Universidad Central “Marta Abreu” de Las Villas
Facultad de Matemática, Física y Computación

Trabajo de Diploma

Selección de rasgos a partir de grupos homogéneos de documentos

Autor:
Damny Magdaleno Guevara

Tutores:
Msc. Leticia Arco García
Dr. Rafael E. Bello Pérez

Julio 2006

Hago constar que el presente trabajo fue realizado en la Universidad Central Marta Abreu de Las Villas como parte de la culminación de los estudios de la especialidad de Ciencias de la Computación, autorizando a que el mismo sea utilizado por la institución, para los fines que estime conveniente, tanto de forma parcial como total y que además no podrá ser presentado en eventos ni publicado sin la autorización de la Universidad.

_____

Firma del autor

Los abajo firmantes, certificamos que el presente trabajo ha sido realizado según acuerdos de la dirección de nuestro centro y el mismo cumple con los requisitos que debe tener un trabajo de esta envergadura referido a la temática señalada.

_____ _____

Firma del tutor

Firma del jefe del Seminario
de Inteligencia Artificial

“Muchas cosas se juzgan imposibles de hacer, antes de que estén hechas”

Plinio

A mi mamá, la persona que más quiero y le debo todo en esta vida.

A mi hermanita, ni el tiempo ni la distancia nos separaran.

Agradecimientos

A Dios.

A Leticia, más que tutora has sido compañera, guía, amiga. Este trabajo no existiera sin tu apoyo. Gracias por tu paciencia sin límites.

A mi mamá, que lo ha dado todo por ver su sueño hecho realidad.

A mi abuela, mi segunda madre.

A mi papá, por hacer posible que esté escribiendo estas líneas.

A mis tíos, en especial a Eduardo, no se que sería de mí sin ti.

A Yuneisy, que ha estado a mi lado todo el tiempo brindándome su apoyo incondicional y dándome fuerzas para seguir adelante.

A Aylin, por todas las cosas que hemos compartido.

A Yanet, por ayudarme a tener este proyecto.

A Diego, por toda su preocupación

A "AI-TEXTLYNX", por no fallarme nunca.

Resumen

El objetivo general de la investigación consiste en desarrollar un modelo para la aplicación de técnicas de selección de rasgos para la extracción de términos relevantes que caractericen los grupos de documentos afines, soportado por un módulo implementado en el software CorpusMiner, que ofrece a los investigadores y desarrolladores en el campo de la minería de textos una herramienta que posibilita la extracción de palabras claves que permiten caracterizar corpus textuales y discernir entre clases.

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 selección de rasgos, y su aplicación en la minería de textos, particularmente inducción de árboles de decisión en la selección de rasgos. Se desarrolla un modelo conceptual flexible que justifica la concepción y posterior aplicación de las etapas del procedimiento general propuesto: discretización de los rasgos que describen los documentos, construcción de las variables lingüísticas asociadas a cada término, aplicación de los algoritmos ID3 duro o ID3 borroso, y extracción de palabras claves de grupos textuales homogéneos.

Finalmente, se muestra la viabilidad del modelo desarrollado a partir de su aplicación en dos casos de estudio utilizando la herramienta CorpusMiner. Se verificaron los resultados comparando con implementaciones en Weka y CorpusMiner del ID3 y C4.5. Se validaron los resultados a partir del análisis de las palabras claves obtenidas y su relación con los tópicos asociados a los grupos textuales que ellas caracterizan. Se demostró de esta forma la hipótesis de investigación planteada.

Abstract

The general aim of this research is to develop a conceptual model and a procedure supported in the software CorpusMiner, which offers researchers and developers in the field of the text mining a tool that makes possible the extraction of keywords that allow to characterize textual corpus and to discern between classes.

In the content of the work the theoretical framework of the research is explained, emphasizing the techniques most widely used at present for the feature selection, and its application in the text mining, particularly the induction of decision trees in the feature selection. A flexible conceptual model is developed that justifies the conception and later application of each of the stages of the general proposed procedure: discretization of the features that describe documents, construction of the linguistic variables associated to each term, application of hard ID3 or Fuzzy ID3 algorithms, and extraction of keywords of homogenous textual clusters.

Finally, the viability of the model developed is shown in two study cases by using the tool CorpusMiner that supports it. The results were verified comparing with implementations in WEKA and CorpusMiner of the ID3 and C4.5 algorithms. The results from the analysis of the keywords obtained and their relations with the topics associated to the textual groups that they characterize were validated. Thus the hypothesis of the research was proved.

Índice

Introducción

Se ha estimado que cada 20 meses aproximadamente la cantidad de información en el mundo se duplica. En la misma forma, herramientas para el uso en varios campos del conocimiento (adquisición, almacenamiento, recuperación, mantenimiento, etc) deben ser desarrolladas para combatir ese crecimiento. El conocimiento tiene valor solamente cuando puede ser usado eficiente y efectivamente; por tal motivo, la manipulación del conocimiento debe ser reconocida incrementalmente como un elemento importante en la extracción de su valor.

Un elemento fundamental es el proceso de descubrimiento de conocimiento (Knowledge Discovery in Databases (KDD)) y se puede definir 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 (Lezcano, 2002). Tradicionalmente, datos pueden ser convertidos en conocimiento a través de un análisis manual e interpretación. Para muchas aplicaciones, esta forma manual es lenta, costosa y tiene un alto grado de subjetividad. Además, como el volumen de los datos crece dramáticamente, este tipo de análisis manual de los datos se torna completamente impracticable en muchos dominios. Esto motiva, para lograr eficiente, la necesidad de automatizar el descubrimiento de conocimiento, sobre todo en fases de la minería de datos y de textos en particular.

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 (Lezcano, 2002). 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) (Dürsteler, 2001), que 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 (Obeso, 2001).

Un paso fundamental en los procesos de descubrimiento de conocimiento, y sobre todo en la minería de textos por el gran número de rasgos que describen los documentos, es la reducción de datos, y esta investigación se centrará en esta etapa. La alta dimensionalidad de los datos puede ser reducida usando técnicas adecuadas, dependiendo de los requerimientos del procesamiento futuro del KDD. Aquí los métodos de selección de rasgos tienen un papel fundamental, donde un subconjunto pequeño de los rasgos originales es escogido basado en una evaluación de ese subconjunto. En el descubrimiento de conocimiento, los métodos de selección de rasgos son particularmente deseables para facilitar la interpretabilidad del conocimiento resultante.

En la actualidad puede constatarse que se han desarrollado métodos y técnicas para la selección de rasgos como parte de la reducción de la dimensionalidad en la etapa de representación textual, sin embargo, no tanto así en otras etapas del procesamiento textual. Por tal motivo, se requiere el la aplicación de técnicas de selección de rasgos en otras etapas del procesamiento textual. Por lo anteriormente expuesto, se deriva el **problema científico** a resolver que se manifiesta en la necesidad de la aplicación de técnicas de selección de rasgos en la extracción de palabras claves que caractericen grupos textuales homogéneos y a la vez logran discernir entre clases, con una utilidad posterior en la calidad de los resúmenes automáticos a obtener de dichos grupos, 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 aplicación de técnicas de selección de rasgos es posible extraer las palabras claves que logran caracterizar grupos homogéneos de documentos y logran discernir entre las

clases con una utilidad posterior en la calidad de los resúmenes automáticos a obtener en dichos grupos.

En conformidad con la hipótesis de investigación identificada, el **objetivo general** de la investigación consiste en desarrollar un modelo que permita la aplicación de técnicas de selección de rasgos para la extracción de términos relevantes que caractericen los grupos de documentos afines.

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. Específicamente, definiciones de relevancia, selección de rasgos, técnicas para la selección de rasgos y selección de rasgos en la minería de textos.
2. Realizar un análisis crítico sobre el estado actual de las técnicas de selección de rasgos, enfatizando en las variantes que existen para la inducción de árboles de decisión borrosos; así como en los diferentes métodos que existen para la estimación de parámetros de las funciones de pertenencia necesarias en la construcción de los árboles de decisión borrosos.
3. Diseñar y proponer un modelo para la selección de palabras claves en grupos textuales homogéneos, que permita la combinación de la relevancia de las palabras obtenida por los métodos de agrupamiento, con la aplicación de la inducción de los árboles de decisión borrosos para lograr que los términos encontrados logren discernir entre clases.
4. Implementar e incorporar a la herramienta CorpusMiner el procedimiento general del modelo propuesto.
5. Evaluar el modelo 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 el software que soporte

el modelo.

6. Mostrar las posibilidades que brinda este módulo dentro del sistema CorpusMiner, encaminado a que los usuarios puedan utilizarlo con facilidad.

La **novedad científica** principal que aporta esta investigación, radica en la creación de un modelo que permite la aplicación de métodos árboles de decisión borrosos para la selección de las palabras claves que caracterizan grupos homogéneos de documentos, cuando se ha utilizado una técnica borrosa en el agrupamiento.

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

El **valor práctico** se relaciona con la aplicación de las palabras claves seleccionadas en procesamiento futuros en la minería de textos como lo es el resumen de múltiples documentos.

Para la presentación de esta investigación, este Trabajo de Diploma se estructuró de la forma siguiente. 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 desarrollado para la selección de palabras claves, así como la implementación e incorporación a CorpusMiner del procedimiento general que lo sustenta. Un Capítulo 3, donde se describe a nivel de usuario 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.

1

LA SELECCIÓN DE RASGOS Y SU USO EN LA MINERÍA DE TEXTOS

CAPÍTULO 1. LA SELECCIÓN DE RASGOS Y SU USO EN LA MINERÍA DE TEXTOS

La selección de rasgos, también conocida como selección de subconjuntos, es un proceso comúnmente usado en el aprendizaje automatizado (machine learning), donde un subconjunto de los rasgos disponibles en los datos se selecciona para el uso de un algoritmo de aprendizaje. La selección de rasgos es necesaria debido a que sería de un gran costo computacional utilizar todos los rasgos disponibles, o pueden surgir problemas de valoración cuando existen muestras limitadas de ejemplos y una gran cantidad de rasgos. En este capítulo se verán algunas técnicas usadas para la selección de rasgos, así como su utilidad en la minería de textos. Se particularizarán en el uso de los árboles de decisión en la selección de rasgos.

1.1 Definiciones de relevancia

Al referirse a la selección de rasgos, implícitamente se habla de selección de rasgos relevantes para el proceso de aprendizaje automático que se llevará a cabo. Sin embargo, existe un número de definiciones diferentes en la literatura de aprendizaje automático para la cual la palabra "relevancia" indica relevancia de rasgos. La razón de esa variedad depende generalmente de la interrogante: ¿relevante para qué? Las definiciones pueden ser más o menos apropiadas dependiendo del objetivo de la selección.

Se considera que existen n rasgos o atributos usados para describir ejemplos y cada rasgo i tiene algún dominio F_i. Un ejemplo es un punto en el espacio de instancias $F_1 \times F_2 \times \ldots \times F_n$. El conjunto de datos de entrenamiento es S, donde cada punto de dato es un ejemplo pareado con una etiqueta o clasificación asociada.

Una noción de relevancia simple es la relevancia al concepto etiquetado (relevant to the target concept), donde un rasgo x_i es relevante a un concepto etiquetado (clase) si existe un par de ejemplos A y B en el espacio de instancia tal que A y B difieran solamente en su asignación a x_i y la clase de A sea diferente a la clase de B. Esta definición no puede necesariamente determinar cuando algún rasgo es relevante o no. Es poco probable encontrar ejemplos con estas características. Esta definición sería útil en análisis teóricos de

algoritmos de aprendizaje, donde la noción de relevancia es usada para probar algunas propiedades de convergencia de un algoritmo.

Para remediar algunas de las desventajas de la definición anterior, John, Kohavi y Pfleger (Blum, 1997) definen dos nociones de relevancia, relevancia respecto a una distribución (relevance with respect to a distribution o relevance with respect to a sample), en sus variantes débil y fuerte. La diferencia fundamental es que no es requerido que *A* y *B* estén en *S* (o tengan una probabilidad distinta de cero). Estas nociones de relevancia son útiles para un algoritmo de aprendizaje porque intentan decidir cuáles rasgos mantener y cuáles ignorar. Rasgos que son fuertemente relevantes se deben mantener, mientras que rasgos que son débilmente relevantes, pueden o no ser importantes dependiendo de cuales otros rasgos son ignorados.

Otra noción es usar la relevancia como una medida de complejidad (relevance as a complexity measure). La idea es preguntar por el menor número de rasgos necesarios para conseguir un funcionamiento óptimo sobre *S* mediante un concepto en *C*. Esta noción de relevancia es independiente del algoritmo de aprendizaje a ser usado. Caruana y Freitag (Caruana, 1994) hacen explícita una noción de relevancia llamada utilidad incremental (incremental usefulness). Esta noción depende del algoritmo de aprendizaje y es especialmente natural para algoritmos de selección de rasgos que busquen el subconjunto de rasgos mediante la adición incremental o la eliminación de rasgos del conjunto actual.

Existe una variedad de extensiones realizadas a las definiciones anteriores. Por ejemplo, una puede ser considerar relevantes combinaciones lineales de rasgos, en lugar de rasgos individuales relevantes. Una pregunta pudiera ser: ¿Cuál es el espacio de menor dimensionalidad tal que la proyección de todos los ejemplos en *S* sobre el espacio preserve la existencia de una buena función en la clase *C*? Esta noción de relevancia es una de las más naturales para enfoques estadísticos de aprendizaje (e.g., Análisis de componentes principales).

1.2 Generalidades de la selección de rasgos

El problema de la selección de subconjuntos de rasgos se refiere a la tarea de identificar y seleccionar un subconjunto de rasgos a ser usados para representar patrones desde un gran conjunto de rasgos usualmente redundantes, posiblemente irrelevantes, y con riesgos.

Los conjuntos de datos consisten en un sistema de información descrito por un conjunto de atributos y los objetos en sí que representan combinaciones de valores válidos dentro del dominio de cada atributo. De esta forma, la selección rasgos en un sistema de información de este tipo, consiste en obtener un subconjunto de atributos tal que describa el sistema como si se tratara del conjunto completo. Esto quiere decir que el proceso se centra en encontrar los atributos más importantes dentro de los que se han utilizado para representar los datos y eliminar aquellos que se consideran irrelevantes y hacen más difícil el proceso de descubrimiento de conocimiento dentro de una base de ejemplos. Dicho de otro modo, la selección de rasgos representa el problema de encontrar un subconjunto óptimo de características (rasgos o atributos) del conjunto de datos y según cierto criterio, tales que se pueda generar un clasificador con la mayor calidad posible a través de un algoritmo inductivo que corra sobre los datos, pero sólo tomando en cuenta el subconjunto de atributos obtenido (Zhong, 2001).

La selección de rasgos puede ser vista como un problema de optimización, si hay m rasgos, el espacio de búsqueda tiene 2^m subconjuntos de rasgos candidatos. Obviamente, realizar una búsqueda exhaustiva es intratable cuando el número de rasgos es muy grande, sobre todo en dominios textuales donde el número de rasgos es alto. Por tal motivo, la selección de rasgos puede ser guiada por heurísticas.

El proceso de selección de rasgos consta de dos componentes principales: una función de evaluación y un método de búsqueda. La función de evaluación permite calcular la calidad de un subconjunto de rasgos; mientras que el método de búsqueda, por lo general una heurística, es el encargado de generar los subconjuntos de rasgos. Seleccionar los rasgos relevantes de un conjunto de datos es una tarea necesaria dentro del aprendizaje automatizado (machine learning), dada su importancia en el descubrimiento de reglas y/o relaciones en grandes volúmenes de datos entre otras aplicaciones, es por eso que la

selección de rasgos relevantes de un conjunto de datos con tiempos y costo de cómputo aceptables, ha sido en los últimos años tema de investigación de muchos autores en diferentes variantes; véase (Choubey, 1996), (Deogun, 1998), (Kohavi, 1994), (Liu, 1998) (Wroblewski, 1995).

1.2.1 Selección de rasgos como búsqueda heurística

Un paradigma conveniente para la selección de rasgos es la búsqueda heurística. Siguiendo este paradigma, la selección de rasgos debe considerar cuatro elementos básicos. Primero, determinar el punto de inicio que influirá en la dirección de la búsqueda (forward selection o backward elimination). La segunda decisión considera la organización de la búsqueda. El tercer elemento es concerniente a la estrategia usada para evaluar los subconjuntos alternativos de atributos. Una estrategia comúnmente usada es medir la habilidad de los atributos para discriminar entre clases. Muchos algoritmos de inducción incorporan un criterio basado en teoría de la información. Y por último, es necesario un criterio para detener la búsqueda.

Los métodos de selección de rasgos pueden ser agrupados en tres clases. A continuación se mencionarán algunas de las técnicas utilizadas en la selección de rasgos: aquellos que empotran la selección en el algoritmo básico de inducción (embed), aquellos que usan la selección para filtrar rasgos antes de la inducción (filter), y, por otra parte, si el algoritmo de inducción está atado al proceso de búsqueda, evaluación y selección de rasgos entonces se dice que la selección de rasgos emplea un modelo wrapper alrededor del proceso de inducción (wrapper).

1.2.1.1 Métodos empotrados para la selección de rasgos (embed)

Estos métodos están embebidos en un algoritmo de inducción básico y generalmente utilizan un ordenamiento parcial para organizar la búsqueda. Por ejemplo, métodos de particionamiento recursivo por inducción, tales como el ID3, (Quinlan, 1993), C4.5 (Quinlan, 1996), y CART (Quinlan, 1993), llevan a cabo una búsqueda greedy a través del espacio del árbol de decisión, cada etapa usa una función de evaluación para seleccionar el

atributo que ha sido elegido como el que tiene la mejor habilidad de discriminar entre clases.

1.2.1.2 Métodos de filtrado para la selección de rasgos (filter)

Estos métodos filtran los atributos irrelevantes antes que el proceso de inducción comience. La etapa de preprocesamiento usa características generales del conjunto de entrenamiento para seleccionar algunos rasgos y excluir otros. Estos métodos son independientes del algoritmo de inducción que será usado en la salida, y pueden ser combinados con otros muchos métodos. Un esquema de filtrado simple es evaluar cada rasgo individualmente basado en su correlación con la función etiquetadora (e.g., usando una medida de información mutua) y entonces seleccionar los *k* rasgos con mayor valor. Estos métodos son comúnmente usados en tareas de categorización de textos, usualmente en combinación con Bayes o un esquema de clasificación de vecinos más cercanos. También es posible evaluar cada rasgo individualmente sin tener en cuenta una correlación con la función etiquetadora en problemas de aprendizaje no supervisado, por ejemplo en un procesamiento previo al agrupamiento de documentos.

1.2.1.3 Métodos de cubierta para la selección de rasgos (wrapper)

Un tercer enfoque genérico para la selección de rasgos también ocurre fuera de los métodos de inducción básicos pero usan tales métodos como una subrutina, más que como un postprocesador. Un algoritmo wrapper típico busca el mismo espacio de subconjuntos de rasgos como los métodos embebidos y de filtrado, pero éste evalúa conjuntos alternativos corriendo algún algoritmo de inducción sobre el conjunto de entrenamiento y usando la precisión estimada del clasificador resultante como su métrica. Realmente, el esquema wrapper tiene una larga historia dentro de la literatura sobre estadística y reconocimiento de patrones, pero en el aprendizaje automático es relativamente reciente. El argumento general para el enfoque wrapper es que el método de inducción que usará el subconjunto de rasgos debe proveer un mejor estimado de la precisión que una medida separada (independiente) que pueda tener una influencia inductiva enteramente diferente. Existen autores que se han mostrado a favor de usar un método wrapper para mejorar el comportamiento de la inducción de árboles de decisión.

La mayor desventaja de los métodos wrapper respecto los métodos de filtrado es el costo computacional de los primeros, cuyos resultados dependen de la llamada del algoritmo de inducción para cada subconjunto de rasgos considerado.

1.2.1.4 Métodos para el pasado de rasgos (Feature Weighting Methods)

Estos son algoritmos que explícitamente intentan seleccionar un subconjunto de rasgos "más relevantes". Sin embargo, otro enfoque, especialmente para algoritmos embebidos, es aplicar una función pesada para seleccionar los rasgos, en efecto asignarles los grados de relevancia percibidos. En (Blum, 1997) separan estos métodos del enfoque de selección de rasgos explícitamente porque las motivaciones y usos para estos dos modos tienden a ser diferentes. La selección de rasgos explícita es generalmente más natural cuando se intenta que el resultado sea entendido por los humanos, o que su resultado sea utilizado dentro de otro algoritmo.

Esquemas pesados pueden ser vistos en términos de búsqueda heurística, como fue visto explícitamente en los métodos de selección de rasgos. Sin embargo, como el espacio pesado requiere de orden parcial del conjunto de rasgos, la mayoría de los enfoques requieren formas diferentes de búsqueda.

1.3 Selección de rasgos en la minería de textos

Un escenario donde la selección de rasgos tiene un interés significativamente práctico es la minería de datos, especialmente en la minería de textos, donde el volumen de rasgos considerados para describir los documentos es extremadamente grande y en muchos casos irrelevante y redundante.

Varias áreas dentro de la minería de textos requieren que se realice un proceso de selección de rasgos, ya sea en la recuperación de información (Dixon, 1997 y Frankes, 1992), en la extracción de la información (Dixon, 1997; Frankes, 1992 y Franke, 2003), análisis de textos (Jackson, 2002), resumen (Jackson, 2002 y Berry, 2004), agrupamiento (Berry, 2004), categorización (Jackson, 2002 y Berry, 2004), así como en la clasificación (Jackson, 2002 y Berry, 2004). Como los textos son datos no estructurados, cualquiera de estas ramas

de la minería de textos requiere que se preprocesen los corpus textuales para poder aplicar estas técnicas.

La representación textual es vital al procesar documentos. 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 (Lewis, 1992). Una de las representaciones más utilizadas es la representación espacio vectorial (Vector Space Model (VSM)) (Salton, 1975) en la comunidad de minería de textos (text mining), especialmente en las áreas de recuperación de información, agrupamiento y clasificación. 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 (Joachims, 1997).

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, independientemente del orden en que aparecen, son usadas como términos indexados. Estos pasos requeridos son los siguientes: (i) transformación del corpus, (ii) extracción de términos, (iii) reducción de dimensionalidad y (iv) normalización y pesado de la representación (Lanquillon, 2001).

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, se obtiene 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 (Lanquillon, 2001). 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" (Duda, 1973). En muchos casos, eliminar los rasgos menos informativos puede realmente aumentar la eficiencia del procesamiento a realizar.

Se usa el término reducción de dimensionalidad para abarcar cualquier técnica que su objetivo sea controlar la dimensionalidad del vector. Por tanto, se encuentran incluidas las técnicas de selección de rasgos y técnicas basadas en la reparametrización (Lanquillon, 2001). A continuación se describirán algunas técnicas de selección de rasgos empleadas habitualmente en la representación textual.

Como se mencionó con anterioridad la selección de rasgos que se realiza en la representación textual utiliza un enfoque de filtrado, así se trata cada rasgo independientemente y se evalúa con una puntuación que permite decidir cuando incluirlo o no en el vocabulario. 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.

Algunos ejemplos de técnicas que aplican un enfoque de filtrado para la selección de rasgos en la etapa de representación textual se mencionan a continuación.

Un enfoque lingüístico ampliamente conocido como eliminación de palabras de parada (stop word elimination) (Yang, 1997 y Mladenic, 1998), estas son las palabras que pueden ocurrir en todos los documentos sin ofrecer información alguna sobre el contenido de los mismos (e.g. artículos, preposiciones, conjunciones y pronombres) (Salton, 1983 y Rijsbergen, 1979), por tanto tienen una alta frecuencia de aparición y poco poder discriminante (Sahami, 1998).

Existen 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.

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[1] 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 (Lanquillon, 2001). Así, términos que raramente aparecen en una colección de documentos tendrán poco poder discriminante y pueden ser eliminados (Rijsbergen, 1979). En contraste, términos con frecuencia de aparición alta se asumen que son comunes y que tampoco tienen poder discriminante[2].

En (Sahami, 1998), a partir de estudios de la Ley de Zipf (Zipf, 1949), se concluye que se puede reducir la dimensionalidad de los rasgos originales en un 50 %, si se eliminan los términos que tienen o muy alta o muy baja frecuencia de aparición, a partir de un cálculo adecuado del umbral.

También es posible considerar un umbral de frecuencia en documentos. 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 (Yang, 1997). 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.

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*. Después de eliminar las palabras de parada, la importancia de un término

[1] 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 (Zipf, 1949).
[2] 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.

se incrementa con su frecuencia de uso. Combinando estas ideas se formuló la medida frecuencia del término / frecuencia inversa de documentos (term frequency / inverse document frequency (tfidf)), 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 (Salton, 1988).

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. Así, la entropía puede ser evaluada como la cantidad de información que se espera recibir sobre el promedio cuando se observa una variable aleatoria particular (Salton, 1983). Mientras más uniforme sea una distribución, mayor es su entropía. Así, la entropía alcanza su máximo valor si todos los términos son igualmente probables. Cuando un término *t* está concentrado en sólo pocos documentos, se puede calcular el ruido del término *t*, como la entropía de la distribución de probabilidad del término *t* entre los documentos (Salton, 1983). En (Nürnberger, 2001) se utiliza la entropía, según Lochbaum y Streeter en 1989, como una medida para el cálculo de la importancia de las palabras. Empíricamente se consideran relevantes las palabras que tienen una alta entropía, dentro de aquellas que tienen una alta frecuencia de aparición (i.e. se prefiere seleccionar aquellas palabras que tienen una entropía alta desde un conjunto de palabras que son igualmente frecuentes).

En (Berry, 2004) 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.

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 (Fukuhara, 1999). Valores altos de Skewness(*t*) y Kurtosis(*t*) indican que el término *t* es más general en el corpus de textos y viceversa.

Hasta aquí se han mostrado ejemplos de formas de selección de rasgos en la etapa de representación textual. Sin embargo, existen otras etapas en el procesamiento textual 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 como resultado de un proceso de agrupamiento, no es posible considerar todas las palabras que se obtuvieron en el proceso de reducción de dimensionalidad de la representación VSM, sino que se hace necesario someter cada grupo a un nuevo proceso de reducción de dimensionalidad.

1.4 Algunas técnicas utilizadas en la selección de rasgos

La selección de rasgos en un conjunto de datos es un problema en cuya solución se han utilizado diversas áreas de la Ciencia de la Computación, son ejemplos: la Inteligencia Artificial, la Estadística y el Aprendizaje Automático. A continuación se mencionan algunas técnicas utilizadas.

Conjuntos aproximados: La teoría de los conjuntos aproximados posee un importante potencial para investigar problemas relacionados con bases de casos del mundo real que con frecuencia suelen ser largas y dinámicas. Los conjuntos aproximados ofrecen un marco formal para el descubrimiento de conocimiento (knowledge discovery) es por eso que se pueden utilizar como una herramienta de selección para descubrir dependencias entre datos y reducir el número de rasgos contenidos en un conjunto de datos obteniendo un reducto del conjunto inicial de rasgos con un mínimo de pérdidas de información (Yao, 1999). Diversos autores han propuesto métodos para el cálculo de reductos a través de los conjuntos aproximados, entre ellos se encuentran: (Yao, 1999; Kohavi, 1994; Zhong, 2001 y Dunstsh, 2000), entre otros. La reducción de rasgos a través de los conjuntos aproximados se basa en comparar las relaciones de equivalencia generadas por conjuntos de rasgos. Son eliminados rasgos de manera sucesiva hasta que se obtenga un conjunto reducido tal que provea la misma calidad de la clasificación que el original.

Algoritmos de Optimización de Colonias de Hormigas (OCH): Los (OCH) reproducen el comportamiento de las hormigas reales en una colonia artificial. Estos han sido aplicados a un gran número de problemas cuyas soluciones generan explosión combinatoria como el

clásico del viajero vendedor, problemas de ruteo en redes de telecomunicaciones, planificación de tareas, etcétera. En (Jensen, 2003) se plantea el uso de estas técnicas para la selección de rasgos debido a que las hormigas pueden descubrir las mejores combinaciones de atributos en la medida en que atraviesan el grafo. La selección de rasgos utilizando técnicas de colonias de hormigas consiste en representar cada atributo en un nodo del grafo y los arcos entre ellos denotan la posibilidad de optar por el siguiente atributo. La búsqueda del subconjunto de rasgos adecuado es entonces un recorrido de una hormiga por el grafo donde sea visitada la cantidad mínima de nodos que satisfaga el criterio de parada. Se han reportado otros trabajos de selección de rasgos a través de los algoritmos de Optimización de Colonias de Hormigas, cuyos resultados son muy positivos (Bello, 2005a y Bello, 2005b).

Algoritmos Genéticos: Los Algoritmos Genéticos (Genetic Algorithm (GA)) son métodos de búsqueda y optimización sobre un espacio de soluciones potenciales, basados en el principio de la selección natural y la evolución de poblaciones. El espacio de soluciones potenciales o población es iterativamente refinado para optimizar la medida de puntaje de la población, esta medida se define a través una función de puntaje (fitness function) de los individuos que se interpreta también como la habilidad de sobrevivir en el ambiente de dichos individuos. Los GA han sido utilizados en la selección de rasgos, algunos ejemplos han sido publicados en (Vinterbo, 1999 y Wróblewski, 1995).

Razonamiento probabilístico y teoría de la información. La motivación central para utilizar esta técnica es la observación que el objetivo de un algoritmo de inducción es estimar las distribuciones de probabilidad sobre los valores de las clases. En la misma forma, la selección del subconjunto de rasgos debe estar dirigida a encontrar aquel subconjunto lo más cercano posible a las distribuciones originales. El algoritmo realiza una búsqueda de eliminación hacia atrás (backward elimination search), en cada etapa elimina el rasgo que causa el menor cambio entre las distribuciones. La búsqueda para cuando se ha obtenido el número de rasgos deseado.

Redes Bayesianas. Las redes Bayesianas codifican las relaciones contenidas en los datos modelados. Pueden ser usadas para describir los datos así como para generar nuevas

instancias de las variables con propiedades similares a las que presentan los datos dados. Estas redes pueden ser usadas en la selección de rasgos, un ejemplo se presenta en (Inza, 2001), con el método Feature Subset Selection by Estimation of Bayesian Network Algorithm (FSS-EBNA) que está basado en la modelación probabilística por redes Bayesianas y brinda soluciones (subconjuntos de rasgos) para guiar exploraciones futura en el espacio de rasgos. Estas redes permiten obtener la próxima generación de subconjuntos de rasgos, garantizando la evolución de la búsqueda para un subconjunto optimal de rasgos. Utilizan un enfoque wrapper para evaluar la calidad de cada subconjunto de rasgos candidato.

Árboles de decisión. Los algoritmos para la inducción de árboles de decisión, pueden utilizarse como métodos se selección de rasgos, porque el proceso de inducción incluye la selección de los atributos que serán incorporados en el árbol de acuerdo a su información teórica en comparación con los demás atributos, como por ejemplo la ganancia de la información (Quinlan, 1993). La selección de atributos con la información más importante provoca que no todos los atributos sean necesarios en la solución del problema. Las medidas de selección pueden variar, algunos criterios para dividir las instancias de cada nodo se presentan en (Ming, 2001): la entropía y sus variantes, el estadístico Chi cuadrado, el estadístico G y el índice de diversidad GINI.

1.5 Árboles de decisión en la selección de rasgos

El aprendizaje por Árboles de Decisión (Decision Trees (DT)) es un método para aproximar funciones de valores discretos. Los DT pueden ser representados también como conjuntos de reglas **If - Then**. Un DT clasifica las instancias ordenándolas top–down (de la raíz a las hojas). Cada nodo interior del árbol de decisión especifica la prueba de algún atributo y las hojas son las clases en las cuales se clasifican las instancias. Cada rama descendiente de un nodo interior corresponde a un valor posible del atributo probado en ese nodo. Un DT representa una disyunción de conjunciones sobre los valores de los atributos. Cada rama de la raíz a un nodo hoja corresponde a una conjunción de atributos y el árbol da así una disyunción de estas conjunciones (Mitchell, 1997).

La inducción de los árboles de decisión es una aproximación muy popular en el análisis de datos para generar modelos de clasificación o regresión. Los árboles de decisión están basados en algoritmos de aprendizaje discriminativo por medio de particionamiento recursivo. El espacio de datos es particionado aplicando un método dirigido por dato (backward) y la partición es representada como un árbol. El árbol de decisión puede ser transformado en una base de reglas donde cada camino desde la raíz a un nodo hoja es una regla.

Los algoritmos para la inducción de árboles de decisión son al mismo tiempo una manera de optimizar la representación de los árboles y crear árboles tan pequeños como se pueda. Esto es dado por la selección de los atributos que serán incluidos en el árbol de acuerdo a su información teórica en comparación con los demás atributos, como por ejemplo la ganancia de la información (Quinlan, 1993). La selección de atributos con la información más importante provoca que no todos los atributos sean necesarios en la solución del problema. Es por esto que los árboles de decisión se pueden utilizar en la selección de rasgos. Las medidas de selección pueden variar, algunos criterios para dividir las instancias de cada nodo se presentan en (Ming, 2001): la Entropía y sus variantes, el estadístico Chi cuadrado, el estadístico G y el índice de diversidad GINI.

El método de selección es una heurística y no garantiza que el árbol sea optimal. La construcción de un árbol de decisión representa una estrategia de búsqueda greedy que implementa una decisión óptima local para cada nodo. La dificultad es que una combinación de decisiones locales óptimas puede no garantizar el óptimo global para el árbol, i.e., un árbol con tamaño más pequeño. Por cierto esto es un problema NP-Duro para encontrar el árbol más pequeño o uno con el número mínimo de niveles (Wang, 2000).

La inducción de DT proporciona uno de las más populares metodologías para la adquisición de conocimiento simbólico. El resultado es, un árbol simbólico de decisión junto con un mecanismo de inferencia simple, el cual se ha elogiado por su comprensibilidad (Janikow, 1996).

La característica más importante de los DT es su capacidad de analizar un proceso de toma de decisiones complejo en una colección de decisiones más simples, proporcionando una

solución fácilmente interpretable (Sushmita, 2002). Los árboles de decisión son las herramientas no paramétricas más comúnmente usadas para la clasificación de patrones, además, tiene un gran uso en la selección de rasgos.

Un algoritmo de aprendizaje de un árbol de decisión, tiene dos componentes principales: la construcción del árbol y la inferencia. La construcción del árbol está basada en el particionamiento recursivo, y usualmente se supone la independencia de todos los atributos. La rutina de partición recursiva selecciona un atributo a la vez, usualmente el primero con máxima medida de la información para los ejemplos de entrenamiento en el nodo. Este atributo es usado para dividir el nodo, usando los valores del dominio del atributo para formar condiciones adicionales dominantes al subárbol. Entonces el mismo procedimiento es recursivamente repetido para los hijos del nodo. Este procedimiento se repite hasta que se cumpla el criterio de parada definido.

Según (Marsala, 1998) y (Zeidler, 1996), algunos de los criterios de parada disponibles en la literatura consideran crear un nodo hoja en los casos siguientes:

- Todos los ejemplos pertenecen a la misma clase.
- La proporción del conjunto de ejemplos de una clase es mayor o igual que un umbral dado.
- No hay más atributos o rasgos para las clasificaciones.
- El calculo de la medida de la información del atributo es menor que un umbral especificado por el usuario o calculado por el propio algoritmo, los cual evita escoger atributos con muy bajo valor de la información (Ming, 2001).

1.5.1 Árboles de decisión borrosos

Los árboles de decisión clásicos (duros) se aplican extensamente a las tareas de clasificación y selección de rasgos. Sin embargo, hay muchos tipos de problemas, especialmente en el caso de atributos numéricos (valores continuos) donde es factible encontrar una solución usando árboles de decisión borrosos (Zeidler, 1996).

En la inducción de los árboles de decisión continuos, el punto dominante es generar una discretización apropiada de los atributos en un nodo. Los métodos usados en la inducción

del árbol clásico son sensibles al ruido, y así, propenso a errores en la clasificación. La robustez y capacidad de generalización son las aplicaciones principales de los árboles de decisión continuos. Una "discretización suave" podría tener la capacidad potencial para mejorar la robustez de la clasificación y para realzar la generalización del clasificador inducido.

A través de los años, se han investigado y se han propuesto metodologías para tratar datos continuos o multi-evaluados, así como la introducción de ruidos. Recientemente, con el renombre cada vez mayor de la representación borrosa, algunos investigadores han propuesto utilizar la representación borrosa en los árboles decisión para ocuparse de situaciones similares. La representación borrosa es un puente entre datos simbólicos y no simbólicos, enlazando términos lingüísticos cualitativos a datos cuantitativos.

Según (Umano, 1994) los árboles de decisión borrosos (Fuzzy Decision Trees (FDT)) difieren de los DT tradicionales en los siguientes aspectos:

- Hay un grado de pertenencia para todos los ejemplos de entrada a cada clase.
- Más de una clase puede asignarse a un nodo hoja.
- Cada atributo es considerado como una variable lingüística.

Esta fusión permite combinar la incertidumbre y el razonamiento aproximado de los conjuntos borrosos con la capacidad de los árboles de decisión de comprensibilidad y facilidad de uso. Esto realza el poder representativo de los árboles de decisión, naturalmente, con el componente de conocimiento inherente en la lógica borrosa, ventaja para mejorar la robustez, inmunidad ante el ruido, y aplicabilidad en contextos inciertos o imprecisos.

Muchos criterios son expuestos para la construcción de los árboles de decisión borrosos. En estos, los atributos continuos necesitan ser repartidos en varios sistemas borrosos antes de la inducción del árbol, heurísticamente basados en las experiencias de expertos y las características de los datos, así como construidos automáticamente (Peng, 2001).

El proceso de construcción de árboles de decisión borrosos es basado en el conocimiento de una partición borrosa para cada atributo numérico. Sin embargo, puede ser difícil obtener

una partición borrosa dado un atributo numérico. Por tanto, no es una tarea trivial la obtención de tales particiones y existen varios métodos que permiten la construcción automática de las mismas. Antes de realizar un estudio de dichos métodos, se definirán elementos esenciales del proceso de discretización y de la lógica borrosa.

1.5.1.1 Discretización

En algunos casos, teniendo en cuenta que la mayoría de los problemas de la vida real son con datos no simbólicos (numérico, continuo), los rasgos que describen el problema deben ser discretizados antes de la selección. Aunque también existen árboles de clasificación y de regresión los cuales no necesitan la discretización.

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

La discretización es un caso específico del agrupamiento. El agrupamiento consiste, en esencia, en hallar la estructura interna de un conjunto de descripciones de objetos en el espacio de representación. Esta estructura interna obviamente depende en una primera instancia, de la selección del propio espacio de representación y de la forma en que los objetos se comparen, es decir, del concepto de similitud que se utilice y de la forma en que éste se emplee (Ruiz-Shulcloper, 1995).

Existen varios métodos de discretización, dos de los clásicos y más sencillos son la discretización por intervalos de igual tamaño y la discretización considerando igual frecuencia por intervalo (Arco, 2003). Estos métodos son muy usados por su sencillez y rapidez. Sin embargo, en la actualidad se han desarrollados métodos que logran mejores resultados en la discretización, por ejemplo, aquellos que tienen en cuenta la distribución real de los datos en la base de información. Algunos ejemplos se mencionan a continuación. El método Chi^2 que tiene como objetivo discretizar atributos numéricos basados en el estadístico χ^2, y además permite eliminar los atributos redundantes y chequear inconsistencias (Liu, 1997). El algoritmo CAIM, es otro ejemplo que discretiza un

atributo en el menor número posible de intervalos y maximiza la interdependencia entre el atributo y el rasgo objetivo (clase). Es el propio algoritmo quien selecciona de manera automática (al igual que el Chi^2) el número de intervalos discretos en los que quedará finalmente particionado el atributo (Kurgan, 2004). Los discretizadores Chi^2 y CAIM son eficientes pero lamentablemente sólo pueden emplearse en problemas de clasificación donde el rasgo objetivo pueda tomar un solo valor a la vez. No pasa así con la discretización por intervalos de igual tamaño, la discretización considerando igual frecuencia por intervalo ni con el algoritmo de agrupamiento *K-means* (o su variante *C-means*), quien sí implementa una heurística mucho más compleja (Jyh, 1998).

1.5.1.2 Principales definiciones de la lógica borrosa

La teoría de los conjuntos borrosos fue inicialmente propuesta por Zadeh en 1965 (Zadeh, 1965). La teoría de conjuntos borrosos es primeramente concebida para la cuantificación y razonamiento usando lenguaje natural en los cuales muchas palabras tienen significados ambiguos. Esta teoría puede verse también como una extensión de los conjuntos duros tradicionales, en los cuales cada elemento pertenece o no a un conjunto.

Formalmente, el proceso por el cual se determina si valores individuales de un conjunto universal X es miembro o no de un conjunto duro, puede estar definido por una función característica o de discriminación (Klirl, 1992). Para un conjunto duro A, esta función asigna un valor $\mu_A(x)$ para cada $x \in X$ tal que:

$$\mu_A(x) = \begin{cases} 1 & si \quad x \in A \\ 0 & si \quad x \notin A \end{cases} \tag{1.1}$$

Así, los elementos de la función van del conjunto universal al conjunto que sólo tiene los elementos 0 y 1. Esto se puede indicar por:

$$\mu_A(x): X \rightarrow \{0,1\} \tag{1.2}$$

Este tipo de función puede ser generalizada de forma tal que el valor asignado a los elementos del conjunto universal se encuentra dentro de un rango específico y son

referenciados como el grado de pertenencia de esos elementos al conjunto. Grandes valores denotan mayores grados de pertenencia al conjunto. Tal función es llamada función de pertenencia μ_A, para la cual, un conjunto borroso A es usualmente definido. Ver fórmula 1.3

$$\mu_A(x):X\rightarrow[0,1] \tag{1.3}$$

donde [0,1] denota el intervalo de números reales desde 0 a 1, incluyendo a ambos extremos(Buckley, 2002).

Los conjuntos borrosos son funciones que acotan un valor que puede ser un miembro del conjunto a un número entre cero y uno, indicando su actual grado de pertenencia. Un grado cero significa que ese valor no está en el conjunto, y un grado de uno significa que el valor es completamente representativo del conjunto. Esto produce una curva a través de los miembros del conjunto.

El centro de las técnicas de modelación borrosa es la idea de una variable lingüística. En su origen, una variable lingüística es el nombre de un conjunto borroso. Pero una variable lingüística también lleva consigo el concepto de calificadores de conjuntos borrosos. Una variable lingüística encapsula las propiedades de aproximación o conceptos imprecisos en una forma sistemática y conveniente computacionalmente. Estas reducen la aparente complejidad de describir un sistema por la correspondencia a una etiqueta semántica para el concepto fundamental.

Una variable lingüística se caracteriza por un quíntuplo $(x,T(x),X,G,M)$ en el cual x es el nombre de la variable, $T(x)$ es el conjunto de términos, o sea, el conjunto de sus valores o términos lingüísticos, X es el universo de discurso, G es la regla sintáctica la cual genera los términos en $T(x)$, y M es una regla semántica la cual asocia a cada valor lingüístico A su significado $M(A)$, donde $M(A)$ denota un conjunto borroso en X (Buckley, 2002).

Una definición formal de conjuntos borrosos y funciones de pertenencia se presenta a continuación (Buckley, 2002):

Si X es una colección de objetos denotados genéricamente por x, entonces un conjunto borroso A en X se define como un conjunto de pares ordenados:

$$A = \{(x, \varphi_A(x)) | x \in X\} \tag{1.4}$$

donde $\varphi_A(x)$ es llamada la función de pertenencia (FP) para el conjunto A. La función de pertenencia asigna a cada elemento de X un grado de pertenencia en el intervalo [0,1]. A X se le llama universo de discurso y puede ser un espacio discreto o continuo. El conjunto de pares ordenados puede ser también denotado como $\varphi_A(x_1)/x_1 + \varphi_A(x_2)/x_2 + \ldots + \varphi_A(x_n)/x_n$.

Un conjunto borroso consiste de tres componentes: un eje horizontal con el dominio de números reales monótonamente crecientes que constituye la población del conjunto borroso, un eje vertical con la pertenencia entre cero y uno indicando el grado de pertenencia en el conjunto borroso, y la superficie del conjunto borroso por sí misma que conecta un elemento en el dominio con el grado de pertenencia en el conjunto.

El α-corte o conjunto de nivel α de un conjunto borroso A es un conjunto duro definido por:

$$A_\alpha = \{x | \varphi_A(x) \geq \alpha\} \tag{1.5}$$

El α-corte fuerte se define similarmente:

$$A'_\alpha = \{x | \varphi_A(x) > \alpha\} \tag{1.6}$$

1.5.1.3 Funciones de pertenencia principales

Los números borrosos son una clase importante de contornos borrosos que representan aproximaciones de un valor central y se visualizan gráficamente como una clase de curvas "campana". En general, existen tres clases importantes de curvas "campana" – los conjuntos borrosos PI, Beta y Gaussianos. La diferencia entre los tres tipos de curvas está dada por la pendiente de la curva así como por los valores de los puntos finales de la curva (Zadeh, 1994).

El espacio borroso puede ser definido en el caso de las curvas PI, o ser infinito, en el caso de las curvas Beta y Gaussianas. La anchura y la pendiente de las curvas campanas indican el grado de compactación asociado con el número borroso.

Las curvas PI. Una curva PI es la preferida y se toma como representación de números borrosos. Esta provee un gradiente descendiente suave desde el valor central hasta los puntos de pertenencia cero a lo largo del dominio. La curva PI simétrica es centrada en un único valor del dominio (x) con un único parámetro que indica el ancho de la base de la curva (β). El valor de la curva para los puntos x del dominio esta dada por la expresión 1.7,

$$\prod(x;\beta;\gamma) = \begin{bmatrix} S(x;\gamma-\beta,\gamma-\beta/2,\gamma) & \longrightarrow & x \le \gamma \\ 1-S(x;\lambda;\gamma+\beta/2,\gamma+\beta) & \longrightarrow & x > \gamma \end{bmatrix} \tag{1.7}$$

Los puntos de inflexión son automáticamente determinados. Las curvas PI tienen una característica importante: su valor de pertenencia se hace cero en un punto discreto y específico, y no son asintóticas (Buckley, 2002).

Las curvas Beta. La curva Beta es una curva con forma de campana más estrecha que la curva PI. Los conjuntos borrosos Beta son definidos, como la curva PI, con dos parámetros: el único valor del dominio alrededor del cual la curva es construida (γ) y un valor que indica la mitad del ancho de la curva en el punto de inflexión (β). Los valores de la curva para los puntos x del dominio están dados por la expresión 1.8,

$$B(x;\gamma,\beta) = \frac{1}{1+\left(\frac{x-\gamma}{\beta}\right)^2} \tag{1.8}$$

La curva producida desde esta fórmula se parece a la curva PI con una principal diferencia, el grado de la FP va hasta cero solo en los valores extremadamente grandes de Beta (β); esto es, a infinito. La función de la curva Beta es mucho más directa que el generador de la curva PI. Se encuentra un espacio del dominio y entonces para cada punto a lo largo de la curva Beta después de calcular un valor del dominio en la i-ésima posición del arreglo de pertenencias y localizar su distancia del centro de la curva, se genera un grado de

pertenencia resolviéndose la función logística Beta. La figura 1.1 representa la representación gráfica de la curva Beta (Buckley, 2002).

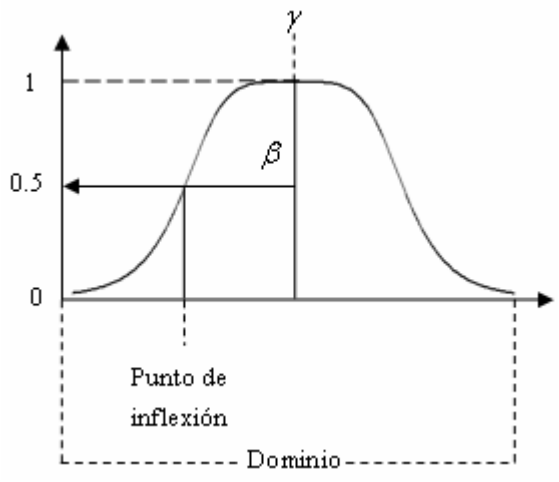


Figura 1.1 Gráfico de la curva campana Beta.

Las curvas Gaussianas. La curva Gaussiana o exponencial es un método menos popular de representar números borrosos. La curva Gaussiana es definida, como la curva PI, con dos parámetros: el valor único del dominio alrededor del cual la curva es construida (γ) y un valor que indica la anchura de la forma de la campana de la curva (k). El valor de la curva para un valor x del dominio, está dado por la expresión 1.9,

$$G(x;k,\gamma) = \ell^{-k(\gamma-x)^2} \tag{1.9}$$

La curva producida desde su fórmula se parece a la curva Beta con una diferencia – la pendiente de la FP va hasta cero muy rápidamente. Hay que tener en cuenta que la curva Gaussiana tiene alguna de las propiedades de las curvas PI. Naturalmente el parámetro de anchura (k) juega un role crítico en la forma y alcance del conjunto borroso. El mayor de los valores para k, hace la curva más ancha y viceversa. La inhabilidad de predecir exactamente la forma completa de la curva de campana resultante para un valor particular de K hace que se dificulte el uso de la función Gaussiana (Buckley, 2002).

Funciones de pertenencia triangulares. Existen algunos campos donde los conjuntos borrosos se representan con funciones triangulares. Los conjuntos borrosos triangulares,

semejantes a las curvas PI y Beta, representan valores de pertenencia. Su centro es la punta del triángulo, ahí está la máxima pertenencia. Las aristas izquierda y derecha de la región borrosa especifican un descenso lineal desde el centro hasta los puntos donde la pertenencia es cero. Obsérvese la figura 1.2:

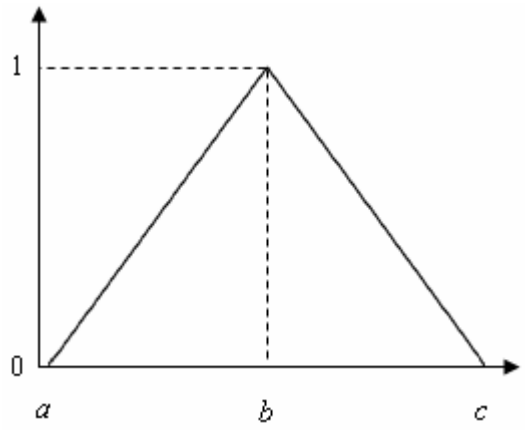


Figura 1.2 Gráfico de la función triangular.

La expresión de esta función está dada por ecuación 1.10:

$$Triángulo(x,a,b,c) = \begin{cases} 0 & x \le a \\ \dfrac{x-a}{b-a} & a \le x \le b \\ \dfrac{c-x}{c-b} & b \le x \le c \\ 0 & c \le x \end{cases} \tag{1.10}$$

donde a, b y c son los vértices del triángulo ($a < b < c$) (Buckley, 2002).

Funciones de pertenencia trapezoidales. Finalmente se define la función trapezoidal con los parámetros (a_2, b_2, b_3, c_3). La FP trapezoidal es especificada por cuatro parámetros {a, b, c, d}, donde $a < b < c < d$. Note que esta función se reduce a la función triangular cuando b es igual a c (Buckley, 2002). Observe la expresión 1.11:

$$Trapecio(xa,b,c,d) = \begin{cases} 0 & x \le a \\ \dfrac{x-a}{b-a} & a \le x \le b \\ 1 & b \le x \le c \\ \dfrac{d-x}{d-c} & c \le x \le d \\ 0 & d \le x \end{cases} \tag{1.11}$$

1.5.1.4 Construcción automática de funciones de pertenencia

Algunos autores utilizan el término discretización borrosa (Fuzzy Discretization) para referirse a la construcción de funciones de pertenencia. Según (Zeidler, 1996), *e*l principio de discretización es simplemente una condición lógica (usando uno o más atributos) que sirven para dividir los datos en por lo menos dos subconjuntos. La mayor interrogante sería en qué parte del conjunto se pondrán los puntos de corte en la partición de los valores de los atributos continuos. De ahí que ha sido necesario el desarrollo de métodos que permiten la construcción de funciones de pertenencia.

Los métodos más comunes empleados en la construcción de funciones de pertenencia son:

Evaluación subjetiva y construcción a partir de expertos. Los conjuntos borrosos pueden determinarse a partir de procedimientos simples o complejos de extracción, dado que usualmente modelan el estado cognoscitivo de las personas. Los expertos en el dominio de aplicación simplemente dibujan o especifican diferentes curvas de pertenencia de las cuales eligen una. En algunos casos, la elección puede estar determinada mediante métodos que tienen su base en la psicología.

Frecuencias convertidas o probabilidades. Algunas veces, la información tomada a partir de histogramas de frecuencias u otras curvas de probabilidad se emplea como base para construir la función de pertenencia. Existe una gran variedad de métodos de conversión posibles y cada uno posee sus propias fortalezas y debilidades, tanto matemáticas como metodológicas. Sin embargo, es necesario recordar que las funciones de pertenencia no son necesariamente probabilidades.

Medición física. Muchas aplicaciones de lógica borrosa usan la medición física, pero casi ninguna mide el grado de pertenencia directamente. En su lugar, la función de pertenencia se obtiene mediante otro método y los grados de pertenencia individuales se calculan a partir de ella.

Aprendizaje y adaptación. La aplicación de las técnicas de aprendizaje automatizado posibilita construir de forma automática a partir de datos numéricos y simbólicos las funciones de pertenencia, generalmente las funciones de pertenencia construidas por esta vía son representadas como modelos matemáticos. Con frecuencia, una vez construidas, se les ajusta para mejorar su efectividad por medio de técnicas que posibilitan su adaptación.

Usualmente, la FP asociada a un conjunto borroso se define de acuerdo al conocimiento experto o simplemente se asigna por el desarrollador del sistema computacional que la implemente. En estos casos, se construye de forma manual, aunque la tendencia en los últimos tiempos ha sido prescindir del conocimiento experto, que puede ser limitado o sujeto a fallos. Novedosas técnicas capaces de proponer uno o varios tipos de funciones de pertenencia a partir de los ejemplos de la base de conocimiento, e inclusive estimar con calidad los parámetros de estas, han aflorado en la etapa más reciente.

El método de interpolación permite la construcción automática de FP. Como premisa para la aplicación de este método es necesario conocer la pertenencia para un conjunto finito de puntos, información que podría ser suministrada por un experto. Luego por alguna forma de interpolación podría determinarse la pertenencia de un elemento no suministrado previamente por el experto (Chen, 1995). Dadas las características de las FP aplicando métodos de interpolación basados en los mínimos cuadrados y los spline no siempre se logra construir buenas funciones de pertenencia, de ahí que se piense en una interpolación más avanzada que preserve la monotonía local y la convexidad. En (Chen, 1995) se propone realizar la interpolación utilizando polinomios de Bernstein.

Narazaki y Ralescu (Narazaki, 1994) proponen un método basado en la determinación de los centros de gravedad de los términos lingüísticos de una variable lingüística. En estos centros de gravedad la FP asociada alcanza el máximo valor de pertenencia.

En (Hong, 1998) se propone un método de aprendizaje para derivar automáticamente reglas borrosas y FP desde un conjunto dado de casos de entrenamiento facilitando la adquisición de conocimiento. Las principales desventajas de este algoritmo son que requiere que los valores del rasgo objetivo sean ordenables, solo admite datos numéricos y permite construir FP triangulares solamente. Además, la construcción de estas FP implica la obtención de unas FP iniciales donde se van a obtener muchas regiones iniciales. Si la diferencia entre dos valores adyacentes del conjunto de entrenamiento es muy pequeña, entonces la cantidad de regiones, que a su vez es la cantidad de FP iniciales, es muy grande, lo que hace a este proceso complejo. Como ventaja fundamental se señala que este algoritmo construye de forma automática las FP a partir de los casos.

Los Algoritmos Genéticos (AG) también han sido utilizados en la construcción automática de FP. Al usar esta técnica el problema de determinar los parámetros que describen cada función se transforma en un problema de optimización donde se pretende minimizar el error que se produce con la selección de diferentes valores de los parámetros. Los cromosomas son arreglos lineales donde en cada escaque hay un parámetro de la función en cuestión, la función de evaluación es una función de error que depende de los parámetros que describen a la función de pertenencia y que están representados en el cromosoma; el criterio de parada por su parte es generalmente determinado por el número de generaciones (Piñero, 2005). Como principal desventaja del uso de los AG se señala que no siempre convergen a un óptimo global sino a un elemento casi óptimo.

En ocasiones las funciones de pertenencia que se quieren obtener son funciones con propiedades deseables para el análisis, por ejemplo continuas, derivables, etc. En estos casos generalmente es factible aplicar métodos numéricos clásicos o simples análisis matemáticos que permitan determinar los parámetros de las funciones de pertenencia. La aplicación de estos métodos depende de las características de las funciones de pertenencia y de la naturaleza del problema en cuestión (Arco, 2001).

Otro método es el descrito en (Zhou, 1997), este hace uso de técnicas estadísticas para construir las FP que posteriormente se usarán en una red neuronal borrosa. Este enfoque se encarga de discretizar por el método Chi2 (Liu, 1997) el conjunto de valores continuos de

un cierto atributo, determinando los intervalos obtenidos la cantidad de términos lingüísticos y la longitud de cada FP. Posteriormente se construyen funciones trapezoidales para provocar que cada valor de entrada pertenezca como máximo a dos FP.

En (Botzheim, 2001) se presenta un método que usa Algoritmos Bacterianos para extraer las reglas de un sistema borroso. La clase de FP utilizadas se restringe a la trapezoidal, pues es bastante general y ampliamente utilizada. El algoritmo contiene el paso bacteriano de la mutación, permitiendo el cambio de más de una función de pertenencia a la vez, y el ajuste de parámetros.

En (Marsala, 1996) se presenta un algoritmo para inferir una partición borrosa sobre un conjunto de valores numéricos. Este algoritmo es basado en morfología matemática y es expresado en teoría de lenguaje formal. Además, es usado cuando los valores numéricos están asociados a clases.

En (Varela, 2005) se proponen varias heurísticas para la estimación de los parámetros de las FP de tipo Triangular, Trapezoidal, Gaussiana y Sigmoidal (Buckley, 2002). En todos los casos se aplica una generalización de la heurística propuesta por Hong (Hong, 1998).

1.5.2 Ejemplos de métodos de inducción de árboles de decisión

Un clásico en la inducción de árboles de decisión es el algoritmo ID3 para valores discretos (Quinlan, 1993 y Mitchell, 1997) y su extensión C4.5 para valores continuos (Quinlan, 1996). Anterior al ID3 surgió el CART en 1984 que crea árboles de decisión para clasificación y regresión (Quinlan, 1993). Un problema de estos algoritmos es que no pueden proveer ninguna información de las regiones de intersección cuando las clases son solapadas. Una extensión del ID3 es el llamado GID3 (Wang, 1998). Este construye un árbol de decisión, pero sobre la base de atributos continuos y maneja valores borrosos en lugar de duros. Es importante destacar que en el algoritmo GID3 la generación del árbol de decisión no depende de la selección de las funciones de pertenencia.

En (Sushmita, 2002) proponen una variante borrosa del algoritmo ID3 para la inducción de FDT y obtienen los términos lingüísticos usando cuantiles. Otra variante del algoritmo ID3 es presentada en (Ming, 2001) donde se combinan el método que proponen (look-ahead) y

la ganancia de la información en la selección del atributo ganador. En (Marsala, 1999) se introduce un modelo para estudiar medidas de discriminación usadas en la inducción de árboles de decisión, específicamente se presenta el sistema Salammbo para construir árboles de decisión borrosos con varios tipos de medidas de discriminación. Finalmente, en (Wang, 2003) se propone una variante borrosa del algoritmo ID3 a partir del aporte de nuevas medidas de entropía y ganancia de la información considerando los grados de pertenencia de cada valor a los términos correspondientes a las variables lingüísticas asociadas a cada atributo.

Este último método referenciado es el que se utilizará en la selección de los términos que caracterizan los grupos homogéneos de documentos afines, cuando se ha agrupado utilizando métodos que aplican técnica borrosa.

1.6 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:

- Al referirse a la selección de rasgos, implícitamente se habla de selección de rasgos relevantes para el proceso de aprendizaje automático que se llevará a cabo. Sin embargo, existen varias definiciones de relevancia y ellas son o no apropiadas dependiendo del objetivo de la selección. La inducción de árboles de decisión como una forma de selección de rasgos relevantes aplica los conceptos de relevancia respecto a una distribución y de utilidad incremental.
- Si bien existen tres formas fundamentales dentro de la selección de rasgos como una búsqueda heurística, métodos embebidos, de filtrado y wrapper, se reporta en la literatura que los métodos de filtrado son los más utilizados en la selección de rasgos como parte de la aplicación de técnicas de reducción de dimensionalidad en la etapa de representación textual al preprocesar documentos. Sin embargo, no sólo es necesaria la selección de rasgos en la etapa de representación de los documentos, este proceso es útil en otras etapas de la minería de textos, por ejemplo en la desambiguación de términos o en la obtención de palabras claves en grupos homogéneos de documentos. Se utilizará

en este trabajo un método embebido para la extracción de palabras claves que caracterizan grupos de documentos afines.

- Existen diversas técnicas de la Inteligencia Artificial, Estadística y Aprendizaje Automático que han permitido el desarrollo de métodos de selección de rasgos. En este trabajo se ha seleccionado la inducción de árboles de decisión borrosos para extraer las palabras claves que caracterizan y logran discriminar entre grupos homogéneos de documentos. Esta selección se ha realizado porque los DT tienen la capacidad de ir seleccionando los rasgos relevantes y que logran discriminar entre clases en el propio proceso de inducción. Se han seleccionado los árboles borrosos porque estos permiten el trabajo con atributos continuos y cuando existe solapamiento entre clases. Además, si las palabras claves a seleccionar se encuentran en grupos de documentos provenientes de un método de agrupamiento que utilizó una técnica borrosa, se hace imprescindible el uso de árboles borrosos.
- En la inducción de árboles de decisión borrosos se hace necesaria la estimación de los parámetros de las funciones de pertenencia asociadas a las variables lingüísticas correspondientes a cada rasgo del problema. El análisis del estado del arte ha permitido conocer que existen diferentes formas de estimar estos parámetros y diversas funciones de pertenencia a aplicar. Dos comúnmente utilizadas son las campanas Beta y las triangulares, que son las que se han seleccionado, estimándolas con métodos del análisis matemático por ser sencillos y fáciles de aplicar y reutilizar.
- En la literatura se reportan varias formas de inducción de árboles de decisión borrosos, se ha seleccionado la variante propuesta en (Wang, 2003) por el buen funcionamiento de las nuevas medidas de entropía y ganancia de la información considerando los grados de pertenencia de cada valor a los términos correspondientes a las variables lingüísticas asociadas a cada atributo.
- Si bien existen numerosas técnicas y algoritmos que permiten la selección de rasgos, y específicamente la selección de rasgos en la etapa de representación textual, se han desarrollado en menor medida métodos que permitan la selección de rasgos en otras etapas del procesamiento de documentos, particularmente en la extracción de palabras

claves que logran caracterizar grupos homogéneos de documentos. Lo anterior constituye un problema aún no resuelto y evidencia que la falta de nuevos métodos en esta área limitan el desarrollo de investigaciones en este campo, por ejemplo, en el resumen automático de documentos.

2

MODELO PARA LA SELECCIÓN DE PALABRAS CLAVES EN GRUPOS TEXTUALES HOMOGÉNEOS

Capítulo 2. MODELO PARA LA SELECCIÓN DE PALABRAS CLAVES EN GRUPOS TEXTUALES HOMOGÉNEOS

A partir de la revisión bibliográfica realizada acerca de las técnicas de selección de rasgos y su aplicación en la minería de textos, especialmente el uso de los árboles de decisión borrosos como una forma de selección de rasgos, en este capítulo se propone un modelo que permite realizar la selección de palabras claves en grupos textuales homogéneos. También se muestra la implementación del procedimiento general que sustenta este modelo y su incorporación al sistema CorpusMiner.

2.1 Modelo conceptual propuesto que permite la selección de palabras claves en grupos textuales

En esta investigación se propone un modelo que explica el problema científico formulado. Obsérvese el anexo 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 la selección de palabras claves que logren caracterizar grupos homogéneos de documentos afines y a la vez logre discernir entre los grupos.

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.

La premisa fundamental del modelo es que para su aplicación se hace necesario el resultado de algoritmos de agrupamiento que utilicen técnicas de agrupamiento borrosas y que obtengan la relevancia de las palabras en el propio proceso de agrupamiento para poder utilizar el modelo en toda su dimensión.

La entrada al modelo es el resultado del agrupamiento de documentos, donde las clases a las cuales corresponde cada documento son los grupos resultantes del proceso de agrupamiento y la salida principal son las palabras claves que caracterizan y logran discernir entre los grupos homogéneos de documentos. Dos salidas secundarias, pero también de gran utilidad son: el árbol de decisión y las reglas de inducción.

En este modelo, a partir de los resultados del agrupamiento, se requiere seleccionar aquellos términos que son relevantes y caracterizan cada grupo obtenido, útil, por ejemplo, en la obtención de resúmenes extractos de los múltiples documentos que componen un grupo homogéneos. En el modelo se han considerado tres variantes de selección de los términos relevantes de cada grupo: seleccionar los términos más relevantes a partir de los resultados del agrupamiento (esto sólo es posible para los algoritmos de agrupamiento que devuelven la relevancia de los términos a cada cluster obtenido; e.g. SKWIC y *Fuzzy* SKWIC), seleccionar los términos que logran discernir entre clusters a partir de la aplicación del algoritmo ID3, variantes dura (Mitchell, 1997 y Valdés, 2005) y borrosa (Wang, 2003), y a partir de la intersección de los dos resultados anteriores (Valdés, 2005 y Arco, 2005).

Como se hizo alusión con anterioridad, al modelo se le definen, además, procedimientos que posibilitan su implementación.

2.2 Procedimiento general para extraer las palabras claves en grupos textuales

Como parte del modelo conceptual se desarrolla un procedimiento 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 el anexo 2):

1. Discretización de los rasgos que describen los documentos.
2. Construcción de las variables lingüísticas asociadas a cada término.
3. Aplicación de los algoritmos ID3 duro o ID3 borroso.
4. Extracción de palabras claves de grupos textuales homogéneos.

Al describir las etapas del procedimiento general para seleccionar las palabras claves, se enfatizará en las técnicas empleadas cuando la entrada al procedimiento es el resultado de un agrupamiento que empleó una técnicas borrosa, por tanto, se hará énfasis especial en la inducción del árbol de decisión borroso según el algoritmo *Fuzzy* ID3 (Wang, 2003).

2.2.1 Entrada al procedimiento general

Como ya se ha mencionado, la entrada al modelo es el resultado del agrupamiento de documentos, donde las clases a las cuales corresponde cada documento son los grupos resultantes de este proceso. Se consideran salidas de métodos de agrupamiento que contemplen alguna de las tres técnicas siguientes: duras y deterministas, borrosas, y duras y con solapamiento (Höppner, 1999).

Este procedimiento general es incorporado a CorpusMiner (Valdés, 2005) (Mederos, 2005), sistema que parte 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, y agrupa los documentos siguiendo alguna de las variantes siguientes: algoritmo *Simultaneous Keyword Identification and Clustering of Text Documents* (SKWIC) (Berry, 2004), algoritmo *Simultaneous Keyword Identification and Fuzzy Clustering of Text Documents* (Fuzzy SKWIC) (Berry, 2004), y algoritmo *Extended Star* (Gil-García, 2003), o las variantes concatenadas *Extended Star* – SKWIC y *Extended Star – Fuzzy* SKWIC (Arco, 2005).

2.2.2 Etapa 1: Discretización de los rasgos que describen los documentos

Los valores discretos tienen importancia en el descubrimiento de conocimiento desde datos (Hussain, 1999). 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, un ejemplo lo constituye el algoritmo ID3.

El procedimiento general propuesto considera en la tercera etapa la aplicación del algoritmo ID3, ya sea en su variante dura o borrosa. En el primer caso se requieren que los rasgos que describen el problema estén discretizados, en el segundo se hace necesaria la construcción de las variables lingüísticas asociadas a los términos que describen los documentos agrupados. Es por eso, que la etapa 1 es importante, tanto si se comienza el procesamiento a partir del resultado de un agrupamiento duro o borroso.

El procedimiento propuesto considera en esta primera etapa una discretización por amplitud, es decir, considerando 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 2.1:

$$Amplitud = \frac{\max - \min}{K} \tag{2.1}$$

donde, *max* y *min* son los valores máximo y mínimo del intervalo inicial [*a*,*b*], y *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. Este método se ha seleccionado por su sencillez y rapidez. No obstante, el diseño concebido en la implementación de este procedimiento da la posibilidad de incorporar nuevos métodos de discretización o incluso considerar en esta etapa métodos de agrupamiento, como pudiera ser el algoritmo k-means para obtener los intervalos asociados a cada atributo (Jyh-Shing, 1998).

2.2.3 Etapa 2: Construcción de las variables lingüísticas asociadas a cada término.

En este trabajo se propone construir automáticamente dos tipos de funciones de pertenencia: funciones triangulares y funciones campana Beta. Las funciones triangulares

por ser sencillas y fácil de obtener los tres parámetros que la definen. Las campanas Beta por ser más suaves y ajustarse mejor a las características de los atributos que describen el problema. Como se mencionó en el capítulo 1, existen diversos métodos para obtener funciones de pertenencia, en esta etapa se ha considerado un método analítico para construirlas definido en (Arco, 2001), a continuación se comentará brevemente.

2.2.3.1 Obtención de las funciones de pertenencia campanas Beta

Para construir funciones de pertenencia Beta se parte de los puntos que son extremos de los intervalos del atributo discreto del cual se quiere obtener la variable lingüística. Para obtener la primera función de pertenencia es necesario tener en cuenta el primer intervalo y el segundo. Para obtener las funciones de pertenencia de la segunda en adelante sólo es necesario considerar la función de pertenencia obtenida en el instante anterior.

Las campanas Beta son asintóticas a las abscisas; por tanto, se establece una cota para la imagen de la función, considerando que los valores del dominio que tengan esa cota como imagen serán, por convenio, los extremos de la función.

Obsérvese en el anexo 3 la representación gráfica de las funciones de pertenencia considerando el solapamiento. Para obtener la primera función de pertenencia se consideró que la primera función es *f* y la segunda es *g*. Se parte de un porciento de solapamiento (*val*) y los extremos de los intervalos (a_{f_o} , b_{f_o} , a_{g_0} y b_{g_o}) que son conocidos, y el objetivo es obtener los valores de Beta (β_f y β_g). Las medias μ_f y μ_g se pueden calcular con facilidad considerando como puntos extremos los valores a_{f_o} , b_{f_o} , a_{g_0} y b_{g_o}, respectivamente. El cálculo de los valores β_f y β_g parte de considerar el porciento del área que se debe solapar y establecer una ecuación que relaciona el área debajo de la curva f y el área que se solapa, donde el objetivo principal es obtener β_f y β_g. El trabajo algebraico desarrollado sobre esta expresión permite obtener la expresión final 2.2 para β_f (Arco, 2001).

$$\beta_f = \frac{\mu_g \beta_{f_0} - \mu_f \beta_{g_0}}{\left(\beta_{f_0} + \beta_{g_0}\right)\tan\left[\arctan\sqrt{\frac{1-val}{val}}\left(\frac{100\beta_{f_0} + 100\beta_{g_0} - 2p\beta_{f_0}}{100\left(\beta_{f_0} + \beta_{g_0}\right)}\right)\right]} \qquad (2.2)$$

Ya obtenido β_f según la ecuación 2.2, y considerando la proporción que existe del área bajo la curva y el área que se solapa, según la expresión 2.3, es posible obtener el valor de β_g.

$$\frac{\beta_f}{\beta_g} = \frac{\beta_{f_0}}{\beta_{g_0}} \tag{2.3}$$

Este análisis se ha hecho para obtener las dos primeras funciones de pertenencia Beta, correspondientes a los dos primeros intervalos del atributo discreto. Para el resto de los intervalos sólo es necesario conocer β y β_0 relativos a la función anterior a la que se quiere calcular y según ecuación 2.2 se obtiene el valor de β_f deseado. Obtenidos los valores de β y μ para cada uno de los intervalos de un atributo discretizado, se pueden crear las funciones de pertenencia de la variable lingüística correspondiente.

2.2.3.2 Obtención de las funciones de pertenencia triangulares

Ahora se analizará la obtención de las funciones de pertenencia triangulares según (Arco, 2001). Al igual que en las funciones Beta, se parte de los puntos que son extremos de los intervalos del atributo discretizado del cual se quiere obtener la variable lingüística. Para obtener la primera función de pertenencia es necesario tener en cuenta el primer intervalo y el segundo. Para obtener las funciones de pertenencia de la segunda en adelante sólo es necesario considerar la función de pertenencia obtenida en el instante anterior. Para obtener la primera función de pertenencia, se considera *f* como primera función y *g* es la segunda.

Obsérvese en el anexo 4 la representación gráfica de las funciones de pertenencia considerando el solapamiento. Se parte de un porciento de solapamiento y los extremos de los intervalos (a_{f_o} , b_{f_o} , a_{g_0} y b_{g_o}) que son conocidos, y el objetivo es obtener los valores extremos y la media de los triángulos. Las medias se pueden calcular con facilidad porque se calcula a partir de los puntos extremos de cada intervalo. Es necesario obtener los valores a_f, a_g, b_f y b_g, para esto se crea una proporción considerando las áreas debajo de las curvas y el porciento de solapamiento. El trabajo algebraico desarrollado sobre esta ecuación permite obtener la expresión final 2.4 para b_f (Arco, 2001)

$$b_f = \frac{f(cut)\left(\mu_f a_{g_0} - \mu_g b_{f_0}\right) + \frac{p}{50}\mu_f\left(b_{f_0} - \mu_f\right)}{f(cut)\left(a_{g_0} - \mu_g\right) + \left(\frac{p}{50} - f(cut)\right)\left(b_{f_0} - \mu_f\right)} \tag{2.4}$$

Hasta este momento se tienen b_f y μ_f, y aplicando la ecuación 2.5 se obtiene el último parámetro que falta para crear la primera función de pertenencia de una determinada variable lingüística. Para calcular el resto de las funciones de pertenencia se trabaja con los valores ya calculados a_f, μ_f y b_f, así se puede calcular a_g y con el valor de este y μ_g se puede obtener b_g.

$$a_f = 2\mu_f - b_f \tag{2.5}$$

2.2.4 Etapa 3: Aplicación de los algoritmos ID3 duro o ID3 borroso.

Esta etapa considera la selección de los términos que caracterizan cada grupo homogéneo de documentos utilizando el algoritmo ID3, en su variante dura (Quinlan, 1986) (Mitchell, 1997) cuando se trabaja con resultados del agrupamiento duro determinista o con solapamiento, y en su variante borrosa cuando se agrupó con una técnica borrosa (Wang, 2003).

En ambas variantes se genera un árbol de decisión y a partir del árbol generado se obtienen las reglas que describen cada grupo de documentos considerando el valor de los intervalos resultantes del proceso de discretización según la frecuencia de los términos o de las variables lingüísticas que describen cada término, para las variantes dura y borrosa respectivamente. La extracción de palabras claves se realiza a partir del análisis de las reglas obtenidas y este proceso será descrito en la etapa 4.

El algoritmo ID3 realiza un aprendizaje supervisado, es decir, parte de ejemplos previamente clasificados. Se podría pensar que no es posible aplicar ID3 a una colección de documentos, ya que los documentos originalmente no están etiquetados. En el problema que se resuelve, la clasificación de cada documento lo constituyen el o los grupos a los cuales pertenece después del proceso de agrupamiento.

Ya se conoce la idea básica de la inducción de árboles de decisión, sin embargo, es necesario identificar y considerar en esta etapa las características fundamentales y especificidades del algoritmo ID3 en su variante dura, así como en la variante borrosa.

2.2.4.1 Algoritmo ID3 en su variante dura

La entrada requerida al algoritmo ID3 en su variante dura se muestra en el anexo 5 (a). 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. Dada una colección S que contiene ejemplos, donde el atributo objetivo puede tomar c valores diferentes, entonces la entropía de S relativa a las c posibles clasificaciones se muestra en la expresión 2.6 (Mitchell, 1997):

$$Entropy(S) = \sum_{i=1}^{n} - p_i \log_2 p_i \tag{2.6}$$

donde p_i es la proporción de S que pertenecen a la clase i.

Dada la entropía como una medida de la impureza en una colección de ejemplos de entrenamiento, es posible calcular la 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 2.7:

$$Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{S_v}{S} Entropy(S_v) \tag{2.7}$$

donde $Values(A)$ es el conjunto de todos los posibles valores para el atributo A, y S_v es el subconjunto de S para el cual 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*.

2.2.4.2 Algoritmo ID3 borroso

La entrada requerida al algoritmo ID3 en su variante borrosa se muestra en el anexo 5 (b). Un método estándar para seleccionar un atributo de prueba en la inducción de un árbol de decisión clásico es escoger aquel que tenga mayor ganancia de la información. Sin embargo, aparecen problemas si se aplica esta medida directamente en la inducción de árboles de decisión borrosos. Por tanto, la inducción de árboles de decisión borrosos difiera de la variante dura, es por eso que se requiere reformular las variantes de cálculo de la entropía y la ganancia. Para lograr una mejor comprensión de la variante borrosa, se presentan las notaciones siguientes según (Wang, 2003):

- $C=\{C_1, \ldots, C_m\}$, conjunto de las clases. En el procedimiento propuesto cada clase se hace corresponder con cada grupo obtenido por el algoritmo *Fuzzy* SKWIC.
- $A=\{A_1, \ldots, A_n\}$, es el conjunto de los atributos de entrada con dominio $dom(A_i)$, $1 \leq i \leq n$. En el procedimiento propuesto A_i se corresponden con las palabras o términos que describen los documentos.
- Para cada variable $A_i \in A$, $1 \leq i \leq n$:
  - $u^i \in dom(A_i)$, es un valor duro del atributo A_i.
  - D_i es la partición borrosa de A_i.
  - a_p^i denota el conjunto borroso (término lingüístico) p para el atributo A_i.

Por ejemplo:

- A_i = palabra ‘RESEARCH’.
- $u^{RESEARCH}$ = 7.58 (frecuencia de de la palabra ‘RESEARCH’ en Doc_j).
- $D_{RESEARCH}$ = {Frecuencia Baja, Frecuencia Normal, Frecuencia Alta}.

- $a^{RESEARCH}_{Frecuencia\ Baja}$ significa el término lingüístico Frecuencia Baja de la variable lingüística asociada al atributo palabra 'RESEARCH'.

- $E=\{e_1, \ldots, e_s\}$, conjunto de referencia (ejemplos). En el procedimiento propuesto los ejemplos constituyen todos los documentos que pertenecen al corpus a procesar.
  - $e_k = \left(\overrightarrow{u_k}, \overrightarrow{y_k}\right)$, $1 \le k \le s$.
  - $\overrightarrow{u_k}$, es el vector de entrada. En el problema que se trata, este vector coincide con las frecuencias de los términos que describen el corpus en el documento *k*.
  - $\overrightarrow{y_k} \in [0,1]^m$, es el vector de salida de e_k. Este valor se hace corresponder con los grados de pertenencia del documento *k* a cada grupo resultado del agrupamiento.
  - u_k^i (i.e. el *i*-ésimo elemento de $\overrightarrow{u_k}$) es un valor duro del atributo A_i en el ejemplo e_k.
  - y_k^j (i.e. el *j*-ésimo elemento de $\overrightarrow{y_k}$) es el grado de pertenencia del ejemplo e_k a la clase C_j.

- El conjunto de entrenamiento es un conjunto borroso sobre *E* definido por unos pesos de confidencia inicial $X=\{X_1, \ldots, X_s\}$, $\forall k$, $0 \le k \le s$: $0 \le X_k \le 1$. Si no hay información provista por el usuario, se pone $\forall k$, $1 \le k \le s$: $X_k=1$.

En esta etapa se consideran dos formas de obtener información para pesar los documentos:

1. El mayor grado de pertenencia del documento a un grupo.
2. La varianza de los grados de pertenencia del documento a todos los grupos.

El mayor grado de pertenencia de cada documento a los grupos es una forma que se propone en este trabajo para pesar los documentos, porque mientras mayor sea este grado de pertenencia, el documento es más típico o representativo en ese grupo, por tanto debe tener una mayor responsabilidad en la inducción del árbol de decisión borroso.

Aquellos documentos que tengan un grado de pertenencia similar a cada grupo, son documentos que no son típicos o característicos de ningún grupo en particular, por tanto,

estos documentos deben tener una menor influencia en la selección de los algoritmos para inducir el árbol borroso. De ahí que los documentos que tengan una mayor varianza de los grados de pertenencia a los grupos, deben tener un protagonismo mayor en la inducción. Es por eso, que uno de los criterios se que proponen en este trabajo para pesar los documentos es considerar la varianza de los grados de pertenencia de cada documento a todos los grupos como una forma de pesar los documentos.

- Para cada nodo N en el árbol borroso, $X_N = \{X_1^N,...,X_s^N\}$, es el conjunto de ejemplos borrosos (un conjunto borroso sobre E) en N. En la raíz, este conjunto de ejemplos borrosos coincide con el conjunto de entrenamiento, i.e. $\forall k, \quad 1 \le k \le s: \quad X_k^{Root} = X_k$.

- $Z_j^N = \sum_{k=1}^{s} \mathsf{T}(X_k^N, Y_k^j)$, será el contador de ejemplos para la clase C_j en el nodo N.

Nótese que en la expresión anterior se ha introducido el cálculo de una T-norma. Las T-normas son las funciones usadas para la intersección de los conjuntos borrosos ($i(a,b)$) (Buckley, 2002). Una T-Norma es una función $z = \mathsf{T}(a,b)$, $0 \le a,b,z \le 1$ que cumple las propiedades siguientes:

1. $\mathsf{T}(a,1)=a$
2. $\mathsf{T}(a,b)=\mathsf{T}(b,a)$
3. Si $b_1 \le b_2$, entonces $\mathsf{T}(a,b_1) \le \mathsf{T}(a,b_2)$
4. $\mathsf{T}(a,\mathsf{T}(b,c))= \mathsf{T}(\mathsf{T}(a,b),c)$

Las T-Normas básicas son:

$\mathsf{T}_m(a,b) = \min(a,b)$, llamada intersección estándar.

$\mathsf{T}_b(a,b) = \max(0,a+b-1)$, llamada la suma límite.

$\mathsf{T}_p(a,b) = a \cdot b$, llamada el producto algebraico.

$$\mathsf{T}_*(a,b)=\begin{cases} a, & si \quad b=1 \\ b, & si \quad a=1 \\ 0, & en\ otro\ caso \end{cases}$$, llamada la intersección drástica.

Nótese que $\mathsf{T}_*(a,b) \le \mathsf{T}_b(a,b) \le \mathsf{T}_p(a,b) \le \mathsf{T}_m(a,b)$, $\forall a,b \in [0,1]$

- $Z^N = \sum_{j=1}^{s} Z_j^N$, es el contador total para todos los ejemplos de todas las clases.

- $I(X^N)$, denota la entropía de Shannon de la clase de distribución con respecto al conjunto de ejemplos borrosos X^N en el nodo N.
- $I(X^N \mid A_i)$, es la suma pesada de las entropías de todos los nodos hijos, si A_i es usado como atributo de prueba en el nodo N.
- $Gain(X^N, A_i) = I(X^N) - I(X^N \mid A_i)$, es la ganancia de la información con respecto al atributo A_i.
- $SplitI(X^N, A_i)$, denota la información dividida, la entropía con respecto a la distribución del valor del atributo A_i (en lugar de la distribución de la clase).
- $GainR(X^N,A_i) = Gain(X^N,A_i)/SplitI(X^N, A_i)$, es la razón de la ganancia de la información con respecto al atributo A_i.
- $\mu_{ai}\left(u_k^A\right)$, es el grado de pertenencia del valor del atributo u_k^A al conjunto borroso a_i.
- $X_k^{N|ai} = T_1\left(X_k^N, \mu_{ai}\left(u_k^A\right)\right)$, $1 \le k \le s$, es el grado de pertenencia del ejemplo e_k al subconjunto de ejemplos borrosos para el conjunto borroso a_i.
- $Z_{Cj}^{N|ai}$, contador para los ejemplos que pertenecen al conjunto borroso a_i y la clase C_j.

Para determinar el mejor atributo de prueba, se crea una tabla de contingencia para cada atributo candidato A en el nodo N, desde la cual se puede calcular la medida de la información para el atributo A. Obsérvese en el anexo 6 la tabla de contingencia, cuyos valores se calculan según las expresiones 2.8 y 2.9:

$$Z_{C_j}^{N|a_i} = \sum_{k=1}^{S} T_2\left(X_k^{N|a_i}, Y_k^{j}\right) \tag{2.8}$$

$$X_k^{N|a_i} = T_1\left(\mu_{a_i}\left(u_k^A\right), X_K^N\right), \quad 1 \le k \le S \tag{2.9}$$

Si A fue el atributo de prueba en N; los subconjuntos de ejemplos borrosos en los dos nodos hijos son $X^{N|ai}, i = 1,..,n$. Desde la fila "CSum" se obtiene la distribución de la frecuencia de clases y su entropía en N:

$$I\left(\hat{X}^N\right) = -\sum_{j=1}^{m}\left(Z_{cj}^N / Z^N\right)\log_2\left(Z_{cj}^N / Z^N\right) \tag{2.10}$$

Cada fila a_i representa un nodo hijo N^{ai}. Línea a línea se puede obtener la entropía de cada subconjunto de ejemplos para el conjunto de ejemplos borrosos a^i según 2.11:

$$I\left(X^{N|ai}\right) = -\sum_{j=1}^{m}\left(Z_{cj}^{N|ai} / Z^{N|ai}\right)\log_2\left(Z_{cj}^{N|ai} / Z^{N|ai}\right) \tag{2.11}$$

donde $Z^{N|ai}$, es el contador para todos los ejemplos en el nodo hijo N^{ai}. Seguidamente se calcula la suma pesada de las entropías la cual se define según la expresión 2.12, teniendo en cuenta que Z^N se calcula según la expresión 2.13.

$$I\left(X^{N|A}\right) = \sum_{i=1}^{n} Z^{N|ai} \Big/ Z^N \; I\left(X^{N|ai}\right) \tag{2.12}$$

$$Z^N = \sum_{i=1}^{n} Z^{N|ai} = \sum_{j=1}^{m} Z_{cj}^N \tag{2.13}$$

Obsérvese en la expresión 2.14 el cálculo de la ganancia de la información del atributo *A*:

$$\overline{Gain}\left(X^N, A\right) = I\left(\hat{X}^N\right) I\left(X^N \mid A\right) \qquad \hat{X}_k^N = \sum_{i=1}^{n} X_k^{N|ai}, 1 \le k \le s \tag{2.14}$$

La información dividida $SplitI\left(X^N, A\right)$ del atributo A es calculada desde $Z^{N|ai}$ y corresponde a la suma de los grados de pertenencia de los ejemplos de los subconjuntos de ejemplos borrosos para los conjuntos borrosos a_i. Obsérvese la expresión 2.15.

$$SplitI\left(X^N, A\right) = -\sum\left(Z^{N|ai} \Big/ Z^N\right)\log_2\left(Z^{N|ai} \Big/ Z^N\right) \tag{2.15}$$

La razón de la ganancia de la información del atributo *A* se muestra en 2.16:

$$GainR\left(X^N, A\right) = \overline{Gain}\left(X^N, A\right) \Big/ SplitI\left(X^N, A\right) \tag{2.16}$$

(Si $\overline{Gain}\left(X^N, A\right)$ es no negativa, entonces $GainR\left(X^N, A\right)$ es también no negativa).

Con los pasos descritos anteriormente, se puede estimar la medida de la información para los candidatos actuales en el nodo N y escoger el mejor atributo de prueba.

En los árboles de decisión duros, es sencillo seleccionar qué ejemplos pertenecen al nodo que se esté construyendo asociado a un valor de un determinado atributo. Sin embargo, la lógica borrosa considera que todos los elementos pertenecen a todos los conjuntos pero con un grado de pertenencia dado. Por tanto, en los FDT al ramificar un atributo por un determinado término lingüístico, todos los ejemplos tienen un grado de pertenencia a ese término lingüístico. Supóngase que se tiene un nodo correspondiente a la variable lingüística *i* y que dicha variable está compuesta por *k* términos lingüísticos. Supóngase además, que el nuevo nodo a formar es el correspondiente al término lingüístico a_j^i (término lingüístico *j* de la variable lingüística *i*). ¿Qué ejemplos considerar en el nodo a ramificar asociado a a_j^i? Para resolver este problema en este trabajo se proponen dos variantes.

1. Aplicar el Principio de Máxima Membresía, de forma tal que se incluirán en el nodo correspondiente a a_j^i aquellos ejemplos para los cuales $\mu_{a_j^i}(u_x^i) \geq \mu_{a_y^i}(u_x^i)$, $\forall y \in [1,k]$; $y \neq j$, es decir, se incluirán los ejemplos que de todos los grados de pertenencia a los términos de la variable lingüística *i*, el mayor grado sea el correspondiente al término *j*.
2. Dada la especificación de un umbral α, aplicar α-corte para incluir en el nodo correspondiente a a_j^i aquellos ejemplos para los cuales se cumpla que $\mu_{a_j^i}(u_x^i) \geq \alpha$, es decir, se incorporarán al nodo todos aquellos ejemplos que cumplan el α-corte para el término a_j^i.

¿Cuándo terminar la ramificación? En el capítulo 1 se mencionaron algunos criterios de parada a tener en cuenta en la inducción de árboles de decisión. En esta etapa se han

incluido tres criterios de parada. El más general es detener la ramificación cuando no hay más atributos o rasgos para las clasificaciones. El segundo criterio incluido es considerar un nodo hoja cuando todos los ejemplos pertenecen a la misma clase. El tercer y último criterio considerado en la inducción compara el valor de la medida de la información del atributo con un umbral especificado, si el valor calculado es menor que umbral dado por el usuario o calculado por el propio algoritmo, se detiene la ramificación. Este criterio de parada evita escoger atributos con muy bajo valor de la información. La variante automatizada calcula el umbral como la media de las ganancias de los atributos que describen inicialmente el conjunto de ejemplos.

Tanto para verificar ciertos criterios de parada, como para determinar las clases asociadas a un nodo hoja, es necesario especificar cómo se identifica a qué clase o conjunto de clases pertenece un ejemplo. Nótese que en este caso, también se presenta la lógica borrosa, ya que todos los ejemplos tienen un grado de pertenencia a cada una de las clases. Se han tenido en cuenta dos formas de escoger las clases correspondientes a un ejemplo o nodo hoja:

1. Aplicar el Principio de Máxima Pertenencia, de forma tal que se considerará una única clase asociada al ejemplo y ésta será aquella clase a la cual el ejemplo tenga el mayor grado de pertenencia. Para identificar la clase asociada a un nodo hoja se aplica este principio a cada ejemplo del nodo y se selecciona la clase que tenga el mayor número de ejemplos asociados.
2. Definir un umbral α y aplicar α-corte, de esta forma se incluyen en la clasificación de un ejemplo todas aquellas clases para las cuales el ejemplo perteneció con un grado mayor que el umbral α. Para identificar las clases asociadas a un nodo hoja se aplica el α-corte a cada ejemplo del nodo y las clases asociadas a ese nodo serán todas aquellas obtenidas de los ejemplos que pertenecen al nodo siguiendo este criterio. Esta es una de las ventajas de los FDT, al permitir en un nodo hoja más de un valor del rasgo objetivo.

Otro elemento a tener en cuenta al obtener un nodo hoja es la definición de su certidumbre, aspecto importante al generar y aplicar las reglas a partir del FDT (véase subepígrafe

2.2.4.3). Se han considerado dos formas de calcular la certidumbre de un nodo hoja (certidumbre de la regla que se genera desde la raíz hasta dicho nodo hoja):

1. Calcular la certidumbre del nodo hoja como la media de los grados de pertenencia de los ejemplos que están en el nodo a las clases seleccionadas para ese nodo.
2. Considerar la suma pesada de los grados de pertenencia de los ejemplos existentes en el nodo hoja a las clases seleccionadas para ese nodo. La ponderación se basa en el peso asociado a cada ejemplo.

2.2.4.3 Generación de reglas que describen un corpus textual a partir de las variantes dura y borrosa del algoritmo ID3

Después de construido el árbol se pueden generar las reglas que describen el corpus de textos, teniendo en cuenta que 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 o términos lingüísticos para ID3, variante dura o borrosa, respectivamente) y el consecuente es el nodo hoja (i.e. clases asociadas y certidumbre de la regla).

La generación de reglas a partir de una variante dura del algoritmo ID3 es presentada en (Valdés, 2005). La generación a partir de un ID3 borroso se muestra en el anexo 7 a partir de un ejemplo.

2.2.5 Etapa 4: Extracción de palabras claves de grupos textuales.

En esta etapa del modelo se proponen 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, por ejemplo, para posible etapa posterior de generación automática 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.

Tanto la selección de las palabras de mayor relevancia resultante de métodos de agrupamiento, como la selección de los términos con mayores valores de calidad en el grupo, son formas de selección que coinciden tanto para cuando el agrupamiento se realizó aplicando una técnica dura o borrosa, con la única diferencia que cuando la técnica es borrosa se hace necesario especificar un umbral para determinar qué documentos pertenecen a cada grupo. Estas variantes de selección se encuentran descritas en (Valdés, 2005) y (Arco, 2005). Sin embargo, la elección de los términos a partir de las reglas generadas por el algoritmo ID3 depende si la variante fue dura o borrosa. A continuación se describe como se obtienen las palabras claves a partir de las reglas generadas de un árbol borroso inducido a partir de una colección textual previamente agrupada.

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, término lingüístico asociado a la variable lingüística correspondiente al atributo que describe el nodo) sea uno de los *n* mejores valores que puede alcanzar ese término, donde *n* es un valor de entrada al algoritmo. Obsérvese en el anexo 8 un ejemplo.

Si el algoritmo que generó la colección de grupos de documentos generó también la relevancia de cada término por grupo, se pueden interceptar las listas de palabras claves que se obtienen con el ID3 borroso 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.

2.3 Generalidades del sistema CorpusMiner

El procedimiento general propuesto en el epígrafe 2.2 que soporta el modelo propuesto en el epígrafe 2.1 ha sido implementado e incorporado a la herramienta para el procesamiento textual CorpusMiner. Por tal motivo, en este epígrafe se describirá el diseño general de este sistema y los módulos que lo componen.

La herramienta CorpusMiner permite obtener un resumen extracto de un corpus textual partiendo de la verificación de la homogeneidad del mismo utilizando métodos de

agrupamiento. Para lograr este propósito en el sistema se representa un corpus textual según VSM y se parte de una etapa de representación textual, donde se incluyen subetapas de transformación del corpus, extracción de términos, reducción de la dimensionalidad, y normalización y pesado de los vectores documentos. Observe en el anexo 9 todas las etapas incluidas en CorpusMiner para el procesamiento textual (Valdés, 2005) (Arco, 2005).

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 éstas, es posible agrupar aquellos documentos que sean similares por su contenido. Los métodos de agrupamiento incluidos en CorpusMiner son, como se mencionó anteriormente, los algoritmos SKWIC y *Fuzzy* SKWIC (Berry, 2004), el algoritmo *Extended Star* (Gil-García, 2003), o las variantes concatenadas *Extended Star* – SKWIC y *Extended Star* – *Fuzzy* SKWIC (Arco, 2005). En cualquier caso se obtiene como resultado una colección de grupos de documentos. La descripción, diseño e implementación de los algoritmos involucrados en ambos módulos aparece detallada en (Mederos, 2005) (Arco, 2005).

En CorpusMiner, a partir de los resultados del agrupamiento se requiere seleccionar aquellos términos que son relevantes y caracterizan cada grupo obtenido, para así, obtener un resumen extracto de cada tema que abordan los documentos de la colección. Es precisamente en esta etapa donde se incluye el módulo con la implementación del procedimiento general que soporta el modelo que se ha propuesto en este trabajo.

2.4 Diseño del sistema CorpusMiner

El diseño del sistema CorpusMiner se dividió en tres capas fundamentales como se muestra en la figura 2.1. 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 (Valdés, 2005).

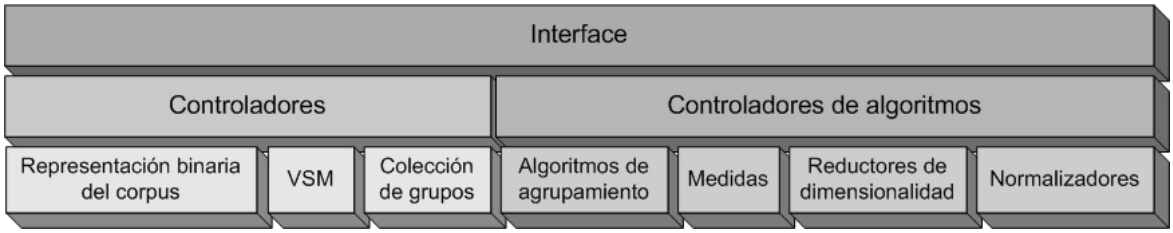


Figura 2.1 Diseño general del sistema CorpusMiner.

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 cluster, 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; esta al igual que la primera capa, esta dividida en dos tipos de clases fundamentales las controladoras de las clases de datos y las controladoras de algoritmos. Observe anexos 10 y 11.

2.5 Implementación del procedimiento general del modelo propuesto y su incorporación a CorpusMiner

El procedimiento general que sustenta el modelo propuesto para la selección de palabras claves en grupos homogéneos de documentos fue diseñado e implementado siguiendo el diseño general de CorpusMiner. A continuación se especificarán las clases e interfaces que permiten la discretización de los rasgos, la construcción de las variables lingüísticas y las funciones de pertenencia asociadas, la inducción de los árboles de decisión duro y borroso según algoritmo ID3 en cada una de sus variantes y la extracción de los términos relevantes a partir de los árboles y reglas obtenidos.

2.5.1 Diseño e implementación de la discretización de los rasgos que describen los documentos

Como se mencionó en el epígrafe 2.2.2 se consideró que el método de discretización que se implementara fuera por amplitud, es decir, por intervalos de igual tamaño, no obstante, el uso de las interfaces en el diseño brindan la posibilidad de incorporar nuevos métodos de discretización o incluso considerar en esta etapa métodos de agrupamiento. A continuación se describirá como se implementó el mismo. Obsérvese el anexo 12.

Todos los dicretizadores deben implementar la interfaz *IDiscretize*. Esta posee dos funciones llamadas *Discretize* y *DiscretizefF*, las cuales implementan la discretización por intervalos de igual tamaño (ver epígrafe 2.2.2). Dichas funciones reciben como parámetro la representación VSM pero se diferencian en que la primera recodifica dicha matriz de dispersión (VSM), ver (Valdés, 2005) y la segunda devuelve un objeto de la clase *TDiscretizerForFuzzy_Output* que es una colección de atributos discretizados. Para el caso de la variante borrosa del algoritmo ID3 se utiliza *DiscretizefF*.

Cada atributo discretizado es una instancia de la clase *TInterval*, la cual implementa la interfaz *IMetricDiscreteAttribute* y no es más que una colección de los intervalos del atributo discretizado, que a su vez, estos intervalos son objetos de la clase *TOutput*, que posee como atributos el nombre del intervalo y los límites del mismo.

2.5.2 Diseño e implementación de la construcción de las variables lingüísticas asociadas a cada término

Al igual que en el caso de los discretizadores, el diseño de las funciones de pertenencia, mediante el uso de las interfaces, permite que puedan ser implementados otros tipos de funciones de pertenencia y el sistema se mantenga inalterable.

En esta versión del CorpusMiner se desarrollan dos tipos de funciones de pertenencia clásicas, las curvas Beta (como un tipo de funciones campana) y las funciones triangulares. Tanto en las funciones de pertenencia Beta (*TAbsBellMembership*) como en las funciones triángulo (*TAbsTriangleMembership*), es necesario distinguir tres tipos de funciones: las relacionadas con el intervalo del extremo izquierdo de la variable lingüística (*TLowLimitBellMembership* y *TLowLimitTriangleMembership*), las correspondientes al

intervalo del extremo derecho de la variable lingüística (*THighLimitBellMembership* y *THighLimitTriangleMembership*) y las que corresponden al resto de los intervalos (*TBellMembership* y *TTriangleMembership*), (Arco, 2001). Obsérvense los anexos 13, 14 y 15.

La concepción general del diseño consiste en una clase *TMembership* que tiene todos los atributos y métodos comunes de las funciones de pertenencia, por ejemplo, su nombre y los métodos que permiten acceder a éste. Esta clase hereda de *TInterfacedObject* y de ella heredan las clases correspondientes a los tres tipos de funciones de pertenencia implementados, aunque se permite que de ella hereden tantas funciones de pertenencia como se quieran implementar. La clase *TAbsBellMembership* tiene los atributos Beta y media que son los parámetros de las funciones Beta y la clase *TAbsTriangleMembership* tiene los atributos límite izquierdo, media y límite derecho que son los atributos de una función triangular. Todas estas clases tienen métodos que permiten manipular sus atributos. (Arco, 2001)

Obsérvese en el anexo 15 que están creadas las interfaces *IBellMembership* e *ITriangleMembership* que permiten el trabajo con los métodos comunes de las curvas Beta y las funciones triángulo respectivamente. *Evaluator* es un método común para todas las funciones y es a su vez el principal. Esta función permite dado un valor del dominio, evaluarlo en la función de pertenencia y devolver el valor de pertenencia entre 0 y 1. Todas las funciones de pertenencia necesitan tener implementado el método pero realizan la evaluación de una manera diferente. Surge entonces la interfaz *IMembership* con el método *Evaluator*, y esta interfaz la deben implementar todas las clases de funciones de pertenencia existentes y las nuevas que se quieran adicionar al sistema (Arco, 2001). Es por eso, que el diseño está preparado para asumir nuevos tipos de funciones de pertenencia sin alterarlo. El diseño muestra cómo el uso de esta interfaz facilita el trabajo con las funciones de pertenencia. Obsérvese en el anexo 16 el uso de las interfaces en la creación y evaluación de las funciones de pertenencia campanas Beta., de una manera similar se realiza con las funciones triangulares.

2.5.3 Diseño e implementación de los algoritmos ID3 duro e ID3 borroso.

La implementación del ID3 variante dura se encuentra detalla en (Valdés, 2005), por lo que en este epígrafe se enfatizará en la descripción de los elementos fundamentales en la implementación de la variante borrosa.

En el diseño se concibió la clase *TID3_Fuzzy* para la implementación de la generación de reglas que caractericen una colección de grupos de documentos utilizando el algoritmo ID3 borroso. Esta clase recibe tres entradas principales como argumento. La primera, es una representación VSM de la cual, se toman los términos como rasgos y los documentos como ejemplos. La segunda, es una colección borrosa de grupos obtenida del algoritmo *Fuzzy* SKWIC, considerando cada grupo como la clase a la que pertenece cada ejemplo (i.e., cada documento). La tercera es una colección de intervalos, producto de la discretización de los términos de la representación VSM. Esta clase es la encargada de crear el árbol de decisión borroso. A continuación se describirán brevemente los pasos para la construcción del mismo.

Primeramente, es necesario construir las variables lingüísticas correspondientes a cada atributo, todas las variables lingüística será del mismo tipo. Obsérvese en el epígrafe 2.5.2 el tipo de variables lingüísticas consideradas en esta implementación.

Como se especificó en el epígrafe 2.2.4.2, el método de inducción del árbol de decisión borroso según la variante borrosa del algoritmo ID3 implementada, permite asociar pesos a los ejemplos. Las clases *TMax_Membership_ToThe_Cluster* y *TMax_Variance_Value* son las que permiten el cálculo de los pesos asociados a los ejemplos. La primera se utilizará si la forma de calcular dichos pesos es por el mayor grado de pertenencia a los grupos, mientras que la segunda, si el cálculo es a partir del valor de la varianza de los grados de pertenencia. Ambas clases implementan el método *Weigth* de la interfaz *IWeigths* el cual devuelve un objeto de la clase *TWeigth* que tendrá una lista de los documentos con sus respectivos pesos.

Obtenidas las variables lingüísticas asociadas a término y asignados los pesos a los ejemplos, es posible inducir el árbol de decisión borroso. Según la descripción realizada en el epígrafe 2.2.4.2, en cada nodo se pondrá el atributo que "gane", de todos los "candidatos", para esto se construyen las tablas de contingencias de dichos atributos, para

las cuales se implementó la clase *TContingencyTableC* que no es más que una lista de objetos de la clase *TContingencyTable*. Con la información dada por las tablas de contingencias se puede calcular la ganancia de cada atributo y así saber quien resultó ganador. Obsérvese el anexo 17.

Después que se tiene el nodo con el atributo "ganador" se ramifica el árbol, teniendo por cada nodo tantos hijos como términos lingüísticos tenga el atributo presente en dicho nodo.

Para cada nodo hijo se seleccionarán los ejemplos en el que coincidan: el valor del atributo "ganador" con el término lingüístico por el cual se está ramificando. Para reducir la cantidad de ejemplos presentes en cada nodo, y de esta forma no introducir atributos innecesarios en el árbol, se diseñó la clase *TExamplesInNode*. En esta clase se implementaron dos métodos: *Examples_AlfaCourt* y *Examples_MaxMS*; los que seleccionarán para ese nodo hijo los ejemplos que cumplan con el criterio de α-corte -tendrá en cuenta los ejemplos para los que el valor del atributo "ganador" en el nodo padre, evaluado en el término lingüístico correspondiente, es mayor que determinado umbral(α)- o el de máxima membresía, respectivamente -tendrá en cuenta los ejemplos para los cuales el valor del atributo "ganador" en el nodo padre, evaluado en el término lingüístico correspondiente, sea el mayor de todos-, ver subepígrafe 2.2.4.2.

Para el segundo criterio de parada, 2.2.4.2, se tuvo en cuenta los criterios de máxima membresía y el de alfa-corte, para esto se implementaron las clases que heredan de *TClassBelong*: *TMaxMemberShipForStop* y *TAlfa_Court.* Estas clases tienen un método llamado *GetCluster*, el cual devuelve la clase a que pertenecen todos los ejemplos si es que todos estos pertenecen a la misma clase y -1 en caso contrario.

Después que se generan todos los nodos con sus respectivos atributos, para ese camino, se selecciona la clase a la que pertenece dicho camino, y su grado de certidumbre. Para seleccionar la clase se toma la misma idea que se utilizó para el segundo criterio de parada de construcción del árbol, ver subepígrafe 2.2.4.2, y se implementó la clase *TFillLeaf* la cual tiene dos métodos que implementan los criterios de máxima pertenencia, *Cluster_MaxMS*; y alfa-corte, *Cluster_AlfaCourt.*

La clase *TCertitude* fue implementada para saber el grado de certidumbre de una regla o nodo hoja. Esta clase posee los métodos *MeanofBelong* y *WeigthSum*, que calculan la

media de los grados de pertenencia de los documentos a la clase y la suma pesada de los documentos presentes en este nodo hoja, respectivamente, considerando como pesos el peso de los documentos.

Así se tiene el árbol de decisión borroso. Para almacenar toda esta información se hizo necesario el diseño e implementación de la clase *TFuzzy_DecisionNode,* la misma representa un árbol de decisión que es el resultado del algoritmo ID3 borroso. Los nodos que no son hojas representan términos y tienen tantos hijos como cantidad de términos lingüísticos tenga la variable lingüística del atributo (i.e. del término) correspondiente al nodo que se va a generar. Los nodos hojas tienen la certidumbre de la regla resultante y el grupo que clasifica los términos que se hallan en el camino desde la raíz del árbol hasta el nodo hoja.

2.5.4 Diseño e implementación de la extracción de palabras claves de grupos textuales

En el presente epígrafe se describe como se concibió y diseñaron las clases relacionadas con la obtención de palabras claves de una colección de grupos de documentos a partir de las reglas generadas por el algoritmo ID3; la relevancia de las palabras, resultado de métodos de agrupamiento y los valores de calidad de los términos.

En primer lugar, se diseñó la clase *TKeyword* que representa una lista de palabras claves y el grupo al que pertenece la lista. Se creó además, la clase *TKeywordList* que es una lista de *TKeyword*. El resultado que se obtiene a partir de la aplicación de los métodos de extracción de palabras claves es una instancia de *TKeywordList* (Valdés, 2005).

Los dos primeros métodos, es decir, el basado en la relevancia y el basado en la calidad de los términos, pueden ser aplicados a los grupos de documentos por separado, mientras que el método basado en el ID3 necesita la colección de grupos de documentos completa (i.e., no se puede aplicar a un grupo aislado) (Valdés, 2005).

Es por eso que para los dos primeros métodos se diseñaron clases que operan sólo sobre un grupo de documentos e implementan la interfaz *IClusterKeyWordGenerate*. Esta interfaz tiene una función que recibe como parámetro la representación VSM del corpus de textos y un grupo de documentos, y devuelve la lista de palabras claves del grupo. Por otra parte, es

posible extraer las palabras claves a partir de toda la colección de grupos, es por eso que se diseñaron clases que operan sobre la colección de grupos de documentos para los tres métodos implementados. Estas implementan la interfaz *IKeyWordGenerate* la cual posee una función que se le pasa como parámetro una matriz VSM y una colección de grupos, y devuelve un lista palabras por cada grupo (Valdés, 2005).

2.5.4.1 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. Dos de los algoritmos de agrupamiento implementados en CorpusMiner tienen esta característica: SKWIC y *Fuzzy* SKWIC (Mederos, 2005).

Considerando la relevancia por grupo se pueden escoger las palabras claves. En CorpusMiner existen dos formas; las palabras que su relevancia sea mayor (o menor) que cierto umbral y las *n* palabras con mejor valor de relevancia. El primero, recorre por grupos todos los términos y escoge aquellos que su valor de relevancia sea mayor (o menor) que un umbral especificado. El segundo, ordena según la relevancia todos los términos utilizando el método *QuickSort* y selecciona los *n* primeros términos de la lista. La clase *TCCKeywordGeneratorWordRelevanceThreshold* fue implementada para el primer método, y la clase *TCCKeywordGeneratorWordRelevanceBestCount* fue implementada para el segundo (Valdés, 2005).

2.5.4.2 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 (Mederos, 2005) y en (Valdés, 2005) se muestra 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, y como extender esta idea a la selección de palabras por cada grupo de documentos obtenido por un método de agrupamiento. La clase diseñada e implementa es *TCLusterCReductorKWGenerator*.

2.5.4.3 Extracción de palabras claves utilizando el algoritmo ID3

En primer lugar, se genera el árbol de decisión utilizando el ID3. 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. A partir de las reglas obtenidas es posible generar las palabras que logran discernir entre grupos.

Para obtener las palabras claves se implementó la clase *TFuzzyId3KW* la cual se encarga de realizar todo el proceso, el cual sería:

- Construir un árbol de decisión
- Generar las palabras claves teniendo en cuenta las reglas obtenidas del árbol de decisión.

Si el algoritmo que generó la colección de grupos de documentos generó también la relevancia de cada término por grupo, se pueden 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 (Valdés, 2005).

2.6 Conclusiones parciales

- El modelo propuesto permite la selección de palabras claves que logran caracterizar grupos homogéneos de documentos y a la vez logran discernir entre las clases.
- Las características que presenta el procedimiento general del modelo desarrollado confieren ventajas respecto a la consideración en la entrada de nuevas formas de agrupamiento de los documentos, así como la inclusión de otras variantes de discretización o agrupamiento de los rasgos y de funciones de pertenencia asociadas a las variables lingüísticas construidas para cada rasgo.

- La etapa de extracción de palabras claves permite la combinación de la relevancia de los rasgos obtenida de los procesos de agrupamiento, con las palabras seleccionadas a partir del proceso de inducción de los árboles de decisión duros o borrosos.
- El diseño del procedimiento general que soporta el modelo es extensible y se ajusta al diseño general del sistema CorpusMiner. La implementación realizada permite utilizar este módulo como parte de CorpusMiner.

3

EVALUACIÓN DEL MODELO Y DESCRIPCIÓN A NIVEL DE USUARIO DE LA IMPLEMENTACIÓN DEL PROCEDIMIENTO GENERAL

Capítulo 3. EVALUACIÓN DEL MODELO Y DESCRIPCIÓN A NIVEL DE USUARIO DE LA IMPLEMENTACIÓN DEL PROCEDIMIENTO GENERAL

A continuación se describirá como es posible utilizar a nivel de usuario el módulo para la selección de palabras claves de grupos homogéneos de documentos en CorpusMiner. Además, se presentarán los casos de estudio definidos para la evaluación del modelo propuesto. Se presentarán los resultados de la verificación y validación del modelo.

3.1 Interfaz de usuarios de CorpusMiner para la selección de las palabras claves de grupos homogéneos de documentos

En este epígrafe se describirá cómo utilizar en CorpusMiner las opciones asociadas al procedimiento general que soporta el modelo de selección de palabras claves propuesto. Se hará énfasis en las opciones referidas a la colección de grupos borrosos y la extracción de las palabras claves que caracterizan dichos grupos, las referidas a la colección de grupos duros se describen en detalles en (Valdés, 2005). Se asume que el usuario domina el resto de las opciones de CorpusMiner que le permiten representar el corpus adecuadamente.

3.1.1 ¿Cómo obtener la entrada al procedimiento general en su implementación en CorpusMiner?

A partir de la representación VSM pueden obtenerse colecciones de grupos duros y borrosos de documentos sobre las cuales es posible realizar un conjunto de operaciones. A continuación se describirán los pasos a seguir para obtener dichas colecciones de grupos.

La opción "Agrupar los documentos o sentencias" (*Clusterize Documents or Sentence*): Permite seleccionar una de las técnicas de agrupamiento implementadas en CorpusMiner.

El sistema permite realizar agrupamientos con técnicas duras o borrosas, por tanto al querer realizar un proceso de agrupamiento se despliega un submenú que permite decidir si la técnica a aplicar es dura o borrosa, opciones 1 ó 2 respectivamente (ver figura 3.1).

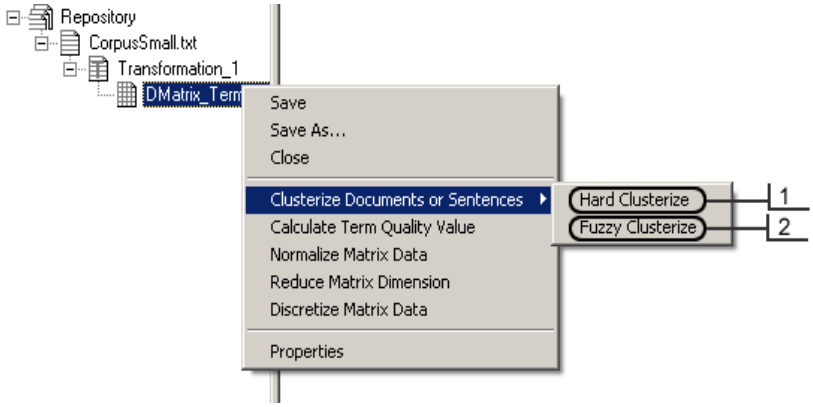


Figura 3.1 Selección de agrupamiento duro o borroso.

Posteriormente se elige el algoritmo de agrupamiento mediante el cual se desea agrupar, en dependencia de la técnica seleccionada. Si decidió por una técnica dura, entonces se presenta un diálogo como el de la figura 3.2. Donde la zona seleccionada con un 1 representa los dos algoritmos duros que ofrece CorpusMiner: SKWIC (Hard SK) o Extended Star (Overlapped Star). Si la decisión fue métodos borrosos, sólo se muestra el algoritmo Fuzzy SKWIC (Fuzzy SK) (Mederos, 2005).

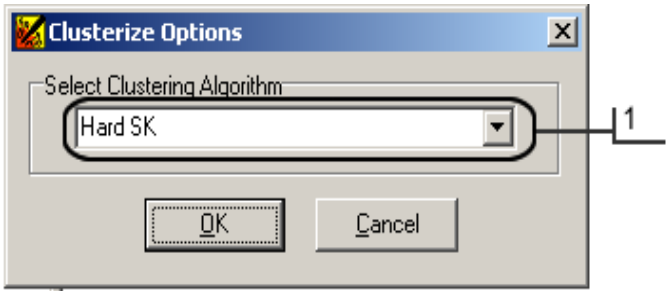


Figura 3.2 Algoritmos de agrupamiento duros.

1. Permite escoger el tipo de algoritmo de agrupamiento.

Una descripción detallada de cómo especificar los parámetros necesarios para los algoritmos SKWIC, Fuzzy SKWIC y Extended Star aparece en (Mederos, 2005).

3.1.2 ¿Cómo discretizar los rasgos que describen los documentos en CorpusMiner?

Como se mencionó en el epígrafe 2.2.2, para esta etapa se eligió la discretización por amplitud, de la cual se implementaron dos variantes descritas en 2.5.1, una para recodificar

la representación VSM y otra para crear la colección de rasgos discretizados; en el diálogo de la figura 3.3 CorpusMiner da la opción de elegir cual variante escoger.

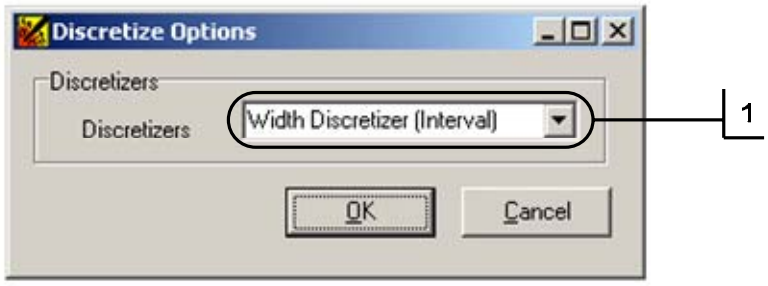


Figura 3.3 Seleccionar el discretizador.

1. Permite seleccionar la variante de discretización por amplitud.

Posteriormente el usuario debe especificar los intervalos en los cuales quiere particionar el rasgo. Esta opción se puede especificar en el diálogo de la figura 3.4.

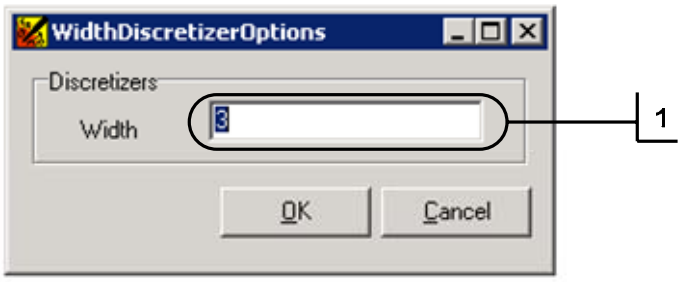


Figura 3.4 Opciones del discretizador por amplitud.

1. Línea de entrada de texto para especificar la cantidad de intervalos a generar (el valor por defecto es 3).

Para aplicar el algoritmo ID3, ya sea en su variante dura o borrosa es de carácter obligatorio discretizar los rasgos, por tanto antes de aplicar este algoritmo, CorpusMiner muestra los diálogos de las figuras 3.3 y 3.4 para llevar a cabo dicha discretización. Además el modelo en esta etapa da la posibilidad de ver los resultados de la discretización aunque no se vaya a utilizar el algoritmo ID3. Para esto después de obtener la representación VSM es posible discretizar estos rasgos de la forma que se muestra en la figura 3.5.

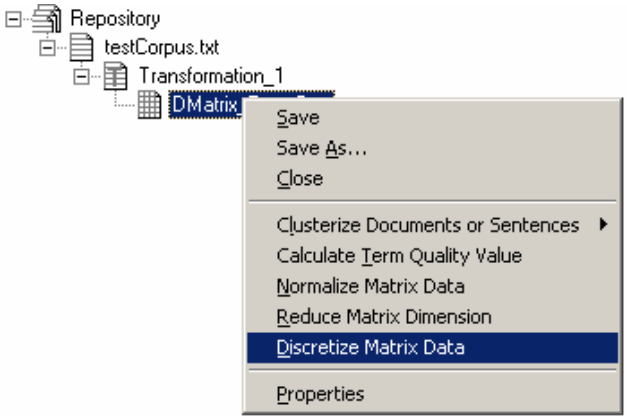


Figura 3.5 Opción que permite obtener una discretización de la representación VSM

3.1.3. ¿Cómo construir las variables lingüísticas asociadas a cada término en CorpusMiner?

En esta etapa las variables lingüísticas son utilizadas por el algoritmo ID3 en su variante borrosa.

Como fue descrito en 2.5.2 en esta versión de CorpusMiner se propone construir automáticamente dos tipos de funciones de pertenencia: funciones triangulares y campana Beta. El tipo de función de pertenencia a escoger para la construcción de las variables lingüísticas, así como el porciento de solapamiento entre estas funciones se especifica en el diálogo de la figura 3.7, en el cual se encuentran todas las opciones que brinda CorpusMiner para la construcción del árbol de decisión borroso.

3.1.4 ¿Cómo aplicar el algoritmo ID3 en sus variantes dura y borrosa en CorpusMiner?

Una de las variantes que implementa CorpusMiner para la selección de palabras claves es a través de la aplicación del algoritmo ID3 en sus variantes dura y borrosa. A continuación se describirá como aplicar este algoritmo.

Variante dura

Después que se obtuvo la colección de grupos de documentos, aplicando un método de agrupamiento duro, el usuario puede seleccionar la opción de construir el árbol de decisión

como se especifica en la figura 3.6 para así obtener un conjunto de reglas que describen los grupos, ver detalles en (Valdés, 2005) .

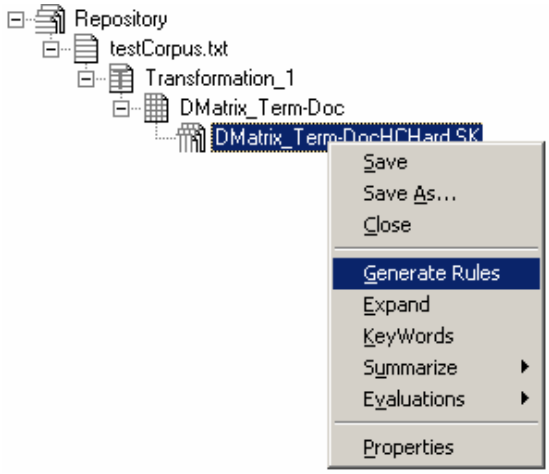


Figura 3.6 Opción para generar las reglas a partir del árbol de decisión obtenido del algoritmo ID3.

Después de esto el usuario tendrá que elegir el algoritmo de discretización como se describió en el epígrafe 3.12.

Variante Borrosa

Para esta variante es necesario que se haya utilizado un método de agrupamiento borroso. Posteriormente el usuario puede seleccionar la misma opción de la figura 3.6, y como el método de agrupamiento fue borroso la variante del algoritmo ID3 que se ejecutará es la borrosa.

Al igual que con la variante dura el usuario tendrá que elegir el algoritmo de discretización.

Para el uso del ID3 en su variante borrosa es necesario que se especifiquen una serie de parámetros los cuales son solicitados en el diálogo de la figura 3.7, entre ellos, el tipo de función de pertenencia que se quiere construir; es importante aclarar que todas las funciones de pertenencia serán del mismo tipo para todas las variables lingüísticas.

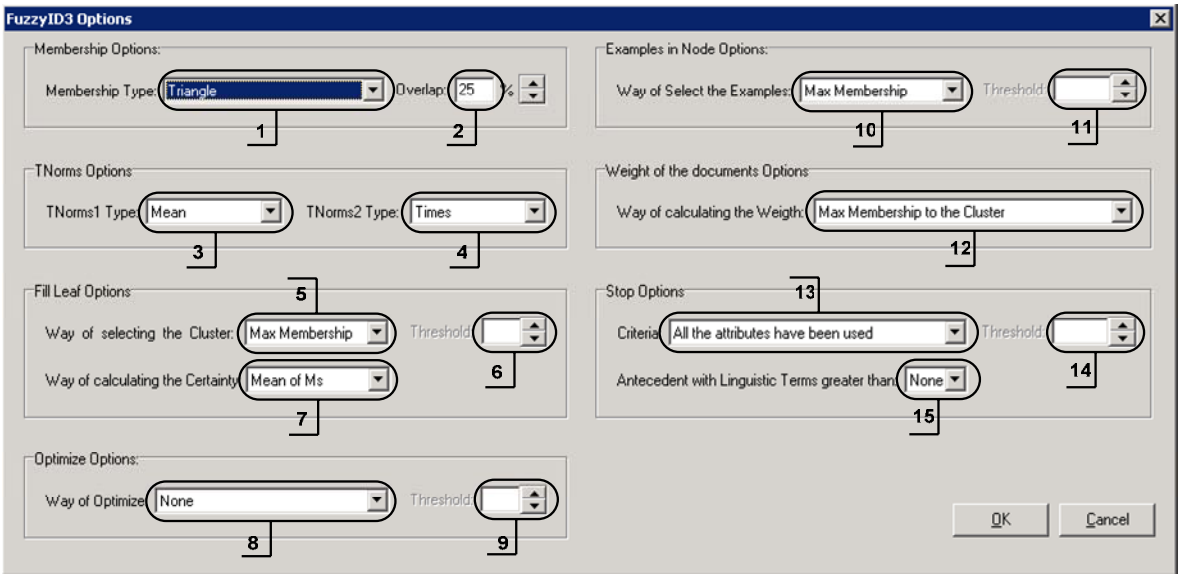


Figura 3.7 Parámetros del algoritmo ID3 variante borrosa.

1. Permite seleccionar el tipo de FP a construir. (*Triangle, Bell*).
2. Porciento de solapamiento de los términos lingüísticos.
3. Permite seleccionar el tipo de T-Norma 1.(*Mean, Min, Times*)
4. Permite seleccionar el tipo de T-Norma 2. (*Mean, Min, Times*)
5. Permite seleccionar la forma de escoger la(s) clase(s) correspondiente(s) a la regla *n*.; las mismas son:
    a. Máxima Pertenencia (*Max Membership*)
    b. Alfa-Corte (*Alfa-court*): Si se elige esta variante se activará el incrementador 6.
6. Es un incrementador para definir el umbral de la opción 5b. (incrementa o decrementa su valor en 0.1, tiene como valor mínimo 0 y como máximo el mayor grado de pertenencia de los documentos a las clases, valor resultante del agrupamiento borroso).
7. Permite seleccionar la forma de calcular la certidumbre de la regla *n*.; las mismas son:
    a. La media de los grados de pertenencia de los ejemplos a los grupos existentes en la hoja (*Mean of Ms*).

b. La suma pesada de los grados de pertenencia de los ejemplos a los grupos existentes en la hoja (*Weigth Sum*).

Como el algoritmo ID3 variante borrosa, genera un árbol de decisión bastante grande se hizo necesario implementar algunos criterios para optimizar el tamaño de éste.

8. Permite seleccionar la forma de optimizar el tamaño del árbol de decisión; la variante propuesta en esta etapa es:
   a. De los grupos obtenidos en el proceso de agrupamiento, desechar los que tengan una media de grado de pertenencia menor al umbral especificado en 9 (*Cluster with low Membership value*).
9. Es un incrementador para definir el umbral de los grados de pertenencia de los grupos que se van a desechar. (incrementa o decrementa con un paso de 0.1, tiene como valor mínimo 0 y como máximo 1).
10. Permite seleccionar los ejemplos que van a estar presentes en cada nodo hijo; las variantes son:
    a. Máxima Pertenencia (*Max Membership*): El o los ejemplos para el cual el atributo seleccionado tiene su mayor grado de pertenencia.
    b. Alfa-Corte (*Alfa-court*): El o los ejemplos para el cual el atributo seleccionado tiene grado de pertenencia mayor que el umbral especificado en 11.
11. Es un incrementador para definir el umbral de los grados de pertenencia del atributo elegido al ejemplo candidato a estar en el nodo hijo. (incrementa o decrementa su valor con un paso de 0.1, tiene como valor mínimo 0 y como máximo 1)
12. Permite seleccionar el peso que van a tener los ejemplos; las variantes son:
    a. El máximo grado de pertenencia de ese ejemplo a los grupos de la colección. (*Max Membership to the Cluster*)
    b. La varianza de los grados de pertenencia de ese ejemplo a los grupos de la colección (*Variance to the Cluster*).

13. Permite seleccionar uno de los criterios de parada de la construcción del árbol de decisión borroso; las variantes son:
    a. Cuando todos los atributos ya han sido usados en el camino actual (*All Attributes have been used*).
    b. Cuando ya todos los ejemplo pertenezcan a la misma clase (*All Examples belong to the same class*).
    c. Cuando el valor de la ganancia de la información del atributo "ganador" esté por debajo de un umbral definido en 14 (*By Threshold of Gain*).
14. Es un incrementador para definir el umbral del valor de la ganancia de la información del atributo "ganador".

Al ser muy grande el árbol de decisión construido, las reglas que se generen de éste también tendrán un gran tamaño, por lo que se le da la posibilidad al usuario que elija que parte de las reglas quiere ver, o sea, definir a partir de que término lingüístico correspondiente a los atributos que están en el antecedente de la regla quiere que se muestren. Esto no cambia el sentido de la regla, solamente muestra los atributos de la regla que tengan un término lingüístico mayor que el especificado.

15. Permite seleccionar a partir de qué término lingüístico quiere que se muestren las reglas (la cantidad de variantes a escoger está en dependencia de la cantidad de intervalos en los que se particionaron los atributos).

3.1.5 ¿Cómo extraer las palabras claves que caracterizan los grupos textuales homogéneos en CorpusMiner?

CorpusMiner cuenta con tres métodos para la extracción de las palabras claves a partir de una colección dura de grupos de documentos y dos métodos para la colección borrosa, estos son: extraer palabras claves a partir de su relevancia (común para los dos tipos de colecciones), por reducción de dimensionalidad (para el agrupamiento duro) y a partir de ID3 variante dura y borrosa para las colecciones duras y borrosas respectivamente.

La selección del algoritmo a utilizar se realiza a través de un diálogo con la lista de todos los métodos (observe la figura 3.8).

Figura 3.8 Seleccionar el método para la selección de las palabras claves.

1. Permite la selección del algoritmo para la extracción de palabra claves.

A continuación se describirá como extraer las palabras claves a partir del algoritmo ID3 variante borrosa, los otros métodos son explicados en (Valdés, 2005).

Extraer palabras claves a partir de ID3 variante borrosa (Fuzzy-ID3 Keyword Finder): Esta opción posibilita obtener, a partir de las reglas generadas por el ID3, aquellas palabras que formen parte de antecedentes de las reglas asociadas a términos lingüísticos que correspondan a uno de los *n* mayores términos de su variable lingüística correspondiente, y se le asignan al grupo que identifica el consecuente de la regla, donde *n* es un valor que puede ser dado al sistema. Si no se desea especificar esta cantidad, entonces se toman todas las palabras que aparecen en las reglas independientemente de su valor. CorpusMiner permite además, obtener las palabras claves interceptando la selección de palabras claves utilizando las reglas generadas por el ID3 y las palabras claves obtenidas a partir de la relevancia a partir de la especificación de un umbral.

El sistema muestra un diálogo que le permite al usuario la entrada de los parámetros relacionados con la extracción de palabras claves utilizando las reglas generadas por el ID3. En primer lugar, contiene una lista desplegable para la especificación del discretizador a utilizar antes de aplicar el ID3. Muestra además, dos casillas de verificación. La primera determina si se entrará la cantidad de mejores valores a tomar por cada palabra y en caso que se active se habilita un incrementador para la entrada de este valor. La segunda lista desplegable define si se interceptarán los resultados de este método con los obtenidos

utilizando la relevancia de las palabras, si se activa, se habilita una línea de entrada de texto para especificar el umbral que se utilizará en la extracción de palabras claves según la relevancia (observe la figura 3.9).

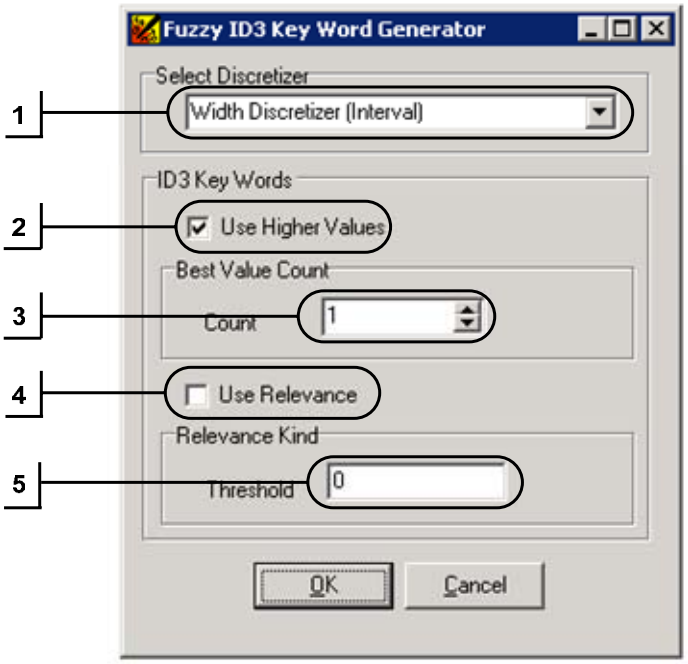


Figura 3.9 Especificación de parámetros para la selección de palabras claves utilizando el ID3.

1. Permite seleccionar el discretizador a emplear para la generación de las reglas utilizando ID3.
2. Determina si se desea introducir una cantidad de mejores valores para la selección de las palabras de las reglas, si está activado se habilita el incrementador 3.
3. Incrementador para la entrada de la cantidad de mejores intervalos a tomar de las reglas.
4. Determina si se interceptarán los resultados con los generados por la relevancia de las palabras utilizando un umbral superior, si se activa se habilita la línea de entrada de texto 5.
5. Línea de entrada de texto para introducir el umbral superior que se utilizará en la generación de las palabras claves según su relevancia.

Al elegir los parámetros que necesita el algoritmo ID3 variante borrosa, se mostrará una ventana como la de la figura 3.7, lo que este caso carecerá del parámetro 15.

3.1.6 ¿Cuáles son las salidas posibles de este módulo? ¿Cómo obtenerlas?

En esta etapa se tienen dos salidas importantes: las reglas generadas a partir de algoritmo ID3 y las palabras claves obtenidas a partir de los diferentes métodos de extracción de palabras claves implementados en CorpusMiner.

Generación de reglas a partir del ID3

Después que se ejecuta el algoritmo ID3 variantes dura o borrosa se puede ver el resultado de éste. En CorpusMiner se expresa a través de reglas, las que se obtienen recorriendo el árbol de decisión resultante desde la raíz a cada una de las hojas del mismo. Las reglas se pueden ver seleccionando la figura con forma de árbol que se muestra en CorpusMiner (Ver figura 3.10).

Las reglas obtenidas del algoritmo ID3 variante dura tienen el formato siguiente:

Término 1 Value X ^ Término 2 Value Y ^...^ Término N Value Z -> grupo k

Ejemplos de estas reglas se pueden ver en (Valdés, 2005).

Las reglas obtenidas por ID3 variante borrosa son reglas que podrían formar un sistema Sugeno grado cero y tienen forma siguiente:

Término 1 es término lingüístico X ^ Término 2 es término lingüístico Y ^... ^ Término N es término lingüístico Z -> grupo k -> Certidumbre C

Ejemplo de regla generada por CorpusMiner:

```
(GIVE Is 1) ^ (AGREE Is 3) ^ (LETTER Is 3) -> Cluster: 5 -> Certainty:
0.7300000000000233
```

es una regla que significa que si en un documento el término “GIVE” es término lingüístico “1” y los términos “AGREE” y “LETTER” son términos lingüísticos “3” entonces dicho documento pertenece al “cluster 5” con un grado de certidumbre de 0,7300000000000233.

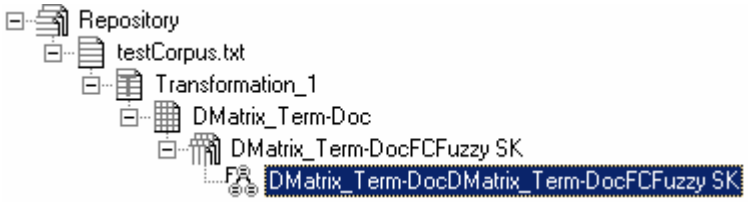


Figura 3.10 Opción que permite mostrar las reglas obtenidas del algoritmo ID3.

En la figura 3.11 se puede observar un ejemplo de reglas obtenidas por el algoritmo ID3 variante borrosa de un corpus de texto con documentos de la agencia de noticias Reuters.

```
(TECH Is 1) ^ (MEAN Is 1) ^ (DOLLAR Is 1)  -> Cluster: 3 -> Certainty: 0.33477170698794
(TECH Is 1) ^ (MEAN Is 1) ^ (DOLLAR Is 2) ^ (INTERVENE Is 1) -> Cluster: 3 -> Certainty: 0.33477170698794
(REFLECT Is 1) ^ (SLIGHTLY Is 1) ^ (HISTORICALLY Is 3)  -> Cluster: 18 -> Certainty: 0.909929825164689
(REFLECT Is 1) ^ (SLIGHTLY Is 2)  -> Cluster: 18 -> Certainty: 0.909929825164689
(REDUCE Is 2) ^ (DEPENDENT Is 1)  -> Cluster: 19 -> Certainty: 0.92461843878485
(REDUCE Is 2) ^ (DEPENDENT Is 2)  -> Cluster: 19 -> Certainty: 0.92461843878485
(BANKER Is 1) ^ (PRESENCE Is 3)  -> Cluster: 2 -> Certainty: 0.884556040117709
(BANKER Is 2)  -> Cluster: 2 -> Certainty: 0.884556040117709
```

Figura 3.11 Fragmento de reglas generadas por el algoritmo ID3 variante borrosa.

Extracción de palabras claves

La forma en que se muestran las palabras claves extraídas es común para todos los métodos. Para cada clase se pone la lista de sus palabras claves con su valor (en el caso de la variante borrosa del ID3 sería el término lingüístico). Mostrándose de la forma siguiente:

Clase 1

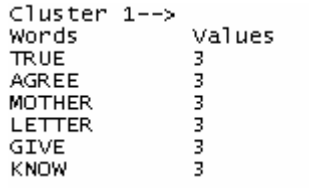
Palabras	Valor
Palabra 1	X
…	…
Palabra N	Y

……

Clase N

Palabras	Valor
Palabra 1	X
…	…
Palabra N	Y

En la figura 3.12 se pueden observar palabras obtenidas por el método de extracción de palabras claves a través del ID3 variante borrosa.



```
Cluster 1-->
Words          Values
TRUE           3
AGREE          3
MOTHER         3
LETTER         3
GIVE           3
KNOW           3
```

Figura 3.12 Fragmento de palabras claves.

3.2 Evaluación

La evaluación incluye la verificación y validación del modelo. Se debe verificar que el sistema esta correctamente construido y que efectivamente es el producto que satisface los requerimientos. Verificar y validar son labores arduas en el campo de la minería de textos. A continuación se mostrará la evaluación realizada del modelo propuesto, donde se tienen en consideración algoritmos reportados en la literatura y el criterio de expertos. Para ello se detallan los casos de estudio diseñados, así como los principales resultados de la verificación y validación, particularizando en la variante borrosa del algoritmo ID3 para la extracción de palabras claves en grupos homogéneos de documentos afines.

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

Entre los corpus textuales publicados en Internet que se referencian en los artículos para evaluar algoritmos en el área 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]

[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

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, trabajar con solo dos clases para evaluar el modelo propuesto no es aconsejable. 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; sin embargo, tiene el problema que los textos tienen mucha información no útil y el formato en que se presentan 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 como su nombre lo indican 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 135 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.

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

A partir de esta colección de la agencia Reuters de noticias, se conformaron cuatro corpus textuales con un tamaño promedio de 135 KB, con 67 documentos incluidos como promedio que abordan 32 tópicos aproximadamente, excepto el cuarto corpus que posee

[4] http://www.daviddlewis.com/resources/testcollections/reuters21578

sólo cuatro tópicos. Nótese que el número elevado de tópicos se debe a que las noticias están multiclasificadas en los tres primeros corpus.

Una descripción detallada de cada corpus se presenta a continuación:

Corpus 1: Tiene un tamaño de 130KB. Posee 59 documentos, previamente etiquetados. Estos documentos abordan 20 tópicos. Obsérvese anexo 20.

Corpus 2: Tiene un tamaño de 132KB. Posee 79 documentos, previamente etiquetados. Estos documentos abordan 29 tópicos. Obsérvese anexo 21.

Corpus 3: Tiene un tamaño de 190KB. Posee 96 documentos, previamente etiquetados. Estos documentos abordan 43 tópicos. Obsérvese anexo 22.

Corpus 4: Tiene un tamaño de 82KB. Posee 31 documentos, previamente etiquetados, éste corpus tiene una clasificación simple. Estos documentos abordan 4 tópicos. Obsérvese anexo 23.

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

Este caso de estudio considera un corpus textual que se construyó intencionalmente por expertos lingüistas para validar agrupamiento y extracción de palabras claves.

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.

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 35 documentos (correspondientes a las 540 palabras más frecuentes del corpus original) y ocupa 2.78 MB. Obsérvese en el anexo 24 fragmentos de documentos del corpus de textos asociados a palabras.

3.2.2 Verificación de los resultados

Verificar los resultados pretende asegurarse que el sistema sea consistente y correcto en cuanto a sintaxis. De todas las etapas que componen el procedimiento general que soporta el modelo, se ha particularizado en la verificación de la extracción de palabras claves siguiendo el algoritmo ID3 en su variante borrosa. La estrategia que se ha seguido es comparar los resultados obtenidos con el ID3 borroso con implementaciones de variantes duras discreta y continua disponibles en WEKA[5] –software que contiene una extensa colección de algoritmos de máquinas de conocimiento, desarrollado por la universidad de Waikato (Nueva Zelanda)–. Además, se comparan los resultados con los obtenidos por el algoritmo ID3 en su variante dura implementado en CorpusMiner (Valdés, 2005):

- Algoritmo ID3, implementado en WEKA.
- Algoritmo C45 (J48), implementado en WEKA.
- Algoritmo ID3 (variante dura), implementado en CorpusMiner.

Tanto para la aplicación de una u otra variante para la inducción de árboles de decisión para la extracción de palabras claves que caractericen los grupos homogéneos de documentos afines, fue necesario preprocesar los corpus textuales. Los experimentos realizados en esta investigación incluyeron en la transformación del corpus las operaciones siguientes: 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

[5] http://www.cs.waikato.ac.nz/ml/weka/

símbolos, la lematización y verificación de la homogeneidad ortográfica. 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. La reducción de la dimensionalidad se realizó a partir de la eliminación de las palabras gramaticales y la selección de aquellos 600 mejores términos, es decir, que tengan una calidad superior a determinado umbral dada la medida de calidad de términos aplicada (Valdés, 2005) (Mederos, 2005). Se aplicó para el agrupamiento el método concatenado *Extended Star – Fuzzy* SKWIC, que permitió la inicialización con *Extended Star* y considerando el método externo *Fuzzy* SKWIC.

Para el uso de los algoritmos implementados en WEKA, previamente se conformaron los archivos tipo *arff* –denominación del formato del fichero nativo del Weka, acrónimo de Attribute-Relation File Format– con la información proveniente de las etapas de construcción de la representación VSM y la clasificación de los ejemplos realizadas por CorpusMiner, utilizándose para la última un agrupamiento borroso seleccionando las clases correspondientes a cada ejemplo aplicando el principio de Máxima Membresía, ver subepígrafe 2.2.4.2.

Después de haber construido los árboles de decisión correspondientes a cada método, se extrajeron las palabras claves que forman parte de los antecedentes de las reglas extraídas de cada grupo interceptadas con aquellos términos que tienen una alta relevancia por grupo (relevancia que supera un umbral determinado). De esta forma, se obtienen por grupos aquellas palabras relevantes y que logran discernir entre ellos.

Obsérvese en los anexos del 24 al 28 las tablas donde se reflejan las coincidencias en las palabras obtenidas para cada uno de estos métodos. Puede observarse que las coincidencias son altas, por lo que se ha comprobado que el sistema es consistente y correcto, aunque por supuesto no idéntico, porque los algoritmos con los que se ha comparado son ID3 en su variante dura discreta y dura continua. Además, hubo que utilizar el principio de máxima membresía para seleccionar los documentos que pertenecen a cada cluster y distorsiona un poco el resultado del agrupamiento.

3.2.3 Validación de los resultados

La validación consiste en asegurarse que el sistema hace lo que se supone que debe hacer, cumpliendo las especificaciones, o sea, que su semántica se correcta. En el área de la minería de textos es realmente difícil validar los resultados. Es por eso que en este trabajo se presentan las tablas donde se muestran las palabras claves seleccionadas en cada uno de los clusters obtenidos para cada uno de los cinco corpus textuales, realizando la selección de rasgos a partir de la inducción de los árboles de decisión borrosos con la variante borrosa del algoritmo ID3.

Obsérvese en los anexos del 30 al 34 el listado de palabras claves seleccionado para cada grupo de los cinco corpus textuales. Nótese que estas palabras tienen relación con los principales tópicos que abordan estos grupos homogéneos de documentos, por tanto, la semántica de la selección es correcta.

3.3 Conclusiones parciales

- La interfaz de usuario que presenta el módulo que incluye la implementación del procedimiento general que soporta el modelo propuesto es amigable y ha sido descrita en detalles.
- Los casos de estudio definidos permitieron demostrar la factibilidad del modelo propuesto y su procedimiento general.
- La verificación de la implementación del algoritmo ID3 en su variante borrosa se realizó a partir de la comparación de los resultados de la extracción de palabras claves de grupos homogéneos de documentos afines según el algoritmo ID3 en su variante borrosa con las palabras extraídas después de aplicar los algoritmos ID3 duro y para datos discretos implementado en WEKA, algoritmo C45 (J48) duro y para datos continuos implementado en WEKA y algoritmo ID3 variante dura, implementado en CorpusMiner. Para cada una de las variantes existen muchas coincidencias en las palabras claves obtenidas, por lo que el algoritmo ID3 en su variante borrosa está implementado de una manera correcta.

- Los resultados de la validación arrojan que las palabras seleccionadas tienen relación con los principales tópicos que abordan estos grupos homogéneos de documentos, por tanto, la semántica de la selección es correcta.

Conclusiones

Como resultado de esta investigación se desarrolló un modelo y un procedimiento soportado por un módulo del software CorpusMiner, que ofrece la posibilita de la aplicación de técnicas de selección de rasgos para la extracción de términos relevantes que caractericen los grupos de documentos afines, cumpliéndose de esta forma el objetivo general planteado, ya que:

- Existe una creciente base teórica conceptual sobre definiciones de relevancia, selección de rasgos, técnicas para la selección de rasgos y selección de rasgos en la minería de textos. Sin embargo, los principales trabajos reportados en la literatura están focalizados al desarrollo de algoritmos para la selección de rasgos en la etapa de representación textual, y no así, al desarrollo y aplicación de técnicas de selección en otras etapas del procesamiento textual, por ejemplo en la selección de palabras claves de grupos homogéneos de documentos. Por tal motivo, el problema científico formulado para la presente investigación se considera de gran actualidad y pertinencia.
- El análisis crítico sobre el estado actual de las técnicas de selección de rasgos, para la extracción de palabras claves de grupos de documentos afines a partir de un agrupamiento duro o borroso, arrojó que es necesaria la utilización de la inducción de árboles de decisión duros y borrosos respectivamente. El estudio de la literatura sobre métodos automáticos de construcción de funciones de pertenencia permitió la selección de un método analítico para la construcción y estimación automáticas de las funciones de pertenencia campanas Beta y triángulo, necesarias en la construcción de las variables lingüísticas asociadas a rasgos en los árboles borrosos.
- El diseño del modelo explica y fundamenta un procedimiento general que permite la selección de palabras claves en grupos textuales homogéneos, y la combinación de la relevancia de las palabras obtenida por los métodos de agrupamiento, con la aplicación de la inducción de los árboles de decisión borrosos para lograr que los términos encontrados logren discernir entre clases. 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 general consta de cuatro etapas. Es posible identificar variantes para la construcción automática de funciones de membresía e inducción de árboles de decisión. Este procedimiento fue implementado e incorporado como un módulo en la herramienta CorpusMiner.
- La interfaz de usuario que presenta el módulo que incluye la implementación del procedimiento general que soporta el modelo propuesto es amigable y se adapta al diseño general de CorpusMiner.
- Los casos de estudio definidos permitieron demostrar la factibilidad del modelo propuesto y su procedimiento general. Se verificó la implementación del algoritmo ID3 en su variante borrosa a partir de la comparación de los resultados de la extracción de palabras claves de grupos homogéneos de documentos afines según el algoritmo ID3 en su variante borrosa con las palabras extraídas después de aplicar los algoritmos ID3 duro y para datos discretos implementado en WEKA, algoritmo C45 (J48) duro y para datos continuos implementado en WEKA y algoritmo ID3 variante dura, implementado en CorpusMiner. Los resultados de la validación arrojaron que las palabras seleccionadas tienen relación con los principales tópicos que abordan estos grupos homogéneos de documentos, por tanto, la semántica de la selección es correcta.

Recomendaciones

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

- Incorporar nuevos métodos que permitan la poda antes y después del proceso de inducción de los árboles de decisión.
- Incorporar nuevos métodos de discretización tributando a una mejor calidad de las funciones de pertenencia a obtener.
- Realizar una validación supervisada de las palabras claves extraídas a partir del uso de corpus textuales que incluyan previamente las palabras claves que lo caracterizan.

Referencias bibliográficas

Arco, L (2001), Machine *Learning para la construcción de reglas fuzzy*. Tesis de Grado. Universidad Central "Marta Abreu" de Las Villas.

Arco, L (2005), *Modelo para el agrupamiento de documentos afines y su ulterior resumen a través de la representación espacio vectorial de un corpus textual*. Tesis de Maestría. Universidad Central "Marta Abreu" de Las Villas.

Bello, P.R. et al., (2005a) *A model based on Ant Colony System and Rough Set Theory to Feature Selection*. "Genetic and Evolutionary Conference (GECCO05)". June 25-29. Washington, USA.

Bello, P.R. et al., (2005b) *Using ACO and Rough Set Theory to Feature Selection*. 6th WSEAS Evolutionary Computing Conference (EC05). June16-18. Lisbon, Portugal.

Berry, M., (2004) *Survey of Text Mining. Clustering, Classification, and Retrieval*. Springer-Verlag. ISBN 0-387-95563-1.

Blum, A. L. Langley, P., (1997) *Selection of Relevant Features and Examples in Machine Learning*.

Botzheim, J., (2001) *Extracting Trapezoidal Membership Functions of a Fuzzy Rule System by Bacterial Algorithm*. Proceeding in LNCS Computational Intelligence.

Buckley J. J., Eslami E., (2002) *An Introduction to Fuzzy Logic and Fuzzy Sets.* pp 31-35

Caruana, R. Freitag, D., (1994) How *Useful Is Relevance?* AAAI Fall Symposium on Relevance, New Orleans, Louisiana.

Chen, J.E. Otto, K. N., (1995) *Constructing membership functions using interpolation and measurement theory*. Fuzzy Sets and systems 73. Pp. 313-327.

Choubey, S.K. et al., (1996) A *comparison of feature selection algorithms in the context of rough classifiers*. In Proceedings of Fifth IEEE International Conference on Fuzzy Systems, vol. 2, pp. 1122-1128.

Deogun, J.S. et al., (1998) *Feature selection and effective classifiers*. Journal of ASIS 49, 5, pp. 423-434.

Dixon, M., (1997) *An Overview of Document Mining Technology*, disponible en: www.geocities.com/ResearchTriangle/Thinktank/1997/mark/writings/dixm97_dm.ps

Duda, R.O. Hart, P.E. Pattern, (1973) *Classification and Scene Analysis*. New York: Wiley-Interscience.

Dunstsh, Ivo and Gunter, Gediga., (2000) *Rough set data analysis*, Disponible en: http://citeseer.nj.nec.com/dntsch00rough.html.

Dürsteler, J.C., (2001) *Minería de Textos*. Inf@Vis! La revista digital de InfoVis.net. Mensaje No. 27.

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

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

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

Gil-García, R. Badía-Contelles, J.M. Pons-Porrata, A., (2003) *Extended Star Clustering Algorithm*. Proceedings of CIARP.

Hong, T. Lee, C.Y., (1998) *Learning Fuzzy Knowledge from Training Examples*. CIKM. pp 161-166.

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

Hussain, F. Liu, H., (1999) *Discretization: An Enabling Technique*. TRC6/99., disponible en: http://www.comp.nus.edu.sg/~liuh

Inza, I.; Larrañaga,P. y B. Sierra, (2001) *Feature Subset Selection by Bayesian networks: a comparison with genetic and sequential algorithms*. International Journal of Approximate Reasoning, 27/2, 143-164.

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

Janikow, C. Z., (1996) *Exemplar Learning in Fuzzy Decision Trees* Mathematics and Computer Science University of Missouri

Jensen, Richard and Qiang Shen., (2003) *Finding rough sets reducts with Ant colony optimization*. Disponible en: http://www.inf.ed.ac.uk/publications/online/0201.pdf

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

Jyh-Shing, R. et. al., (1998) *Neuro-Fuzzy and Soft Computing*. Prentice Hall

Klir G. J. and T. A. Folger,, (1992) *Fuzzy Sets, Uncertainty, and Information*, pp. 4-14

Kohavi, R. and Frasca, B., (1994) *Useful feature subsets and Rough set Reducts*. Proceedings of the Third International Workshop on Rough Sets and Soft Computing.

Kurgan L.; et. al, (2004): *CAIM Discretization Algorithm.* IEEE Transactions on Knowledge and Data Engineering. Vol 16. No. 2

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

Lewis, D.D., (1992) *Representation and Learning in Information Retrieval*. Ph. D. thesis, Department of Computer and Information Science, University of Massachusetts.

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

Lezcano, R.D., (2002) *Minería de Datos*., disponible en: http://www.google.com.cu/depar/areas/informatica/SistemasOperativos/MineriaDatosLezcano.pdf

Liu H. and Setiono R., (1997): *Chi^2: Attribute selection and discretization of numeric attributes*. In Proceedings of the IEEE 7th Conference on tools with AI.

Liu, H y H. Motoda., (1998) Feature Selection Boston, MA : Kluwer academic Publishers. Disponible en: http://citeseer.ist.psu.edu/321378.html

Maloney, J., (2004) *Text Mining Solutions*. Services & Solutions.

Marsala, C., (1996) *Fuzzy Partitioning Using Mathematical Morphology in a Learning Scheme* Universit´e Pierre et Marie Curie.

Marsala, C., (1998) *Application of Fuzzy Rule Induction to Data Mining*, In Proc. of the 3rd Int. Conf. FQAS'98, mayo, Roskilde (Denmark), LNAI nr. 1495, pp. 260-271. Disponible en: www-poleia.lip6.fr/~marsala/publications

Marsala, C. Bouchon-Meunier, B., (1999) *An Adaptable System to Construct Fuzzy Decision Trees.* Universit´e Pierre et Marie Curie.

Mederos, J.M., Pérez, Y., (2005) *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.

Ming D., Student Member; IEEE, Ravi Kothari, Senior Member; IEEE, (2001) *Look-Ahead Based Fuzzy Decision Tree Induction*, transactions on fuzzy systems, vol. 9, no. 3, june.

Mitchell, T.(1997) *Machine Learning*. McGraw-Hill Science, Engineering, Math. ISBN 0070428077.

Mladenic, D. Grobelnik, M., (1998) *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).

Narazaki, H. Ralescu, A., (1994) *Iterative Induction of a Category Membership Function*, International Journal of Uncertainty, Fuzziness and Knowledge-based Systems, 2.1, March, pp. 91-100.

Nürnberger, A. Klose, A. Kruse, R., (2001) *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.

Obeso, C., (2001) *Apuntes de éxito*, disponible en www.infonomia.com

Peng, Y. Flach, P. A., (2001) *Soft Discretization to Enhance the Continuous Decision Tree Induction*

Piñero P. P. Y., (2005) *Un modelo para el aprendizaje y la clasificación automática basado en técnicas de softcomputing*. Tesis de Maestría. Universidad de Ciencias Informáticas.

Quinlan, J.R., (1986) *Induction of Decision Trees*. Machine Learning. Pp 81-106.

Quinlan, J.R., (1993) *C4.5: Programs for Machine Learning*. Morgan Kaufman, San Mateo, CA.

Quinlan, J. R., (1996) *Improved Use of Continuous Attributes in C4.5*.Journal of Artificial Intelligence Research 4. Pg 77 – 90.

Rijsbergen, C.J., (1979) *Information Retrieval*. London: Butterworths, disponible en: http://www.dcs.gla.ac.uk/Keith

Ruiz-Shulcloper, J. Alba-Cabrera, E. Lazo-Cortés, M., (1995) *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.

Sahami, M., (1998) *Using Machine Learning to Improve Information Access*. Ph. D. thesis, Department of Computer Science, Stanford University.

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

Salton, G., McGill, M., (1983) *Modern Information Retrieval*. New York: McGraw-Hill.

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

Skalak, D., (1994) *Prototype and feature selection by sampling and random mutation hill_climbing algorithms*. In Proceedings of the Eleventh International Conference on Machine Learning, pages 293-301. New Brunswick. NJ. Morgan Kaufmann.

Sushmita, M. Kishori, M. K. Sankar, K., (2002) *Fuzzy decision tree, linguistic rules and fuzzy knowledge-based network: generation and evaluation* transactions on systems, man, and cybernetics—part c: applications and reviews, vol. 32, no. 4, November

Umano, M. Okamoto, H. Hatono, I. Tamura, H. Kawachi, F. Umedzu, S. Kinoshita, J. , (1994) *Fuzzy decision trees by fuzzy ID3 algorithm and its applications to diagnosis systems*. In Proc. of 3th IEEE International Conference on Fuzzy System, pp 2113-2118, Orlando, FL.

Valdés, L., (2005) *Representación de textos y su reducción de dimensionalidad*. Trabajo de diploma. Universidad Central “Marta Abreu” de Las Villas.

Varela, A. J., (2005), *Estimación automática de parámetros de funciones de pertenencia.* Tesis de Grado. Universidad Central "Marta Abreu" de Las Villas.

Vinterbo, Staal V., (1999) *Predictive Models in Medicine: Some Methods for Construction and Adaptation.* Tesis presentada para obtener el grado de Doctor Ingenior. Asesor: Jan Komorowski. Department of Computer and Information Science, Norwegian University of Science and Technology (NTNU). Trondheim, Diciembre. ISBN 82-7984-011-7. ISSN 0802-6394

Wang, X. Borgelt, C., (2003) *Information Measures in Fuzzy Decision Trees*., disponible en: http://fuzzy.cs.uni-magdeburg.de/~borgelt/papers

Wang, X. B. Chen, G. Qian, and F. Ye, (2000) "On the optimization of fuzzy decision trees," *Fuzzy Sets Syst.*, vol. 112, pp. 117–125.

Wang, X Hong, J., (1998) *On the handling of fuzziness for continuous valued attributes in decision tree generation*. Fuzzy Sets and Systems 99. pp 283-290.

Wroblewski, J., (1995) *Finding minimal reducts using genetic algorithms*. In Wang, P.P. (Ed). Proceedings of the International Workshop on Rough Sets Soft Computing at Second Annual Joint Conference on Information Sciences, North Carolina, USA, p. 679, pp. 186-189.

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

Yao, Y. S. Wong y C. J. Butz., (1999) *Methodologies for Knowledge Discovery and Data Minin*. En: On Information-Theoretic Measures of attribute importante. Zhong y Zhou (Eds.) pp. 231-238. Disponible en:
http://citeseer.ist.psu.edu/yao99informationtheoretic.html

Zadeh, L. A., (1965) *Fuzzy Sets, Information and Control*, 8

Zadeh, Lotfi A., (1994) *The Fuzzy Systems Handbook: a practitioner's guide to building, using, and maintaining fuzzy systems*. Academic Press.

Zeidler, J. Schlosser, M., (1996) *Continuous-Valued Attributes in Fuzzy Decision Trees*

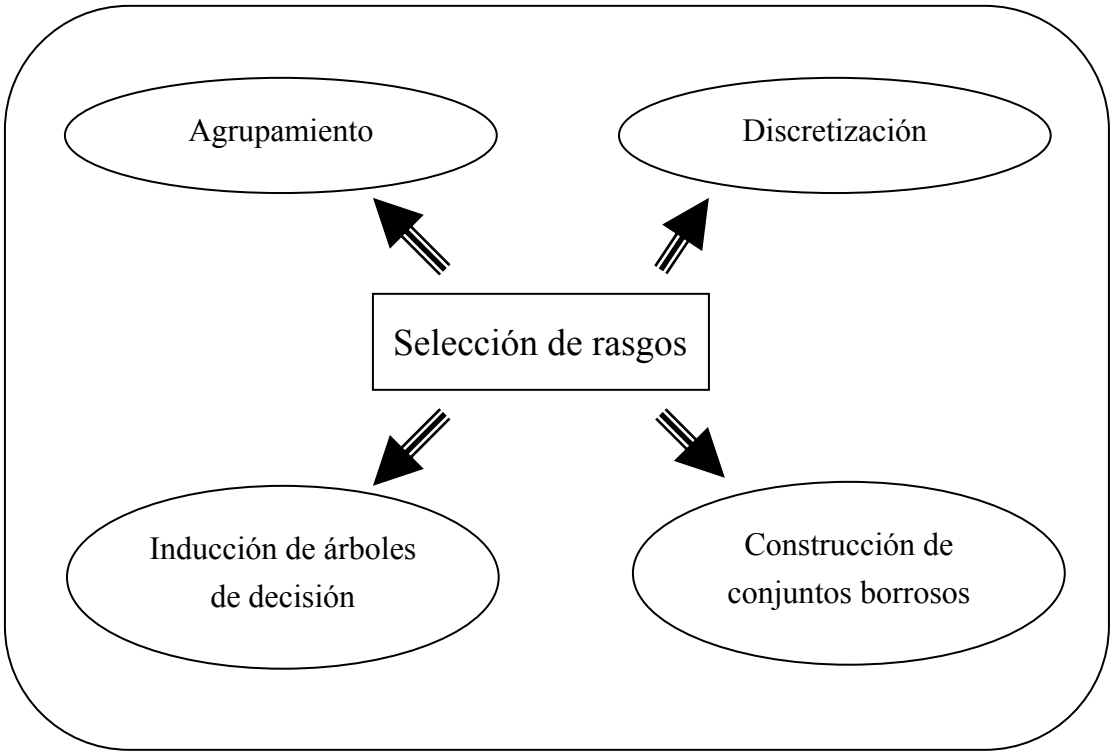
Zhong, N. et al., (2001) *Using Rough sets with heuristics for feature selection*. Journal of Intelligent Information Systems, 16, 199-214.

Zhou, Q. Purvis, M. Kasabov, N., (1997) *A Membership Function Selection Method for Fuzzy Neural Networks* Proceedings of the International Conference on Neural Information Processing and Intelligent Systems, Springer, Singapore, pp. 785-788.

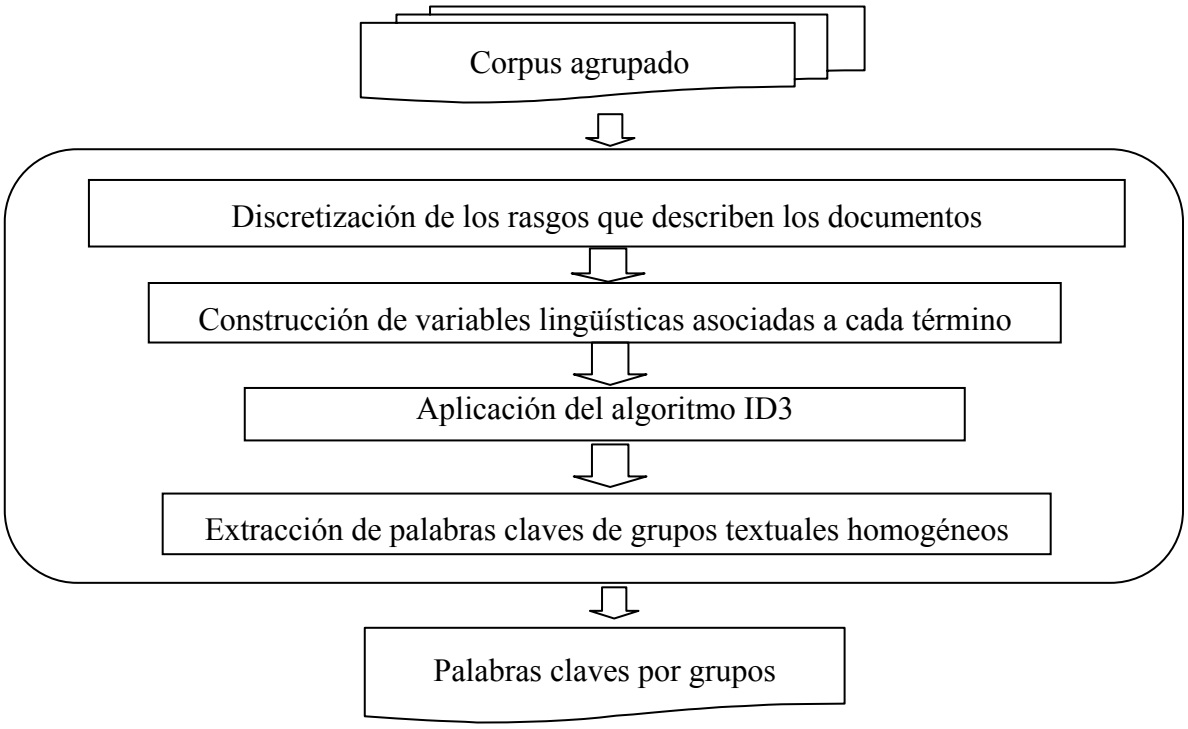
Zipf, G.K., (1949) *Human Behaviour and the Principle of Least Effort*. Reading, MA: Addison-Wesley.

Anexos

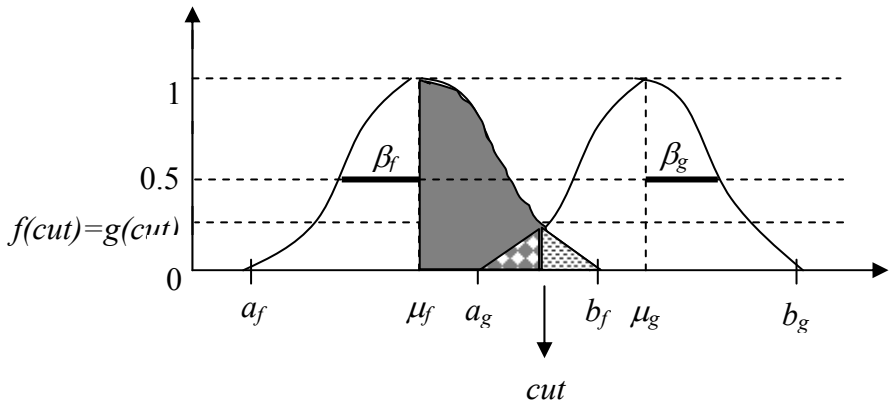
Anexo 1. Modelo propuesto para la selección de palabras claves que logran caracterizar grupos homogéneos de documentos afines.



Anexo 2. Procedimiento general que soporta el modelo propuesto.



Anexo 3. Solapamiento de las funciones de pertenencia campanas Beta en la estimación de parámetros.



val: valor mínimo de imagen que alcanzan los extremos de la función según lo considerado.

a_f : valor del dominio correspondiente al extremo izquierdo de la primera función de pertenencia.

b_f : valor del dominio correspondiente al extremo derecho de la primera función de pertenencia.

a_g : valor del dominio correspondiente al extremo izquierdo de la segunda función de pertenencia.

b_g valor del dominio correspondiente al extremo derecho de la segunda función de pertenencia.

cut: valor del dominio donde se intersectan las funciones *f* y *g*.

β_f: valor de Beta para la función *f*.

β_g : valor de Beta para la función *g*.

μ_f : es la media de la función *f*.

μ_g: es la media de la función *g*.

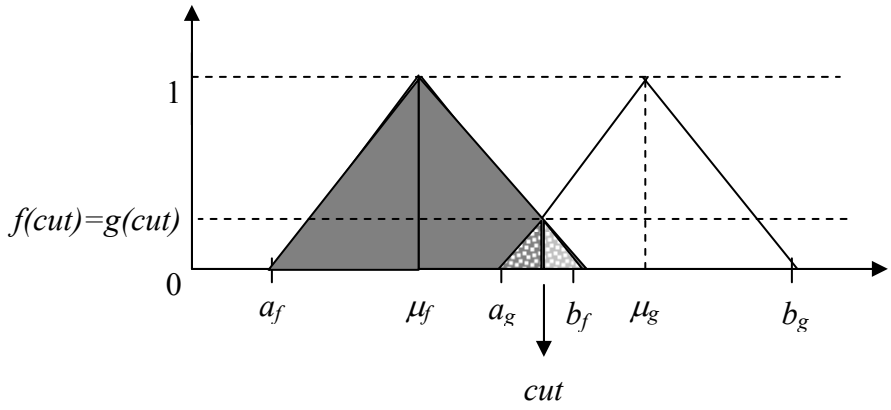
p: porciento de las áreas que se solapa.

A_{s1}: parte izquierda del área que se solapa.

A_{s1}: parte derecha del área que se solapa.

A_f: área debajo de la curva *f*.

Anexo 4. Solapamiento de las funciones de pertenencia triangulares en la estimación de parámetros.



a_f : valor del dominio correspondiente al extremo izquierdo de la primera función de pertenencia.

b_f : valor del dominio correspondiente al extremo derecho de la primera función de pertenencia.

a_g : valor del dominio correspondiente al extremo izquierdo de la segunda función de pertenencia.

b_g : valor del dominio correspondiente al extremo derecho de la segunda función de pertenencia.

cut : valor del dominio donde se intersectan las funciones *f* y *g*.

μ_f : es la media de la función *f*.

μ_g : es la media de la función *g*.

p : porciento de las áreas que se solapa.

α : es el ángulo que se forma opuesto a la recta que va de los puntos (μ_g, 0) y (μ_g, 1).

β : es el ángulo que se forma opuesto a la recta que va de los puntos (μ_f, 0) y (μ_f, 1).

A_s : área que se solapa.

A_f : área del triángulo debajo de la curva *f*.

Anexo 5. Matriz de entrada a los algoritmos ID3 en sus variantes duras y borrosas.

(a) Matriz de entrada al algoritmo ID3 duro.

	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}

Nótese que las filas de la tabla se hacen corresponder con cada documento previamente agrupado, mostrándose los valores de frecuencia normalizados y pesados de cada término que describe el corpus textual.

(b) Matriz de entrada al algoritmo ID3 borroso.

	Tér 1	**Tér 2**	**…**	**Tér *m***	**Grupo**
Doc_1	$tf_{d_1}(t_1)$	$tf_{d_1}(t_2)$		$tf_{d_1}(t_m)$	$(\delta_{\text{Grupo1}}(\text{Doc}_1), \ldots, \delta_{\text{Grupo}k}(\text{Doc}_1))$
Doc_2	$tf_{d_2}(t_1)$	$tf_{d_2}(t_2)$		$tf_{d_2}(t_m)$	$(\delta_{\text{Grupo1}}(\text{Doc}_2), \ldots, \delta_{\text{Grupo}k}(\text{Doc}_2))$
…			…		…
Doc_n	$tf_{d_n}(t_1)$	$tf_{d_n}(t_2)$		$tf_{d_n}(t_m)$	$(\delta_{\text{Grupo1}}(\text{Doc}_2), \ldots, \delta_{\text{Grupo}k}(\text{Doc}_2))$

Nótese que las filas de la tabla se hacen corresponder con cada documento previamente agrupado, donde a cada documento se le asigna su grado de pertenencia a cada grupo obtenido como resultado del agrupamiento, se muestran en cada celda los valores de frecuencia normalizados y pesados de cada término que describe el corpus textual.

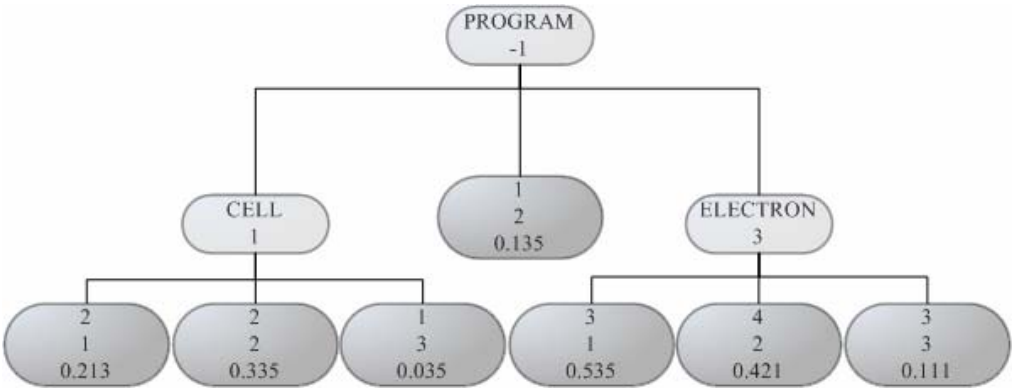
Anexo 6. Tabla de contingencia para un atributo candidato.

A	Cluster 1	…	Cluster m	ASum
a_1	$Z_{C1}^{N\|a1}$	…	$Z_{Cm}^{N\|a1}$	$Z^{N\|a1}$
…	…	…	…	…
a_n	$Z_{C1}^{N\|an}$	…	$Z_{Cm}^{N\|an}$	$Z^{N\|an}$
CSum	Z_{C1}^{N}	…	Z_{Cm}^{N}	Z^{N}

Cada fila de la tabla corresponde a los términos lingüísticos que conforman a la variable lingüística del atributo A y las columnas a los cluster obtenidos por el algoritmo de agrupamiento.

Anexo 7. Generación de reglas a partir de un árbol de decisión borroso.

Dado un árbol de decisión resultante de aplicar el algoritmo ID3 borroso, como el que se muestra en el siguiente ejemplo,



es posible obtener las reglas que describen la colección previamente agrupada. Dichas reglas se obtienen recorriendo el árbol de decisión de la raíz a cada hoja. Cada camino de la raíz a una hoja determina una regla.

Dado el árbol del ejemplo, se obtienen las reglas siguientes. Nótese que son reglas tipo Sugeno Grado 0.

(PROGRAM Is 1) ^ (CELL Is 1) -> Cluster: 2 -> Certainty: 0.213
(PROGRAM Is 1) ^ (CELL Is 2) -> Cluster: 2 -> Certainty: 0.335
(PROGRAM Is 1) ^ (CELL Is 3) -> Cluster: 1 -> Certainty: 0.035
(PROGRAM Is 2) -> Cluster: 1 -> Certainty: 0.135
(PROGRAM Is 3) ^ (ELECTRON Is 1) -> Cluster: 3 -> Certainty: 0.535
(PROGRAM Is 3) ^ (ELECTRON Is 2) -> Cluster: 4 -> Certainty: 0.421
(PROGRAM Is 3) ^ (ELECTRON Is 3) -> Cluster: 3 -> Certainty: 0.111

Anexo 8. Generación de las palabras claves a partir del árbol y reglas del anexo 7.

A partir de las reglas mostradas en el anexo 7 y seleccionando un valor de *n* igual a 1 (i.e., tomando sólo los términos que tienen mayor frecuencia de aparición), se obtienen las palabras claves siguientes:

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

Teniendo en cuenta el ejemplo anterior, cada atributo puede tomar tres valores distintos (1,2,3), por tanto, los *n* mejores valores para *n* igual a 1 es solamente el valor 3. Se toman por cada regla aquellas palabras que tienen valor 3. Como en ninguna regla con consecuente grupo 2 se encontró una palabra con valor 3, la lista de palabras claves de este grupo esta vacía.

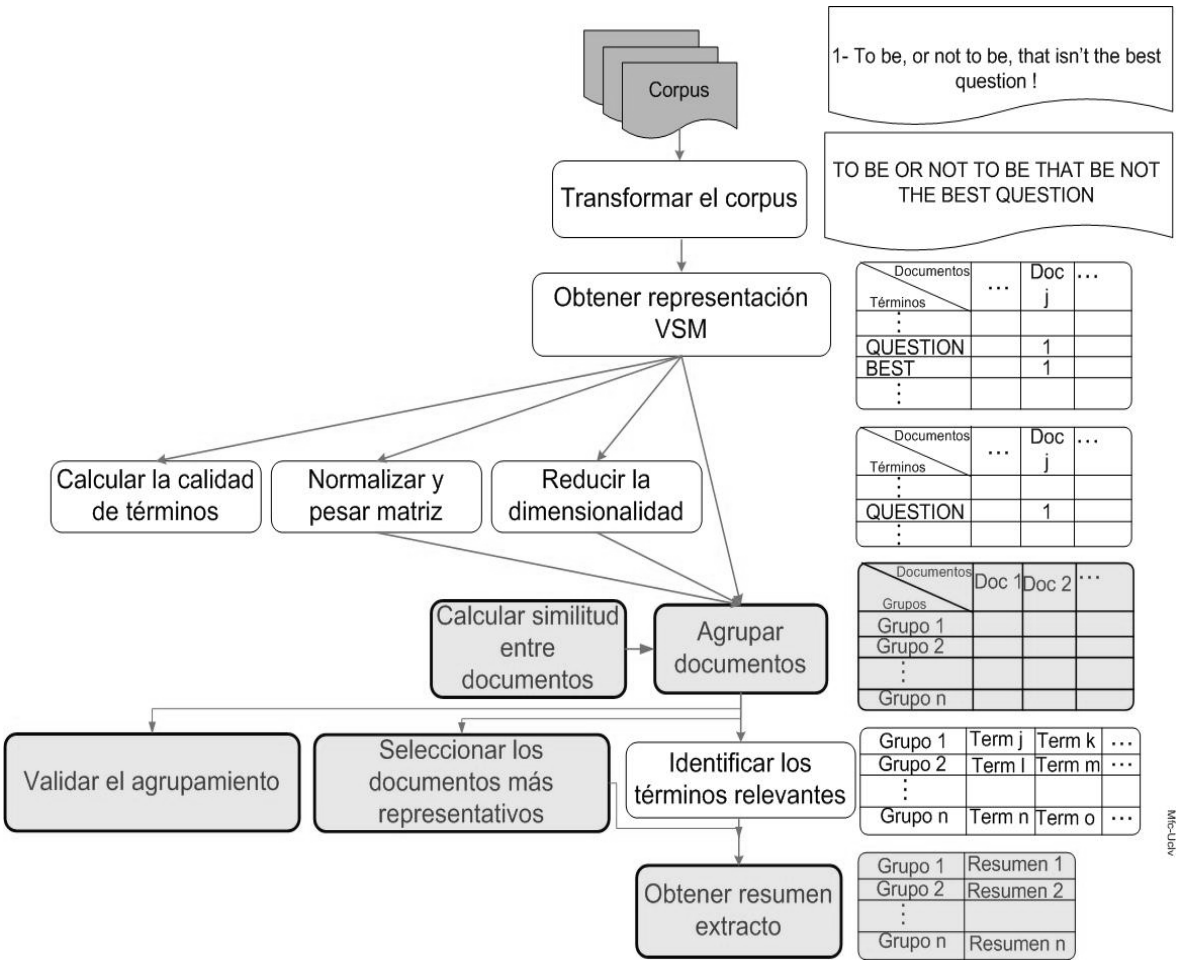
Si en lugar de escoger *n*=1 se escoge *n*=2, los *n* mejores serían los valores 3 y 2, por lo que se seleccionarían de las reglas las listas de palabras claves siguientes:

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

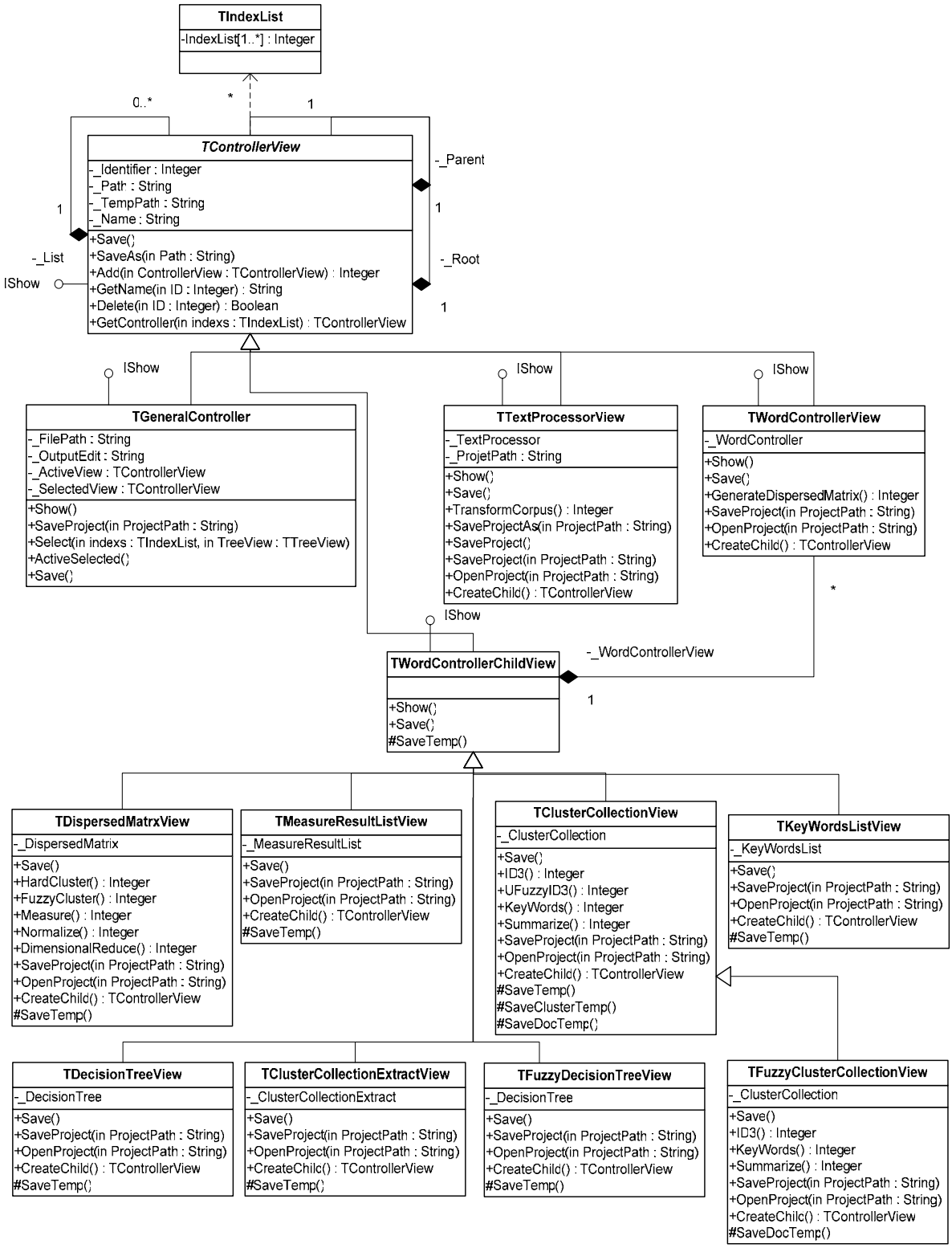
Si no se especifica un *n*, se seleccionan todas las palabras de la regla. Continuando con el mismo ejemplo se tiene:

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

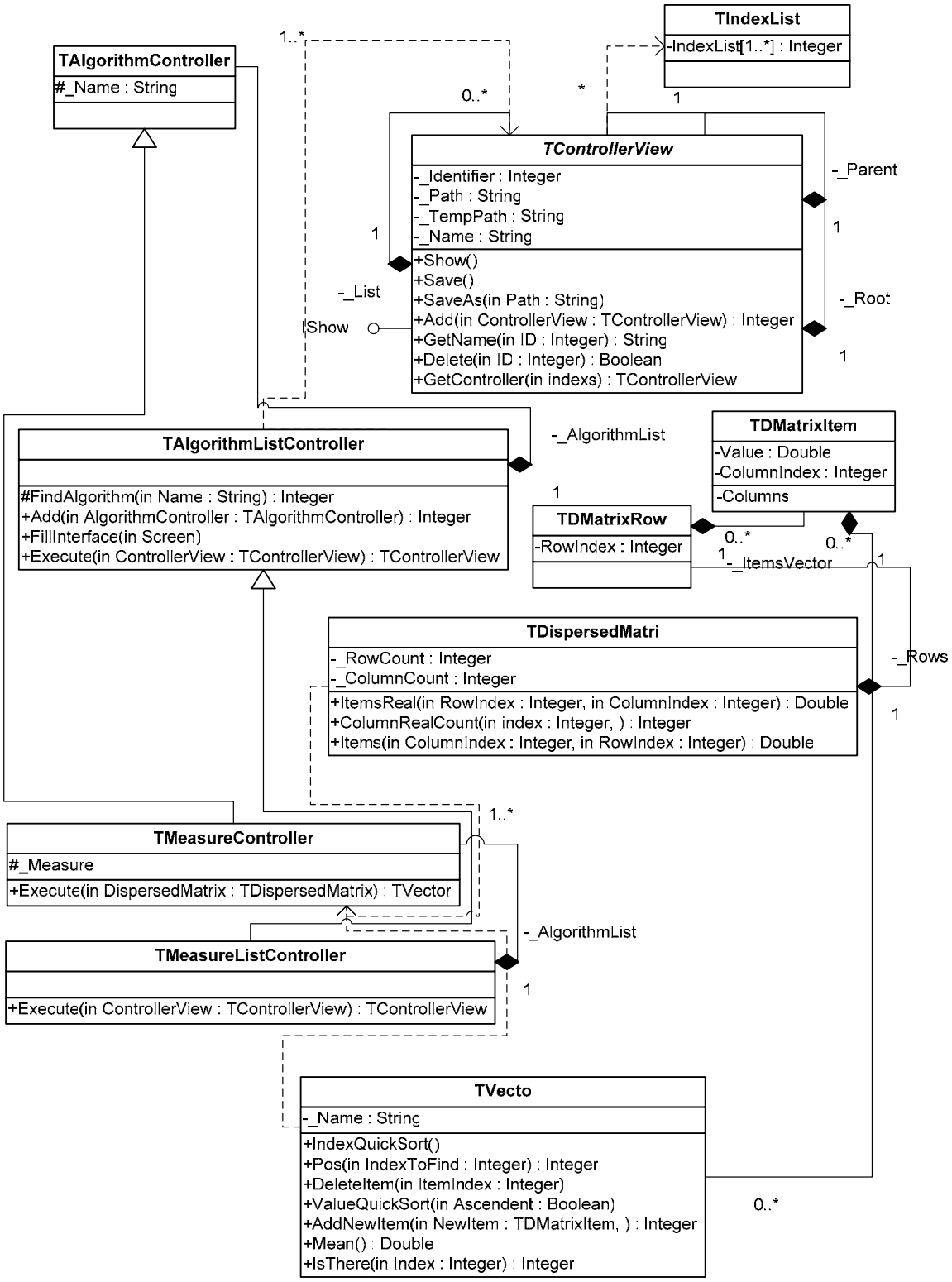
Anexo 9. Etapas en el procesamiento de una colección de textos en CorpusMiner.



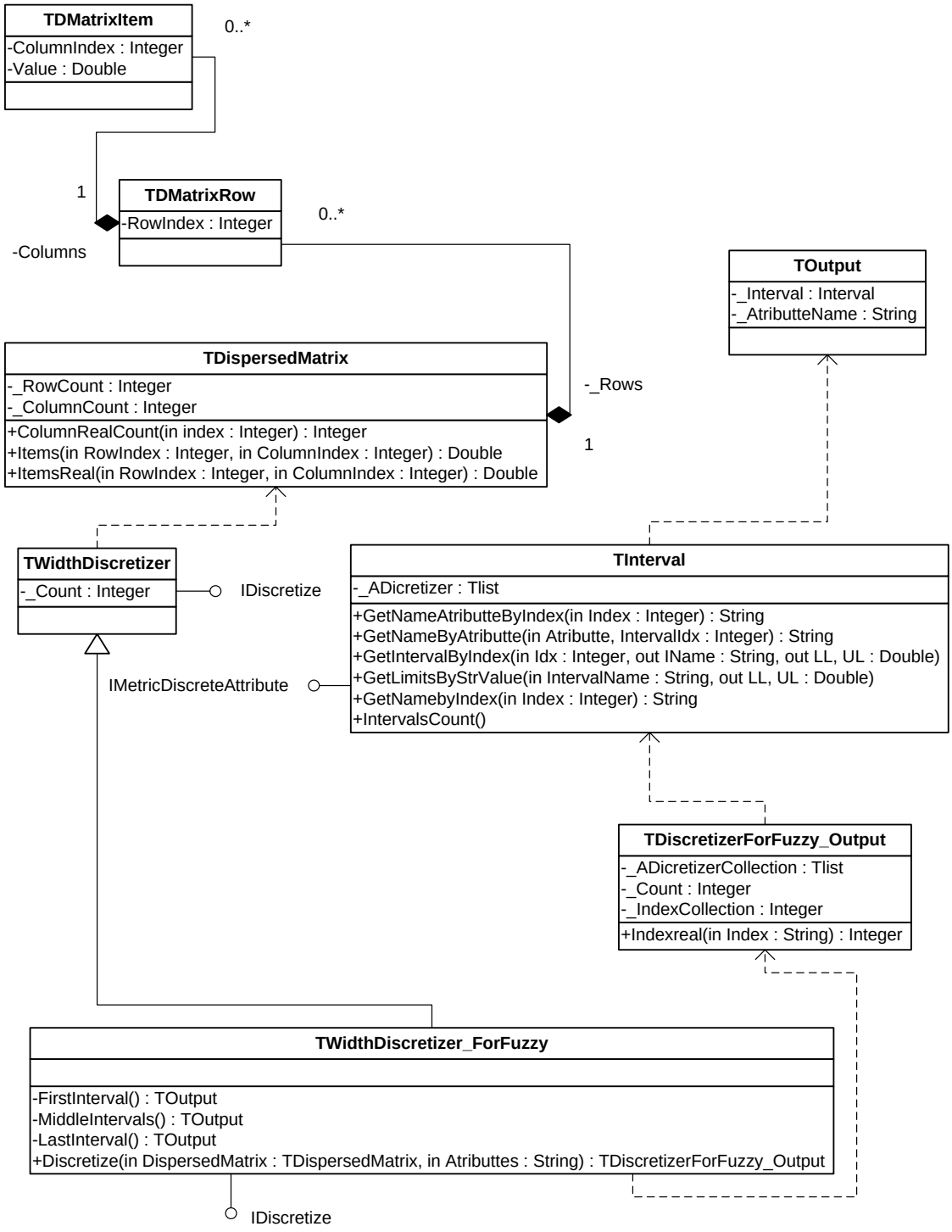
Anexo 10. Diseño general de los controladores en CorpusMiner.



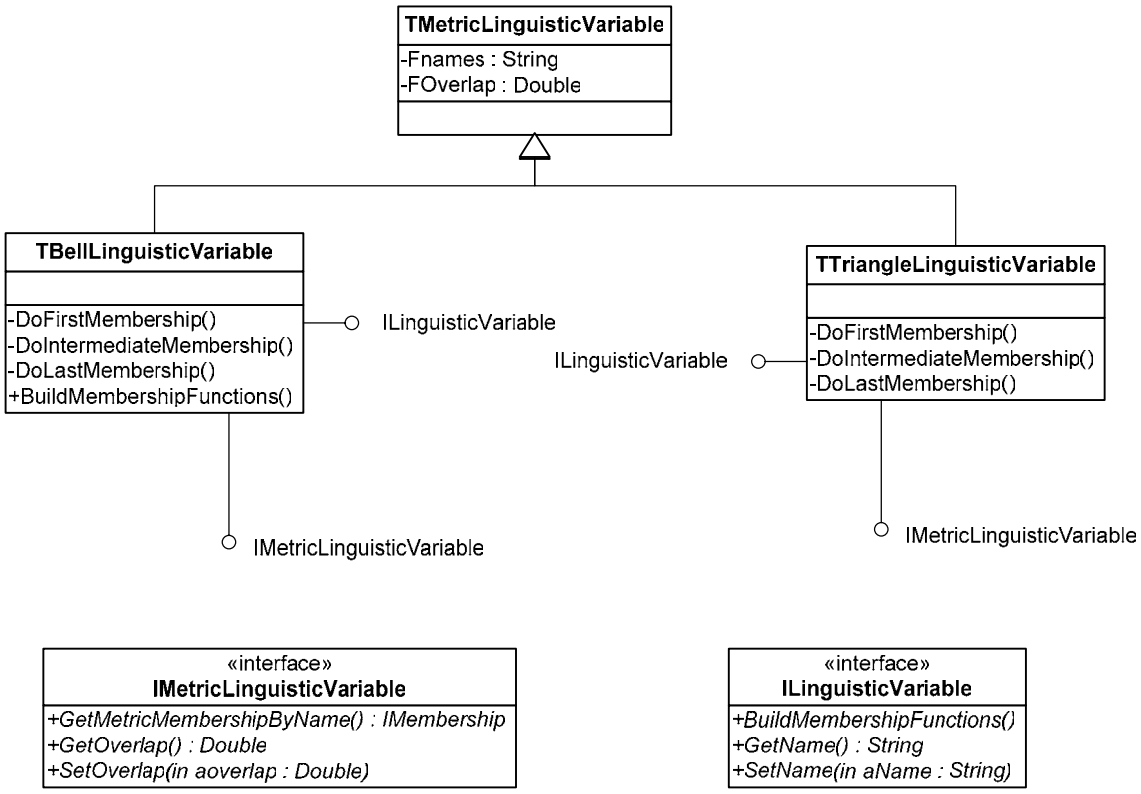
Anexo 11. Diseño de las clases controladoras.



Anexo 12. Diseño de las clases relacionadas con el proceso de discretización de los rasgos.

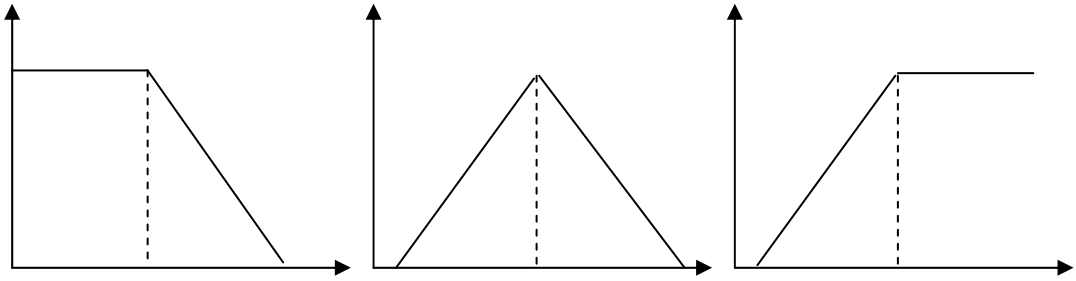


Anexo 13. Diseño de las clases relacionadas con el proceso de construcción de las variables lingüística.



Anexo 14. Funciones de pertenencia asociadas a variables lingüísticas.

(a) Funciones de pertenencia triangulares para una variable lingüística.

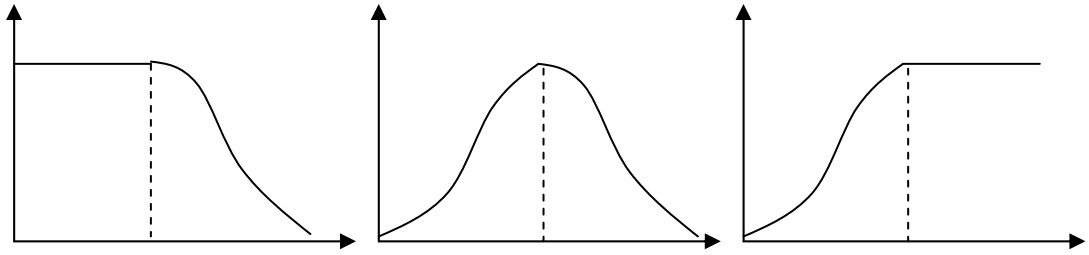


(a) primer intervalo del atributo discreto

(b) intervalos intermedios del atributo

(c) último intervalo del atributo discreto

(b) Funciones de pertenencia campanas Beta para una variable lingüística.

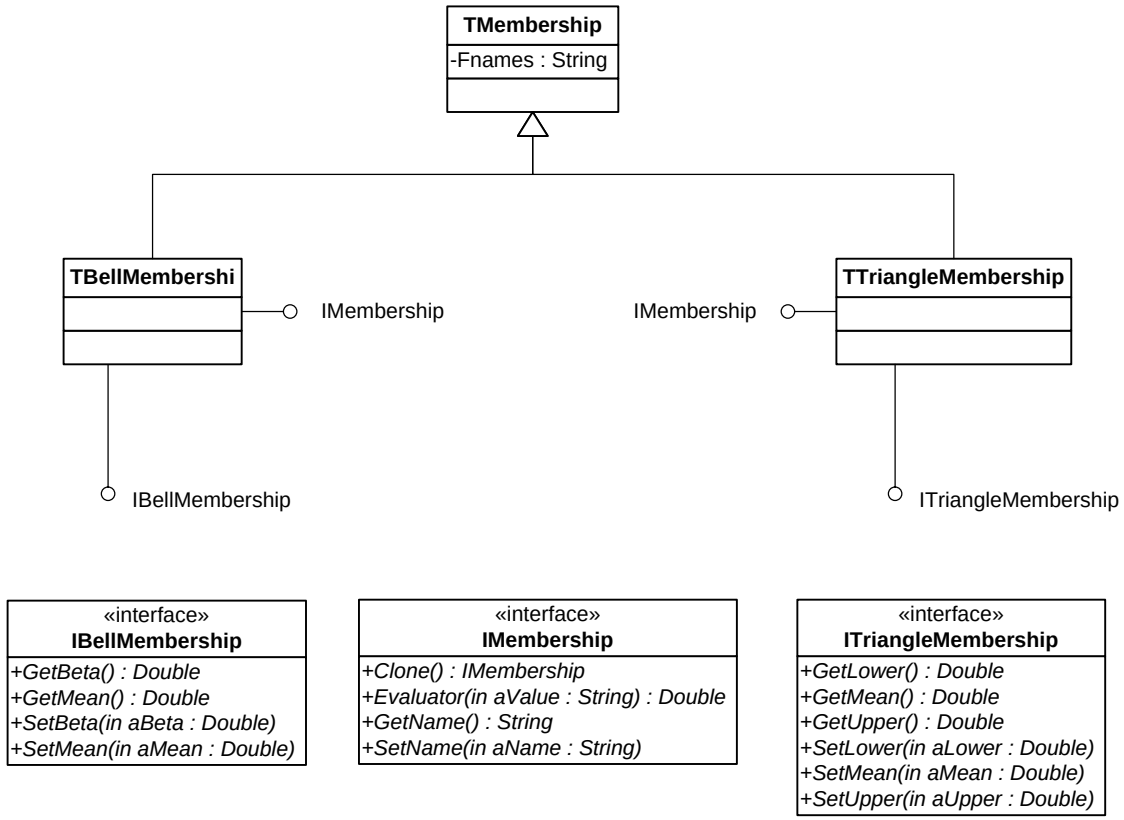


(a) primer intervalo del atributo discreto

(b) intervalos intermedios del atributo

(c) último intervalo del atributo discreto

Anexo 15. Diseño de las clases relacionadas con el proceso de construcción y evaluación de las funciones de pertenencia.

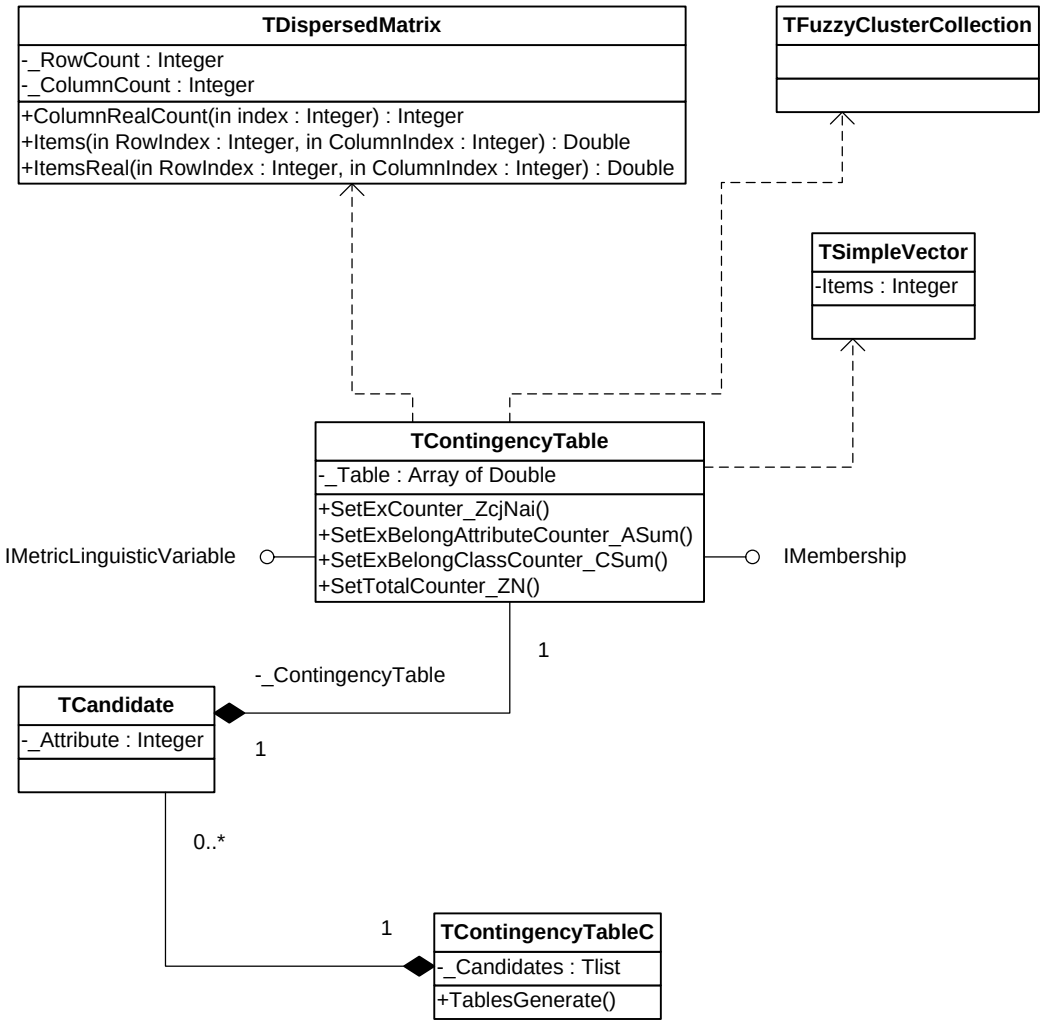


Anexo 16. Uso de las interfaces en la creación y evaluación de las funciones de pertenencia campanas Beta.

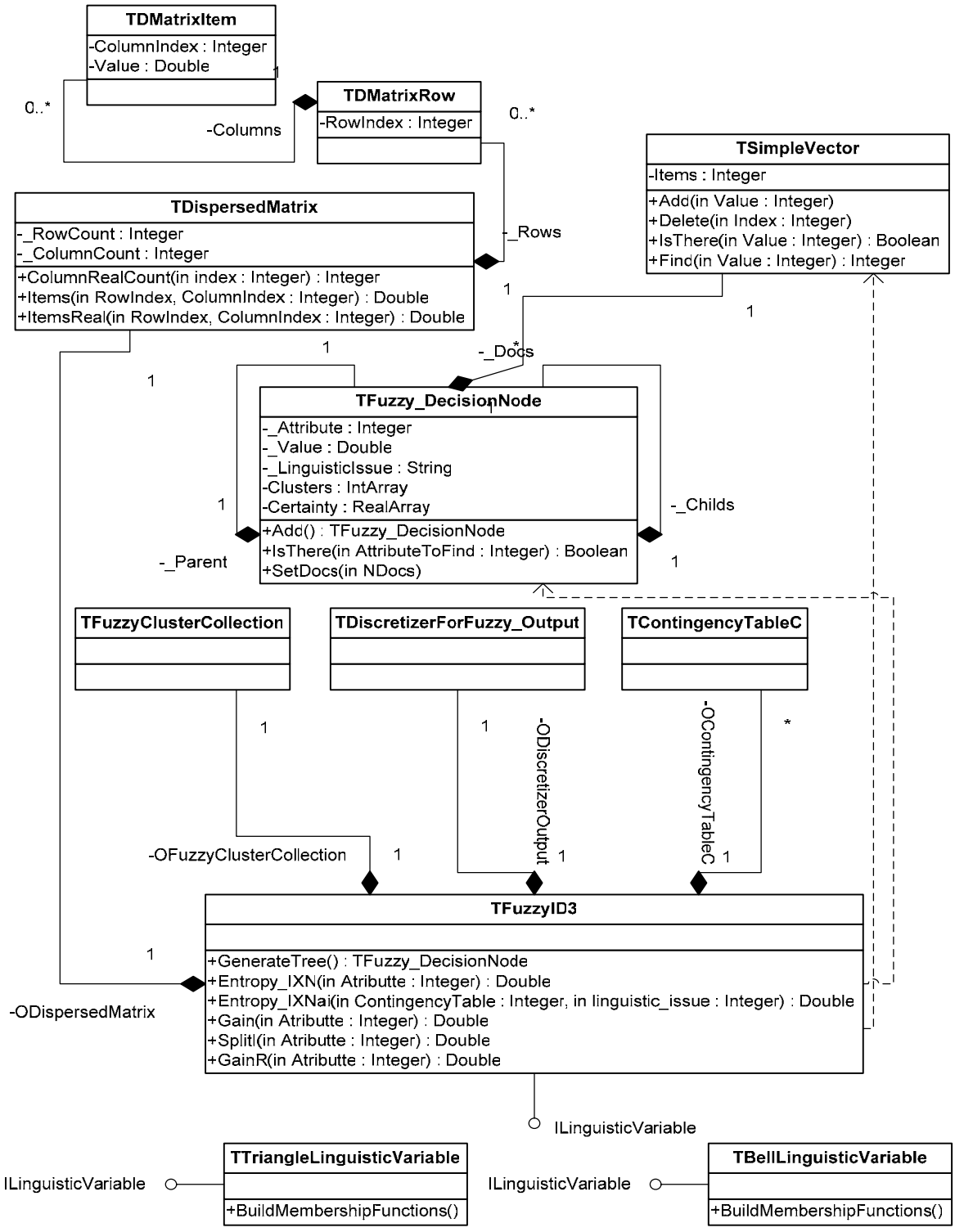
```
var
  MyMembership : IMembership;
begin
  if crear función de pertenencia Beta para el intervalo del límite izquierdo
    then
       MyMembership := TLowLimitBellMembership.Create;
  if crear función de pertenencia Beta para el intervalo del límite derecho
    then
       MyMembership := THighLimitBellMembership.Create;
  if crear función de pertenencia Beta para los intervalos intermedios
    then
       MyMembership := TBellMembership.Create;

  MembershipValue := MyMembership.Evaluator(MyDomainValue);
end;
```

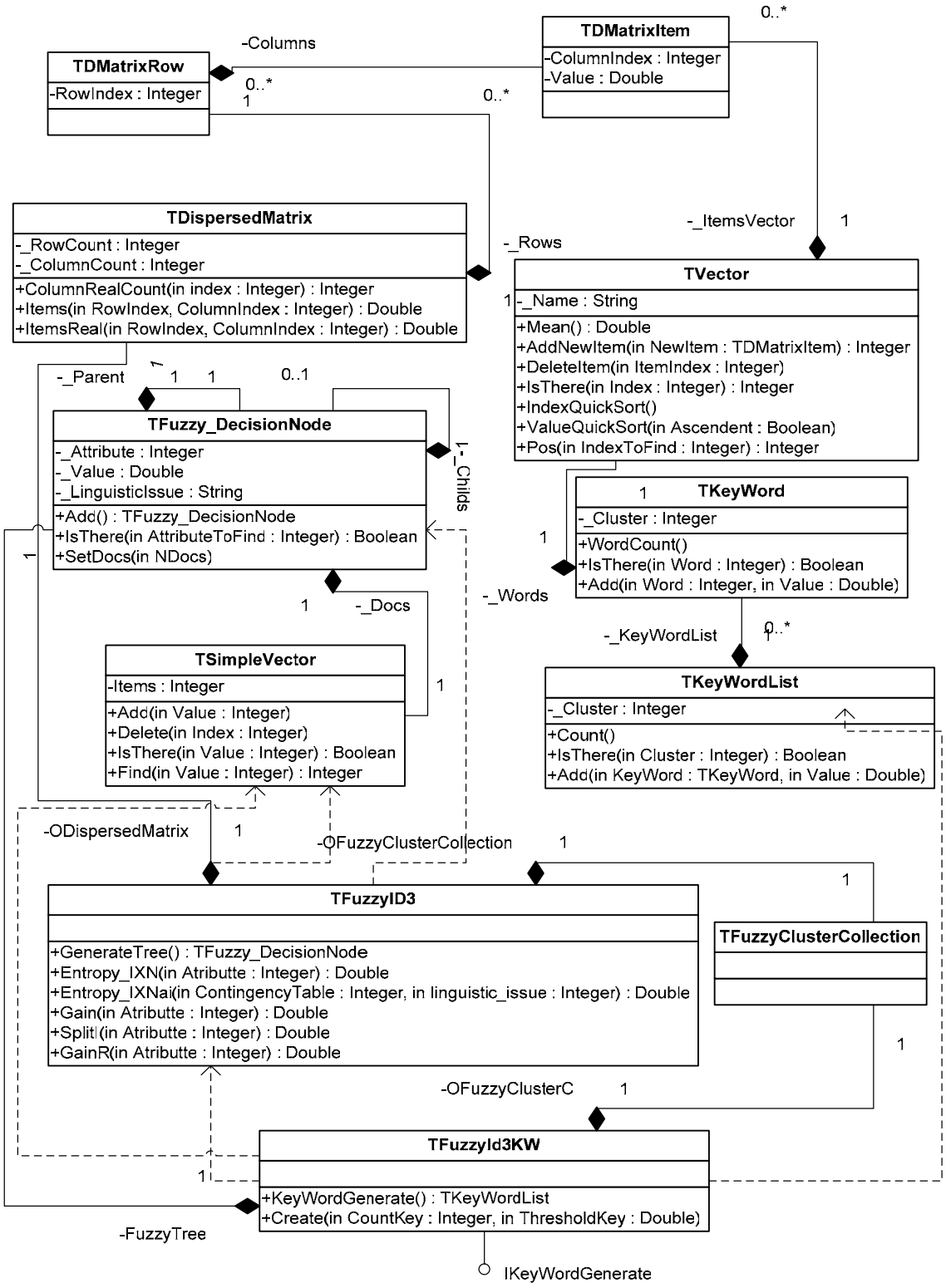
Anexo 17. Diseño de las clases relacionadas con el proceso de construcción de las tablas de contingencia.



Anexo 18. Diseño de las clases relacionadas con la generación de reglas borrosas.



Anexo 19. Diseño de las clases relacionadas con la extracción de palabras claves.



Anexo 20. Descripción de los tópicos asociados a los documentos que forman el corpus 1.

Tópicos	**Número de documentos**
cocoa	7
Zinc	9
Yen	1
Trade	2
strategic-metal	2
silver	8
Ship	9
saudriyal	1
Retail	1
reserves	7
Money-fx	7
Lead	7
Jobs	7
Ipi	1
income	1
Grain	1
Gnp	1
Dlr	8
Crude	10
copper	4

Anexo 21. Descripción de los tópicos asociados a los documentos que forman el corpus 2.

Tópicos	Número de documentos
acq	7
gnp	1
grain	5
Housing	1
Interest	3
iron-steel	1
jobs	3
lei	1
livestock	1
meal-feed	1
money-fx	4
money-supply	1
Reserves	8
ship	1
Sorghum	1
soy-meal	1
tin	1
trade	7
veg-oil	1
wheat	3
alum	3
bop	2
Carcass	1
cocoa	6
coffee	6
corn	2
crude	12
earn	12
barley	1

Anexo 22. Descripción de los tópicos asociados a los documentos que forman el corpus 3.

Tópicos	Número de documentos
acq	16
Housing	1
interest	3
jobs	2
meal-feed	2
money-fx	4
money-supply	2
Naphtha	1
nickel	1
oilseed	3
palmkernel	1
palm-oil	2
pet-chem	1
plywood	1
propane	1
reserves	3
rice	1
Rubber	1
ship	2
Soybean	4
soy-meal	3
soy-oil	1
sugar	1
tea	2
trade	8
veg-oil	5
Wheat	3
alum	1
Coffee	6
Copper	3
copra-cake	1
corn	3
Cotton	1
cpi	1
crude	22
earn	3
gas	1
gnp	3
grain	10
heat	1
Cocoa	9

Anexo 23. Descripción de los tópicos asociados a los documentos que forman el corpus 4.

Tópicos	**Número de documentos**
acq	10
money-supply	6
cocoa	7
trade	9

Anexo 24. Fragmento del corpus 5.

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.
…

Anexo 25. Verificación de la extracción de palabras claves según ID3 variante borrosa con Corpus 1.

Grupos / Palabras	1	2	3	4	5
TIME			B		
FOLLOW			B		
TELL	B		B		
LATE	B				
BIG	B				
RESERVE				B D C A	
COCOA	B D A				
WEEK				B	
YEAR		B		B	A
TOPIC		B			
STATE	D	B D C A		B	
CURRENCY				B A	
RATE		B			B D A
OPERATE	D				
COMPARE	D				
MANCERA				D	

A – ID3 variante borrosa

B – ID3 variante dura (WEKA)

C – ID3 variante dura (CorpusMiner)

D – C4.5 (WEKA)

Anexo 26. Verificación de la extracción de palabras claves según ID3 variante borrosa con Corpus 2.

Grupos / Palabras	1	2	3	4	5	6	7	8	9	10	11	12	13	14
SHARE								B C A						
RISK														C A
CHAIRMAN	C A			C A										
TRADE	B C A				B A							B C A		
REVENUE										C A				
PLAN							C A							
DECREASE											C A			
EUROPEAN							C A							
IMPORTANT												C A		
RISE			B A							C A				
DLRS			C A							D A				
DLR										C A				
TOTAL										C A				
WEEK					B C A									

COMPANY				B C A										
EARN								B A		B A			B C A	
COFFEE	D B C A													
PCT										B A	B C A			
PRICE				B C A										
GOVERNMENT											C A			
LTD						B C A	B C A				C A		C A	
SERVICE											C A			
LEAST														
ISSUE	B A			B A										
PREDICT														B A
AUGUST				B A										
CONDITION			B A	B A					D A					
ARRANGEMENT		B A												
VOLUME									B A					
SPOKESMAN			B								D	B		

			A								A	A		
CLOSE					B A									
GOOD			B A											
EXPORT												B A		
ACQUISITION							B A							
GROW														
PAYMENT					D A									
MARK					D A									
INSURANCE									D A					
CTS								D A						
STRONG			D A											
OPEC				D A										
CENT						D A								
CONGRESS												D A		
NUMBER	D A													
COCOA			D A											
ECONOMIST														
IRAN														D A
INTANGIBLE													D A	
SUM							D							

							A							
TONE		B A												

A – ID3 variante borrosa

B – ID3 variante dura (WEKA)

C – ID3 variante dura (CorpusMiner)

D – C4.5 (WEKA)

Anexo 27. Verificación de la extracción de palabras claves según ID3 variante borrosa con Corpus 3.

Grupos / Palabras	1	2	3	4	5
LOW				B	
EXCHANGE				B D A	
EXPECT			B	B	B
LONG				B	
MONTH				B	
TALK	B				
REDUCE	B				
EXPECT	B				
DUE		B			
DELEGATE		B			
SHARE		B A	A		
PRICE				B	
HOUSE					B
MAKE					B
GET			B		
INCREASE			B		
DLRS			B C A		
MLN			B A		
BANKER				D	
COPPER				D	
WARN				D	
DEPOSIT				D A	
BOND	D				
GRAIN	D A				
PACT	D				
ICO	D A				
SPOKESMAN					D
AMERICAN					D A
FAMILY					D
SHORTFALL					D
IRAN					D A
PRE		D			
WORK		D			
CROWN		D			

FILE		D A			
COFFEE	A				
COCOA					
EXPORT					
OIL		C	A		
GULF					C A
SAY	A		C A		C
QUOTA					
HOLD		A			
BARREL					
RISE			A		
AREA				C	
YEAR				C	

A – ID3 variante borrosa

B – ID3 variante dura (WEKA)

C – ID3 variante dura (CorpusMiner)

D – C4.5 (WEKA)

Anexo 28. Verificación de la extracción de palabras claves según ID3 variante borrosa con Corpus 4.

Grupos / Palabras	1	2	3	4	5	6	7	8	9
AGREE					B				
MONEY	B D A								
TRADE		C A						B C A	
RECENT					B A				
OFFER			B						
PORTION								B	
CONSUMER					D A				
COCOA					D A				
RAISE			D A						
SHAREHOLDER			D						D
WEEK							C		
GROUP			C A						
YESTERDAY									
ADD		C							
MAIN							C		
LATE							C		
PRICE	C A								
SUBSIDIARY									C A

A – ID3 variante borrosa

B – ID3 variante dura (WEKA)

C – ID3 variante dura (CorpusMiner)

D – C4.5 (WEKA)

Anexo 29. Verificación de la extracción de palabras claves según ID3 variante borrosa con Corpus 5.

Grupos / Palabras	1	2	3	4	5	6
SEE				B		
SYSTEM	B A	B	D	B D A		
AIR						B
THINK			B A			
COMPUTER		B A				
PROVIDE	B					
BROADBAND				D		
SERIOUSLY				D A		
VIRTUAL	D A					
MOBILE		D				
TROPOSPHERE						D A
YEAR						C
USER		A				
WIRELESS		A				

A – ID3 variante borrosa

B – ID3 variante dura (WEKA)

C – ID3 variante dura (CorpusMiner)

D – C4.5 (WEKA)

Anexo 30. Palabras claves extraídas según ID3 variante borrosa para los cluster obtenidos en el corpus 1.

Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
<COCOA> (0.88)	<SHIP> (0.92)	<ZINC> (0.96)	<RESERVES> (0.95)	<JOBS> (0.89)
<CRUDE> (0.10)	<CRUDE> (0.92)	<LEAD> (0.96)	<SILVER> (0.19)	<SILVER> (0.19)
<SILVER> (0.10)	<JOBS> (0.40)	<COPPER> (0.96)	<JOBS> (0.10)	<CRUDE> (0.13)
<SHIP> (0.10)	<SILVER> (0.37)	<SHIP> (0.85)	<MONEY-FX> (0.10)	<IPI> (0.10)
	<MONEY-FX> (0.10)	<GRAIN> (0.85)		<GNP> (0.10)
	<DLR> (0.10)	<SILVER> (0.62)		<INCOME> (0.10)
	<SAUDRIYAL> (0.10)	<STRATEGIC (0.10)		<TRADE> (0.10)
	<YEN> (0.10)	-METAL>		<RETAIL> (0.10)
DETAIL	BRINK	LEAD	FRENCH	UNEMPLOYMENT
PROPOSAL	UNITE	SAY	REVALUE	YEAR
COCOA	STATE	PORT	RESERVE	RECORD
SAY	AMERICA	TONE	JANUARY	PCT
BUFFER	ATTACK	PREDICT	MLN	RATE
STOCK	SPOKESMAN	CANADA	MARCH	ECONOMIST
DELEGATE	WAR	CONCENTRATE	CURRENCY	
ICCO	AMERICAN	STRIKE	FALL	
RULE	GULF	ZINC	EUROPEAN	
CONSUMER	SHIP	METAL	BANK	
GROUP	TANKER	SUPPLY	REPAYMENT	
	IRANIAN	NAVY	FOREIGN	
	IRAN	CAPACITY	MONETARY	
	ARM		FRANC	
	RHETORIC			

Para cada grupo se muestran los tópicos que tienen un grado de pertenencia mayor o igual a 0.1.

Anexo 31. Palabras claves extraídas según ID3 variante borrosa para los cluster obtenidos en el corpus 2.

Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6	Cluster 7
<COFFEE> (0.90) <TRADE> (0.55)	<GRAIN> (0.86) <WHEAT> (0.86) <CRUDE> (0.80)	<COCOA> (0.91) <RESERVES> (0.50)	<CRUDE> (0.86) <IRON -STEEL> (0.43) <MONEY -FX> (0.14)	<RESERVES> (0.99) <INTEREST> (0.84) <MONEY -FX> (0.84) <ACQ> (0.76) <JOBS > (0.58)	<EARN> (0.93) <ALUM> (0.93)	<EARN> (0.86) <CRUDE> (0.86) <ACQ> (0.10)
CUT	AREA	COCOA	CHAIRMAN	WEEK	LTD	ARM
UNDERSTAND	WHEAT	STRONG	BARREL	PAYMENT	CENT	WARN
PRESIDENT	GRAIN	CARRY	WORLD	MARK	LOSS	HISTORICAL
FACTOR	PURCHASE	LAND	HARD	BANK	CLOSURE	OPERATE
USE	SUBSTANTIAL	POUND	DEPOSIT	HOLD	GOLDENDALE	IMPROVE
COFFEE	EUROPEAN	OFFER	STAFF	BUNDESBANK		BASIS
CASE	MEMBER	BEAN	DISAGREE	MERGER		BENEFIT
NATION	IMPORT	COST	CONTRACT	AUTHORITY		PLAN
FAIL	ARRANGEMENT	RISE	SUBROTO	TRADE		EUROPEAN
AGRICULTURE	TONE	DLRS	PRICE	CLOSE		LTD
DRUG		CONDITION	COMPANY			ACQUISITION
CHAIRMAN		GOOD	ISSUE			SUM
TRADE			AUGUST			
ISSUE			CONDITION			
NUMBER			OPEC			

Cluster 8	Cluster 9	Cluster 10	Cluster 11	Cluster 12	Cluster 13	Cluster 14
<GRAIN> (0.91) <CORN> (0.91) <SORGHUM>(0.91) <TRADE> (0.83) <EARN> (0.80)	<TRADE>(0.76) <ACQ> (0.45) <LEI> (0.35) <MONEY (0.22) -FX>	<EARN> (0.91) <BOP> (0.78) <GNP> (0.53) <TRADE> (0.53) <CRUDE> (0.50) <RESERVES> (0.48) <EARN> (0.39) <MONEY (0.27) -SUPPLY>	<HOUSING>(0.82) <ACQ> (0.61)	<TRADE> (0.76) <ACQ> (0.44)	<EARN> (0.72)	<CRUDE> (0.86)
RECEIVE	SHARE	LEASE	SALE	SYSTEM	DOLLAR	MILITARY
CERTIFICATE	EARN	TOTAL	HOME	FOREIGN	REPRESENT	PLATFORM
KIND	OPTION	SPOT	SINGLE	TENDER	CAUSE	WASHINGTON
RATIO	INSURANCE	PERFORMANCE	UNIT	SIGHT	FINANCIAL	TARGET
MANAGEMENT	SHAREHOLDER	PCT	SEASONALLY	GULF	PARTLY	IRAN
DEPUTY	POSSIBLE	REVENUE	SUPPLY	IMPORTANT	ASSET	POLITICAL
HALF	BUFFER	RISE	ADJUST	EXPORT	EXCHANGE	RISK
CASH	GROUP	DLRS	DEPARTMENT	TRADE	PAY	INVOLVE
EARN	VOLUME	DLR	SERVICE	SPOKESMAN	EARN	PREDICT
SHARE	CONDITION	EARN	DECREASE	CONGRESS	LTD	
CTS	INSURANCE		PCT		INTANGIBLE	
			GOVERNMENT			
			LTD			
			SPOKESMAN			

Para cada grupo se muestran los tópicos que tienen un grado de pertenencia mayor o igual a 0.1.

Anexo 32. Palabras claves extraídas según ID3 variante borrosa para los cluster obtenidos en el corpus 3.

Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
<COFFEE> (0.87)	<ACQ> (0.79)	<EARN> (0.87)	<RESERVES> (0.88)	<SHIP> (0.95)
<GRAIN> (0.78)	<CRUDE> (0.42)	<VEG-OIL> (0.81)	<PROPANE> (0.77)	<CRUDE> (0.95)
<CORN> (0.77)	<COCOA> (0.22)	<PALM-OIL> (0.81)	<HEAT> (0.77)	<INTEREST> (0.43)
<OILSEED> (0.77)		<ACQ> (0.77)	<GAS> (0.77)	<HOUSING> (0.43)
<SOYBEAN> (0.77)		<RESERVES> (0.72)	<MONEY-FX> (0.74)	<VEG-OIL> (0.17)
<MEAL-FEED> (0.55)		<BOP> (0.72)	<MONEY - (0.67)	<SOY-OIL> (0.17)
<SOY-MEAL> (0.55)		<MONEY-	SUPPLY>	<OILSEED> (0.17)
<ALUM> (0.18)		SUPPLY> (0.68)	<COPPER> (0.25)	<SOYBEAN> (0.17)
		<CRUDE> (0.54)		
		<TRADE> (0.49)		
		<NAPHTHA> (0.45)		
		<PET-CHEM> (0.45)		
		<COPPER> (0.45)		
		<CPI> (0.42)		
		<COCOA> (0.41)		
		<COFFEE> (0.33)		
		<JOBS> (0.26)		
PATTERN	SEC	TRANSCANADA	SIGHT	WASHINGTON
COFFEE	ALLY	EARN	BANK	DIPLOMAT
ICO	MERGER	QUARTER	FOREIGN	IRANIAN
SAY	COMPANY	RISE	CONTRACT	SOURCE
GRAIN	GROUP	COMPANY	DEPOSIT	UNITE
CORN	SHARE	SAY	RESERVE	STATE
CROP	PCT	SHARE	DAY	AMERICA
WINTER	RIGHT	OIL	EXCHANGE	AL
SOVIET	HOLD	MLN	NATIONAL	GULF
WEATHER	FIRM	TONE	FUND	CARRY
HARVEST	LT	DECEMBER	FEED	ATTACK
EASTERN	LAW	DLRS	DELIVERY	IRAN
RAIN	INC	GENERAL		WAR
INCH	STAKE	CTS		AMERICAN
	VOTE	DOME		SHIP

	TENDER			
	COMMON			
	CARE			
	FILE			

Para cada grupo se muestran los tópicos que tienen un grado de pertenencia mayor o igual a 0.1.

Anexo 33. Palabras claves extraídas según ID3 variante borrosa para los cluster obtenidos en el corpus 4.

Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6	Cluster 7	Cluster 8	Cluster 9
<MONEY -SUPPLY> (0.92)	<TRADE> (0.89)	<ACQ> (0.91)	<TRADE> (0.88)	<COCOA> (0.95)	<ACQ> (0.90)	<MONEY-SUPPLY> (0.86) <TRADE> (0.38)	<TRADE> (0.83)	<ACQ> (0.97)
GROWTH	STUDY	REDSTONE	CONSISTENT	DELEGATE	CONTACT	SMALL	GAP	OPERATION
DATA	CANADA	RAISE	ASSOCIATION	COCOA	SEC	BORROW	JAPAN	PUROLATOR
RISE	CANADIAN	GROUP	PROMOTE	COAST	UNIT	WEDNESDAY	OFFICIAL	FRIDAY
MONEY	TIE		INDUSTRIAL	CROP		RESERVE	MINISTRY	COURIER
RESERVE	PACT		PROTECTIONISM	FLOWER		FEED	CERTIFICATE	E.F.
FEED	TRADE		START	TRADER			JAPANESE	HUTTON
ECONOMIST			SECRETARY	RAIN			TRADE	LBO
RATE			PROTECTIONIST	BUFFER				SUBSIDIARY
PRICE			BUDGET	DEALER				PC
			FORM	TONE				WARRANT
			SEE	MALAYSIAN				
			COOPERATION	ICCO				
			FIND	BAG				
			SIT	CONSUMER				
			NEWSPAPER	NEGOTIATION				
			SPEECH	RECENT				
			ROLE					
			BROADLY					
			CIRCUMSTANCE					

Para cada grupo se muestran los tópicos que tienen un grado de pertenencia mayor o igual a 0.1.

Anexo 34. Palabras claves extraídas según ID3 variante borrosa para los cluster obtenidos en el corpus 5.

Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6
<VIRTUAL> (0.89)	<WIRELESS> (0.77) <USERS> (0.74)	<WORKING> (0.94) <VACCINE> (0.94)	<VIRUS> (0.84)	<WOMAN > (0.83)	<WARMING> (0.90) <WEATHER> (0.83)
VIRTUAL	WIRELESS	VACCINE	VIRUS	WOMAN	WEATHER
SYSTEM	USER	WORK	SYSTEM		WARM
	COMPUTER	THINK	SERIOUSLY		TROPOSPHERE

Para cada grupo se muestran los tópicos que tienen un grado de pertenencia mayor o igual a 0.1.