos respecto a cada medida. A partir de estos resultados, donde existieran diferencias significativas, se aplicaron las pruebas de los rangos de Wilcoxon para comparar los algoritmos dos a dos y verificar dónde existen las diferencias significativas entre los algoritmos.

Como se observa en el anexo 12, teniendo en cuenta los valores de significación de la prueba de Friedman, existen diferencias altamente significativas entre los algoritmos respecto a cada uno de las métricas consideradas. Es por eso que fue necesario aplicar pruebas no paramétricas de Wilcoxon para detectar en qué algoritmos existen diferencias significativas.

Importante es analizar si el *Extended Star* – SKWIC logra superar al SKWIC y si el *Extended Star* – *Fuzzy* SKWIC logra superar al Fuzzy SKWIC.

En el anexo 13 a) se muestra que respecto a la *precision* la diferencia entre SKWIC y *Extended Star* SKWIC es medianamente significativa, ya que la significación es de 0.064. Nótese en el anexo 13 b) que según *precision*, *Extended Star* – SKWIC supera a SKWIC en 8 corpus textuales. Teniendo en cuenta el resto de los criterios, no se aprecian diferencias, sino que se puede decir que los resultados son comparables. No obstante es digno destacar que según F-*Measure Extended Star* – SKWIC supera a SKWIC en 7 corpus, así como en 8 corpus teniendo en cuenta la entropía y en 7 corpus considerando la media según los valores de *Overall Similarity* por clusters.

Una situación similar se aprecia en el anexo 14 al comparar los algoritmos *Fuzzy* SKWIC y *Extended Star – Fuzzy* SKWIC. A partir de los valores de significación teniendo en cuenta la métrica de entropía se aprecia que la diferencia entre los algoritmos es medianamente significativa, ya que la significación es 0.052. Nótese en el anexo 14 b) a partir del análisis de los rangos que el algoritmo *Extended Star – Fuzzy* SKWIC supera al algoritmo *Fuzzy* SKWIC respecto a la entropía en 8 corpus. Teniendo en cuenta el resto de las métricas y sus valores de significación, ambos algoritmos son comparables. No obstante, en el anexo 14 b) se aprecia que *Extended Star – Fuzzy* SKWIC obtiene mejores resultados que el algoritmo *Fuzzy* SKWIC para 8 corpus según F-*Measure* y *Precision*, y para 7 corpus según la media por *Overall Similarity*.

Estos resultados demuestran que utilizar la salida del algoritmo *Extended Star* para inicializar SKWIC y *Fuzzy* SKWIC, reporta mejores resultados que con una inicialización aleatoria de los algoritmos y donde es necesario tener en cuenta conocimiento del dominio. Es importante señalar que en CorpusMiner se consideran en la incialización los centros de aquellos clusters que no son aislados, este es un elemento que contribuye a obtener mejores resultados, debido a que se inicializa

con aquellos centros que se corresponden altamente con los tópicos que aborde el corpus. Un análisis que facilita la comparación de estos algoritmos es comprar *Extended Star* con SKWIC y *Fuzzy* SKWIC, así como *Extended Star* con *Extended Star* – SKWIC y *Extended Star* – *Extended Star* – *Fuzzy* SKWIC. Particularizaremos estos análisis teniendo en cuenta los resultados de algoritmos duros.

Observe en el anexo 15 a) que existen diferencias altamente significativas entre los algoritmos *Extended Star* y SKWIC para las métricas *Precision*, *Recall*, Entropía, y la media y desviación estándar según *Overall Similarity*. El observar el anexo 15 b) se aprecia que las métricas que miden cuán compactos y homogéneos son los clusters (*Precision*, Entropía, media de *Overall Similarity*) reportan que el algoritmo *Extended Star* supera a SKWIC para los 10 corpus analizados. Sin embargo, SKWIC reporta los mejores resultados para los 10 corpus teniendo en cuenta *Recall* y la desviación estándar según *Overall Similarity*. Esto demuestra por qué no existen diferencias significativas según F-*Measure*, con un valor de significación de 0.432 porque esta métrica es una combinación de *Precision* y *Recall*. Estos resultados se deben a que *Extended Star* forma clusters muy homogéneos y compactos, sin embargo forma muchos clusters aislados y clusters pequeños que no cubren las clases. De ahí que haya sido conveniente utilizar los centros de clusters compactos de *Extended Star* para inicializar SKWIC y como éste tiene un buen cubrimiento de las clases, se logra obtener buenos resultados de F-*Measure*.

Observe en el anexo 16 que *Extended Star* continúa siendo superior a *Extended Star* – SKWIC respecto a *Precision*, Entropía y *Overall Similarity*. Esto es lógico, porque *Extended Star* logra clusters muy compactos y homogéneos. No obstante, nuestro objetivo con la combinación de métodos no es superar el algoritmo *Extended Star*, sino lograr mejorar los algoritmos SKWIC y *Fuzzy* SKWIC que son los que simultáneamente calculan la relevancia de las palabras en el proceso de agrupamiento, y por tanto, son de gran utilidad en la obtención de resúmenes extractos.

3.4.2 Análisis de clusters para el caso de estudio corpus de textos asociados a palabras

Como se describió en el epígrafe 3.3 el caso de estudio corpus de textos asociados a palabras fue construido para reflejar claramente los resultados del agrupamiento. El tópico que aborda cada documento está en correspondencia con la palabra que lo identifica, ya que cada documento está compuesto por las oraciones que contienen dicha palabra.

El corpus fue representado considerando todas las variantes de transformación posibles, la representación VSM fue normalizada por frecuencia y pesado siguiendo la variante TF-IDF I, se redujo la dimensionalidad seleccionando los 1000 mejores términos según la medida calidad de

término 1 y se realizó el agrupamiento con la combinación de algoritmos que ha reportado los mejores resultados Extended Star – SKWIC, donde la similitud Coseno fue utilizada para comparar los vectores documentos.

La colección de clusters obtenida a partir del agrupamiento tiene 91 clusters, de ellos 56 fueron vacíos, no hubo clusters con un único documento y la composición del resto de los clusters se presenta a continuación:

Cluster 4--> 17 documentos
Cluster 5--> 9 documentos
Cluster 6--> 12 documentos
Cluster 11--> 31 documentos
Cluster 19--> 6 documentos
Cluster 20--> 9 documentos
Cluster 22--> 16 documentos
Cluster 23--> 10 documentos
Cluster 24--> 8 documentos
Cluster 28--> 12 documentos
Cluster 32--> 35 documentos
Cluster 33--> 11 documentos
Cluster 36--> 6 documentos
Cluster 37--> 21 documentos
Cluster 41--> 11 documentos
Cluster 47--> 8 documentos
Cluster 48--> 10 documentos
Cluster 50--> 21 documentos
Cluster 52--> 10 documentos
Cluster 53--> 16 documentos
Cluster 57--> 7 documentos
Cluster 60--> 8 documentos
Cluster 61--> 12 documentos
Cluster 63--> 120 documentos
Cluster 65--> 13 documentos
Cluster 67--> 12 documentos
Cluster 70--> 6 documentos
Cluster 77--> 7 documentos
Cluster 79--> 9 documentos
Cluster 80--> 6 documentos
Cluster 81--> 7 documentos
Cluster 86--> 20 documentos
Cluster 87--> 14 documentos
Cluster 88--> 7 documentos
Cluster 89--> 13 documentos

A continuación se muestra un fragmento de los clusters obtenidos:

Cluster 4	**Cluster 22**	**Cluster 53**	**Cluster 67**
Doc 22: ASTRONOMER	Doc 15: ANIMAL	Doc 14: AMERICANS	Doc 31: BASED
Doc 27: AWAY	Doc 16: ANIMALS	Doc 17: ANTIBIOTICS	Doc 90: COMPUTER
Doc 137: EARTH	Doc 78: CLOCK	Doc 30: BACTERIA	Doc 91: COMPUTERS
Doc 156: EVIDENCE	Doc 79: CLONING	Doc 61: CASES	Doc 92: COMPUTING
Doc 178: GALAXIES	Doc 129: DNA	Doc 62: CAUSE	Doc 216: IBM
Doc 192: GRAVITY	Doc 157: EVOLUTION	Doc 84: COMMON	Doc 315: NETWORK
Doc 278: MARTIAN	Doc 180: GENE	Doc 111: DEATH	Doc 316: NETWORKS
Doc 329: ORBIT	Doc 181: GENES	Doc 125: DISEASE	Doc 328: OPERATING
Doc 355: PLANET	Doc 182: GENETIC	Doc 126: DISEASES	Doc 418: SECURITY
Doc 356: PLANETS	Doc 184: GENOME	Doc 204: HEART	Doc 470: SYSTEM
Doc 385: RADIATION	Doc 213: HUMAN	Doc 210: HOSPITAL	Doc 471: SYSTEMS
Doc 405: ROCK	Doc 214: HUMANS	Doc 227: INFECTION	Doc 480: TELEVISION
Doc 442: SOLAR	Doc 297: MODERN	Doc 228: INFECTIONS	
Doc 447: SPACECRAFT	Doc 377: PROJECT	Doc 346: PERCENT	
Doc 454: STAR	Doc 423: SEQUENCE	Doc 396: RESISTANCE	
Doc 455: STARS	Doc 448: SPECIES	Doc 397: RESISTANT	
Doc 467: SUN			

Tabla 3.2 Fragmento de los clusters obtenidos a partir del segundo caso de estudio.

En la tabla 3.2 se muestra como el algoritmo de agrupamiento aplicado logra obtener una colección de clusters que abordan temas afines.

3.5 Resultados experimentales de la fase de reducción de dimensionalidad correspondiente a la etapa de representación textual

Un elemento que influye notablemente en los resultados del agrupamiento de documentos es la fase de reducción de dimensionalidad al representar los corpus textuales. En el tópico 2.2.1.5 se hizo referencia a cuatro formas de reducir la dimensionalidad de una representación VSM: reducir basándonos en un umbral de calidad, seleccionar los *n* mejores términos, seleccionar los términos mayores que la media de calidad y reducir por frecuencia de rango. En este epígrafe se analizará, a partir de los resultados del agrupamiento, si existen diferencias significativas entre las formas de reducir la dimensionalidad y qué variante reporta los mejores resultados. Para ello consideramos la reducción de dimensionalidad seleccionando los 1000 mejores términos teniendo en cuenta las medidas calidad de términos 1, calidad de términos 2 y entropía, así como el reductor por frecuencia de rango.

En este estudio comparativo, al igual que se hizo en el subepígrafe 3.4.1, se realizó el procesamiento de los 10 corpus textuales, según el primer caso de estudio, siguiendo los mismos criterios. La etapa de representación consideró todas las opciones de transformación, las palabras gramaticales no fueron incluidas en la representación VSM, ésta se normalizó por la frecuencia de aparición de términos y se pesó según TF-IDF II. El algoritmo de agrupamiento que se utilizó fue *Extended Star* – SKWIC y la medida para comparar documentos en *Extended Star* fue la similitud Coseno.

La estructura de la matriz construida para analizar las variantes de reducción de dimensionalidad se muestra en el anexo 17. Este análisis también se realizó utilizando el paquete de análisis estadístico SPSS versión 11.0. A partir de estos datos se realizaron pruebas no paramétricas. Inicialmente se aplicó la prueba de Friedman para verificar si existen diferencias significativas de todos los reductores de dimensionalidad respecto a cada métrica. A partir de estos resultados, donde existieran diferencias significativas, se aplicaron las pruebas de los rangos de Wilcoxon para comparar los reductores dos a dos y verificar dónde existen las diferencias significativas entre éstos.

Como se observa en el anexo 18, teniendo en cuenta los valores de significación de la prueba de Friedman, existen diferencias significativas entre todos los reductores respecto a las métricas entropía y la media según *Overall Similarity*. Es por eso que fue necesario aplicar pruebas no paramétricas de Wilcoxon para detectar en qué reductores existen diferencias significativas comparando los resultados dos a dos.

En el anexo 19 se evidencia que existen diferencias significativas en los resultados del agrupamiento cuando se reduce la dimensionalidad con calidad de término 1 y calidad de término 2. Los valores de significación según entropía y la media de *Overall Similarity* son 0.015 y 0.029, respectivamente. Al analizar los rangos, según la métrica entropía, la reducción según calidad de términos 2 supera a calidad de términos 1 en 9 casos, igual sucede con la media *Overall Similarity*, donde en 8 casos es mejor.

Las diferencias cuando se reduce la dimensionalidad por entropía y calidad de término 1 son altamente significativas si consideramos la significación 0.012 teniendo en cuenta la métrica de evaluación de entropía. En el anexo 21 se observa como el en 8 corpus la entropía supera a la calidad de términos 1. Según la media de *Overall Similarity* los resultados son comparables, ya que la significación tiene el valor 0.059.

A partir de los valores de la media de *Overall Similarity* que se muestran en los anexos 20, 22 y 24 el reductor por frecuencia de rango supera a los reductores por calidad de término 1, calidad de término 2 y entropía. En el primer caso la significación es 0.005 por lo que existen diferencias altamente significativas, en las dos últimas comparaciones las diferencias también son altamente significativas con valores de significación 0.002 en ambos casos. Sin embargo, considerando la métrica de entropía los resultados son comparables.

En el anexo 23 se evidencia que los resultados del agrupamiento reduciendo la dimensionalidad de la representación textual con calidad de términos 2 y entropía son comparables, tanto cuando consideramos la métrica entropía como la media de *Overall Similarity*, con valores de significación 0.320 en ambos casos.

Un reductor que se destaca al comparar los resultados del agrupamiento es el reductor por frecuencia de rango propuesto. Además, este reductor logra reducir el número de términos de la colección considerablemente, nótese que no es necesario especificar el número de términos final en el vocabulario. La reducción de dimensionalidad según calidad de términos 2 supera a la reducción de dimensionalidad según calidad de términos 1 al analizar los resultados del agrupamiento, reafirmándose de esta forma lo que se reportó en la literatura [BER04]. Nótese que estos resultados son considerando las métricas entropía y media de *Overall Similarity*, porque para el resto de las métricas los resultados son comparables a partir de los valores de significación que se muestran en el anexo 18.

3.6 Análisis de la extracción de palabras claves y resúmenes extractos a partir del caso de estudio corpus textual de la agencia de noticia Reuters

Validar la calidad de la extracción de palabras claves y resúmenes extractos es una tarea ardua y requiere del criterio de expertos. En este epígrafe sólo pretendemos presentar un fragmento de las palabras claves por clusters obtenidas y fragmentos de resúmenes extractos de dichos clusters seleccionados a partir del corpus conformado para el tercer caso de estudio.

Haciendo referencia al capítulo 2, el modelo propuesto contempla la selección de las palabras de mayor relevancia resultante de métodos de los métodos de agrupamiento SKWIC y *Fuzzy* SKWIC, selección de los términos con mayores valores de calidad en el grupo considerando los reductores de dimensionalidad y selección de los términos a partir de las reglas generadas por el algoritmo ID3, así como la combinación de éste con las palabras relevantes.

Observe en las tablas 3.3, 3.4 y 3.5 las palabras relevantes obtenidas por el algoritmo SKWIC para los clusters 2, 5 y 296 respectivamente.

Cluster 2	
Palabras claves	**Relevancia**
COFFEE	639.998066002203
QUOTA	150.978797476516
EXPORT	48.3416311356147
ICO	44.8852920851479
BRAZIL	40.5550270037344
BAG	33.6864444414485
PRODUCER	28.884181926699
SAY	22.022676963465
PRICE	12.5912047884285
DELEGATE	9.11956629787066
IBC	8.60120720273399
TRADE	7.75617380325565
MLN	7.61179668404557
CONSUMER	7.53584386750696
MARKET	7.14701860494342
YEAR	6.22358743456569
BRAZIL'S	5.90051423147361
EXPORTER	5.76528108836532

Tabla 3.3 Palabras claves obtenidas por el algoritmo SKWIC para el cluster 2 de la colección.

Cluster 5	
Palabras claves	**Relevancia**
COCOA	671.551268009175
BUFFER	251.468453926574
DELEGATE	81.1817041291764
STOCK	62.9513924738284
PRODUCER	19.3338555058136
TONE	18.7105111636895
CONSUMER	16.3410166799598
MANAGER	14.0526374430781
BUY	13.7283869457331
SAY	13.524513556072
RULE	13.255157135113
PRICE	13.0558422559357
COUNCIL	12.1578919266104
TRADER	11.9617084741012
BAG	10.5298505408877
MEMBER	9.61515355499522
CROP	6.93703968267803

Tabla 3.4 Palabras claves obtenidas por el algoritmo SKWIC para el cluster 5 de la colección.

Cluster 296	
Palabras claves	**Relevancia**
VENEZUELA	3.18377827158001
TODAY	0.597335018490733
SAY	0.437322991195588
PRICE	1.29614842695025
OPEN	0.925744505096108
OPEC	4.36025082765431
OIL	3.94599951862008
MINISTER	6.59371508523503
MEXICO	2.85759035788277
MEMBER	1.00531602452641
MEET	1.98128575160266
MARKET	1.66073657331461
GROUP	0.661305071774677
EXPORTER	1.70333258443692
ENERGY	9.61879185738887
ECUADOR	3.2324408365707
DLRS	0.602348124410253
BARREL	1.35118905137334
AVERAGE	0.922911742496263

Tabla 3.5 Palabras claves obtenidas por el algoritmo SKWIC para el cluster 296 de la colección.

Nótese que las palabras obtenidas logran trasmitir la idea general de los grupos. No obstante, la selección de las palabras según la relevancia no logra, en su totalidad, distinguir entre grupos. Ejemplos de esta situación es la palabra SAY, que no caracteriza ninguno de estos grupos y sin embargo, por aparecer mucho en los mismos, el algoritmo la considera relevante. Esta situación se resuelve en CorpusMiner interceptando estos resultados con los obtenidos por el algoritmo ID3, de esta forma se logra obtener un conjunto de palabras claves que sean relevantes y que permitan distinguir entre grupos.

Línea 959: <TOPICS><D>coffee</D></TOPICS>
Línea 960: The failure of the International Coffee Organization (ICO) to reach agreement on coffee export quotas could trigger a massive selloff in London coffee futures of at least 100 stg per tone today, coffee trade sources said.
Línea 961: Prices could easily drop to as low as 1.00 dlr or even 80 cents a lb this year from around 1.25 dlrs now, they said.
Línea 962: A special meeting between importing and exporting countries ended in a deadlock late yesterday after eight days of talks over how to set the quotas.
Línea 963: No further meeting to discuss quotas was set, delegates said.
Línea 964: Quotas, the major device used to stabilize prices under the International Coffee Agreement, were suspended a year ago after prices soared following a damaging drought in Brazil.
Línea 965: With no prospects for quotas in sight, heavy producer selling initially and a price war among commercial coffee roasting companies will ensue, the trade sources predicted.
Línea 966: Lower prices are sure to trickle down to the supermarket shelf this spring, coffee dealers said.
Línea 967: The U.S. And Brazil, the largest coffee importer and exporter respectively, each laid the blame on the other for the breakdown of the talks.
Línea 969: Assistant trade representative and delegate to the talks, said in a statement after the council adjourned, "A majority of producers, led by Brazil, were not prepared to negotiate a new distribution based on objective criteria.
Línea 970: "We want to insure that countries receive export quotas based on their ability to supply the market, instead of their political influence in the ICO." Brazilian Coffee Institute (IBC) President Jorio Dauster countered, "Negotiations failed because consumers tried to dictate quotas, not negotiate them."
Línea 971: Previously, quotas were determined by historical amounts exported, which gave Brazil a 30 pct share of a global market of about 58 mln 60-kilo bags.
Línea 972: A majority of producers wanted quotas to continue under this basic scheme.
Línea 973: But most consumers and a maverick group of eight producers proposed carving up the export market on the basis of exportable production and stocks, which would reduce Brazil's share to 28.8 pct.
. . .

Tabla 3.6 Resumen extracto del Cluster 2.

Línea 7496: <TOPICS><D>cocoa</D></TOPICS>
Línea 7497: The credibility of government efforts to stabilise fluctuating commodity prices will again be put to the test over the next two weeks as countries try to agree on how a buffer stock should operate in the cocoa market, government delegates and trade experts said.
Línea 7498: Only two weeks ago, world coffee prices slumped when International Coffee Organization members failed to agree on how coffee export quotas should be calculated.
Línea 7499: This week, many of the same experts gather in the same building here to try to agree on how the cocoa pact reached last summer should work.
Línea 7500: The still unresolved legal wrangle surrounding the International Tin Council (ITC), which had buffer stock losses running into hundreds of millions of sterling, is also casting a shadow over commodity negotiations.
Línea 7503: Some importing countries insist the International Cocoa Organization (ICCO) buffer stock rules must not be muddied with quota type subclauses which might dictate the type of cocoa to be bought.
Línea 7504: One consumer country delegate said this would "distort, not support" the market.
Línea 7505: Trade and industry sources blame uncertainty about the ICCO for destabilising the market as the recent collapse in coffee prices has made traders acutely aware that commodity pacts can founder.
Línea 7506: On Friday this uncertainty helped push London cocoa futures down to eight month lows.
Línea 7507: The strength of sterling has also contributed to the recent slip in prices.
Línea 7508: The ICCO daily and average prices on Friday fell below the "must buy" level of 1,600 SDRs a tonne designated in the pact, which came into force at the last ICCO session in January but without rules for the operation of the buffer stock.
Línea 7509: Consumers and producers could not agree on how it should operate and what discretion it should be given.
Línea 7510: The agreement limits it to trading physical cocoa and expressly says it cannot operate on futures markets.
Línea 7511: A cash balance of some 250 mln dlrs and a stock of almost 100,000 tonnes of cocoa, enough to mount large buying or selling operations, were carried forward from the previous agreement.
Línea 7512: Members finance the stock through a 45 dlrs a tonne levy on all cocoa they trade.
. . .

Tabla 3.7 Resumen extracto del Cluster 5.

Los resúmenes extractos que se muestran en las tablas 3.6, 3.7 y 3.8 son fragmentos de resúmenes obtenidos a partir de la selección de las oraciones que contienen las palabras relevantes de los clusters sólo considerando el documento más representativo de cada grupo. Nótese que los resúmenes obtenidos son una enumeración de las oraciones que contienen palabras claves, por tanto, es posible encontrar anáforas y redundancia en ellos.

Línea 15843: Energy and Mines Minister Arturo Hernandez Grisanti today told a meeting of regional oil exporters the next few months will be critical to efforts to achieve price recovery and stabilize the market.
Línea 15844: Hernandez said while OPEC and non-OPEC nations have already made some strides in their efforts to strengthen the market, the danger of a reversal is always present.
Línea 15845: "March and the next two or three months will be a really critical period," Hernandez said.
Línea 15846: He said, "We will be able to define a movement, either towards market stability and price recovery or, depending on the market, a reversal."
Línea 15847: Earlier this week, Hernandez said Venezuela's oil price has averaged just above 16 dlrs a barrel for the year to date.
Línea 15848: If OPEC achieves its stated goal of an 18 dlrs a barrel average price, he said, Venezuela's should move up to 16.50 dlrs.
Línea 15849: Hernandez spoke today at the opening of the fifth ministerial meeting of the informal group of Latin American and Caribbean oil exporters, formed in 1983.
Línea 15850: Ministers from member states Ecuador, Mexico, Trinidad-Tobago and Venezuela are attending the two day conference, while Colombia is present for the first time as an observer.
Línea 15851: Hernandez defined the meeting as an informal exchange of ideas about the oil market.
Línea 15852: However, the members will also discuss ways to combat proposals for a tax on imported oil currently before the U.S. Congress.
Línea 15853: Following the opening session, the group of ministers met with President Jaime Lusinchi at Miraflores, the presidential palace.
Línea 15854: The delegations to the conference are headed by Hernandez of Venezuela, Energy Minister Javier Espinosa of Ecuador, Energy Minister Kelvin Ramnath of Trinidad-Tobago, Jose Luis Alcudiai, assistant energy secretary of Mexico and Energy Minister Guilermno Perry Rubio of Colombia.

Tabla 3.8 Resumen extracto del Cluster 296.

3.7 Conclusiones parciales

- El diseño del sistema CorpusMiner contempla el procedimiento general del modelo y es flexible y extensible, permitiendo de esta forma una rápida incorporación de nuevas variantes a las funcionalidades brindadas.
- Los casos de estudio definidos permitieron demostrar la factibilidad del modelo propuesto y su procedimiento general.
- La evaluación de la calidad del agrupamiento y de las técnicas de reducción de dimensionalidad en la etapa de representación textual se realiza a través de las métricas entropía, *F-Measure* (*precision y recall*) y *Overall Similarity*.

- La aplicación de las pruebas estadísticas no paramétricas para el estudio de los métodos de agrupamiento permitió determinar que los algoritmos *Extended Star* – SKWIC y *Extended Star* – *Fuzzy* SKWIC reportan mejores resultados que los algoritmos SKWIC y *Fuzzy* SKWIC respectivamente. Además, que el algoritmo *Extended Star* logra los mejores valores de *precision*, entropía y la media de *Overall Similarity*. Por tanto, los resultados de esta evaluación permiten trazar las políticas respecto al uso de los algoritmos de agrupamiento para el desarrollo de aplicaciones de usuario final.
- La evaluación de los reductores de dimensionalidad se realizó a partir de los resultados del agrupamiento. Las pruebas estadísticas no paramétricas aplicadas evidencian que el reductor por frecuencia de rangos obtiene los mejores resultados según la media de *Overall Similarity*, y que la calidad de términos 2 logra un agrupamiento de mayor calidad que calidad de términos 1. Estos resultados permiten decidir que reductores utilizar en aplicaciones futuras.
- La definición del caso de estudio corpus de textos asociados a palabras permitió reflejar de una forma clara la calidad del agrupamiento.
- La definición del caso de estudio corpus textual de la agencia Reuters de noticias permitió mostrar los resultados de las etapas referentes a la extracción de palabras claves y generación de resúmenes extractos, evidenciándose de esta forma una correspondencia entre agrupamiento, palabras claves y resumen asociado.
- El modelo conceptual para la obtención de resúmenes extractos a partir de los resultados del agrupamiento de corpus textuales demostró, asociado al sistema CorpusMiner, su factibilidad como estrategia de solución al problema científico planteado a través de los casos de estudio definidos. Mediante su concepción y desarrollo pudo derivarse la solución metodológica general al problema científico planteado. Se logró la aplicación integral del modelo y su procedimiento general, constatándose la factibilidad y conveniente utilización de ambos como instrumentos metodológicos efectivos, lo que permitió validar la hipótesis general de investigación planteada.

Conclusiones

Como resultado de esta investigación se desarrolló un modelo conceptual y un procedimiento metodológico, soportado por el software CorpusMiner, que ofrece a los investigadores y desarrolladores en el campo de la minería de textos una herramienta que posibilita el agrupamiento de textos para la obtención de resúmenes extractos a partir de grupos homogéneos de documentos afines, con un enfoque de integración, cumpliéndose de esta forma el objetivo general planteado, ya que:

- Existe una creciente base teórica conceptual sobre la representación, agrupamiento y extracción de resúmenes de corpus textuales. Sin embargo, los principales trabajos reportados en la literatura están focalizados al desarrollo de algoritmos muy específicos o a aplicaciones dirigidas a usuarios finales, y no así, al desarrollo de modelos y procedimientos dirigidos a la integración de técnicas que contribuyan a investigaciones en el área de la minería de textos. Por tal motivo, el problema científico formulado para la presente investigación se considera de gran actualidad y pertinencia, tanto en el plano conceptual – metodológico como práctico.
- El análisis crítico sobre el estado actual de las técnicas para la representación, agrupamiento y obtención de resúmenes extractos de corpus textuales arrojó que es necesaria la combinación de algoritmos que permitan la integración de estas tres áreas de la minería de textos. Las técnicas que facilitan esta integración son la representación VSM a partir de una transformación del corpus textual, el pesado y normalización de dicha representación, así como la reducción de dimensionalidad. Dentro de los algoritmos de agrupamiento, aquellos que obtienen simultáneamente la relevancia de las palabras en el proceso de agrupamiento son muy útiles para la obtención de resúmenes extractos de documentos afines: SKWIC y *Fuzzy* SKWIC. Las técnicas de obtención de resúmenes extractos seleccionadas hallan el documento más representativo, permitiéndose de esta forma obtener resúmenes más o menos compactos.
- El diseño del modelo conceptual explica y fundamenta un procedimiento metodológico general que permite agrupar textos para obtener resúmenes extractos a partir de grupos homogéneos de documentos afines, con un enfoque de integración que contribuye al desarrollo de investigaciones en la minería de textos. La definición del modelo contiene las premisas, los objetivos, las entradas, salidas y los procedimientos, así como los principios que lo caracterizan. El modelo conceptual reúne todos los elementos considerados relevantes e imprescindibles cuando se pretende por cada funcionalidad brindar varios métodos que la sustenten. El modelo es flexible y extensible.

- El procedimiento metodológico general desarrollado y los procedimientos específicos asociados, permiten agrupar textos para obtener resúmenes extractos a partir de grupos homogéneos de documentos afines, a través de cuatro etapas que permiten la representación del corpus textual, el agrupamiento de los documentos, la extracción de palabras claves de los grupos y la obtención de un resumen extracto por cada grupo obtenido. En cada una de las etapas es posible identificar las mejores variantes de las técnicas de agrupamiento y representación textual seleccionadas, así como las políticas a seguir para obtener resultados de mayor calidad y desarrollar herramientas dirigidas a usuarios finales.
- La validación del modelo conceptual y el procedimiento propuesto se realizó a partir de corpus textuales representativos del universo investigado mediante la definición de tres casos de estudio, como vía de comprobación y factibilidad de la investigación realizada. Los resultados fueron obtenidos a partir del software CorpusMiner que soporta el modelo propuesto. La aplicación de las pruebas estadísticas no paramétricas para el estudio de los métodos de agrupamiento permitió determinar que los algoritmos *Extended Star* – SKWIC y *Extended Star* – *Fuzzy* SKWIC reportan mejores resultados que los algoritmos SKWIC y *Fuzzy* SKWIC respectivamente. Además, que el algoritmo *Extended Star* logra los mejores valores de *precision*, entropía y la media de *Overall Similarity*. Por otra parte, la evaluación de los reductores de dimensionalidad evidencian que el reductor por frecuencia de rangos obtiene los mejores resultados según la media de *Overall Similarity*, y que la calidad de términos 2 logra un agrupamiento de mayor calidad que calidad de términos 1. Por tanto, los resultados de esta evaluación permiten trazar las políticas respecto al uso de los algoritmos de agrupamiento y técnicas de reducción de dimensionalidad de la representación textual para el desarrollo de aplicaciones de usuario final. De esta forma queda validada la hipótesis de investigación planteada en los términos en que se declaró en el proyecto de la investigación originaria.

Recomendaciones

Teniendo en consideración que el modelo propuesto es extensible se recomienda:

- Adicionar otras formas de reducción de dimensionalidad en la representación textual (e.g. *Latent Semantic Indexing*)
- Incluir el algoritmo *Fuzzy* ID3 para la selección de palabras claves a partir de los resultados de agrupamientos borrosos.
- Incorporar métodos de agrupamiento jerárquico que permitan inicializar SKWIC y *Fuzzy* SKWIC y además determinar cercanía entre los clusters formados.

Referencias bibliográficas

[ASL00] Aslam, J. Pelekhov, K. Rus, D. Scalable Information Organization. Proceedings of RIAO. 2000.

[ASL98] Aslam, J. Pelekhov, K. Rus, D. Static and Dynamic Information Organization with Star Clusters. Proceedings of the Conference of Information Knowledge Management, Baltimore, MD. 1998.

[BAL98] Baldwin, B. Morton, T. Co-reference-Based Summarization. T. Firm Hand & B. Sundheim (Eds), TIPSTER-SUMMAC Summarization Evaluation. Proceedings of the TIPSTER Text Phase III Workshop. 1998.

[BER01] Bergo, A. Text Categorization and Prototypes. 2001.

www.illc.uva.nl/Publications/ ResearchReports/MoL-2001-08.text.pdf

[BER04] Berry, M. Survey of Text Mining. Clustering, Classification, and Retrieval. Springer-Verlag. ISBN 0-387-95563-1. 2004.

[BEZ73] Bezdek, J.C. Fuzzy Mathematics in Pattern Classification. Ph. D. Thesis, Applied Math. Center, Cornell University, Ithaca. 1973.

[BLA92] Blair, D. Information retrieval and the philosophy of language. The Computer Journal 35(3). Pp. 200-2007. 1992.

[BOL98] Boley, D.L. Principal direction divisive partitioning. Data Mining and Knowledge Discovery. 2(4). Pp. 325-344. 1998.

[CAL97] Callan, J. Characteristics of texts. 1997.

http://www-ciir.cs.umass.edu/~allan/cs646-f97/char_of_text.html

[CAV94] Cavnar, W.B. Trenkle, J.M. N-gram based text categorization. Proceedings of the Symposium on Document Analysis and Information Retrieval, Las Vegas. Pp. 161-175. 1994.

[CHE98] Cherkassky, V. Mulier, F. Learning from Data. Concepts, Theory, and Methods. A Wiley-Interscience Publication. John Wiley & Sons, Inc. 1998.

[COH96] Cohen, W.W. Singer, Y. Context-sensitive learning methods for text categorization. Proceedings of the Nineteenth Annual International ACM SIGIR conference on Research and Development in Information Retrieval. Pp. 307-315. 1996.

[CRO02] Cross, V. Sudkamp, T. Similarity and Compatibility in Fuzzy Set Theory. Assessment and Applications. Phisica-Verlag. Springer-Verlag Company. ISSN 1434-9922. 2002.

[DAS97] Dash, M. Liu, H. Feature selection for classification. Intelligent Data Analysis 1(3). 1997.

http://www-east.elsevier.com/ida/browse/0103/ida00013/article.html.

[DIX97] Dixon, M. An Overview of Document Mining Technology. 1997.

www.geocities.com/ResearchTriangle/Thinktank/1997/mark/writings/dixm97_dm.ps

[DUC00] Duch, W. Similarity-based methods: a general framework for classification, approximation and association. Control and Cybernetics. Vol. 29. No. 4. 2000.

[DUD73] Duda, R.O. Hart, P.E. Pattern Classification and Scene Analysis. New York: Wiley-Interscience. 1973.

[DUR01] Dürsteler, J.C. Minería de Textos. Inf@Vis! La revista digital de InfoVis.net. Mensaje No. 27. 2001.

[FEB02] Febles, J.P., González, A. Aplicación de la minería de datos en la Bioinformática. Centro Nacional de Bioinformática. 2002.

[FOR03] Forman, G. An Extensive Empirical Study of Feature Selection Metrics for Text Classification. Journal of Machine Learning Research 3. pp. 1289-1305. 2003.

[FRA03] Franke, J., Nakhaeizadeh, G., Renz, I. Advances in Soft Computing. Text Mining. Theoretical Aspects and Applications. Physica-Verlag. ISBN 3-7908-0041-4. 2003.

[FRA92] Frankes, W. B. Baeza-Yates, R. Information Retrieval. Data Structures & Algorithm. Prentice Hall PTR. ISBN 0-13-463837-9. 1992.

[FUK03] Fukumoto, J. Text summarization based on itemized sentences and similar parts detection between documents. Proceedings of the Third NTCIR Workshop. 2003.

http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings3/NTCIR3-TSC-FukumotoJ.pdf

[FUK99] Fukuhara, T. Takeda, H. Nishida, T. Multiple-text Summarization for Collective Knowledge Formation. Proceedings of Workshop of Social Aspects of Knowledge and Memory. IEEE Systems, Man and Cybernetics Conference. 1999.

[FUN02] Fung, B. Hierarchical Document Clustering using Frequent Itemsets. Master of Science in the School of Computing Science. Simon Fraser University. 2002.

[GAB04] Gabrielsson, S. MOVSOM-II – analysis and visualization of movieplot clusters. 2004. http://www.pcpinball.com/movsom

[GAR99] García, M.M. Monografía de reconocimiento de patrones. Universidad Central "Marta Abreu" de Las Villas. 1999.

[GAS01] Gasperin. C, Gamallo. P. López, G. Lima, V. Using Syntactic Contexts for Measuring Word Similarity. 2001. www.cl.cam.ac.uk/users/cvg20/esslli01.pdf

[GIL03] Gil-García, R. Badía-Contelles, J.M. Pons-Porrata, A. Extended Star Clustering Algorithm. Proceedings of CIARP. 2003.

[GOL00] Goldstein, J. Mittal, V. Carbonell, J. Callan, J. Creating and Evaluating Multi-document Sentences Extract Summaries. CIKM 2000. pp. 165-172. 2000.

[GOL99] Goldstein, J. Automatic Text Summarization of Multiple Documents. Thesis Proposal. 1999.

[HEE82] de Heer, T. The application of the concept of homeosemy to natural language information retrieval. Information Processing Manangement 18(5). Pp. 229-236. 1982.

[HOP99] Höppner, F. Klawonn, F. Rudolf, K. Runkler, T. Fuzzy Cluster Analysis. Methods for Classification, Data Analysis and Image Recognition. John Wiley & Sons Ltd. 1999.

[HUS99] Hussain, F. Liu, H. Discretization: An Enabling Technique. TRC6/99. Junio, 1999. http://www.comp.nus.edu.sg/~liuh

[IND01] Inderjeet, M. Gates, B. Bloedorn, E. Improving Summaries by Revising Them. 2001.

acl.ldc.upenn.edu/P/P99/P99-1072.pdf

[JAC02] Jackson, P. Moulinier, I. Natural Language Processing for Online Applications. Text Retrieval, Extraction and Categorization. John Benjamins Publishing Company. ISBN 902724988. 2002.

[JAM01] James, A. Gupta, R. Center for Intelligent Information Retrieval Temporal Summaries of News Topics. 2001. www-ciir.cs.umass.edu/~allan/Papers/2001-sigir.pdf

[JAN97] Jang, J.S. NeuroFuzzy and Soft Computing. Prentice Hall. ISBN: 0-13-261066. 1997.

[JOA97] Joachims, T. Text categorization with support vector machines: Learning with many relevant features. Technical Report LS-8 Report 23, Fachbereich Informatik, Universität Dortmund, Dortmund. 1997.

[KAR01] Karypis, G. Evaluation of Item-Based Top-N Recommendation Algorithms. 2001. http://www.cs.umn.edu/~karypis/suggest

[KEN97] Kendall, K.E., Kendall, J.E. Análisis y diseño de sistemas. Tercera Edición. Pretice Hall. 1997.

[KEO99] McKeown, K.R. Klavans, J.L. Towards Multidocument Summarization by Reformulation: Progress and Prospects. 1999. http://newsblaster.cs.columbia.edu/papers/stim2.pdf

[LAN01] Lanquillon, C. Enhancing Text Classification to Improve Information Filtering. Dissertation Ph.D. Fakultat fur Informatik der Otto-von-Guericke Universitat Magdeburg. 2001.

[LAR00a] Larocca, J. Santos, A. Kaestner, C. Freitas, A. Generating Text Summaries Through the Relative Importance of Topics. Int. Joint Conf. IBERAMIA 2000 (7th Ibero-American Conf. on Artif. Intel.) & SBIA-2000 (15th Brazilian Symp. On Artif. Intel.). Brazil. Pp 300-309. 2000.

[LAR00b] Larocca, J. Santos, A. A trainable algorithm for summarizing news stories. 4th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'2000). 2000.

[LAR04] Larocca, J. Santos, A. Celso, A. A trainable algorithm for summarizing news stories. 2004. http://www.chirouble.univ-lyon2.fr/~pkdd2000/Download/WS4_04.pdf

[LAR99] Larocca, J. Santos, A. DocClustering: um sistema para agrupamento de documentos. International Seminar on Document Management. Curitiba, Brazil. 1999.

[LEW92] Lewis, D.D. Representation and Learning in Information Retrieval. Ph. D. thesis, Department of Computer and Information Science, University of Massachusetts. 1992.

[LEW94] Lewis, D.D. Ringuette, M. A comparison of two learning algorithms for text classification. In Third Annual Symposium on Document Analysis and Information Retrieval. Pp. 81-93. 1994.

[LEZ02] Lezcano, R.D. Minería de Datos. 2002.

http://www.google.com.cu/depar/areas/informatica/SistemasOperativos/MineriaDatosLezcano.pdf

[MAL04] Maloney, J. Text Mining Solutions. Services & Solutions. 2004.

[MAN00] Manning, C. Shütze, H. Foundations of Statistical Natural Language Processing. MIT Press, Cambridge, MA. 2000.

[MAN02] Mani, I. House, D. Klein, G. Hirschman, L. The TIPSTER SUMMAC Text Summarization Evaluation. The MITRE Corporation. 2002. http://acl.ldc.upenn.edu/E/E99/E99-1011.pdf

[MAR00] Marcu, D. The Theory and Practice of Discourse Parsing and summarization. Cambridge, MA: MIT Press. 2000.

[MCK99] McKeown, K. Klavans, J. Hatzivassiloglou, V. Towards Multidocument Summarization by Reformulation: Progress and Prospects. American Association for Artificial Intelligence. 1999.

[MED05] Mederos, J.M., Pérez, Y. Estudio de métodos de agrupamiento de documentos en el contexto de resúmenes de corpus textuales. Trabajo de diploma. Universidad Central "Marta Abreu" de Las Villas. 2005.

[MIT97] Mitchell, T. Machine Learning. McGraw-Hill Science, Engineering, Math. ISBN 0070428077. 1997.

[MLA98] Mladenic, D. Grobelnik, M. Feature selection for classification based on text hierarchy. In Working Notes of Learning from Text and the Web, Conference on Automated Learning and Discovery (CONALD98). 1998.

[MOE00] Moens, M.F. Automatic Indexing and Abstracting of Document Texts, Chapter 7. Norwell, MA: Kluwer Academic. 2000.

[NAN00] Nanba, H. Okumura, M. Producing More Readable Extracts by Revising Them. Proceedings of the 18th International Conference on Computational Linguistics (COLING-2000), pp. 1071-1075. 2000.

[NER01] Neri, F. A new way to Explore Patents Databases. Text Mining Solutions – Lexical Systems. 2001. http://www.synthema.it/textmining

[NIG00] Nigam, K. McCallum, A. Thrun, S. Mitchell, T. Text classification from labelled and unlabeled documents using EM. Machine Learning 39(2/3). Pp. 103-134. 2000.

[NUR01] Nürnberger, A. Klose, A. Kruse, R. Clustering of Document Collection to Support Interactive Text Exploration. Studies in Classification, Data Analysis and Knowledge Organization. Exploratory Data Analysis in Empirical Research. Proceedings of the 25th Annuals Conference of the Gesellschaft für Klassification. Pp 291-299. 2001.

[OBE01] Obeso, C. Apuntes de éxito. 2001. www.infonomia.com

[POR80] Porter, M.F. An algorithm for suffix stripping. Program 14(3). Pp. 130-137. 1980.

[PRO03] Proyecto México-Cuba. Redes Neuronales para la Minería de Datos y Textos: Aplicación al Análisis Exploratorio y Descubrimiento de Conocimiento en Grandes Bases de Datos de Información Biomédica. 2003.

[QUE67] McQueen, J. Some methods for classification and analysis of multivariate observations. Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability. Pp. 182-297. 1967.

[QUI86] Quinlan, J.R. Induction of Decision Trees. Machine Learning. Pp 81-106. 1986.

[RAD00] Radev, D. Hongyan, J. Malgorzata, B. Centroid-based summarization of multiple documents: sentence extraction,utility-based evaluation, and user studies. 2000. http://acl.ldc.upenn.edu/W/W00/W00-0403.pdf

[RIJ79] Rijsbergen, C.J. Information Retrieval. London: Butterworths. 1979.

http://www.dcs.gla.ac.uk/Keith

[ROD99] Rodríguez, M. El concepto de tipo y la teoría de programación actual. Revista GIGA. No. 4. 1999.

[RUI95] Ruiz-Shulcloper, J. Alba-Cabrera, E. Lazo-Cortés, M. Introducción al Reconocimiento de Patrones. Grupo de Reconocimiento de Patrones Cuba México. Centro de Investigación y de estudios avanzados del IPN. Departamento de Ingeniería Eléctrica. Verde No 51. México. 1995.

[RUN01] Runkler, T.A. Bezdek, J.C. Relational Clustering for the Analysis of Internet Newsgroups. Proceedings of the 25th Annuals Conference of the Gesellschaft für Klassifikation. University of Munich. Pp. 291-299. 2001.

[SAH98] Sahami, M. Using Machine Learning to Improve Information Access. Ph. D. thesis, Department of Computer Science, Stanford University. 1998.

[SAL71] Salton, G. The SMART Retrieval System. Prentice-Hall, Englewood Cliffs, NJ. 1971.

[SAL75] Salton, G. Wong, A. Yang, C.S. A vector space model for automatic text retrieval. Communications of the ACM. 18(11). Pp. 613-620. 1975.

[SAL83] Salton, G., McGill, M. Modern Information Retrieval. New York: McGraw-Hill. 1983.

[SAL88] Salton, G. Buckley, C. Term Weighting approaches in automatic text retrieval. Information Processing and Management 24(5). Pp. 513-523. 1988.

[SAR00] Sarwar, B. Karypis, G. Konstan, J. Riedl, J. Analysis of Recommendation Algorithms for E-Commerce. 2000. http://www.grouplens.org/papers/padf/ec00.pdf

[SCH02] Schiffman, B. Nenkova, A. McKeown, K. Experiments in Multidocument Summarization. Proceedings of HLT 2002 Human Language Technology Conference. San Diego, CA. 2002.

[SCH96] Schürmann, J. Pattern Clasification: A unified view of statistical and neural approaches. New York: Wiley-Interscience. 1996.

[SEB99] Sebastiani, F. A Tutorial on Automated Text Categorisation. 1999

http://net.pku.edu.cn/~webg/papers/sebastiani99tutorial.pdf

[SHA48] Shannon, C. A mathematical theory of communication, Bell System Technical Journal. 1948.

[SME97] Smeaton, A.F. Information retrieval: Still butting heads with natural language processing. In Information Extraction, International Summer School SCIE 1997. pp. 115-139. Springer-Verlag. 1997.

[STE00a] Stein, G. Bagga, A. Wise, G.B. Multi-document summarization: Methodologies and evaluations. Proceedings of TALN-2000. 2000.

[STE00b] Steinbach, M. Karypis, M. A Comparison of Document Clustering Techniques. 2000. http://www.l.cs.cmw.edu/~dunja/KDDpapers/steinbach-IR.pdf

[TAL04] Taller de Herramientas y Recursos Lingüísticos para el desarrollo del Español y el Portugués. IX Congreso Iberoamericano de Inteligencia Artificial (IBERAMIA). 2004.

[TAN99] Tan, A. Text Mining: The state of the art and the challenges. Proceedings of PAKDD'99. pp 65-70. 1999.

[TAU00] Tauritz, D.R. Kok, J.N. Sprinkhuisen-Kuyper, I.G. Adaptive information filtering using evolutionary computation. Information Sciences 122. pp. 121-140. 2000.

[VAL05] Valdés, L. Representación de textos y su reducción de dimensionalidad. Trabajo de diploma. Universidad Central "Marta Abreu" de Las Villas. 2005.

[WAN03] Wang, X. Borgelt, C. Information Measures in Fuzzy Decision Trees. 2003.

http://fuzzy.cs.uni-magdeburg.de/~borgelt/papers

[WHI02] White, M. Selecting Sentences for Multidocument Summaries using. Randomized Local Search.CoGenTex, Inc. 840 Hanshaw Road. Ithaca, NY 14850, USA 2002. http://www.cogentex.com/papers/acl-02-summ-wkshop.pdf

[WIL97] Wilson, D.R. Martinez, T.R. Improved Heterogeneous Distance Functions. Journal of Artificial Intelligence Research. No. 6. pp. 1-34. 1997.

[YAN97] Yang, Y. Pedersen, J.O. A comparative study on feature selection in text categorization. In Machine Learning: Proceedings of the Fourteenth International Conference (ICML'97). Pp. 412-420. 1997.

[YAR92] Yarowsky, D. Word sense disambiguation using statistical models of roget's categories trained on large corpora. In Proceedings of the Fourteenth International Conference on Computational Linguistics. Pp. 454-460, Nantes, France. 1992.

[ZIP49] Zipf, G.K. Human Behaviour and the Principle of Least Effort. Reading, MA: Addison-Wesley. 1949.

Anexos

Anexo 1. Resumen de los principales elementos de frecuencias de términos y número de documentos usados para la evaluación de las métricas numéricas de calidad de términos.

N	Número de documentos de la colección
$n(t)$	Número de documentos en los cuales el término *t* aparece al menos una vez
$\bar{n}(t)$	Número de documentos en los cuales el término *t* no aparece
tf	Número de ocurrencias de todos los términos en todos los documentos
$tf(t)$	Número de ocurrencias del término *t* en todos los documentos
$tf_{d_j}(t)$	Número de ocurrencias del término *t* en el documento d_j

Anexo 2. Fragmento de una lista de palabras de parada frecuentemente usada.

a	be	each	If	last	near	that
about	but	else	In	late	no	the
all	by	…	Is	like	…	they
an	…	for	It	…	of	to
and	did	from	Into	many	often	…
are	do	further	Itself	much	on	with
as	down	…	…	more	once	which
at	during	get	Just	must	or	whether

Anexo 3. Resumen de algunos componentes pesados de los términos.

Código	**Descripción**	**Expresión**
Componente local del término *t* en el documento *d*:		$w_{local}(d,t)$ = **(if $tf_d(t)>0$)**
N	No conversión (frecuencia del término plana)	$tf_d(t)$
B	Indicador del término binario	1
M	Normalizar por el término de mayor frecuencia $t_{\max}$	$\frac{tf_d(t)}{tf_d(t_{\max})}$
A	Frecuencia del término normalizada aumentada	$\frac{1}{2}+\frac{1}{2}\frac{tf_d(t)}{tf_d(t_{\max})}$
L	Logaritmo de la frecuencia del término	$1+\log tf_d(t)$
Componente global para el término *t*:		$w_{global}(t)$ =
N	No peso global	1
T	tfidf estilo de frecuencia inversa de documentos	$\log\frac{n}{n(t)}$
P	Probabilidad de la frecuencia inversa de documentos	$\log\frac{\bar{n}(t)}{n(t)}$
Componente de normalización para el documento *d*:		$w_{normalización}(d)$ =
N	No normalización	1
S	Normalización por la suma de todos los pesos de los términos	$\sum_{i=1}^{m} w_{local}(d,t_i)w_{global}(d,t_i)$
C	Normalización coseno	$\sqrt{\sum_{i=1}^{m}\left(w_{local}(d,t_i)w_{global}(d,t_i)\right)^2}$

Anexo 4. Diferentes enfoques lingüísticos que intentan capturar, o ignorar, una cierta extensión del significado con respecto al contexto.

Estos enfoques se dividen en cinco niveles [LAN01]:

1. Nivel de grafema: Análisis sobre un nivel de sub-palabra, comúnmente concerniente a las letras.
2. Nivel léxico: Análisis concerniente a palabras individuales.

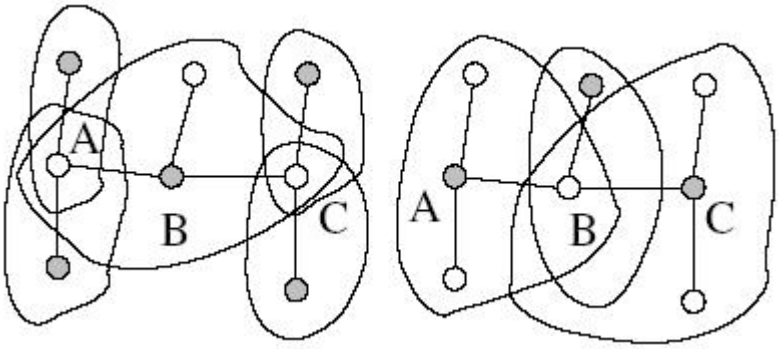
Note que los dos primeros niveles operan solamente con un plano estadístico sobre el texto, i.e. básicamente sobre frecuencias de combinaciones de términos, que pueden ser letras o palabras. Para determinadas aplicaciones de la minería de textos, estas representaciones basadas en esas frecuencias de términos no pueden completamente capturar el significado de los documentos.

3. Nivel sintáctico: Análisis concerniente a la estructura de oraciones.
4. Nivel semántico: Análisis relativo al significado de palabras y frases.
5. Nivel pragmático: Análisis relativo al significado, tanto dependientes del contexto como independientes del contexto (e.g., aplicaciones específicas, contextos).

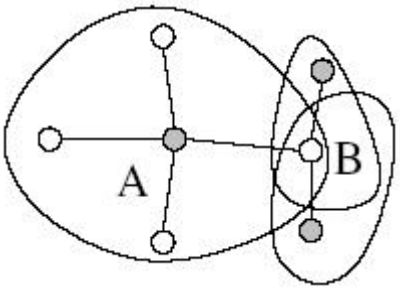
Los niveles más altos del análisis de textos existen para tratar de capturar mayor contenido semántico a través de la explotación de la cantidad creciente de información contextual tal como la estructura de las oraciones, párrafos o documentos. En general, las técnicas de procesamiento de lenguaje natural pueden proveer una representación mucho más rica a través de un análisis sintáctico y semántico de documentos.

Anexo 5

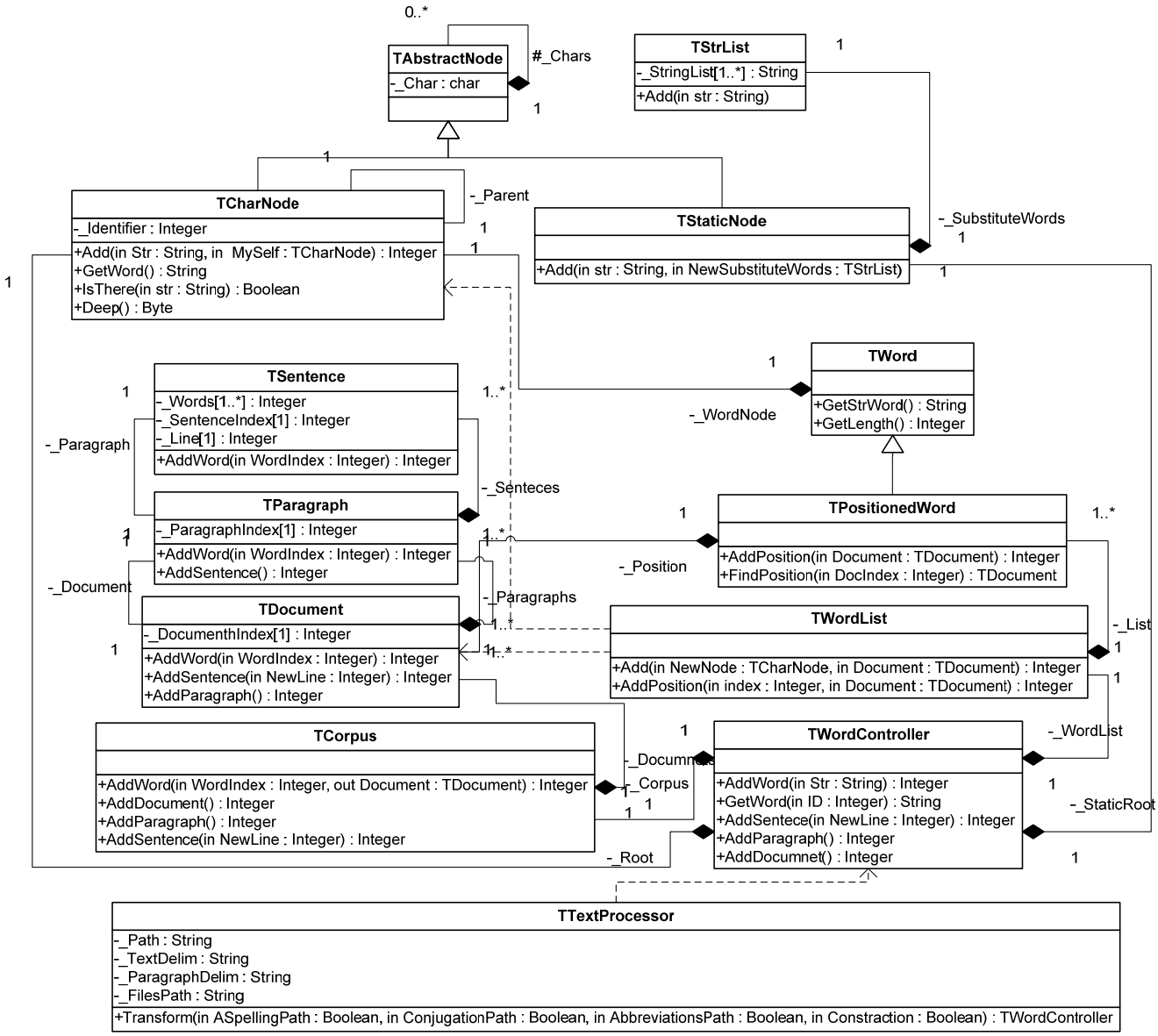
a) Ejemplo de agrupamiento con el algoritmo *Star* donde influye el orden de los datos.



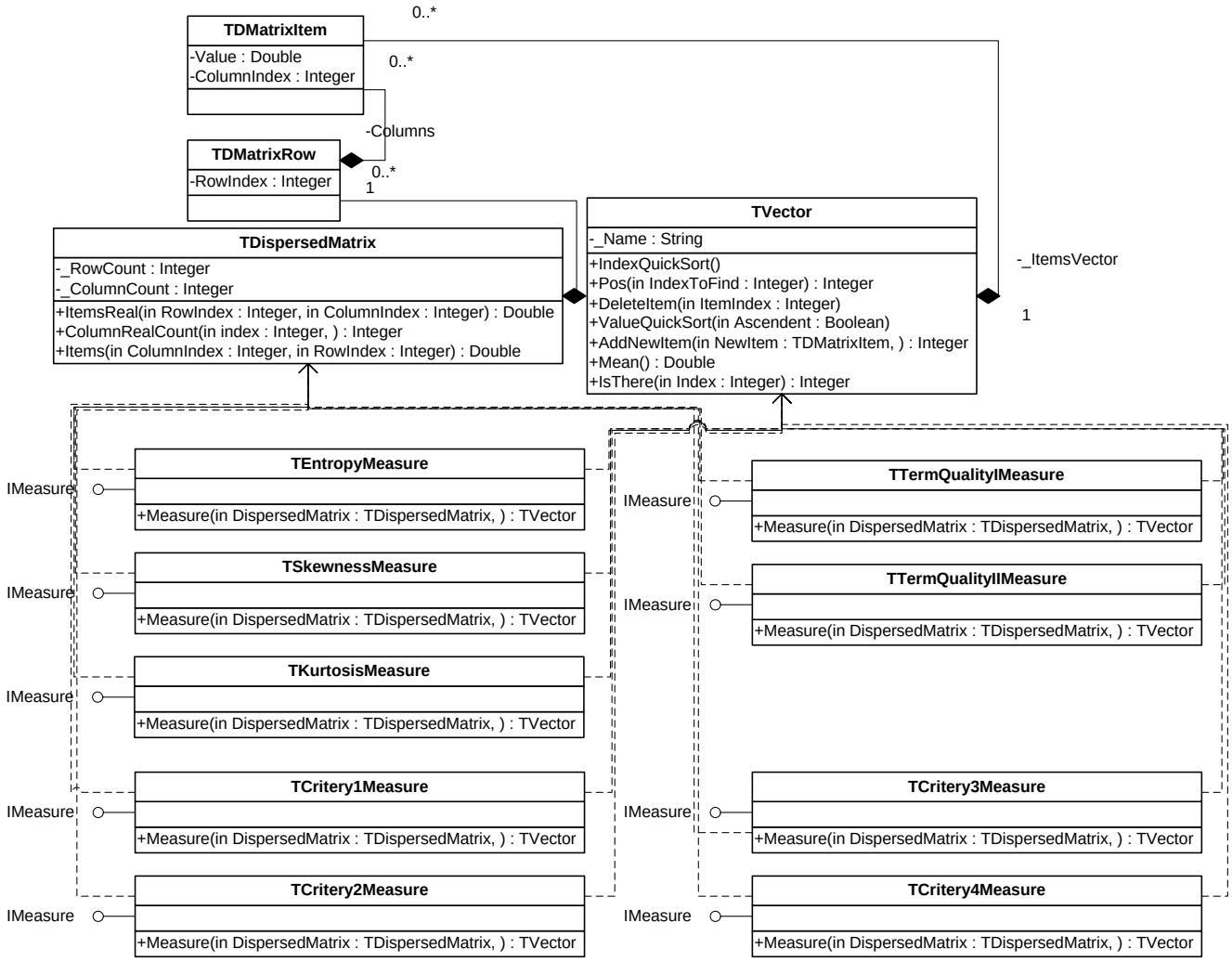
b) Ejemplo de la obtención de clusters ilógicos según agrupamiento *Star* debido a que dos estrellas nunca pueden ser vecinas.



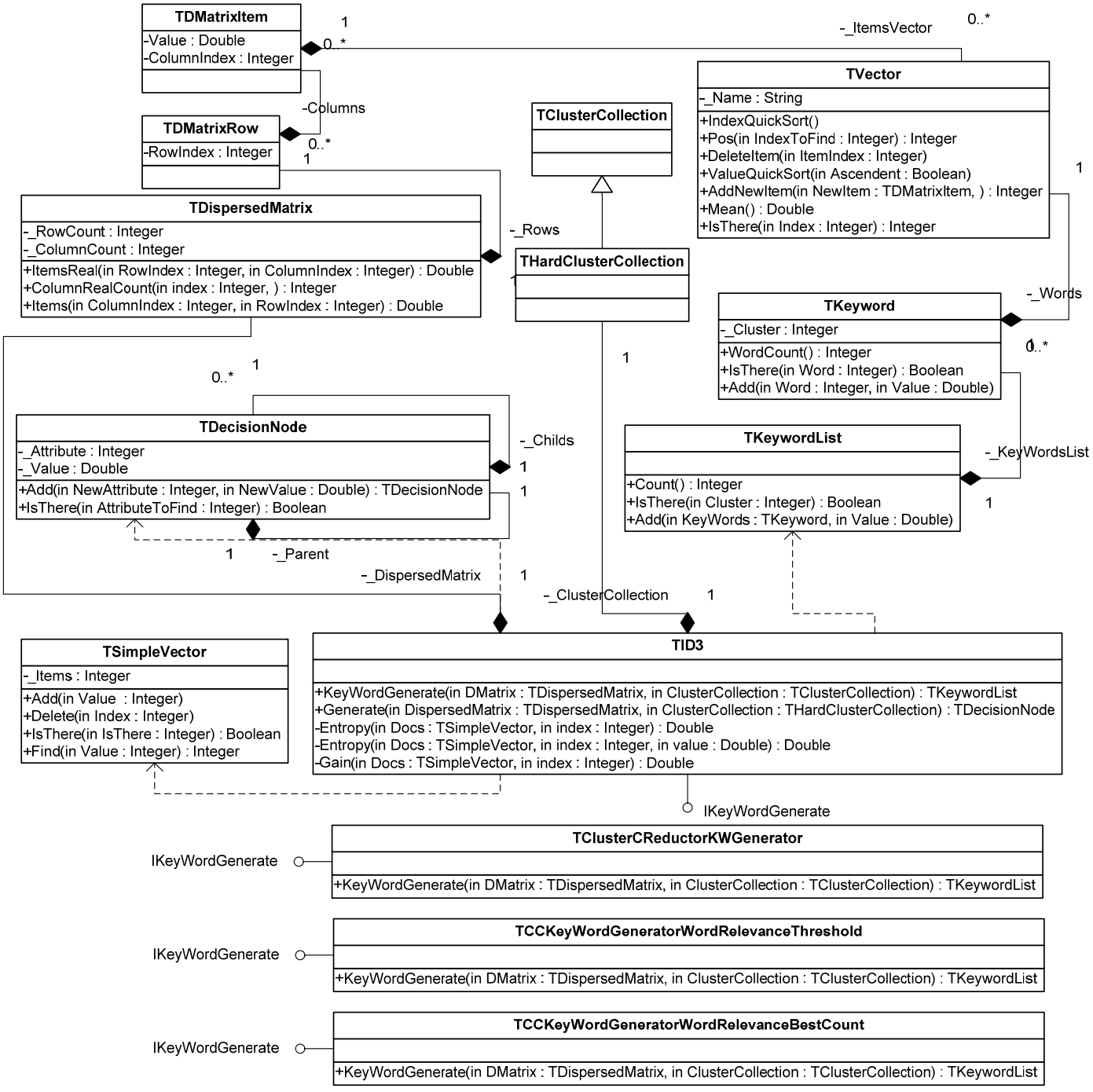
Anexo 6. Diseño UML de las clases relacionadas con la transformación del corpus.



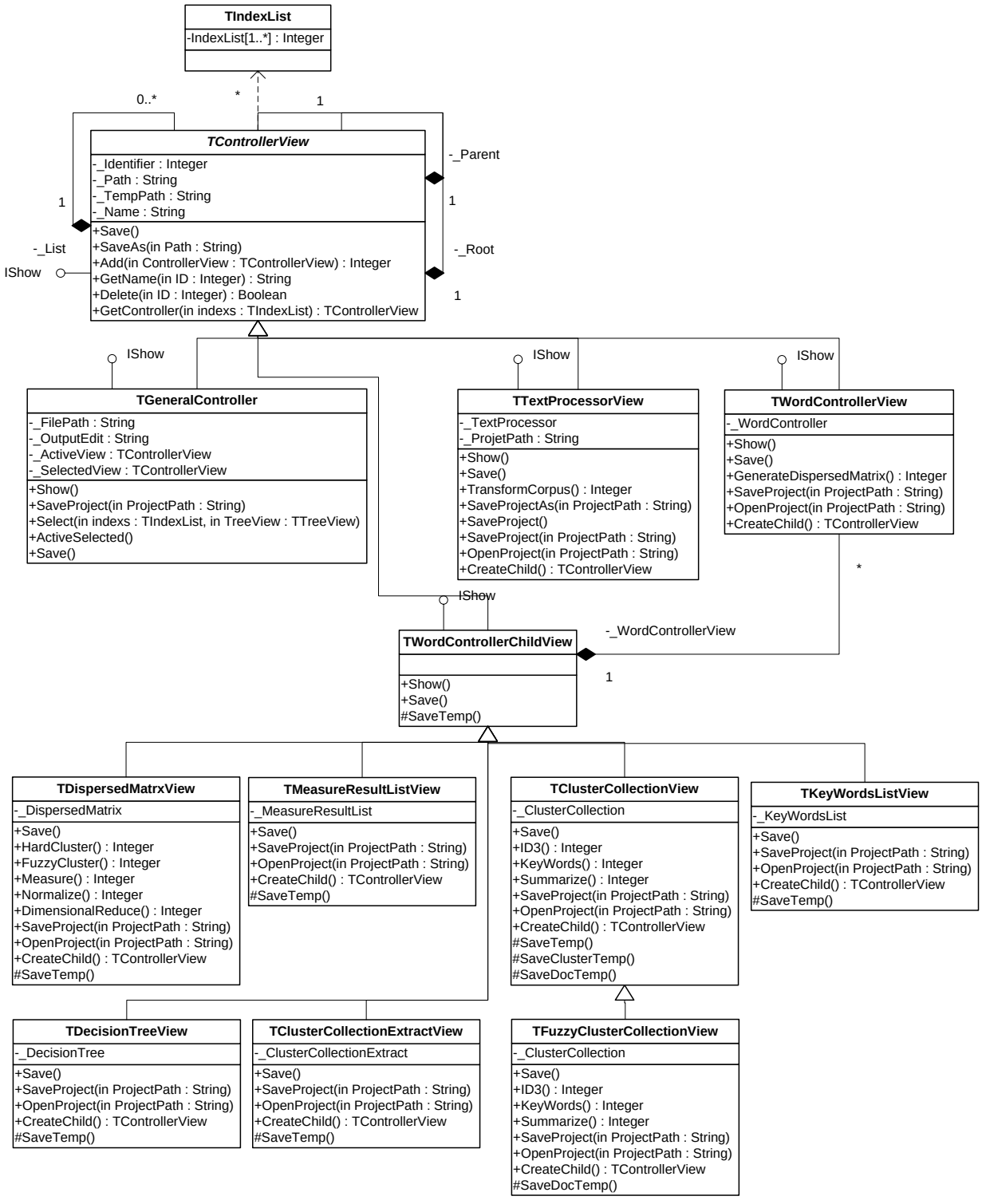
Anexo 7. Diseño UML de las clases que permiten medir la calidad de los términos.



Anexo 8. Diagrama de clases de la extracción de palabras claves.



Anexo 9. Diseño general de los controladores en CorpusMiner.



Anexo 10. Fragmentos de documentos del corpus de textos asociados a palabras.

…
Documento: WEATHER
This is always true; the **weathe**r is, by its nature, capricious, as farmers and sailors know.
Because many **weathe**r processes are simply too small to be captured in such models, a lot of their predictions have to be generated using empirical rules.
And some **weather** – forming events in the ocean happen on a scale big enough for models to get hold of.
…
Documento: WEB
By looking at the frequency of references to **web** pages, Lycos searches out the most popular, and in this way hopes to find the most significant documents.
The unhappy fly whose impact is with a spider's **web** has seen no such improvement in its chances, however.
Peter Rice, Ove Arup's chief engineer until his death last year, saw work on the diversity of spider – **web** designs being done by Fritz Vollrath, a zoologist at the University of Oxford, and by his physicist colleague Donald Edmonds, as an opportunity for the company's engineers to practice a little lateral thinking.
…
Documento: WEEK
Many of the familiar arguments will be aired again during a **week** of protest and discussion in the United States and Europe that starts on April 24th which an alliance of animal – welfare groups is calling International Laboratory Animal Day.
The results were made public last **week** at the annual congress of the Society of Automotive Engineers in Detroit.
Once a bug has been found, identifying it accurately generally takes a **week** of analysis.
…
Documento: WOMAN
Only one egg is normally selected for final maturation and ovulation per menstrual cycle of which a **woman** can expect perhaps 400 in her life.
The precision that allowed him to remove immature eggs without damaging the woman distinguishes Dr Trounson's work from a previous case in which a **woman** bore a child grown from an egg matured in a test tube.
In 1991 K.Y. Cha and his colleagues at the Cha Woman's Hospital in Seoul, South Korea, removed ovarian tissue from a **woman** undergoing surgery for fibroid tumours, then extracted and igitized five eggs.
Some might provide messages on liquid crystal displays for example, warning a **woman** it is more than a day since she last took a contraceptive pill.
…
Documento: WORDS
In human beings the ability is highly evolved, allowing them to sift **words** from a clatter of noise entering their ears.
In the 1970s a group of classicists at the University of California, Irvine, thought up a then extraordinary goal: having every extant **word** of ancient Greek literature in a single database; 3,000 authors, 66m words – all searchable, accessible and printable.
Shakespeare's **words** have been done as well as God's indeed, the entire canon of British poetry prior to this century is now igitized.
…

Anexo 11. Matriz construida para realizar el análisis estadístico del agrupamiento.

Corpus	SKWIC						Fuzzy SKWIC						Extended Star – SKWIC						Extended Star – Fuzzy SKWIC						Extended Star					
	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
C1																														
C2																														
C3																														
C4																														
C5																														
C6																														
C7																														
C8																														
C9																														
C10																														

Métricas consideradas en la evaluación:

1. *Precision*
2. *Recall*
3. F-*Measure*
4. Entropía
5. Media *Overall Similarity*
6. Desviación Estándar *Overall Similarity*

Las métricas son calculadas a partir de los resultados del agrupamiento al aplicar cada algoritmo o combinación de éstos a los 10 corpus textuales.

Nótese que a partir de *Overall Similarity* se derivan dos formas de evaluar el agrupamiento: Media *Overall Similarity* y Desviación Estándar *Overall Similarity*, ya que esta métrica reporta valores por cada clusters, y se utiliza la media y la desviación estándar para tener un criterio general de la colección.

Anexo 12. Resultados de aplicar las pruebas de Friedman según las métricas de calidad del agrupamiento a cada algoritmo.

a) Métricas *Presicion*, *Recall* y F-*Measure*

Rangos

	Rango promedio
Precision SKWIC	3,20
Precision F-SKWIC	1,20
Precision ES-SKWIC	3,80
Precision ES-F-SKWIC	1,80
Precision ES	5,00

Estadísticos de contraste[a]

N			10
Chi-cuadrado			37,440
gl			4
Sig. Monte Carlo	Sig.		,000
	Intervalo de confianza de 99%	Límite inferior	,000
		Límite superior	,000

a. Prueba de Friedman

Rangos

	Rango promedio
Recall SKWIC	2,50
Recall F-SKWIC	4,55
Recall ES-SKWIC	2,50
Recall ES-F-SKWIC	4,45
Recall ES	1,00

Estadísticos de contraste[a]

N			10
Chi-cuadrado			36,569
gl			4
Sig. Monte Carlo	Sig.		,000
	Intervalo de confianza de 99%	Límite inferior	,000
		Límite superior	,000

a. Prueba de Friedman

Rangos

	Rango promedio
F-Measure SKWIC	3,80
F-Measure F-SKWIC	1,20
F-Measure ES-SKWIC	4,30
F-Measure ES-F-SKWIC	1,80
F-Measure ES	3,90

Estadísticos de contraste[a]

N			10
Chi-cuadrado			31,280
gl			4
Sig. Monte Carlo	Sig.		,000
	Intervalo de confianza de 99%	Límite inferior	,000
		Límite superior	,000

a. Prueba de Friedman

Anexo 12. Resultados de aplicar las pruebas de Friedman según las métricas de calidad del agrupamiento a cada algoritmo.

b) Métricas Entropía, Media *Overall Similarity* y Desviación Estándar *Overall Similarity*

Rangos

	Rango promedio
Entropía SKWIC	2,80
Entropía F-SKWIC	4,85
Entropía ES-SKWIC	2,20
Entropía ES-F-SKWIC	4,15
Entropía ES	1,00

Estadísticos de contraste[a]

N			10
Chi-cuadrado			37,889
gl			4
Sig. Monte Carlo	Sig.		,000
	Intervalo de confianza de 99%	Límite inferior	,000
		Límite superior	,000

a. Prueba de Friedman

Rangos

	Rango promedio
Media OS SKWIC	2,60
Media OS F-SKWIC	1,90
Media OS ES-SKWIC	3,00
Media OS ES-F-SKWIC	2,60
Media OS ES	4,90

Estadísticos de contraste[a]

N			10
Chi-cuadrado			20,560
gl			4
Sig. Monte Carlo	Sig.		,000
	Intervalo de confianza de 99%	Límite inferior	,000
		Límite superior	,000

a. Prueba de Friedman

Rangos

	Rango promedio
DesvEstandar OS SKWIC	2,40
DesvEstandar OS F-SKWIC	2,20
DesvEstandar OS ES-SKWIC	3,10
DesvEstandar OS ES-F-SKWIC	2,90
DesvEstandar OS ES	4,40

Estadísticos de contraste[a]

N			10
Chi-cuadrado			11,920
gl			4
Sig. Monte Carlo	Sig.		,013
	Intervalo de confianza de 99%	Límite inferior	,010
		Límite superior	,016

a. Prueba de Friedman

Anexo 13. a) Comparación de los algoritmos SKWIC y *Extended Star* – SKWIC.

Estadísticos de contraste[c,d]

			Precision ES-SKWIC - Precision SKWIC	Recall ES-SKWIC - Recall SKWIC	F-Measure ES-SKWIC - F-Measure SKWIC	Entropía ES-SKWIC - Entropía SKWIC	Media OS ES-SKWIC - Media OS SKWIC	DesvEstandar OS ES-SKWIC - DesvEstandar OS SKWIC
Z			-1,886[a]	-,051[b]	-1,580[a]	-1,580[b]	-1,478[a]	-,866[a]
Sig. Monte Carlo (bilateral)	Sig.		,064	,977	,128	,128	,160	,422
	Intervalo de confianza de 99%	Límite inferior	,055	,951	,116	,116	,146	,401
		Límite superior	,073	1,000	,141	,141	,174	,443

a. Basado en los rangos negativos.

b. Basado en los rangos positivos.

c. Prueba de los rangos con signo de Wilcoxon

d. Basado en 10000 tablas muestrales con semilla de inicio 2000000.

Anexo 13. b) Comparación de los algoritmos SKWIC y *Extended Star* – SKWIC.

Rangos

		N	Rango promedio	Suma de rangos
Precision ES-SKWIC - Precision SKWIC	Rangos negativos	2[a]	4,50	9,00
	Rangos positivos	8[b]	5,75	46,00
	Empates	0[c]		
	Total	10		
Recall ES-SKWIC - Recall SKWIC	Rangos negativos	5[d]	5,60	28,00
	Rangos positivos	5[e]	5,40	27,00
	Empates	0[f]		
	Total	10		
F-Measure ES-SKWIC - F-Measure SKWIC	Rangos negativos	3[g]	4,00	12,00
	Rangos positivos	7[h]	6,14	43,00
	Empates	0[i]		
	Total	10		
Entropía ES-SKWIC - Entropía SKWIC	Rangos negativos	8[j]	5,38	43,00
	Rangos positivos	2[k]	6,00	12,00
	Empates	0[l]		
	Total	10		
Media OS ES-SKWIC - Media OS SKWIC	Rangos negativos	3[m]	4,33	13,00
	Rangos positivos	7[n]	6,00	42,00
	Empates	0[o]		
	Total	10		
DesvEstandar OS ES-SKWIC - DesvEstandar OS SKWIC	Rangos negativos	3[p]	6,33	19,00
	Rangos positivos	7[q]	5,14	36,00
	Empates	0[r]		
	Total	10		

a. Precision ES-SKWIC < Precision SKWIC

b. Precision ES-SKWIC > Precision SKWIC

c. Precision SKWIC = Precision ES-SKWIC

d. Recall ES-SKWIC < Recall SKWIC

e. Recall ES-SKWIC > Recall SKWIC

f. Recall SKWIC = Recall ES-SKWIC

g. F-Measure ES-SKWIC < F-Measure SKWIC

h. F-Measure ES-SKWIC > F-Measure SKWIC

i. F-Measure SKWIC = F-Measure ES-SKWIC

j. Entropía ES-SKWIC < Entropía SKWIC

k. Entropía ES-SKWIC > Entropía SKWIC

l. Entropía SKWIC = Entropía ES-SKWIC

m. Media OS ES-SKWIC < Media OS SKWIC

n. Media OS ES-SKWIC > Media OS SKWIC

o. Media OS SKWIC = Media OS ES-SKWIC

p. DesvEstandar OS ES-SKWIC < DesvEstandar OS SKWIC

q. DesvEstandar OS ES-SKWIC > DesvEstandar OS SKWIC

r. DesvEstandar OS SKWIC = DesvEstandar OS ES-SKWIC

Anexo 14. a) Comparación de los algoritmos *Fuzzy* SKWIC y *Extended Star – Fuzzy* SKWIC.

Estadísticos de contraste[c,d]

			Precision ES-F-SKWIC - Precision F-SKWIC	Recall ES-F-SKWIC - Recall F-SKWIC	F-Measure ES-F-SKWIC - F-Measure F-SKWIC	Entropía ES-F-SKWIC - Entropía F-SKWIC	Media OS ES-F-SKWIC - Media OS F-SKWIC	DesvEstandar OS ES-F-SKWIC - DesvEstandar OS F-SKWIC
Z			-1,070[a]	-,676[b]	-1,070[a]	-1,955[b]	-1,274[a]	-1,070[a]
Sig. Monte Carlo (bilateral)	Sig.		,326	,581	,326	,052	,232	,326
	Intervalo de confianza de 99%	Límite inferior	,306	,557	,306	,044	,215	,306
		Límite superior	,345	,604	,345	,060	,248	,345

a. Basado en los rangos negativos.

b. Basado en los rangos positivos.

c. Prueba de los rangos con signo de Wilcoxon

d. Basado en 10000 tablas muestrales con semilla de inicio 1502173562.

Anexo 14. b) Comparación de los algoritmos *Fuzzy* SKWIC y *Extended Star – Fuzzy* SKWIC.

Rangos

		N	Rango promedio	Suma de rangos
Precision ES-F-SKWIC - Precision F-SKWIC	Rangos negativos	2[a]	8,50	17,00
	Rangos positivos	8[b]	4,75	38,00
	Empates	0[c]		
	Total	10		
Recall ES-F-SKWIC - Recall F-SKWIC	Rangos negativos	4[d]	4,50	18,00
	Rangos positivos	3[e]	3,33	10,00
	Empates	3[f]		
	Total	10		
F-Measure ES-F-SKWIC - F-Measure F-SKWIC	Rangos negativos	2[g]	8,50	17,00
	Rangos positivos	8[h]	4,75	38,00
	Empates	0[i]		
	Total	10		
Entropía ES-F-SKWIC - Entropía F-SKWIC	Rangos negativos	8[j]	4,88	39,00
	Rangos positivos	1[k]	6,00	6,00
	Empates	1[l]		
	Total	10		
Media OS ES-F-SKWIC - Media OS F-SKWIC	Rangos negativos	3[m]	5,00	15,00
	Rangos positivos	7[n]	5,71	40,00
	Empates	0[o]		
	Total	10		
DesvEstandar OS ES-F-SKWIC - DesvEstandar OS F-SKWIC	Rangos negativos	5[p]	3,40	17,00
	Rangos positivos	5[q]	7,60	38,00
	Empates	0[r]		
	Total	10		

a. Precision ES-F-SKWIC < Precision F-SKWIC

b. Precision ES-F-SKWIC > Precision F-SKWIC

c. Precision F-SKWIC = Precision ES-F-SKWIC

d. Recall ES-F-SKWIC < Recall F-SKWIC

e. Recall ES-F-SKWIC > Recall F-SKWIC

f. Recall F-SKWIC = Recall ES-F-SKWIC

g. F-Measure ES-F-SKWIC < F-Measure F-SKWIC

h. F-Measure ES-F-SKWIC > F-Measure F-SKWIC

i. F-Measure F-SKWIC = F-Measure ES-F-SKWIC

j. Entropía ES-F-SKWIC < Entropía F-SKWIC

k. Entropía ES-F-SKWIC > Entropía F-SKWIC

l. Entropía F-SKWIC = Entropía ES-F-SKWIC

m. Media OS ES-F-SKWIC < Media OS F-SKWIC

n. Media OS ES-F-SKWIC > Media OS F-SKWIC

o. Media OS F-SKWIC = Media OS ES-F-SKWIC

p. DesvEstandar OS ES-F-SKWIC < DesvEstandar OS F-SKWIC

q. DesvEstandar OS ES-F-SKWIC > DesvEstandar OS F-SKWIC

r. DesvEstandar OS F-SKWIC = DesvEstandar OS ES-F-SKWIC

Anexo 15. a) Comparación de los algoritmos SKWIC y *Extended Star*.

Estadísticos de contraste[c,d]

			Precision ES - Precision SKWIC	Recall ES - Recall SKWIC	F-Measure ES - F-Measure SKWIC	Entropía ES - Entropía SKWIC	Media OS ES - Media OS SKWIC	DesvEstandar OS ES - DesvEstandar OS SKWIC
Z			-2,803[a]	-2,805[b]	-,866[b]	-2,803[b]	-2,803[a]	-2,599[a]
Sig. Monte Carlo (bilateral)	Sig.		,002	,002	,432	,002	,002	,005
	Intervalo de confianza de 99%	Límite inferior	,000	,000	,411	,000	,000	,003
		Límite superior	,003	,003	,454	,003	,003	,008

a. Basado en los rangos negativos.

b. Basado en los rangos positivos.

c. Prueba de los rangos con signo de Wilcoxon

d. Basado en 10000 tablas muestrales con semilla de inicio 1585587178.

Anexo 15. b) Comparación de los algoritmos SKWIC y *Extended Star*.

Rangos

		N	Rango promedio	Suma de rangos
Precision ES - Precision SKWIC	Rangos negativos	0[a]	,00	,00
	Rangos positivos	10[b]	5,50	55,00
	Empates	0[c]		
	Total	10		
Recall ES - Recall SKWIC	Rangos negativos	10[d]	5,50	55,00
	Rangos positivos	0[e]	,00	,00
	Empates	0[f]		
	Total	10		
F-Measure ES - F-Measure SKWIC	Rangos negativos	5[g]	7,20	36,00
	Rangos positivos	5[h]	3,80	19,00
	Empates	0[i]		
	Total	10		
Entropía ES - Entropía SKWIC	Rangos negativos	10[j]	5,50	55,00
	Rangos positivos	0[k]	,00	,00
	Empates	0[l]		
	Total	10		
Media OS ES - Media OS SKWIC	Rangos negativos	0[m]	,00	,00
	Rangos positivos	10[n]	5,50	55,00
	Empates	0[o]		
	Total	10		
DesvEstandar OS ES - DesvEstandar OS SKWIC	Rangos negativos	1[p]	2,00	2,00
	Rangos positivos	9[q]	5,89	53,00
	Empates	0[r]		
	Total	10		

a. Precision ES < Precision SKWIC

b. Precision ES > Precision SKWIC

c. Precision SKWIC = Precision ES

d. Recall ES < Recall SKWIC

e. Recall ES > Recall SKWIC

f. Recall SKWIC = Recall ES

g. F-Measure ES < F-Measure SKWIC

h. F-Measure ES > F-Measure SKWIC

i. F-Measure SKWIC = F-Measure ES

j. Entropía ES < Entropía SKWIC

k. Entropía ES > Entropía SKWIC

l. Entropía SKWIC = Entropía ES

m. Media OS ES < Media OS SKWIC

n. Media OS ES > Media OS SKWIC

o. Media OS SKWIC = Media OS ES

p. DesvEstandar OS ES < DesvEstandar OS SKWIC

q. DesvEstandar OS ES > DesvEstandar OS SKWIC

r. DesvEstandar OS SKWIC = DesvEstandar OS ES

Anexo 16. a) Comparación de los algoritmos *Extended Star* - SKWIC y *Extended Star*.

Estadísticos de contraste[c,d]

			Precision ES - Precision ES-SKWIC	Recall ES - Recall ES-SKWIC	F-Measure ES - F-Measure ES-SKWIC	Entropía ES - Entropía ES-SKWIC	Media OS ES - Media OS ES-SKWIC	DesvEstandar OS ES - DesvEstandar OS ES-SKWIC
Z			-2,803[a]	-2,803[b]	-1,478[b]	-2,803[b]	-2,803[a]	-1,682[a]
Sig. Monte Carlo (bilateral)	Sig.		,002	,002	,160	,002	,002	,105
	Intervalo de confianza de 99%	Límite inferior	,000	,000	,146	,000	,000	,093
		Límite superior	,004	,004	,174	,004	,004	,116

a. Basado en los rangos negativos.

b. Basado en los rangos positivos.

c. Prueba de los rangos con signo de Wilcoxon

d. Basado en 10000 tablas muestrales con semilla de inicio 2000000.

Anexo 16. b) Comparación de los algoritmos *Extended Star* - SKWIC y *Extended Star*.

Rangos

		N	Rango promedio	Suma de rangos
Precision ES - Precision ES-SKWIC	Rangos negativos	0[a]	,00	,00
	Rangos positivos	10[b]	5,50	55,00
	Empates	0[c]		
	Total	10		
Recall ES - Recall ES-SKWIC	Rangos negativos	10[d]	5,50	55,00
	Rangos positivos	0[e]	,00	,00
	Empates	0[f]		
	Total	10		
F-Measure ES - F-Measure ES-SKWIC	Rangos negativos	6[g]	7,00	42,00
	Rangos positivos	4[h]	3,25	13,00
	Empates	0[i]		
	Total	10		
Entropía ES - Entropía ES-SKWIC	Rangos negativos	10[j]	5,50	55,00
	Rangos positivos	0[k]	,00	,00
	Empates	0[l]		
	Total	10		
Media OS ES - Media OS ES-SKWIC	Rangos negativos	0[m]	,00	,00
	Rangos positivos	10[n]	5,50	55,00
	Empates	0[o]		
	Total	10		
DesvEstandar OS ES - DesvEstandar OS ES-SKWIC	Rangos negativos	3[p]	3,67	11,00
	Rangos positivos	7[q]	6,29	44,00
	Empates	0[r]		
	Total	10		

a. Precision ES < Precision ES-SKWIC

b. Precision ES > Precision ES-SKWIC

c. Precision ES-SKWIC = Precision ES

d. Recall ES < Recall ES-SKWIC

e. Recall ES > Recall ES-SKWIC

f. Recall ES-SKWIC = Recall ES

g. F-Measure ES < F-Measure ES-SKWIC

h. F-Measure ES > F-Measure ES-SKWIC

i. F-Measure ES-SKWIC = F-Measure ES

j. Entropía ES < Entropía ES-SKWIC

k. Entropía ES > Entropía ES-SKWIC

l. Entropía ES-SKWIC = Entropía ES

m. Media OS ES < Media OS ES-SKWIC

n. Media OS ES > Media OS ES-SKWIC

o. Media OS ES-SKWIC = Media OS ES

p. DesvEstandar OS ES < DesvEstandar OS ES-SKWIC

q. DesvEstandar OS ES > DesvEstandar OS ES-SKWIC

r. DesvEstandar OS ES-SKWIC = DesvEstandar OS ES

Anexo 17. Matriz construida para realizar el análisis estadístico de los reductores de dimensionalidad.

Corpus	**Calidad de término 1**						**Calidad de término 2**						**Reductor por frecuencia de rango**						**Entropía**					
	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
C1																								
C2																								
C3																								
C4																								
C5																								
C6																								
C7																								
C8																								
C9																								
C10																								

Métricas consideradas en la evaluación:

1. *Precision*
2. *Recall*
3. F-*Measure*
4. Entropía
5. Media *Overall Similarity*
6. Desviación Estándar *Overall Similarity*

Las métricas son calculadas a partir de los resultados del agrupamiento al aplicar cada algoritmo o combinación de éstos a los 10 corpus textuales.

Nótese que a partir de *Overall Similarity* se derivan dos formas de evaluar el agrupamiento: Media *Overall Similarity* y Desviación Estándar *Overall Similarity*, ya que esta métrica reporta valores por cada clusters, y se utiliza la media y la desviación estándar para tener un criterio general de la colección.

Anexo 18. Resultados de aplicar las pruebas de Friedman según las métricas de calidad del agrupamiento para cada uno de los reductores.

Rangos

	Rango promedio
Precision Calidad 1	2,10
Precision Calidad 2	3,00
Precision Reductor	2,20
Precision Entropía	2,70

Estadísticos de contraste[a]

N			10
Chi-cuadrado			3,240
gl			3
Sig. Monte Carlo	Sig.		,386
	Intervalo de confianza de 99%	Límite inferior	,374
		Límite superior	,399

a. Prueba de Friedman

Rangos

	Rango promedio
Recall Calidad 1	2,00
Recall Calidad 2	2,90
Recall Reductor	2,80
Recall Entropía	2,30

Estadísticos de contraste[a]

N			10
Chi-cuadrado			3,240
gl			3
Sig. Monte Carlo	Sig.		,370
	Intervalo de confianza de 99%	Límite inferior	,357
		Límite superior	,382

a. Prueba de Friedman

Rangos

	Rango promedio
Entropía Calidad 1	3,10
Entropía Calidad 2	1,70
Entropía Reductor	3,10
Entropía Entropía	2,10

Estadísticos de contraste[a]

N			10
Chi-cuadrado			9,120
gl			3
Sig. Monte Carlo	Sig.		,023
	Intervalo de confianza de 99%	Límite inferior	,019
		Límite superior	,026

a. Prueba de Friedman

Rangos

	Rango promedio
Media OS Calidad 1	1,30
Media OS Calidad 2	2,50
Media OS Reductor	4,00
Media OS Entropía	2,20

Estadísticos de contraste[a]

N			10
Chi-cuadrado			22,680
gl			3
Sig. Monte Carlo	Sig.		,000
	Intervalo de confianza de 99%	Límite inferior	,000
		Límite superior	,000

a. Prueba de Friedman

Rangos

	Rango promedio
DesvEstandar OS Calidad 1	2,00
DesvEstandar OS Calidad 2	2,50
DesvEstandar OS Reductor	3,30
DesvEstandar OS Entropía	2,20

Estadísticos de contraste[a]

N			10
Chi-cuadrado			5,880
gl			3
Sig. Monte Carlo	Sig.		,118
	Intervalo de confianza de 99%	Límite inferior	,110
		Límite superior	,127

a. Prueba de Friedman

Anexo 19. Comparación de los reductores calidad de término 1 y calidad de término 2.

Rangos

		N	Rango promedio	Suma de rangos
Entropía Calidad 2 - Entropía Calidad 1	Rangos negativos	9[a]	5,67	51,00
	Rangos positivos	1[b]	4,00	4,00
	Empates	0[c]		
	Total	10		
Media OS Calidad 2 - Media OS Calidad 1	Rangos negativos	2[d]	3,00	6,00
	Rangos positivos	8[e]	6,13	49,00
	Empates	0[f]		
	Total	10		

a. Entropía Calidad 2 < Entropía Calidad 1

b. Entropía Calidad 2 > Entropía Calidad 1

c. Entropía Calidad 1 = Entropía Calidad 2

d. Media OS Calidad 2 < Media OS Calidad 1

e. Media OS Calidad 2 > Media OS Calidad 1

f. Media OS Calidad 1 = Media OS Calidad 2

Estadísticos de contraste[c,d]

			Entropía Calidad 2 - Entropía Calidad 1	Media OS Calidad 2 - Media OS Calidad 1
Z			-2,395[a]	-2,191[b]
Sig. Monte Carlo (bilateral)	Sig.		,015	,029
	Intervalo de confianza de 99%	Límite inferior	,011	,022
		Límite superior	,020	,035

a. Basado en los rangos positivos.

b. Basado en los rangos negativos.

c. Prueba de los rangos con signo de Wilcoxon

d. Basado en 10000 tablas muestrales con semilla de inicio 334431365.

Anexo 20. Comparación de los reductores por calidad de término 1 y por frecuencia de rango.

Rangos

		N	Rango promedio	Suma de rangos
Entropía Reductor - Entropía Calidad 1	Rangos negativos	4[a]	6,25	25,00
	Rangos positivos	6[b]	5,00	30,00
	Empates	0[c]		
	Total	10		
Media OS Reductor - Media OS Calidad 1	Rangos negativos	0[d]	,00	,00
	Rangos positivos	10[e]	5,50	55,00
	Empates	0[f]		
	Total	10		

a. Entropía Reductor < Entropía Calidad 1

b. Entropía Reductor > Entropía Calidad 1

c. Entropía Calidad 1 = Entropía Reductor

d. Media OS Reductor < Media OS Calidad 1

e. Media OS Reductor > Media OS Calidad 1

f. Media OS Calidad 1 = Media OS Reductor

Estadísticos de contraste[b,c]

			Entropía Reductor - Entropía Calidad 1	Media OS Reductor - Media OS Calidad 1
Z			-,255[a]	-2,803[a]
Sig. asintót. (bilateral)			,799	,005
Sig. Monte Carlo (bilateral)	Intervalo de confianza de 99%	Límite inferior	,816	,001
		Límite superior	,866	,004

a. Basado en los rangos negativos.

b. Prueba de los rangos con signo de Wilcoxon

c. Basado en 10000 tablas muestrales con semilla de inicio 743671174.

Anexo 21. Comparación de los reductores calidad de término 1 y entropía.

Rangos

		N	Rango promedio	Suma de rangos
Entropía Entropía - Entropía Calidad 1	Rangos negativos	8[a]	6,38	51,00
	Rangos positivos	2[b]	2,00	4,00
	Empates	0[c]		
	Total	10		
Media OS Entropía - Media OS Calidad 1	Rangos negativos	1[d]	9,00	9,00
	Rangos positivos	9[e]	5,11	46,00
	Empates	0[f]		
	Total	10		

a. Entropía Entropía < Entropía Calidad 1

b. Entropía Entropía > Entropía Calidad 1

c. Entropía Calidad 1 = Entropía Entropía

d. Media OS Entropía < Media OS Calidad 1

e. Media OS Entropía > Media OS Calidad 1

f. Media OS Calidad 1 = Media OS Entropía

Estadísticos de contraste[c,d]

			Entropía Entropía - Entropía Calidad 1	Media OS Entropía - Media OS Calidad 1
Z			-2,395[a]	-1,886[b]
Sig. Monte Carlo (bilateral)	Sig.		,012	,068
	Intervalo de confianza de 99%	Límite inferior	,008	,059
		Límite superior	,016	,078

a. Basado en los rangos positivos.

b. Basado en los rangos negativos.

c. Prueba de los rangos con signo de Wilcoxon

d. Basado en 10000 tablas muestrales con semilla de inicio 112562564.

Anexo 22. Comparación de los reductores por calidad de término 2 y por frecuencia de rango.

Rangos

		N	Rango promedio	Suma de rangos
Entropía Reductor - Entropía Calidad 2	Rangos negativos	3[a]	4,33	13,00
	Rangos positivos	7[b]	6,00	42,00
	Empates	0[c]		
	Total	10		
Media OS Reductor - Media OS Calidad 2	Rangos negativos	0[d]	,00	,00
	Rangos positivos	10[e]	5,50	55,00
	Empates	0[f]		
	Total	10		

a. Entropía Reductor < Entropía Calidad 2

b. Entropía Reductor > Entropía Calidad 2

c. Entropía Calidad 2 = Entropía Reductor

d. Media OS Reductor < Media OS Calidad 2

e. Media OS Reductor > Media OS Calidad 2

f. Media OS Calidad 2 = Media OS Reductor

Estadísticos de contraste[b,c]

			Entropía Reductor - Entropía Calidad 2	Media OS Reductor - Media OS Calidad 2
Z			-1,478[a]	-2,803[a]
Sig. Monte Carlo (bilateral)	Sig.		,160	,002
	Intervalo de confianza de 99%	Límite inferior	,146	,000
		Límite superior	,174	,003

a. Basado en los rangos negativos.

b. Prueba de los rangos con signo de Wilcoxon

c. Basado en 10000 tablas muestrales con semilla de inicio 303130861.

Anexo 23. Comparación de los reductores calidad de término 2 y entropía.

Rangos

		N	Rango promedio	Suma de rangos
Entropía Entropía - Entropía Calidad 2	Rangos negativos	3[a]	5,67	17,00
	Rangos positivos	7[b]	5,43	38,00
	Empates	0[c]		
	Total	10		
Media OS Entropía - Media OS Calidad 2	Rangos negativos	7[d]	5,43	38,00
	Rangos positivos	3[e]	5,67	17,00
	Empates	0[f]		
	Total	10		

a. Entropía Entropía < Entropía Calidad 2

b. Entropía Entropía > Entropía Calidad 2

c. Entropía Calidad 2 = Entropía Entropía

d. Media OS Entropía < Media OS Calidad 2

e. Media OS Entropía > Media OS Calidad 2

f. Media OS Calidad 2 = Media OS Entropía

Estadísticos de contraste[c,d]

			Entropía Entropía - Entropía Calidad 2	Media OS Entropía - Media OS Calidad 2
Z			-1,070[a]	-1,070[b]
Sig. Monte Carlo (bilateral)	Sig.		,320	,320
	Intervalo de confianza de 99%	Límite inferior	,301	,301
		Límite superior	,339	,339

a. Basado en los rangos negativos.

b. Basado en los rangos positivos.

c. Prueba de los rangos con signo de Wilcoxon

d. Basado en 10000 tablas muestrales con semilla de inicio 1335104164.

Anexo 24. Comparación de los reductores por entropía y por frecuencia de rango.

Rangos

		N	Rango promedio	Suma de rangos
Entropía Entropía - Entropía Reductor	Rangos negativos	8[a]	5,13	41,00
	Rangos positivos	2[b]	7,00	14,00
	Empates	0[c]		
	Total	10		
Media OS Entropía - Media OS Reductor	Rangos negativos	10[d]	5,50	55,00
	Rangos positivos	0[e]	,00	,00
	Empates	0[f]		
	Total	10		

a. Entropía Entropía < Entropía Reductor

b. Entropía Entropía > Entropía Reductor

c. Entropía Reductor = Entropía Entropía

d. Media OS Entropía < Media OS Reductor

e. Media OS Entropía > Media OS Reductor

f. Media OS Reductor = Media OS Entropía

Estadísticos de contraste[b,c]

			Entropía Entropía - Entropía Reductor	Media OS Entropía - Media OS Reductor
Z			-1,376[a]	-2,803[a]
Sig. Monte Carlo (bilateral)	Sig.		,205	,002
	Intervalo de confianza de 99%	Límite inferior	,189	,000
		Límite superior	,221	,004

a. Basado en los rangos positivos.

b. Prueba de los rangos con signo de Wilcoxon

c. Basado en 10000 tablas muestrales con semilla de inicio 1535910591.