UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA

La Universidad Católica de Loja

ÁREA TÉCNICA

TITULACIÓN DE INGENIERO EN SISTEMAS INFORMÁTICOS Y COMPUTACIÓN

Aplicación de técnicas de procesamiento de Lenguaje Natural y Minería de Texto para la clasificación de preguntas dentro de un cuestionario digital.

TRABAJO DE FIN DE TITULACIÓN

AUTOR: Ortega Capa, Walter Rodrigo

DIRECTOR: Reátegui Rojas, Ruth María, Mgs.

LOJA - ECUADOR

2015

APROBACIÓN DEL DIRECTOR DEL TRABAJO DE FIN DE TITULACIÓN

Mgs.

Ruth María Reátegui Rojas

DOCENTE DE LA TITULACIÓN

De mi consideración:

El presente trabajo de fin de titulación; “Aplicación de técnicas de procesamiento de Lenguaje Natural y Minería de Texto para la clasificación de preguntas dentro de un cuestionario digital” realizado por Walter Rodrigo Ortega Capa, ha sido orientado y revisado durante su ejecución, por cuanto se aprueba la presentación del mismo.

Loja, Marzo de 2015

f)……
Mgs. Ruth María Reátegui Rojas

DECLARACIÓN DE AUTORÍA Y CESIÓN DE DERECHOS

Yo Walter Rodrigo Ortega Capa declaro ser autor del presente trabajo de fin de titulación: Aplicación de técnicas de procesamiento de Lenguaje Natural y Minería de Texto para la clasificación de preguntas dentro de un cuestionario digital, de la Titulación Ingeniero en Sistemas Informáticos y Computación, siendo Mgs. Ruth María Reátegui Rojas directora del presente trabajo; y eximo expresamente a la Universidad Técnica Particular de Loja y a sus representantes legales de posibles reclamos o acciones legales. Además certifico que las ideas, conceptos, procedimientos y resultados vertidos en el presente trabajo investigativo, son de mi exclusiva responsabilidad.

Adicionalmente declaro conocer y aceptar la disposición del Art. 88 del Estatuto Orgánico de la Universidad Técnica Particular de Loja que en su parte pertinente textualmente dice: "Forman parte del patrimonio de la Universidad la propiedad intelectual de investigaciones, trabajos científico o técnicos y tesis de grado o trabajos de titulación que se realicen con el apoyo financiero, académico o institucional (operativo) de la Universidad".

f)……..

Ortega Capa Walter Rodrigo
1104609126

DEDICATORÍA

Dedico este trabajo a mis padres, quienes me han respaldado incondiconalmene en todo momento y se han sacrificado para que pueda alcanzar mis metas.

AGRADECIMIENTO

Agradezco en primer lugar al Divino Creador quien guía mis pasos cada día y también por haberme brindado la oportunidad de contar con una maravillosa familia, quienes me han apoyado incondicionalmente a lo largo de este tiempo de estudio.

Un agradecimiento especial a la Mgs. Ruth Reátegui directora de mí Proyecto de Fin de Titulación por su paciencia, por sus conocimientos impartidos y sobre todo por el tiempo invertido a lo largo del desarrollo de este trabajo.

Agradezco a mis compañeros y profesores con quienes tuve la oportunidad de trabajar y compartir a lo largo de todas mis asignaturas en estos años.

ÍNDICE DE CONTENIDOS

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

ÍNDICE DE ANEXOS

RESUMEN

Junto con el creciente número de documentos digitales que se generan día a día en las empresas, organizaciones e instituciones surge la necesidad de analizarlos y de extraer información relevante. Este proceso conlleva a una mejor gestión y organización de estos datos. Por tal motivo este trabajo está enfocado en establecer una guía de referencia para la clasificación automática de cuestionarios digitales de la materia de Matemáticas Discretas del Primer Bimestre de la Modalidad Abierta de la Universidad Técnica Particular de Loja. Para el desarrollo de este proyecto se ha utilizado la metodología CRISP-MD (Siglas en inglés, Cross Industry Standard Process for Data Mining) haciendo uso de técnicas de Minería de Texto y de Procesamiento de Lenguaje Natural (PLN). La representación de los datos se realizó mediante los métodos TDM (Matrix -Term Document). Dentro de los mejores algoritmos de clasificación de texto en Weka, se puede mencionar el DMNtext-I1 and NavieBayesMultinominalUpdateable, ya que entre los resultados obtenidos estos dos algoritmos presentan similitudes en sus valores finales Precisión de 0.847, Recall 0.824 y 0.436 de Accuary, por lo tanto se tiene un Error de 0.177. Estos valores son producto de la configuración Porcentaje Split de 66%, datos de entrenamiento 66 y 34 datos de prueba.

Palabras claves: Minería de Texto (MT), Clasificación de documentos, Procesamiento de Lenguaje Natural (PLN), metodología CRISP-MD, TDM (Matriz- Termino Documento), TF-IDF (Término Frecuencia - Frecuencia Documento Inversa).

ABSTRACT

Along with the increasing number of digital documents that are generated daily in companies, organizations and institutions, arises the necessity to analyze and extract relevant information. This process leads to better management and organization of these data. Therefore this work is focused on establishing a reference guide for the automatic classification of digital questionnaires concerning Discrete Mathematics First Bimestre of the Open Method of the Universidad Técnica Particular de Loja. For the development of this project is the use the CRISP-DM methodology (acronym in English, Cross Industry Standard Process for Data Mining) using text mining techniques (Text Mining) and Natural Language Processing (Natural Language Processing) . The representation of the data is performed by the TDM (Matrix -Term Document) method. Among the best text classification algorithms in Weka, we can mention the DMNtext-I1 and NavieBayesMultinominalUpdateable as between the results of these two algorithms have similarities in their final values Accuracy 0.847, 0.824 and 0.436 Recall of accuary, so both have a 0177 error. These values are the product of the Percentage Split configuration of 66%, 66 training data and 34 test data.

KEYWORDS: methodology CRISP-DM, MT (Text Mining), PLN (Natural Language Processing), TF-IDF (Term Frequency -Inverse Document Frequency), TDM (Matrix -Term Document).

INTRODUCCIÓN

Durante estos últimos años la información ha venido siendo un factor muy importante en las organizaciones, por lo tanto al tener una variedad de datos se hace compleja la identificación de información que realmente se requiere. La aplicación de Procesamiento de Lenguaje Natural y Minería de Texto es importante ya que contribuye con algunos temas como la clasificación de información de acuerdo a diccionarios o algunos otros parámetros que permitan agrupar e indexar una colección de documentos digitales, por tal razón varios autores e interesados en el Procesamiento de Lenguaje Natural y Minería de Texto adquirieron conocimiento e información sobre estos temas (Vallez & Pedraza, 2012). La clasificación o categorización de documentos es una aplicación de la Minería de Texto que asigna a los documentos una o más categorías, etiquetas o clases basadas en el contenido (FECyT, 2012). La forma tradicional para la categorización de textos definían manualmente las reglas de clasificación, pero estas han sido reemplazadas por otras basadas en técnicas de aprendizaje automático o en combinaciones con otras.

Este proyecto representa una guía para la clasificación automática de preguntas en cuestionarios digitales de materia de Matemáticas Discretas.

Para este trabajo hemos aplicado técnicas de Procesamiento de Lenguaje Natural y Minería de texto sobre lo cuestionarios digitales con el fin de generar un modelo que permita ser analizado mediante técnicas de Clasificación o Clusterización. En este contexto se ha elegido Weka ya que es una herramienta que brinda las características y algoritmos necesarios para realizar diversos experimentos con cada uno de los modelos que se ha generado, con el fin de realizar el análisis de los datos y contrastar los resultados para realizar una evaluación en base a parámetros, que permiten seleccionar el mejor algoritmo para la clasificación de preguntas en un cuestionarios digitales.

CAPITULO I
ANTECEDENTES

1.1 Información estructurada y no estructurada

La información estructurada se caracteriza por tener un significado que no tiene ambigüedad y que está representado explícitamente en una estructura o formato de los datos (Montes, 1999), (Pérez & Cardoso, 2007). En cuanto a la información no estructurada es aquella que no es almacenada en tablas de bases de datos relacionales y pueden incluir e-mails, documentos ofimáticos, pdf, hojas de cálculo, presentaciones, imágenes, videos etc.

1.2 Procesamiento de Lenguaje Natural (PLN)

La mayor parte del conocimiento se ha venido guardado y manejado en forma de lenguaje natural, y en la actualidad no es la excepción al contrario sigue existiendo, pero en forma de documentos, libros artículos, revistas, páginas web y sobre todo en formato digital. Pero lo que para las personas es conocimiento, para las computadoras es solo secuencias de caracteres y archivos, entonces para resolver este problema y habilitar a las computadoras para entender el texto, surgen varios nombres: procesamiento de lenguaje natural, tecnologías de lenguaje y lingüística computacional con el fin de procesar el texto por su sentido y no como normalmente se hace en formatos de archivos binarios (Gelbukh, 2010).

El procesamiento de lenguaje es una subárea de la inteligencia artificial y la lingüística que tiene como objetivo estudiar los problemas derivados de la generación y comprensión automática del lenguaje natural, es decir construir sistemas y algunos mecanismos que permitan la comunicación entre personas y maquinas por medio de lenguajes naturales (Rodríguez & Benavides, 2007). El procesamiento de lenguaje natural busca poder crear programas que puedan analizar, entender y generar lenguajes que los humanos utilizan habitualmente, de tal manera que el usuario se comunique con el computador. Según (Marti & Llisterri, 2002) en su libro sobre procesamiento de lenguaje natural expresa que el PLN puede definirse como el conjunto de instrucciones que un ordenador recibe en un lenguaje de programación que le permitirá comunicarse con el humano en su propio lenguaje.

1.2.1 Estructura general para el procesamiento de lenguaje natural.

Cuando hablamos de procesamiento de lenguaje natural podemos referirnos a un esquema general que se utiliza para estos temas; primero se realiza la transformación del texto con el fin de tener una representación formal, pero sin que se pierda las características relevantes, luego se manipula esta representación con la ayuda de

programas o herramientas que permitan buscar las estructuras necesarias del problema y por ultimo validar la representación y transformarlo a un lenguaje natural (Gelbukh, 2010). Por otro lado lograr que las computadoras se comuniquen con las personas involucra todas las ramas de la lingüística computacional así como también procesamiento de voz para la comprensión del lenguaje y el razonamiento lógico sobre las situaciones de la vida diaria. Cuando hablamos de comunicación entre personas y computadoras, no se trata de pronunciar comandos, sino se trata de hablarle a la máquina como cuando hablamos con otra persona.

1.2.2 Lingüística computacional.

Constituye un campo científico que se encuentra vinculado a la informática, con el objetivo de la elaboración de modelos computaciones que representen los diferentes aspectos del lenguaje humano y así faciliten la comunicación entre el usuario y la máquina. La lingüística computacional se centra en la consecución de tres objetivos: la elaboración de modelos lingüísticos en términos formales, la aplicación de los modelos creados, comprobación de modelos y sus predicciones (Perez & Moreno, 2009).

1.2.3 Traducción automática.

Por otro lado la traducción automática ha sido el punto de motivación para el desarrollo de la lingüística computacional, ya que permitirá la comunicación fluida entre la gente, el esquema general de todo traductor automático a nivel general inicia con la transformación del texto a una representación intermedia, luego se realizan cambios a la representación anterior para que sea interpretado por el traductor para que finalmente la representación intermedia se transforme en un lenguaje final (Cervantes, 1992). Por otra el procesamiento de lenguaje natural requiere la realización de las siguientes tareas que se descomponen de esta forma:

Análisis morfológico: Se refiere al análisis para extraer raíces, rasgos flexivos y unidades léxicas, es decir se caracteriza por distinguir el tipo de categoría gramatical al que pertenece cada una de las palabras y describen sus características (Gomez, 2002).

Análisis sintáctico: Es aquel que convierte una frase ambigua en un lenguaje natural, (Fernández, 2009), es decir el análisis de la estructura sintáctica de la frase mediante una gramática.

Análisis semántico: Hace referencia a la extracción del significado de la frase y la resolución de ambigüedades léxicas y estructurales. Se trata del primer componente

interpretativo, el cual consiste en asignar un significado a cada una de las oraciones del contexto (Contreras & Davila, 2001).

Análisis pragmático: Constituye el análisis del texto para determinar los antecedentes referenciales de los pronombres. En general la pragmática incluye aspectos del conocimiento conceptual que van más allá de las condiciones reales de cada oración (Contreras & Davila, 2001).

Planificación de la frase: Esta fase presenta la forma en la cual se refiere a como estructurar cada frase con el fin de obtener el significado adecuado.

Generación de la frase: Esta tarea realiza la generación de cadenas de palabras a partir de la estructura general. Las distintas fases y problemáticas del análisis del lenguaje se encuentran principalmente dentro de las técnicas lingüísticas**,** las cuales se forman con el desarrollo de reglas estructurales. En las siguientes fases las técnicas probabilísticas se refieren al estudio en base a un conjunto de textos de referencia que el mismo que contiene características de tipo probabilístico o asociados a las distintas fases del análisis del lenguaje (Méndez & José, 1999).

1.2.4 Aplicaciones de procesamiento de lenguaje natural.

Las aplicaciones del Procesamiento de Lenguajes Naturales son muy variadas, ya que su alcance es muy grande. El objetivo de un sistema de Procesamiento de Lenguaje Natural es permitir la interacción de las personas con el computador en lenguaje cotidiano (Laredo, 2005). El procesamiento del lenguaje natural presenta múltiples aplicaciones:

- ✓ Traducción automática se refiere a la traducción correcta de un lenguaje a otro, tomando en cuenta los que se quiere expresar en cada oración y no por palabra.
- ✓ Extracción de Información y Resúmenes la misma que consiste en crear un resumen de un documento basándose en los datos proporcionados, realizando una análisis detallado del contenido.
- ✓ Resolución de problemas el mismo que contribuye a solución de dificultades proporcionado datos y demanda de información permitiendo así interactividad entre el usuario y el computador.
- ✓ Tutores inteligentes el cual se refiere a la enseñanza asistida por computadora.
- ✓ Reconocimiento de Voz en cual implica dos posibles usos: identificar a la persona o para procesar lo que la persona le dicte.

Existen de manera general diferentes herramientas que realizan tareas relacionadas con el Procesamiento de Lenguaje Natural (Pino & Nicolas, 2009):

- ✓ Sistemas de consulta en lenguaje natural el mismo que se trata de sistemas que traducen el tipo de consultas que pueden hacerse a la base de datos. Este es el contexto con más éxito dentro del mundo empresarial.
- ✓ Programas de edición de texto que consiste en programas difundidos que ayudan en la corrección ortográfica y gramatical.
- ✓ Máquinas de escribir accionadas por la voz se tratan de sistemas que reconocen textos que se desea escribir, además estos sistemas van transcribiendo un texto dictado a su correspondiente representación escrita.
- ✓ Reconstrucción de objetos como perfiles, sombras, partes ocultas de los objetos así como contenidos de las imágenes analizadas.
- ✓ Establecer relaciones en el espacio y tiempo entre objetos y sucesos.

1.3 Minería de texto

La minería de texto es otra de las áreas de la lingüística computacional que en los últimos años está siendo de gran utilidad para facilitar el procesamiento automático de la semántica del lenguaje natural. Esta área engloba un conjunto de técnicas las mismas que nos permiten el acceso, obtención y organización de información relevante, es decir permite el acceso al conocimiento que no existía en ningún texto, pero surge de relacionar el contenido de los datos (Perez & Ortiz, Linguistica Computacional y Linguistica de Corpus, 2009).

La minera de texto se define como el proceso automático de descubrimiento de patrones en una variedad de texto, por el ello el proceso la minería de texto consiste en dos etapas: la primera etapa que se basa en el reprocesamiento o preparación de los datos donde los textos se transforman a algún tipo de representación ya sea estructurada o semiestructurada que facilite realizar el análisis o cualquier tipo de procesamiento y la segunda es en la cual se realiza las representaciones intermedias donde se analizan los datos con el objetivo de descubrir patrones o conocimientos (Brun & Senso, 2004).

La minería de textos se dedica principalmente a la categorización, clasificación y agrupamiento de textos, donde la categorización se encarga de identificar las categorías, temas, materias o conceptos presentes en los textos y la clasificación se

encarga a asignar una clase o categorización de los textos (Miner & Nisbet, 2012). La minería de datos también implica el proceso de estructuración del texto de entrada, derivado patrones en los datos y finalmente una evaluación e interpretación de los datos. Entonces dependiendo del tipo de métodos aplicados en la primera etapa como es la de preprocesamiento se pueden realizar representaciones intermedias, lo cual de acuerdo a las representaciones se determinan los métodos usados en la etapa de descubrimiento es decir los patrones que se encontraran en los datos (Montes, 1999).

La minería de datos es el proceso de descubrimiento de conocimiento para encontrar información no desconocida que sea útil en grandes repositorios de datos, donde se integran diferentes paradigmas o procesos de computación como arboles de decisión, inducción de reglas, redes neuronales, algoritmos estadísticos etc. Las principales tareas y métodos de la minería de datos son clasificación, agrupamiento, estimación modelado de dependencias y descubrimiento de reglas, en la minería de texto se realiza la búsqueda de conocimiento en colecciones de documentos no estructurados, es decir es el descubrimiento de patrones nuevos que no existan explícitamente en ningún texto, pero que surgen al momento de relacionar el contenido (Hearst, 1999), (Varela, 2006).

1.4 Siete áreas prácticas del análisis de texto

Estas áreas describen a breves rasgos cada uno de los problemas que se presentan dentro de cada una de estas, así como también nos muestran que existen métodos alternativas para identificación de las áreas prácticas (Gary Miner & Nisbet, 2012).

En la figura 1 se representa los siete campos de la Minería de Texto con intersecciones en la Minería de Datos, Inteligencia artificial y Lingüística Computacional haciendo hincapié en el fondo del área práctica de la clasificación de texto y en la recuperación de información.

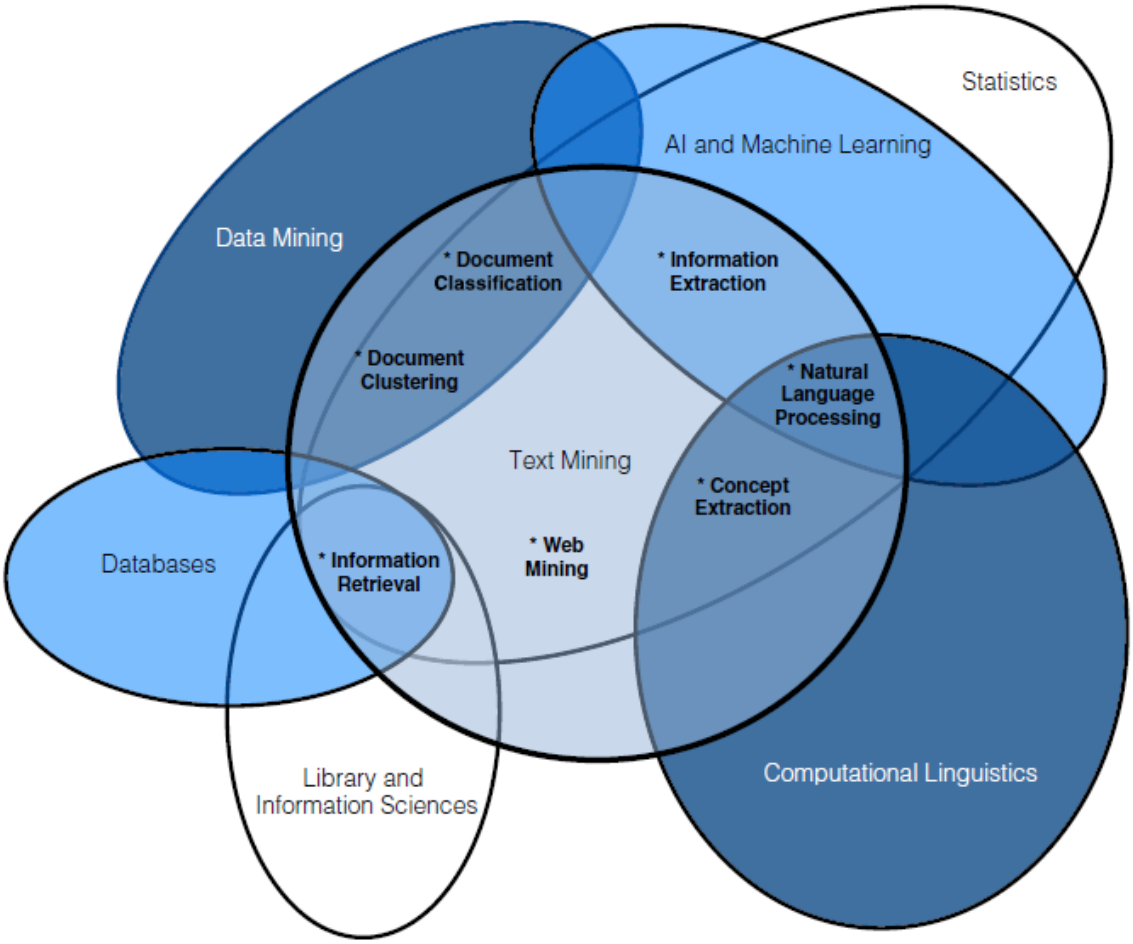


Figura 1. Diagrama de las siete áreas prácticas de la Minería de Texto.

Fuente: Tomado del libro Miner Garly, Practical Text Mining and Statistical Analysis for Non-structured Text Data Applications (Gary Miner & Nisbet, 2012).

La herramienta o aplicación que se utilice para la minería de texto incluye algunas funcionalidades como; Languaje Indentification la misma que ayuda a la identificación del idioma de un documento, Topic Categorization que sirve para la clasificación automática de documentos en categorías definidas al inicio, Feature Extraction la misma que extrae nombres de personas, lugares, organizaciones etc., de los documentos y relaciones que existe entre ellos, Clustering que nos proporciona la facilidad para agrupar automáticamente los documentos y por ultimo lo que denominamos Summarizer la misma que extrae los fragmentos más significativos de un documento (Brun & Senso, 2004).

A continuación se detallan las siete áreas mencionadas en (Gary Miner & Nisbet, 2012).

1.4.1.1 *Búsqueda y recuperación de información (IR).*

Esta área hace referencia al almacenamiento y recuperación de información, así como la indexación de documentos de grandes base de datos incluyendo la búsqueda en internet y consultas por palabra clave e diferentes fuentes de información.

Por otra parte la búsqueda de información realiza búsquedas más avanzadas, es decir centradas en conceptos o características con el fin de ofrecer alternativas al momento de una petición del usuario.

1.4.1.2 *Agrupación de documentos.*

Consiste en clasificar y agrupar términos, párrafos y documentos similares utilizando métodos de agrupación de Minería de Datos.

1.4.1.3 *Clasificación de documentos.*

En esta parte se realiza la categorización de fragmentos de texto, párrafos o documentos utilizando métodos de clasificación de minería de datos basados en los modelos previamente desarrollados.

1.4.1.4 *Minería Web.*

Esta área hace hincapié a los datos y la Minería de Texto en Internet, así como también en un enfoque específico hacia la interconexión con la web. Aquí se presentan grandes desafíos y a la vez oportunidades debido al volumen y estructura de datos aparecen en la web ya que la minería web se basa en la tecnología de clasificación de documentos y comprensión del lenguaje natural.

1.4.1.5 Extracción de Información.

Se refiere a la identificación, extracción de factores relevantes y relaciones de texto, así como también el proceso de toma de datos estructurados, semiestructurados y no estructurados. Esta área tiene como objetivo la construcción de un extracto datos con ayuda algoritmos, herramientas y software especializado. La Extracción de información también se enfoca en ofrecer conceptos o frases de documentos para decidir la relevancia de los mismos, o formar nuevos documentos realizando combinaciones.

1.4.1.6 Procesamiento de Lenguaje Natural (NPL).

Consiste en el procesamiento del lenguaje da bajo nivel y la comprensión de las tareas como; etiquetado de oraciones utilizando la lingüística computacional con el propósito de elaborar modelos computacionales.

1.4.1.7 *Extracción de conceptos.*

Esta área lo busca es realizar la agrupación de palabras y frases de tal manera que pertenezcan a grupos semánticamente similares. Además se realiza la revisión de colecciones de documentos, así como también las tareas de categorización y de clasificación.

1.5 Etapas de la minería de texto

1.5.1 Etapa de preprocesamiento.

En esta primera etapa como es la reprocesamiento los textos se transforman a una representación estructurada o semiestructurada, donde estas representaciones intermedias de los textos deber ser sencillas para facilitar el análisis de los textos y completas con el fin de permitir el descubrimiento de patrones interesantes (Manuel & Gómez, 2005). Las representaciones intermedias más utilizadas en la minera de texto son dos la primera que es a nivel de documento donde cada representación se refiere a un texto y las representaciones se construyen usando métodos de categorización de texto e indexado, y la segunda que es a nivel de concepto donde cada representación indica un objeto, tema o concepto interesante para el dominio, es decir la extracción de términos importantes y la extracción de información (Hearst, 1999),

1.5.2 Etapa de descubrimiento.

En esta etapa se realiza el descubrimiento de texto y por tal motivo sus métodos y sus tareas se clasifican en descriptivos y predictivos, pero es posible clasificarlos de otras maneras, una clasificación alternativa de la minería de texto considera que los textos son una descripción de situaciones y objetos del mundo y que las representaciones intermedias de esos textos obtenidas en la etapa de procesamiento son una descripción estructurada del contenido (Gary Miner & Nisbet, 2012).

1.6 Clasificación de documentos

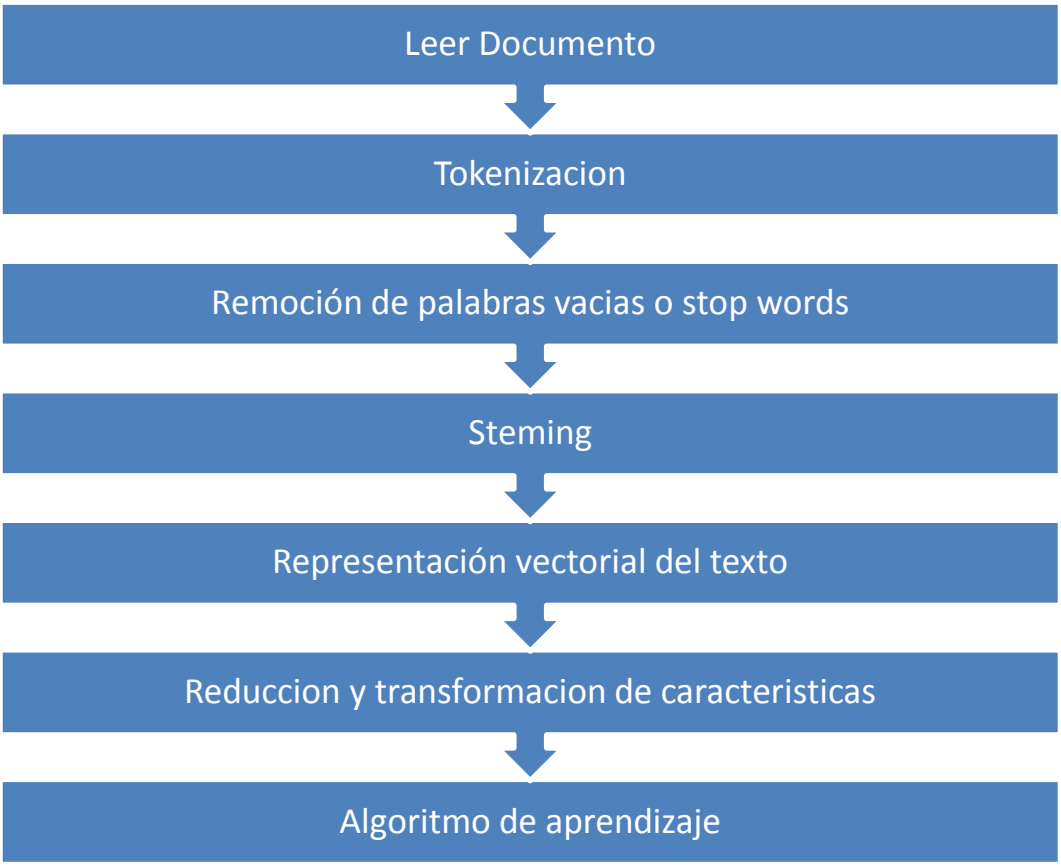


Figura 2. Proceso de clasificación de Texto
Fuente: Basado en el trabajo sobre clasificación automática de respuestas a foros de discusión (Pincay, 2013). www.cib.espol.edu.ec/digipath/d_tesis_pdf/d-83179.pdf

En la figura 2 (Zu G. & Kimura, 2012) se muestra el proceso de clasificación de texto, en la primera fase se hace mención específicamente a la obtención de los datos desde cualquier fuente estas pueden ser; documentos, Bases de Datos y algunos otros archivos, luego se realizan algunas tareas de preprocesamiento de texto con el propósito construir modelos de representación vectorial, y finalmente procede con la clasificación con la ayuda de algoritmos de aprendizaje.

1.6.1 Preprocesamiento de texto.

Esta tarea de reprocesamiento de texto se encuentra compuesta por algunos procesos que permiten realizar de cierta manera una limpieza del texto que vamos a utilizar para desarrollar nuestro trabajo.

- ✓ **Tokenización:** Se refiere a la separación de cadenas de texto con el propósito de identificar palabras y frases en el documento que presenten una característica importante.

- ✓ **Remoción de palabras vacías:** Este consiste en la eliminación de determinadas palabras que poco o nada de información aportan sobe el contenido del documento por el hecho de tratarse de verbos auxiliares, conjunciones, artículos y preposiciones. Por lo general estas palabras aparecen con mucha frecuenta sobre el texto lo cual es una razón principal para la remoción de estas palabras. Revisar Anexo 4.

- ✓ **Stemming:** Es el proceso de reducir la palabra a su raíz gramatical, lo cual contribuye con la reducción del número de términos o palabras diferentes en el documento, quedando luego de este proceso solo palabras que pertenecen a una misma familia léxica.

1.6.2 Representación vectorial del texto.

Un clasificador no puede interpretar el texto en lenguaje natural, por tal motivo es necesario la construcción de modelo que permita la representación de los datos y a su vez pueda tanto un algoritmo como un clasificador realizar el análisis respectivo.

1.6.3 Reducción y transformación de características.

Esta tarea se refiere a la reducción de la dimensionalidad del modelo de datos generalmente en la clasificación de texto de construye la matriz de datos en la cual se puede aplicar distintas técnicas para la reducción del tamaño, este proceso contribuye de manera positiva con un ahorro computacional considerable y su vez los resultados serán más efectivos.

1.6.4 Algoritmos de aprendizaje.

Luego de haber realizado el proceso de reducción de características los documentos se encuentran en una forma en la cual puede ser usada por cualquier algoritmo de aprendizaje automático con el propósito de realizar la clasificación de los mismos.

1.7 Trabajos realacionados a la minería de texto

(Cobo & Martínez, 2009)**,** proponen la combinación de metodologías de minería de texto y técnicas de inteligencia artificial con el fin de optimizar la extracción automática de conocimiento y la agrupación de entidades documentales, es decir hacen mención a un modelo de gestión documental para el proceso de información no estructurada.
Los autores tratan tres problemas principales que pueden abordarse con la implementación de técnicas de minería de texto, inician con la extracción de documentos que son interesantes para el usuario, luego de contar con la recuperación de información, realizan la categorización de documentos asignando a cada documento una o varias categorías. La clasificación se la puede realizar mediante categorización o clustering, en el primer caso se habla de clasificación supervisada, mientas que en segundo se utiliza el concepto de no supervisada.

En este trabajo hacen referencia al modelo vectorial el mismo que permite la representación de documentos a partir de un vector de pesos o palabras que se encuentran en el texto luego de realizar operaciones como eliminación de palabras que tiene poco valor significativo y transformaciones de modo que toda la información se encuentre de la misma forma, con el fin de reducir el tamaño de la lista de términos.

El modelo que han propuesto estos autores tiene como objetivo estructurar, generar información y extraer conocimiento desde datos que se encuentra en forma no estructurada.

Para formar la estructura de la información se utilizan glosarios de términos y tesauros o lista de palabras con sus significados escritos en varios idiomas, estos como elementos de identificación y representación de los documentos independientemente del idioma en que se encuentren escritos. Una vez que se ha estructurado la información se proponen que la nueva información generada debe ser presentada mediante una interfaz de comunicación con el usuario.

El sistema de gestión documental propuesto por estos autores funciona mediante protocolos de comunicación para el acceso a los datos por parte del usuario.
Sobre la parte central del sistema de gestión documental se han implementado en cuanto a la minería texto la extracción automática de conocimiento, utilizando recursos lingüísticos como glosarios para el inicio del análisis, luego de contar con la lista de términos, determinan la forma, clase o categoría gramatical de cada palabra, llamado

también análisis morfológico del texto extraído para identificar sustantivos, adjetivos y verbos para continuar realizando la respectiva lematización.

Estos autores realizan experimentos del uso del modelo en el proceso de clasificación de una colección de 250 documentos asociados a 5 categorías diferentes y escritos en idioma inglés y español, lo cual se puede observar que la minería de texto combinada con modelos de optimización ayudan a una adecuada gestión de grandes volúmenes de información no estructurada.

(Rocha, 2009)**,** utiliza la minería de texto en entornos multilenguaje para la gestión documental en el ámbito empresarial. Realiza operaciones de procesamiento de texto con el objetivo de transformar la información original en alguna forma adecuada para la aplicación de las técnicas minería de texto, es decir realiza la estandarización de los documentos.

Una vez estandarizado el formato de los documentos realiza labores de preprocesado dicho de otra manera prepara los documentos para identificar palabras significativas dentro de ellos.

Este autor para el procesamiento de texto como primer paso realiza el filtrado de los datos donde elimina los caracteres que puedan afectar negativamente a la aplicación de los algoritmos en el contenido de los documentos, en este primer paso también identifica las palabras y frases en el texto que son relevantes o importantes en el dominio de estudio, esto se conoce también como tokenización. Como segundo paso realiza la eliminación de algunas palabras que no aportan información sobre el contenido del documento por ser palabras insignificantes como artículos, preposiciones entre otras, estas palabras se las llama comúnmente stopwords.

Una vez que ha seleccionado los tokens como tercer paso realiza la lematización el mismo que consiste en agrupar palabras que contienen un significado muy parecido asociándolas a un mismo lema, este mismo paso se realiza otra tarea llamada stemming el cual consiste en agrupar palabras comunes con el objetivo de eliminar los prefijos y sufijos y considerar únicamente raíces gramaticales o palabras únicas llamados lexemas.

Este autor (Rocha, 2009) también hace hincapié en uno de los modelos de representación más frecuentes en minería de texto como es el modelo vectorial, donde se representa el conjunto de documentos como una matriz en la que los documentos representan las filas y las palabras del diccionario las columnas.

(Wen-der & Jia-yang, 2013) Señalan que los métodos tradicionales basados en indexación de texto para la recuperación de información no son confiables y derivan dos problemas, el primero que de alguna manera pretende que los diseñadores memoricen los datos, y el segundo problema requiere la ayuda de un humano para ejecutar este método. Basados en estos dos problemas, estos autores proponen una técnica de minería de texto basado en el contenido para la recuperación de texto de documentos, el mismo que se basa en la coincidencia y similitud usando un espacio de modelo vectorial, donde cada expresión se representa como un vector de términos o palabras. Para calcular la similitud entre documentos se realiza mediante el cálculo de la distancia entre uno y otro documento. Esta técnica lo que realiza es la extracción del texto de cada documento y la almacena en una base de datos de indexación, luego se emplea un modelo vectorial para la representación del contenido textual, una vez obtenidos los contenidos se ejecutan tareas de consultas de coincidencias y similitudes en los documentos, todos los documentos más relevantes son extraídos de la base de datos indexada.

En este contexto se realiza un experimento donde se aplican cuatro pasos; en el primer paso se seleccionan un conjunto de términos clave y se almacenan en una base de datos, como segundo paso realiza la recuperación de una lista de documentos, en el tercer paso se realiza la revisión de documentos con el objetivo de determinar la relevancia o importancia de cada uno de los datos textuales, y por último se procede con el conteo de los datos relevantes o no importantes en el contexto del estudio.

Por otro lado estos autores proponen minería de texto basado en contenido como una solución para mejorar la recuperación de información y reutilización de documentos.

(Zhang & Guo, 2013) En este trabajo se propone un clasificador basado en prototipos para la categorización de texto, en el cual un documento que pertenece a una categoría está representado por un conjunto de prototipos. En el proceso de clasificación los prototipos utilizan un algoritmo de ponderación con el objetivo de que los documentos que pertenecen a las subclases se encuentren separados de acuerdo a cada categoría. Existe una ventaja al momento de utilizar estos clasificadores basados en prototipos ya que presentan un resumen de los datos en un pequeño número de prototipos y las estructuras y esquemas de clasificación son interpretables.
La clasificación de un documento se la realiza mediante la búsqueda del prototipo o documento más cercano, esto se calcula por medio de la medida de distancia donde

es representado por K que significa el vecino más cercano, es decir la clasificación basada en prototipos se efectúa de acuerdo a la proximidad y similitud del documento.
El modelo propuesto por estas personas hace referencia a una estructura de clases jerárquicas.
(Ur-Rahman & Harding, 2011) Estos autores se centran en el uso de aplicaciones hibridas de técnicas de minería de datos y de minería de texto, con el fin de clasificar datos de texto de dos clases diferentes. En la primera etapa aplican técnicas de agrupamiento, y en la segunda parte reglas de asociación, donde esta última regla aplica para generar varios términos clave o secuencia de verbos que se utilizan para la clasificación, además esta metodología propuesta se puede utilizar para analizar cualquier tipo de texto sin importar el formato.
El descubrimiento de conocimiento en términos de base de datos textuales es diferente a la forma general, pero existen métodos comunes para la recopilación de información. En este contexto se menciona que la clasificación de texto es muy importante para el manejo textual de datos o información.
Para la clasificación de documentos de texto se utiliza categorías predefinidas o clases que son basadas en muestras del contenido. El método que proponen se refiere a un hibrido el mismo que permite manejar datos textuales en dos clases diferentes lo cual también ha sido puesto en diferentes clasificadores como; arboles de decisión, clasificadores, vectores de soporte con el fin de definir nuevos algoritmos para el manejo de la información textual basado en la realización y tareas de clasificación de texto.
Tanto la preparación de texto como las etapas de procesamiento de texto según estos autores deben funcionar de una manera interactiva de modo que permita encontrar patrones útiles o información relevante.

(Sunikka & Bragge, 2012) Combinan la minería de texto para crear perfiles de personalización de investigación con la revisión textual de literaturas con el propósito de descubrir las principales características de dos categorías de investigación.
Estos perfiles de investigación lo que realizan es la identificación de elementos típicos o similares de la personalización, así como también la recopilación de información para el modelado de otros sistemas.
Este trabajo está relacionado directamente con deseo de comprender la personalización y la forma en cómo se encuentra conceptualizada la literatura. Dentro de este tema se examina la personalización bajo dos enfoques el primero que consiste en la revisión de la literatura tradicional y el segundo enfoque que hace mención a la investigación de perfiles con la ayuda de herramientas de minería de texto.

(Oberreuter & Velásquez, 2013) Utilizan la minería de texto para la detección de plagios, con el fin de explorar el uso de palabras como un rasgo lingüístico o lista de palabras para el análisis de documentos utilizando como modelado el estilo de escritura que se encuentra en estos. En este trabajo se buscan apariciones de las palabras en cada uno de los documentos para analizar si fue escrito o no por otra persona. Dado que este problema se basa en detecciones de plagios no necesita tareas de comparación con otras fuentes de datos y este modelo se basa únicamente en el uso de palabras.

(Delgado, 2009) en su trabajo sobre el enfoque integrado de redes neuronales y algoritmos genéticos para la categorización de documentos, menciona que la categorización automática de documentos se encuentra dentro de la categoría intrínseca o no supervisados, ya que los criterios de categorización se basan en la información contenida en los mismos para determinar sus similitudes, es decir se realiza con base en las características propias de los objetos sin previo conocimiento sobre las clases a las que pertenecen.

El algoritmo “K-means” es el referente típico en el trabajo sobre categorización automática de documentos con mapas auto-organizados de kohonen.
Este algoritmo utiliza a los centroides de cada grupo como sus puntos representantes, partiendo de una selección inicial de K centroides (que pueden ser elementos de la colección seleccionados al azar o los que obtengan mediante la aplicación de alguna técnica de inicialización), donde cada uno de los elementos de la colección se asigna al grupo con el centroide más cercano.
Este algoritmo encuentra una categorización que representa un óptimo local del criterio elegido. En cuanto a las ventajas es mucho más eficiente los tiempos de computo ya que los tiempos requeridos son lineales con la cantidad de documentos a agrupar, pero entre las limitantes podemos mencionar que es dependiente de la selección inicial de centroides y sus resultados pueden ser bastante pobres y pueden variar mucho si se aplica varias veces a la misma colección de documentos, ya que si la selección de centroides al azar es mala, la solución encontrada también lo será (Goldenberg, 2007).

(Hernández, 2013) En su trabajo sobre aplicaciones de Procesamiento de Lenguaje Natural para la categorización automática de documentos experimenta con el algoritmo K-Nearest Neighbour (vecino más cercano) o clasificador kNN, donde se asigna cada documento a la clase de su vecino más cercano, en el cual k es un

parámetro de distancia de los vecinos. Este método es simple ya que trabaja con atributos categóricos en tareas de clasificación de documentos.

Este algoritmo presenta algunas de sus características más importantes entre las más significativas tenemos:

- ✓ Presenta un esquema se clasificación común, el mismo que es basado en el uso de medidas de distancia, se puede decir que es un aprendizaje por analogía.
- ✓ Tiene una técnica que le permite asumir que el conjunto de entrenamiento incluye no solo datos sino también la clasificación deseada.
- ✓ Los datos de entrenamiento los toma por referencia.

Por otra parte este algoritmo es el más frecuente y el más utilizado en el campo de categorización automática de documentos. Entre las desventajas podemos mencionar que requiere más tiempo cuando se tiene un gran número de ejemplos clasificando.

(Sánchez & Antonieta, 2011) Aplican algoritmos genéticos o K-medias en la categorización automática de documentos. Este algoritmo es un método popular para la categorización de documentos de texto ya que sus resultados se basa en la representación de cada uno de los clústeres por la media de sus puntos, es decir su centroide. Por lo tanto la representación mediante centroides tiene la ventaja de que tiene un significado gráfico y estadístico inmediato. Este algoritmo según este autor es uno de los más veloces y eficientes, aunque también se dice que es una de los más limitados, ya que precisa únicamente del número de categorías similares en las que queremos dividir el conjunto de datos.

Como podemos darnos cuenta los distintos autores en sus trabajos lo realizan con algoritmos de clasificación tanto supervisados como no supervisados, aunque también depende del contexto y corpus con el cual se está trabajando para aplicar tal o cual algoritmo. Se puede decir que para el proceso de categorización automática de documentos se necesita un algoritmo que busque soluciones rápidas y eficientes de acuerdo a lo que queremos obtener como resultado.

CAPITULO II
METODOLOGÍA

Cross-Industry Standard Process for Data Mining (CRISP-DM)

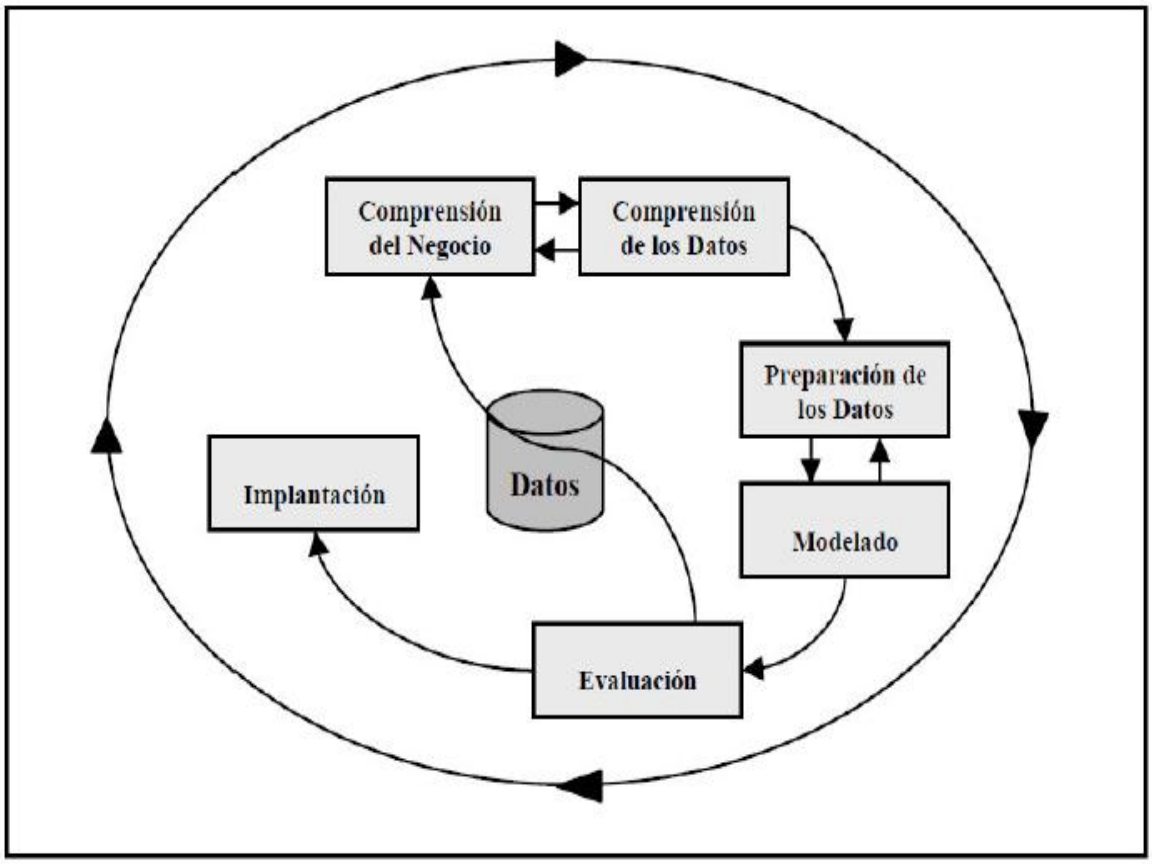


Figura 3. Modelo de proceso CRISP–DM **(CRISP-DM, 2000).**

Para la realización de este proyecto de tesis se ha considerado trabajar con la metodología CRISP-DM (Cross-Industry Standard Process for Data Mining). Como metodología incluye descripciones de la fases normales de un proyecto, las tareas necesarias en cada fase y una explicación de la relaciones entre tareas, y como modelo CRISP-MD ofrece un resumen del ciclo vital de la minería de datos.

Esta metodología como se puede apreciar en la figura 3 es ampliamente utilizada en trabajos de minería de datos pero, también puede usarse para abordar problemas de minería de textos (Santana & Daniela, 2014) ya que se distingue por tener un modelo que está basado en situaciones reales que ocurren en las organizaciones. Además se caracteriza por iniciar su análisis desde una perspectiva global enfatizando el conocimiento del negocio. Este modelo es flexible y se puede personalizar fácilmente para crear modelos que se adapten a las necesidades del proyecto (IMB, 2012).

Esta metodología consta de seis fases las mismas que presentan una cobertura completa de todas sus actividades relacionadas con la ejecución de la minería de datos. La diferencia principal entre la minería de datos y la minería de texto es el tipo

de datos que intervienen en el proceso de descubrimiento. (Gary Miner & Nisbet, 2012). A continuación se da más detalle de esta metodología con un enfoque hacia la minería de texto.

2.1 Fase I: Determinar el propósito del estudio

En esta fase realiza el estudio de minería de texto y la determinación de la finalidad del estudio, así como también un profundo conocimiento del negocio y el objetivo que se logrará.

Con el fin de alcanzar la comprensión del negocio y definir los objetivos se identifica el entorno del problema, interactuando con los expertos en el dominio con el propósito de desarrollar una apreciación profunda sobre su estructura, restricciones y los recursos disponibles para el desarrollo del proyecto. Es decir esta fase se enfoca en la comprensión de los objetivos del proyecto, así como también las exigencias de la perspectiva del negocio, convirtiendo este dominio en la definición del problema con intensión de lograr las metas que se plantearon al inicio (Pete, Khabaza, & Shearer, 2000), (Gary Miner & Nisbet, 2012).

2.2 Fase II: Exploración de la disponibilidad y la naturaleza de los datos

Aquí se avalúa la disponibilidad, así como también la facilidad de la obtención de los datos para el dominio de estudio, además se realiza la recolección de los datos y se identifica problemas de calidad, descubriendo las primeras pautas de los datos, detectando subconjuntos de la información. (Vanrell, 2011). En esta fase se cumplen algunas tareas como; la identificación de fuentes de datos, estos pueden ser digitalizados, en papel, internos o externos a la organización, evaluación del acceso a los datos, recopilación de un conjunto inicial de datos, así como también la exploración y evaluación de los datos necesarios para el estudio de la minería de texto. (Gary Miner & Nisbet, 2012).

2.3 Fase III: Preparación de los datos

La preparación de los datos incluye algunas tareas como la selección de información a la cual se le aplica alguna determinada técnica de modelado, limpieza, generación de variables e integración de diferentes fuentes de datos, así como también cambios de formato de los datos (Gallardo, 2000). Esta fase también incluye actividades

necesarias para construir una estructura de datos final, los mismos que serán procesados en las diferentes herramientas de modelado de datos.

Las tareas que se realizan en esta fase pueden incluir selección tablas, registros y atributos, así como también la transformación y limpieza de los datos (Gary Miner & Nisbet, 2012). En esta fase se realizan algunas actividades con el propósito de decidir los datos a utilizar para el análisis de los mismos.

Para establecer el corpus realizamos la recopilación de documentos más importantes para la solución del problema que se está desarrollando, donde la calidad y cantidad de los datos son los elementos más importantes al momento de la preparación de los mismos. (Miner G. , 2011). Existen varias maneras de recolección de información las cuales permiten conocer y analizar el dominio del problema, es decir la recolección, síntesis, organización y comprensión de los datos que se requieren. La información puede incluir un sin número de formatos y fuentes como; documentos de texto, archivos HTML, correos electrónicos, mensajes web, notas breves, y datos normales de texto y grabaciones de voz (Gonzalez, 1997).

Luego de obtener los datos como primer paso se organizan y transforman los mismos con el objetivo que todos ellos lleven la misma forma y estructura. La organización de los documentos e información contenida, puede ser tan simple como un resumen de texto, una lista de enlaces o una colección de páginas web. Los archivos generados de las actividades mencionadas se los puede preparar fuera de la herramienta o software que se va a utilizar para realizar la minería de texto, para luego presentarlos como entrada a la aplicación seleccionada para la resolución del problema. (Gary Miner & Nisbet, 2012).

El propósito del preproceso de los datos es principalmente corregir la inconsistencias lo cual serán utilizados en diferentes tareas de análisis de minería de texto, aquí se utilizan todos los documentos recolectados con el fin de crear una representación estructurada de los datos. Cuando nos referimos a una representación de datos en Minería de Texto estamos hablando de la Matriz de Documento de Texto (*Text Document Matriz*), la misma que se compone de filas que representan los términos y las columnas que representan las preguntas (Compass, 2013). Las relaciones entre términos y los documentos se realizan mediante índices que muestran la frecuencia que un carácter o palabra aparece en un documento. Cuando realizamos la recolección y estructura de la información recolectada se debe tomar en cuenta que no todos los términos son igual de importantes al momento de categorizar los documentos, algunos términos tales como artículos, verbos auxiliares y otros términos

utilizados, casi en su totalidad no tienen significado en el contexto de estudio, por lo que deben quedar excluidos del proceso que se está llevando a cabo.

Por otro lado la lista de términos también es conocida como stopterms lo cual podemos decir que es utilizada solo para el dominio de estudio y por ende debe ser identificada y validada por personas que conozcan del negocio, porque es una parte muy importante y puede influir mucho al momento de alcanzar o no los objetivos del proyecto (Gary Miner & Nisbet, 2012).

2.3.1 Contrucción de la matriz de documentos de texto (TDM).

Para realizar una estructura de los datos que se emplearán para la solución del problema se aplican varias tareas con el fin de obtener la matriz (TDM). Como primer paso se genera la lista de términos identificados y clasificados para el dominio o problema de estudio. En segundo lugar se procede a reducir los términos que no se van a utilizar en la minería de texto con el fin de dejarlos en sus formas más simples y entendibles, pero siempre conservando la calidad de los datos. La reducción de términos trata de identificar las diferentes formas gramaticales de un verbo, con el objetivo de normalizar la lista de términos como presente, pasado, singular, plural y análisis morfológico, esto con la finalidad de alcanzar una forma sin prefijos, sufijos, y lograr un menor número de términos que no tengan relación con el dominio de estudio para evitar errores en el desarrollo del proyecto (Gary Miner & Nisbet, 2012).

Luego de realizar tanto la generación de la lista de términos, como la reducción de los datos se inicia por la creación de la matriz, la misma que consta de una representación del problema en dos dimensiones lo cual se incluye algunos pasos:

- ✓ Tomar en cuenta y poner mucho énfasis en especificar como filas todos los documentos con los que vamos a trabajar.
- ✓ Luego se realiza identificación de todos los términos singulares que se encuentran en los documentos y se procede a definirlos como columnas en la matriz.
- ✓ Como tercer paso se calcula el número con que se presenta cada término o palabra en cada documento (Salton & Lesk).

2.3.2 Reducción de la dimensionalidad de la Matriz.

El entorno del problema que se está desarrollando puede incluir un número bastante grande de documentos e información lo cual se puede catalogar como normal, pero al momento del procesamiento o aplicación de minería texto podría llevar mucho tiempo en la ejecución y con ello resultados erróneos.

En este contexto se debe evaluar diferentes formas de representación de los índices, una de ellas es transformar las frecuencias o apariciones de los términos en los documentos (Louise & Matt, 2010).

Cuando se realiza la normalización se realiza la reducción de la dimensionalidad de la matriz para que sea consistente al momento del análisis (Feinerer, 2008), por tal motivo se puede aplicar algunas opciones para reducir el tamaño de modo que sea manejable:

- ✓ La verificación de la lista de términos y eliminar aquellos que no tienen relación para el contexto de estudio, este es un proceso manual.
- ✓ En esta opción se eliminan los términos pocas veces apariciones en cada uno de los documentos con los cuales se está trabajando en el proyecto.
- ✓ Se realiza la transformación de la matriz mediante la descomposición del valor singular(DVS) el mismo que permite representar la matriz como una serie de aproximaciones lineales que exponen el significado de la misma, es decir el número de veces que un término aparece en un documento (Gary Miner & Nisbet, 2012).

2.3.3 Extracción del Conocimiento.

En esta actividad se realiza la extracción patrones en el contexto del problema de estudio utilizando la matriz estructurada con variables numéricas y nominales de los documentos utilizados para el análisis. (Gary Miner & Nisbet, 2012). Las técnicas de minería de texto permiten explorar y extraer conocimiento de los datos, los principales métodos en lo referente a estudios de minería de texto se muestran a continuación. (López & Baeza, 2001).

Análisis Estadístico: Cando nos referimos al análisis estadístico la normalización consiste en validar una o varias palabras en varios conjuntos de datos lo cual servirá para eliminar diferentes términos que de alguna manera son innecesarios para el estudio de minería de texto (Gary Miner & Nisbet, 2012).

Clasificación: El objetivo de la clasificación es asignar a cada documento una o varias categorías temáticas de un conjunto de categorías previamente establecidas para el estudio del tema (Cobo & Martínez, 2009). La instancia de datos en un conjunto predeterminado de clases o categorías, en la minería de texto se conoce también como la categorización de texto. (Gary Miner & Nisbet, 2012).

Clusterización: Es un proceso que consiste en la generación de grupos de documentos relacionados, que tienen un mismo tema contribuyendo a la generación de un nuevo conocimiento (Cobo & Martínez, 2009). En la minería de texto y recuperación de información se utilizan vectores con características ponderadas que sirven para describir un documento, vectores contienen una lista de palabras claves con un peso numérico que indica la importancia relativa del tema o término en un documento o conjunto de los mismos (Wakil, 1999).

Asociación: En la asociación se encuentran similitudes entre los diferentes elementos de datos, objetos o eventos, donde la idea principal en la generación de reglas de asociación es identificar los conjuntos frecuentes de datos o términos que se refieren al contexto del estudio (Gary Miner & Nisbet, 2012).

Análisis de tendencias: El objetivo principal de este método es encontrar los cambios dependientes del tiempo para un objeto o evento, además de recoger información y descubrir patrones o comportamientos a partir del procesamiento de esos datos. El análisis de tendencias también permite organizar, cuantificar y disponer la información con el fin de tomar decisiones ante el comportamiento de esas tendencias. (Vickers, 1985). Este es un método exploratorio y por lo tanto siempre es necesario investigar más a fondo con el fin de encontrar mayor conocimiento sobre el entorno de estudio.

2.4 Fase IV: Desarrollo del modelo

En esta fase se seleccionan y se aplican diferentes técnicas de modelado para el mismo tipo de problema, por esta razón es importante elegir la más apropiada para el proyecto, tomando en cuenta criterios como el problema a resolver, los datos con los que se trabaja, el tiempo para la solución y conocimiento sobre proceso (Chapman & Colin, 2000).

La técnica ejecutada sobre los datos genera uno o más modelos, por lo cual llevan un conjunto de parámetros en su configuración, que determinan las características del modelo a generar.

2.5 Fase V: Evaluación de resultados

Una vez que se desarrollan y evalúan la exactitud y la calidad de los datos desde una perspectiva de análisis de datos y modelos, hay que verificar y validar la correcta ejecución de la todas las actividades. Esta fase de evaluación se realiza una conexión para asegurarse que los modelos desarrollados y verificados en realidad están

abordando el problema del negocio y el cumplimiento de los objetivos que fueron plateados al inicio del proyecto (Pete, Khabaza, & Shearer, 2000), (Gary Miner & Nisbet, 2012). Si al evaluar los resultados de los modelos nos damos cuenta que no se están cumpliendo los objetivos del negocio, entonces debemos volver atrás y corregir estos problemas antes de pasar a la fase de desarrollo, es probable que no se haya considerado los datos necesarios, para el desarrollo del proyecto.

2.6 Fase VI: Desarrollo

Una vez que el proceso de modelado ha pasado con éxito ya se puede implementar el despliegue de estos modelos, esto puede ser tan simple como generar un informe que explique las conclusiones del estudio, o puede ser tan complejo como la construcción de un nuevo sistema entorno a estos modelos, o su integración en un sistema que ya existe. (Chapman & Colin, 2000), (Molina, 2002).

Algunos modelos van a perder su exactitud y relevancia con el tiempo lo que significa que deben ser actualizados periódicamente con nuevos datos. El desarrollo de un sistema sofisticado es una tarea difícil para la auto-evaluación y auto-ajuste, pero una vez logrado los resultados son muy satisfactorios (Gary Miner & Nisbet, 2012).

CAPITULO III
DESARROLLO

A continuación se detalla cómo se desarrolló el proyecto de tesis siguiendo cada una de las fases indicadas en la metodología CRISP-MD.

3.1 Fase I: Determinar el propósito del estudio

La Universidad Técnica Particular de Loja es una institución autónoma, con la finalidad social y publica, pudiendo impartir enseñanza, desarrollar investigación con la libertad científica-administrativa y participar en los planes de desarrollo del país. (UTPL, Informacion General)

La UTPL se caracteriza por seguir las líneas generales de los sistemas de educación a distancia mundiales, ofrece la posibilidad de personalizar los procesos de enseñanza-aprendizaje con el fin de promover la formación de habilidades para el trabajo independiente y auto responsable. La eficacia del modelo de educación a distancia se sustenta con la exigencia académica y su sistema de evaluación presencial. (UTPL, Modalidad abierta y a Distancia)

Actualmente la Universidad cuenta con dos modalidades de estudio abierta y a distancia, de tal forma que contribuye con herramientas de apoyo para la enseñanza-aprendizaje entre ellas tenemos Unidad de videoconferencias, Biblioteca virtual, Centros Universitarios y Entorno Virtual de Aprendizaje (EVA), esta última dispone de cuestionarios digitales los mismos que sirven para envío de trabajos en línea por parte de los estudiantes. (Fernández, 2009) En este ámbito los cuestionarios no se encuentran asociados a los recursos educativos abiertos, videos, revistas, links a libros y otras fuentes de información que permitan a los estudiantes contribuir con el estudio, motivo por cual dificulta encontrar recursos educativos que estén relacionados directamente con los temas tratados en dichos cuestionarios digitales.

Por tanto el objetivo principal de este trabajo es clasificar las preguntas de los cuestionarios digitales aplicando técnicas de procesamiento de lenguaje natural y minería de texto. Tener clasificado las temáticas de cada pregunta será un aporte para otros temas de estudio como por ejemplo los sistemas recomendadores o de clasificación automática de preguntas ya que será más fácil identificar los recursos educativos específicos que le permitan al estudiante retroalimentar o brindar una solución a la pregunta planteada.

3.2 Fase II: Exploración de la disponibilidad y la naturaleza de los datos

Este trabajo la Minería de Texto y Aplicación de Técnicas de Inteligencia Artificial para la clasificación de preguntas en cuestionarios digitales contribuye de manera positiva con la Universidad Técnica Particular de Loja ya que presenta una guía de apoyo para la resolución de problemas de esta índole.

Para este trabajo utilizaremos los cuestionarios digitales de Matemáticas Discretas elaborados por los profesores de la asignatura, que se imparte en la titulación de Informática en la modalidad abierta y a distancia de la Universidad Técnica Particular de Loja.

En esta fase una vez que ya hemos determinado el propósito del estudio, evaluamos la disponibilidad de los datos y obtenemos toda la información que vamos a utilizar para desarrollar este proyecto.

Tabla 1. Extracto del cuestionario de Matemáticas Discretas

1. Identifique cuál de las siguientes oraciones es una proposición.
a. Los niños de ecuador.
b. 4 es número primo.
c. Me gustan las margaritas.
2. Cuál de las siguientes oraciones es una proposición.
a. 2 = 4
b. 2 + 5
c. 7 / 5
3. Identifique cuál de las siguientes oraciones es una proposición simple o atómica.
a. Cuatro no es un número primo.
b. Cinco es un número primo.
c. Seis es divide exactamente para 2 y para 3.
4. Identifique cuál de las siguientes oraciones es una proposición compuesta o molecular.
a. Cuatro es mayor de 6.
b. Cuatro no es mayor o igual a 6
c. Cuatro es un número par.
5. Una importante condición para que una oración pueda ser considerada proposición es:
a. La oración debe tener conectores.
b. La oración se le puede asignar valores de verdadero o falso

pero no ambos vez.
c. La oración puede tener signos de interrogación.

La tabla 1 muestra una parte del cuestionario digital con el cual vamos a trabajar el mismo que se encuentra en formato Excel, contiene información no estructurada, ya que las preguntas están formadas por texto, números, caracteres especiales, símbolos y otras expresiones algebraicas, consta de 100 preguntas donde cada pregunta posee 3 alternativas de respuesta las cuales se refieren a dos temas abordados en la materia de Matemáticas Discretas como son Lógica y Circuitos Combinatorios. Revisar Anexo 2, Anexo 3.

3.3 Fase III: Preparación de los datos

3.3.1 Actividad I: Establecer el corpus.

Los cuestionarios correspondientes a Matemáticas Discretas utilizados para los estudiantes de la modalidad abierta y a distancia, contienen preguntas del primer bimestre, estos son trabajados a nivel de la Titulación de Sistemas Informáticos y Computación de la UTPL. Revisar Anexo 1

Como ya se mencionó en la fase anterior para este proyecto utilizaremos los cuestionarios de Matemáticas Discretas

Luego que contamos con los datos, realizamos la preparación de modo que estén de la misma forma, para ello hacemos una limpieza de manera que facilite su posterior análisis. Se realiza la eliminación de espacios en blanco, viñetas, tildes y transforma todos los datos a palabras minúsculas con el propósito que no presenten problemas durante el desarrollo del proyecto.

3.3.2 Actividad II: Preproceso de los datos.

Como preprocesamiento de datos el primer proceso que se realizó fue la obtención de las palabras o tokenización del documento, lo cual consiste en crear un token para cada cadena que se encuentra separada por un espacio en blanco o cualquier signo de puntuación u operación matemática. Como resultado de este proceso obtenemos una cantidad bastante grande de tokens, donde cada token es un atributo del documento.

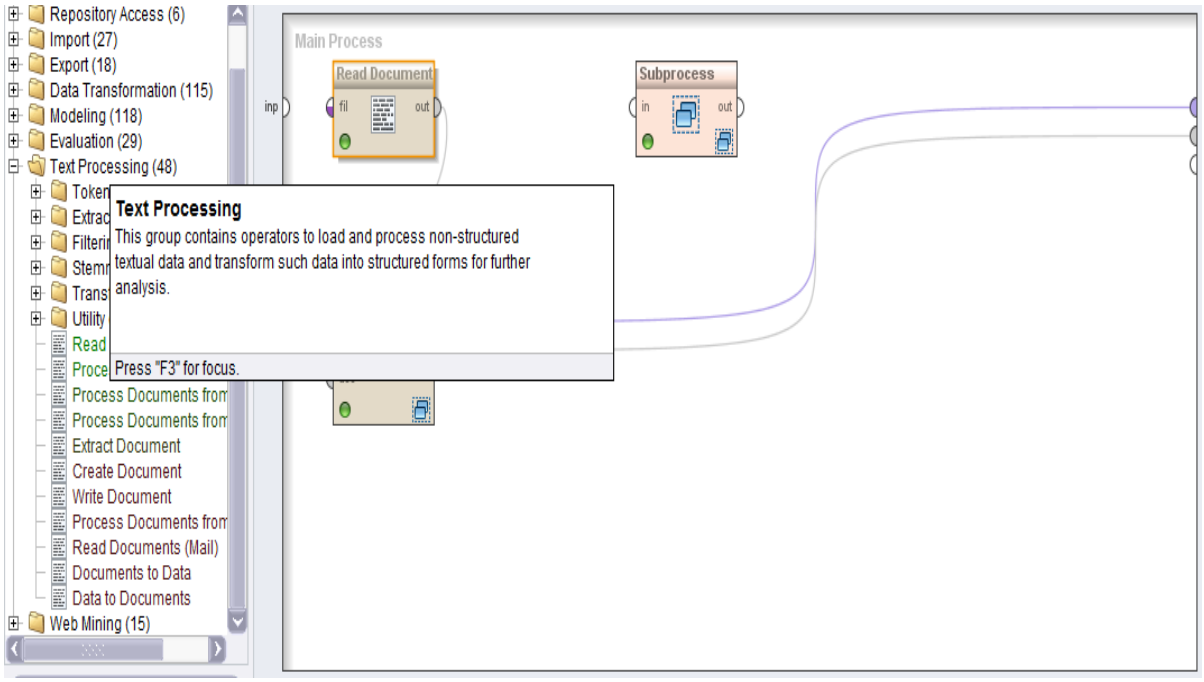


Figura 4. Plug-in de RapidMiner para el procesamiento de texto.
Fuente: Basado en los experimentos realizados en la herramienta RapidMiner.

Para realizar la tokenización se utilizó el plug-in de Rapidminer llamado procesamiento de texto tal como se puede apreciar en la Figura 3.1. Existe un operador para la tokenización que extrae palabras de texto, y tiene un parámetro que especifica cómo identificar palabras.

Dentro de la tokenización podemos utilizar otros operadores para reducir el número de palabras y filtrar palabras que no aportan con información necesaria.

Luego de la tokenización existe un operador filtering o (stop words), el mismo que nos permite filtrar las palabras sin significado, existen diferentes operadores con varios idiomas que también incluye diccionario para agregar otras palabras.

Dentro de este proceso también se ha incluido el operador stemming el mismo que transforma las ocurrencias de la misma palabra en una misma cadena, este proceso ayuda a reducir el número de tokens en el cuestionario. Hemos finalizado con el operador que realiza el filtrado de tokens, estos pueden ser más cortos o más largos en nuestro proyecto le hemos dado una longitud mínima de 3 y una máxima de 100 caracteres.

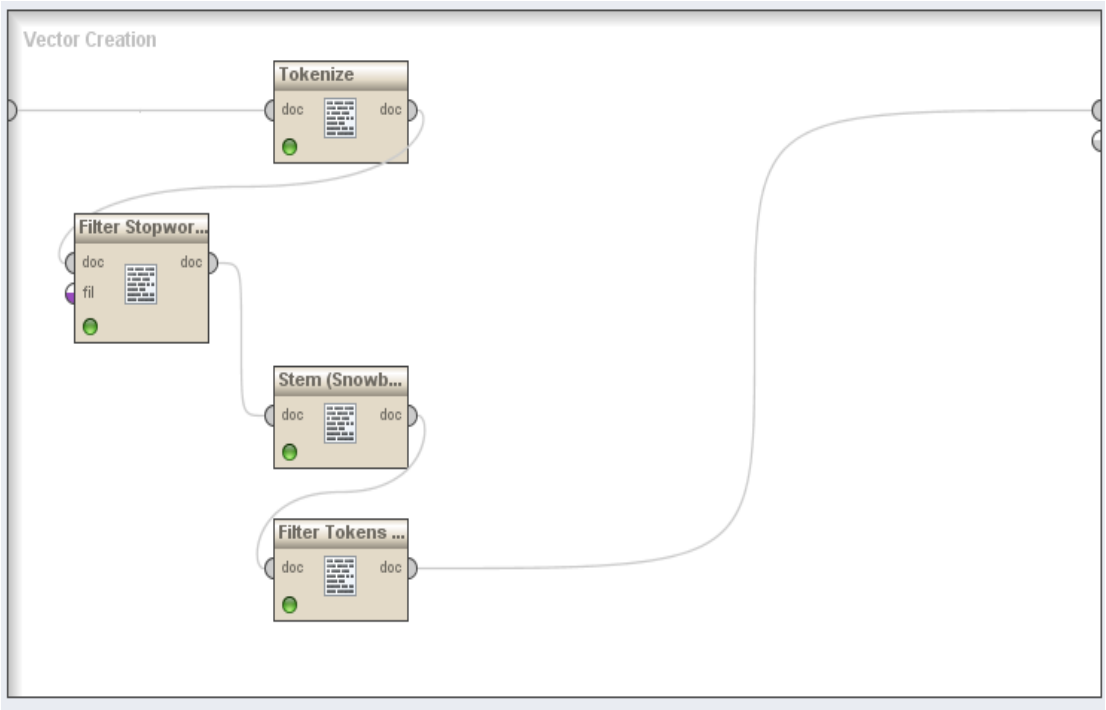


Figura 5. Preprocesamiento de texto
Fuente: Basado en los experimetos realizados en la herramienta RapidMiner.

La figura 5 muestra la ejecución de tareas como; Tokenizer, Stopwords, Stem y Filter Tokens, una vez que se han completado estas tareas se obtuvo la siguiente salida donde se encuentran cada uno de los tokens, los cuales nos ayudaron a crear la matriz TDM.

Word	Attribute Name	Total Occurences	Document Occurences
absorcion	absorcion	1	1
acot	acot	2	1
admision	admision	1	1
algebr	algebr	1	1
algun	algun	13	1
ambas	ambas	3	1
ambos	ambos	1	1
and	and	6	1
anterior	anterior	6	1
aplic	aplic	5	1
aprueb	aprueb	1	1
arbol	arbol	31	1
argument	argument	7	1
asign	asign	1	1
atom	atom	1	1
bicondicional	bicondicional	2	1
binari	binari	1	1
bits	bits	1	1
boolean	boolean	14	1
cad	cad	1	1
cant	cant	10	1
carm	carm	1	1
cas	cas	1	1

Figura 6. Resultados ejecución del Preprocesamiento de texto en RapidMiner.
Fuente: Basado en los experimetos realizados en la herramienta RapidMiner.

Luego de realizado las tareas de preprocesamiento con el software RapidMiner, se puede verificar en la Figura 6 que el modelo obtenido consta solo texto y que las expresiones y símbolos se han eliminado ya que al tener preguntas matemáticas constan de números, letras, símbolos y signos por lo tanto esta herramienta al realizar las tareas de reprocesamiento de datos ha excluido estos datos.

Por lo tanto para no perder datos que son importantes para el análisis de los mismos, se decidió construir el modelo de manera manual, donde se toma en cuenta todos los datos de manera que sean únicos y relevantes.

3.4 Fase IV: Desarrollo del modelo

En esta fase se ha construido el modelo de datos representado en una matriz en la cual se ha colocado como filas las preguntas del cuestionario de Matemáticas Discretas y como columnas los tokens o palabras en su forma base, es decir es el resultado del preprocesamiento de los datos.

Como podemos observar en la tabla 2 se muestra la representación de la matriz símbolos la misma que contiene números, letras, símbolos matemáticos y las respectivas categorías, así mismo en la tabla 3 se representa la matriz expresiones la cual está formada por números, letras, símbolos, expresiones algebraicas, caracteres especiales y al igual que matriz símbolos consta de categorías previamente establecidas, de esta forma hemos construido el modelos de datos.

Con las filas y columnas definidas para la matiz utilizamos Excel, con el objetivo de crear y llenar cada celda, tomando en cuenta que es una matriz que consta de números binarios se coloca el 1 para el caso de que dicho token esté presente en la pregunta y para el caso de que el termino no se encuentre se coloca el 0.

Luego de tener la matriz TDM completa con ceros y unos, eliminamos la primeria columna que contiene las preguntas del cuestionario de Matemáticas Discretas y se crea una columna al final de la matriz la misma que contiene categorías (Lógica, Circuitos), esto con el fin de obtener la matriz lista para realizar el análisis de los datos. Revisar Anexo 5.

La matriz TF-IDF se encuentra compuesta por dos términos;

TF (Término Frecuencia): Consiste en medir la frecuencia con que se produce un término en un documento. Puesto que cada documento posee diferente longitud, es posible que un término aparezca con más frecuencia en los documentos largos o con más contenido que en los cortos (Wu & Kwok, 2008). Por lo cual la frecuencia de los términos se calcula de la siguiente manera:

TF = (Número de veces que el término aparece en un documento) / (Número total de términos).

IDF (Frecuencia de documentos inversa): Mide la importancia de un término. En el cálculo de TF todos los términos se consideran importantes. Sin embargo se conoce que algunos términos como (en, de, que, la) pueden aparecer varias veces, pero tienen poco importancia. Para dar respuesta a esto se realiza en siguiente cálculo:

IDF: = Log (Número total de documentos) / (Numero de documentos con términos en ellas).

Para la construcción de la matriz TF-IDF tal como se muestran la tabla 4 se realizó, como primer paso el cálculo del término frecuencia, luego se procedió a calcular la frecuencia inversa de documentos y finalmente se obtiene mediante el producto TF x IDF que es el peso de los términos que aparecen frecuentemente en las preguntas del cuestionario de Matemáticas Discretas. Revisar Anexo 6.

Tabla 2. Representación de los datos Matriz TDM Símbolos

	absorcion	***acotado***	***…***	**+**	**-**	*****	**/**	**=**	***categorías***
Pregunta 1	0	0	…	0	0	0	0	0	logica
Pregunta 2	0	0	…	1	0	0	1	1	logica
Pregunta 3	0	0	…	0	0	0	0	0	logica

Pregunta 4	0	0	…	0	0	0	0	0	circuitos
Pregunta ..	0	0	…	0	0	0	0	0	circuitos
Pregunta n	0	0	…	0	0	0	0	0	circuitos

Tabla 3. Representación de los datos Matriz TDM Expresiones

	absorcion	***acotado***	**…**	**vx (h(x) v - c(x))**	**- Vx (h(x) -> - c(x))**	**(x1 v x2) v x3**	**x b.y c.x+y**	**(x1 Λ x2) v (x3 Λ x1)**	***categorías***
Pregunta 1	0	0	…	0	0	0	0	0	logica
Pregunta 2	0	0	…	1	0	0	1	1	logica
Pregunta 3	0	0	…	0	0	0	0	0	logica
Pregunta 4	0	0	…	0	0	0	0	0	circuitos
Pregunta ..	0	0	…	0	0	0	0	0	circuitos
Pregunta n	0	0	…	0	0	0	0	0	circuitos

Tabla 4. Representación de los datos Matriz TF-IDF

	absorcion	***Acotado***	**…**	**Verdader**	**vez**	**viaj**	**vicevers**	**xyz**
Pregunta 1	1	2	…	11	0	2	1	1
Pregunta 2	0	0	0	0	0	0	0	0
Pregunta 3	1	4	0	0	2	0	2	0
Pregunta 4	0	0	0	0	0	1		0
Pregunta ..	0	0	0	0	0	0	0	0
Pregunta n	0	0	0	0	0	0	0	0

3.4.1 Weka.

Para la parte de experimentación o aplicación de algoritmos de clasificación se considera la herramienta WEKA (Waikato Environment for Knowledge Analysis), la misma que es una herramienta de experimentación y análisis de datos mediante la aplicación de técnicas y algoritmos relevantes de análisis de datos y aprendizaje automático. Trabaja con un formato denominado arff, acrónimo de Atribute Relación File Formato.

Además proporciona soporte para todo el proceso experimental, es decir permite realizar actividades como; preparación evaluación y visualización de datos ya que cuenta con diferentes métodos de clasificación, regresión, clustering y reglas de asociación.

Otro aspecto importante de Weka es que nos permite cargar información de nuestros datos de diferentes maneras ya sea bases de datos, a partir de URL o a partir de un archivo que se haya creado de manera manual o automática.

Weka implementa algoritmos que pueden aplicarse para realizar preprocesamiento de datos para transformarlos en un esquema de aprendizaje a fin de que los resultados puedan ser analizados de manera fácil, es decir permite aplicar métodos de aprendizaje a conjuntos de datos y analizar los resultados para extraer información (Ian H. Witten, 2011).

3.4.1.1 Ventajas

- ✓ Weka proporciona una interfaz para la comunicación con el usuario CLI (Simple Client Interfaz), es decir es una interfaz que proporciona una consola de tal forma que sea mucho más fácil la introducción de datos (Basilio, 2006).
- ✓ Permite ubicar patrones de comportamiento de la información de tal manera que sea mucho más sencilla la toma de decisiones.
- ✓ Se encuentra disponible de manera libre bajo licencia GNU.
- ✓ Es completamente portable ya que se encuentra implementado en Java y puede correr en cualquier plataforma.
- ✓ Contiene una extensa colección de técnicas para realizar actividades de reprocesamiento, modelado, análisis y evaluación de datos.

3.4.1.2 Desventajas

- ✓ Es muy complicado manejar ya que es necesario tener un conocimiento muy amplio de esta aplicación.
- ✓ No incluye algoritmos para el modelado de secuencias
- ✓ Al utilizar métodos de combinación de modelos, los resultados tienden a complicarse haciendo que su comprensibilidad y sea más difícil.

3.4.2 Algoritmos de clasificación en weka.

Weka permite aplicar, analizar y evaluar las técnicas más relevantes del análisis de datos principalmente en aquellas que provienen del aprendizaje automático sobre un conjunto de datos. Está constituido por una serie de paquetes con diferentes técnicas de preprocesado, clasificación, clusterizacion, asociacion, y visualizacion además de las facilidades de su aplicación y analalis.
En este trabajo utilizaremos algunos de los algoritmos que dispone Weka para la clasificación, entre los que hemos utilizado estan los siguintes:

Naive Bayes: Es un clasificador sencillo probabilístico basado en aplicar el teorema de Bayes que asume la presencia o usencia de una característica particular de una clase. Naive Bayes utiliza una técnica de clasificación y predicción que contribuye modelos que predicen la probabilidad de posibles resultados.
Este algoritmo centra su fundamento en la hipótesis de que todos los atributos son independientes entre sí, además representa una distribución de una mezcla de componentes, donde cada componente dentro de todas las variables se asumen independientes, es decir la hipótesis de independencia da lugar a un modelo de un único nodo raíz correspondiente a la clase y en el cual todos los atributos son nodos hoja que tienen como origen la variable en este caso la clase. (González, Castellón, & M, 2009)**.**
Al ser robusto, rápido, preciso y fácil de implementar se utilizan en muchos campos como; diagnóstico de enfermedades, estudios taxonómicos, filtrado de spam en clientes de correo electrónico entre otros. Una de las ventajas del algoritmo Bayes ingenuo es que requiere una pequeña cantidad de datos de entrenamiento para estimar parámetros necesarios para la clasificación (P & Stutz, 1996)**.**

Naive Bayes Multinomial: Es un algoritmo que asume independencia entre los términos luego de haber conocido la clase a la que pertenecen. Además este algoritmo permite tener en cuenta no solo los términos que aparecen cada documento sino también la frecuencia de aparición de cada término.
Es uno de los modelos probabilísticos más simples y más usados en la clasificación de texto ya que produce resultados muy eficientes (Hernandez, 2009).

IBk: Este algoritmo está basado en instancias por lo cual consiste únicamente en almacenara los datos presentados.
Cuando una nueva instancia es encontrada un conjunto de instancias similares relacionadas es devuelto desde la memoria y usado para clasificar la instancia consultada (Valiente & Cebrián). Se trata de un algoritmo del método lazy learning ya que este método de aprendizaje se basa en que los módulos de clasificación mantienen en memoria una selección de ejemplos sin crear ningún tipo de abstracción en forma de reglas o de árboles de decisión (por lo cual es llamado, lazy perezosos), esto hace que cada vez que una nueva instancia es encontrada, se calcula su relación con los ejemplos previamente guardados con el objetivo de asignar un valor de la función objetivo para la nueva instancia.

Kstar: Este algoritmo determina cuales son las instancias más parecidas, este puede utilizar la entropía o contenido de información de las instancias como medida de distancia entre ellas.
En cuanto a las características principales se destaca que permite trabajas con atributos numéricos y simbólicos así como pesos por cada instancia, además permite que la clase sea simbólica o numérica (Pérez & Carranza, 2008)**.**

J48: Este algoritmo es una implementación realizada por Weka basado en el algoritmo conocido como C4.5, el cual forma parte también de los árboles de decisión, la característica principal es que incorpora una poda del árbol de clasificación una vez que haya sido inducido, es decir una vez que se ha construido el árbol de decisión se podan aquellas ramas del árbol con menor capacidad predictiva (Valiente & Cebrián)**,** (Ian & Eibe, 2005)**.**
Este algoritmo presenta nuevas funcionalidades con respecto del C4.5 tales como permitir la realización del proceso de pos-poda del árbol mediante el método basado en la reducción del error (reducedErrorPruning) o que las divisiones sobre las variables discretas sean siempre binarias (binarySplits).

Weka nos presenta algunos algoritmos de los cuales utilizaremos el algoritmo basado en arboles de decisión (J48), el mismo que es una implementación del algoritmo C4.5 donde la función utilizada es representada mediante un árbol de decisión con reglas "if-then" (Pradenas, 2012) este algoritmo principalmente hace uso del mecanismo de poda con el método *reduceErrorPrunning* que se encarga de la reducción del error, con el fin que el modelo sea más comprensible.

Permite trabajar con valores continuos para los atributos, separando los posibles resultados en dos ramas y escoge un rango de medida apropiada. El algoritmo (J48) se basa en la utilización del criterio de ganancia lo cual consigue evitar que las variables con mayor número de posibles de posibles valores salgan beneficiadas en la selección.

Considera todas las pruebas posibles que puede dividir el conjunto de datos y selecciona la prueba que genera mayor obtención de información.

En cuanto a las ventajas de este algoritmo podemos mencionar su sencillez y la facilidad para la interpretación, permite el manejo instancias ponderadas por pesos, se puede utilizar un árbol podado y no podado. Como principal desventaja no permite ser actualizado de forma incremental, es decir no permite añadir nuevos datos sin reclasificar los anteriores (Kirkby, 2003).

LTM: (Logistic Model Trees) consiste básicamente en una estructura de una árbol de decisión con funciones de regresión logística en las hojas. Para atributos numéricos, el nodo tiene dos nodos hijos y la prueba consiste en comparar el valor del atributo con un umbral, por ejemplo puede clasificar los datos menores en la rama izquierda mientras que los valores mayores en la rama derecha. (Preisach & Decker, 2007).

PART: Este algoritmo genera una lista de decisión sin restricciones usando el procedimiento divide y vencerás. Además construye un árbol de decisión parcial para obtener una regla.

Por otro lado para poder podar una rama es necesario que todas sus implicaciones seas conocida. El algoritmo PART permite manejar instancias ponderadas por pesos, puede procesar datos categóricos y forma reglas a partir de los árboles de decisión parcialmente podados construidos usado los heurísticos de C4.5. (Hall & Eibe, 2011)**.**

Un aspecto importante a tomar en cuenta al utilizar este algoritmo es que no puede ser actualizado de forma incremental, es decir no soporta añadir nuevos datos sin realizar la reclasificación de los datos anteriores.

3.4.3 Algoritmos de cluterización de weka.

K-Medias: Este algoritmo se encuentra clasificado como método de particionado y recolocación. Este método es el más utilizado en aplicaciones científicas e industriales, el nombre viene ya que representa cada uno de los clúster por la media (o media ponderada) de sus puntos, dicho de otra forma por su centroide (Hernández, María, & Aránzazu). La representación mediante centroides tiene una ventaja ya que tiene un significado gráfico y estadístico inmediato.
Este algoritmo usa como función objetivo la suma de las discrepancias entre un punto y su centroide expresado a través de la distancia. El K-Medias es el algoritmo de agrupamiento que s e encuentra entre uno de los más veloces y eficientes aunque como desventaja al que destacar que es uno de los más limitados.

EM: Este algoritmo asigna a cada instancia una distribución de probabilidad de pertenencia de cada clúster. Puede predecir cuantos clúster crear basado en validación cruzada o se le puede especificar a priori cuantos debe generar. Este algoritmo utiliza el modelo Gaussiano finto de mezclas asumiendo que todos los atributos son variables aleatorias independientes (Jiménez, 2008).

Farthest First: Este algoritmo comienza seleccionando aleatoriamente una instancia que pasa a ser el centroide del clúster. Calcula la distancia entre cada una de las instancias y el centro. La distancia que encuentre más alejada del centro es seleccionada como el nuevo centro del clúster (Pinar, 2007).

En tabla 9 y 10 se muestran algunos de los resultados obtenidos de los experimentos realizados mediante algoritmos de Clusterización tanto con la Matriz Símbolos como con la matriz de datos Expresiones.

Tabla 5. Resultados aplicando algoritmos de clasificación de Weka con la Matriz Símbolos.

ALGORITMOS DE CLASIFICACIÓN											
DATOS	***PARÁMETROS DE EVALUACIÓN DE ALGORITMOS***					***CONFIGURACIONES DE PARÁMETROS EN ALGORITMOS***				***MATRIZ DE CONFUSIÓN***	
MATRIZ TDM, compuesta por 0 y 1	**Algoritmo**	**Precisión**	**Recall**	**Accuary**	**Error**	**Porcentaje de Split**	**Datos de entrenamiento**	**Datos de prueba**	**Otras configuraciones**	**Instancias correctamente clasificadas**	**Instancias incorrectamente clasificadas**
Dataset Símbolos	BayessianLogisticRegresion	0.808	0.722	0.028	0.972	66%	66	34	normalizeData = **false**	27 (79.4%)	21 (20.5%)
Dataset Símbolos	ComplementNaiveBayes	0.847	0.824	0,91	0,09	66%	66	34	normalizeWordWeights = **false**	28 (82.3%)	6 (17.6%)
Dataset Símbolos	NavieBayesMultinomial	0.847	0.824	0.436	0,564	66%	66	34	Default	28 (82%)	6 (17.4%)
Dataset Símbolos	DMNtext-I 1	0.847	0.824	0.823	0.177	66%	66	34	Default	28 (82%)	6 (17.6%)

Dataset Símbolos	Naive Bayes Multinomial lUpdateable	0.847	0.824	0.823	0.177	66%	66	34	Default	28 (82%)	6 (17.6%)
Dataset Símbolos	Lazy.IB1	0.786	0.765	0.735	0.265	66%	66	34		25 (73.5%)	9 (26.4%)
Dataset Símbolos	Lazy.IBk	0.793	0.735	0.735	0.265	66%	66	34	nearestNeighbourSearchAlgorithm = **LinearNNSearch**	25 (73.5%)	9 (26.4%)
Dataset Símbolos	Lazy.KStar	0.808	0.794	0.764	0.236	66%	66	34	Default	27 (79%)	7 (20%)
Dataset Símbolos	J48	0.826	0.824	0.823	0,17	66%	66	34	reducedErrorPruning = **false**	28 (82.3%)	7 (20%)
Dataset Símbolos	J48	0.833	0.926	0.86	0.14	50%	50	50	reducedErrorPruning = **false**	43 (86%)	7 (14%)
Dataset Símbolos	LMT	0.826	0.824	0.823	0.17	66%	66	34	minNumInstances = **15**	43 (86%)	6 (17.6%)

Dataset Símbolos	PART	0.826	0.824	0.823	0.17	66%	66	34	reducedErrorPruning = **false**	28 (82.3%)	6 (17%)

Tabla 6. Resultados utilizando algoritmos de clasificación de Weka con la Matriz Expresiones.

ALGORITMOS DE CLASIFICACIÓN											
DATOS	***PARÁMETROS DE EVALUACIÓN DE ALGORITMOS***					***CONFIGURACIONES DE PARÁMETROS EN ALGORITMOS***				***MATRIZ DE CONFUSIÓN***	
MATRIZ TDM, compuesta por 0 y 1	**Algoritmo**	**Precisión**	**Recall**	**Accuary**	**Error**	**Porcentaje de Split**	**Datos de entrenamiento**	**Datos de prueba**	**Otras configuraciones**	**Instancias correctamente clasificadas**	**Instancias incorrectamente clasificadas**
Dataset Expresiones	BayessianLogisticRegresion	0.645	0.647	0.647	0.353	66%	66	34	normalizeData = **false**	22 (64%)	12 (35%)
Dataset Expresiones	ComplementNaiveBayes	0.847	0.824	0,824	0,176	66%	66	34	normalizeWordWeights = **false**	28 (82.3%)	6 (17.6%)
Dataset Expresiones	NavieBayes	0.867	0.853	0.852	0,148	66%	66	34	Default	29 (85%)	5 (14.7%)
Dataset Expresiones	NavieBayesUpdateable	0.867	0.853	0.852	0,148	66%	66	34	Default	29 (85%)	14 (17.6%)

Dataset Expresiones	Lazy.IB1	0.793	0.735	0.735	0.265	66%	66	34	Default	25 (73.5%)	9 (26.4%)
Dataset Expresiones	Lazy.IBk	0.793	0.735	0.735	0.265	66%	66	34	nearestNeighbourSearchAlgorithm = **LinearNNSearch**	25 (73.5%)	9 (26.4%)
Dataset Expresiones	Lazy.KStar	0.793	0.794	0.735	0.265	66%	66	34	Default	25 (73.5%)	9 (26.4%)
Dataset Expresiones	Lazy.LWL	0.682	0.647	0.647	0.353	66%	66		reducedErrorPruning = **false**	22 (64.7%)	12 (35.2%)
Dataset Expresiones	J48	0.799	0.794	0.794	0.206	66%	66	34	reducedErrorPruning = **false**	27 (79.4 %)	7 (20.5%)
Dataset Expresiones	J48	0.831	0.82	0.82	0.118	50%	50	50	reducedErrorPruning = **false**	41 (82 %)	9 (18%)
Dataset Expresiones	LMT	0.765	0.765	0.764	0.236	66%	66	34	minNumInstances = **15**	41 (82 %)	9 (18%)
Dataset Expresio	PART	0.799	0.794	0.794	0.206	66%	66	34	reducedErrorPruning = **false**	26 (76,4 %)	8 (26.5%)

nes											

Tabla 7. Resultados con la configuración Cross-Validation utilizando la Matriz Símbolos.

ALGORITMOS DE CLASIFICACIÓN										
DATOS	***PARÁMETROS DE EVALUACIÓN DE ALGORITMOS***					***CONFIGURACIONES DE PARÁMETROS EN ALGORITMOS***			***MATRIZ DE CONFUSIÓN***	
MATRIZ TDM, compuesta por 0 y 1	**Algoritmo**	**Precisión**	**Recall**	**Accuary**	**Error**	**Cross-Validation**	**Numero de Instancias**	**Otras configuraciones**	**Instancias correctamente clasificadas**	**Instancias incorrectamente clasificadas**
Dataset Símbolos	BayessianLogisticRegresion	0.923	0.92	0.92	0.08	10	100	normalizeData = **false**	92 (92%)	8 (8%)
Dataset Símbolos	Naive Bayes	0.871	0.87	0.87	0.13	10	100	AtributeSelection =30	87 (92%)	13 (8%)
Dataset Símbolos	ComplementNaiveBayes	0.918	0.93	0.93	0,7	10	100	Default	93 (93%)	7 (7%)
Dataset Símbolos	Naive Bayes Multinomial	0.91	0.9	0.9	0,1	10	100	Default	90 (90%)	10 (10%)

Dataset Símbolos	DMNtext-I 1	0.914	0.91	0.91	0,09	10	100	Default	91 (91%)	9 (9%)
Dataset Símbolos	Naive Bayes Multinomial lUpdateable	0.916	0.9	0.9	0,1	10	100	Default	90 (90%)	10 (10%)

Tabla 8. Resultados con la configuración Cross-Validation utilizando la Matriz Expresiones.

ALGORITMOS DE CLASIFICACIÓN										
DATOS	***PARÁMETROS DE EVALUACIÓN DE ALGORITMOS***					***CONFIGURACIONES DE PARÁMETROS EN ALGORITMOS***			***MATRIZ DE CONFUSIÓN***	
MATRIZ TDM, compuesta por 0 y 1	**Algoritmo**	**Precisión**	**Recall**	**Accuary**	**Error**	**Cross-Validation**	**Numero de Instancias**	**Otras configuraciones**	**Instancias correctamente clasificadas**	**Instancias incorrectamente clasificadas**
Dataset Símbolos	Bayes Net	0.845	0.84	0.84	0,16	10	100	Default	84 (84%)	16(16%)
Dataset Símbolos	ComplementNaiveBayes	0.926	0.92	0.92	0,08	10	100	Default	92 (92%)	8 (8%)
Dataset Símbolos	Naive Bayes Updateable	0.91	0.9	0.9	0,1	10	100	Default	90 (90%)	10 (10%)
Dataset Símbolos	Lazy.kstar	0.895	0.88	0.87	0,13	10	100	Default	88 (88%)	12 (12%)
Dataset Símbolos	J48	0.91	0.91	0.91	0,09	10	100	Default	91 (91%)	9 (9%)

Tabla 9. Resultados utilizando algoritmos de Clusterización de Weka con la Matriz Símbolos.

ALGORITMOS DE CLUSTERIZACIÓN									
MATRIZ DE DATOS	***CONFIGURACIONES DE PARÁMETROS DEL ALGORITMO***				***MODO CLÙSTER***		***RESULTADOS***		
	Algoritmo	**Número Clústeres**	**Interacciones Máximas**	**Conservar el orden de las Instancias**	**Porcentaje Split**	**Use training set (Usar el conjunto de entrenamiento)**	**Modo Prueba**	**Interacciones realizadas**	**Instancias Clusterizadas**
Dataset Símbolos	SimpleKMeans	4	500	false	66%	**X**	Divide 66% para entremaniemto y el resto de prueba (34)	5	Cluster# 0 0 (0%) Cluster# 1 8 (24%) Cluster# 2 25 (74%) Cluster# 3 1 (3%)
Dataset Símbolos	SimpleKMeans	3	500	false	66%	**X**	Divide 66% para entremaniemto y el resto de prueba (34)	5	Cluster# 0 0 (0%) Cluster# 1 8 (24%) Cluster# 2 26 (76%)
Dataset Símbolos	SimpleKMeans	2	500	false	66%	**X**	Divide 66% para entremaniemto y el resto de prueba (34)	2	Cluster# 0 0 (0%) Cluster# 1 34 (100%)
Dataset Símbolos	SimpleKMeans	4	500	false	X	✓	Evalua con los datos de	6	Cluster# 0 58 (58%) Cluster# 1 1 (1%)

							entrenamiento		Cluster# 2 1 (1%) Cluster# 3 40 (40%)
Dataset Símbolos	SimpleKMeans	3	500	false	X	✓	Evalua con los datos de entrenamiento	2	Cluster# 0 98 (98%) Cluster# 1 1 (1%) Cluster# 2 1 (1%)
Dataset Símbolos	SimpleKMeans	2	500	false	X	✓	Evalua con los datos de entrenamiento	2	Cluster# 0 99 (99%) Cluster# 1 1 (1%)

Tabla 10. Resultados utilizando algoritmos de Clusterización de Weka con la Matriz Símbolos.

ALGORITMOS DE CLUSTERIZACIÓN									
MATRIZ DE DATOS	***CONFIGURACIONES DE PARÁMETROS DEL ALGORITMO***				***MODO CLUSTER***		***RESULTADOS***		
	Algoritmo	**Número Clústeres**	**Interacciones Máximas**	**Conservar el orden de las Instancias**	**Porcentaje Split**	**Use training set (Usar el conjunto de entrenamiento)**	**Modo Prueba**	**Interacciones realizadas**	**Instancias Clusterizadas**
Dataset Expresiones	SimpleKMeans	4	500	false	X	✓	Evalua con los datos de entrenamiento	4	Cluster# 0 60 (60%) Cluster# 1 1 (1%) Cluster# 2 1 (1%) Cluster# 3 38 (38%)
Dataset Expresiones	SimpleKMeans	3	500	false	X	✓	Evalua con los datos de entrenamiento	3	Cluster# 0 98 (98%) Cluster# 1 1 (1%) Cluster# 2 1 (1%)
Dataset Expresiones	SimpleKMeans	2	500	false	X	✓	Evalua con los datos de entrenamiento	2	Cluster# 0 99 (99%) Cluster# 1 1 (1%)
Dataset Expresiones	SimpleKMeans	4	500	false	66%	**X**	Divide 66% para entremaniemto y el resto de prueba (34)	5	Cluster# 0 0 (0%) Cluster# 1 0 (0%) Cluster# 2 4 (12%) Cluster# 3 30 (88%
Dataset Expresion	SimpleKMea	3	500	false	66%	**X**	Divide 66% para entremaniemto	2	Cluster# 0 0 (0%) Cluster# 1 4 (12%)

es	ns						y el resto de prueba (34)		Cluster# 2 30 (88%)
Dataset Expresiones	SimpleKMeans	2	500	false	66%	**X**	Divide 66% para entremaniemto y el resto de prueba (34)	2	Cluster# 0 3 (9%) Cluster# 1 31 (91%)

Tabla 11. Mejores resultados aplicando filtros de Weka y algoritmos de clasificación.

ALGORITMOS DE CLASIFICACIÓN										
Matriz con 100 preguntas, 201 instancias y compuesta por 0 y 1		**CONFIGURACION FILTROS DE WEKA**				**CONFIGURACIÓN DE PARAMETROS DEL ALGORITMO**			**RESULTADOS**	
FILTROS	**ATRIBUTOS**	**EVALUACIÓN**	**BÚSQUEDA**	**NUMERO FILAS**	**INSTANCIAS**	**ATRIBUTOS**	**Algoritmo**	**SPLIT**	**CORRECTAMENTE**	**INCORRECTAMENTE**
Supervisado	AttributeSelection	CfsSubsetEval	RankSearch	Automático	100	14	J48	-	43 (86%)	7 (14%)
Supervisado	AttributeSelection	FilteredAttributeEval	Ranker	10	100	11	J48	-	41 (82%)	9(18%)
Supervisado	AttributeSelection	FilteredSubsetEval	RankSearch	Automático	100	9	J48	5O%	40 (80%)	10 (20%)
Supervisado	AttributeSelection	OneRAttributeEval	Ranker	10	100	21	J48	5O%	36 (80%)	9 (20%)
Supervisado	AttributeSelection	SVMAttributeEval	Ranker	0	100	201	J48	50%	41 (82%)	9 (18%)
Supervisado	AttributeSelection	SVMAttributeEval	Ranker	10	100	21	J48	50%	43 (86%)	7 (14%)
Supervisado	AttributeSelection	SymmetricalUncertAttrributeEval	Ranker	0	100	201	J48	50%	40 (80 %)	10 (20%)
Supervisado	AttributeSelection	SymmetricalUncertAttrributeEval	Ranker	10	100	21	J48	50%	40 (80 %)	10 (20%)
Supervisado	AttributeSelection	WrapperSubsetEval	Ranker	0	100	201	J48	50%	33 (66 %)	17 (34%)

Tabla 12. Mejores resultados aplicando filtros de Weka y algoritmos de Clusterización

CONFIGURACIONES					RESULTADOS
DATOS	**FILTRO**	**NUMERO CLUSTERS**	**SPLIT**	**ALGORITMO**	**CLUSTER**
Matriz con 100 preguntas y 201 instancias. Contenido binario (0,1)	StringtoNominal	2	50%	SimpleKMeans	Clúster# 0 49 (98%) Clúster# 1 1 (2%)
	StringtoNominal	3	50%	SimpleKMeans	Clúster# 0 4 (8%) Clúster# 1 1 (2%) Clúster# 2 45 (90%)
	StringtoNominal	4	50%	SimpleKMeans	Clúster# 0 1 (2%) Clúster# 1 1 (2%) Clúster# 2 12 (24%) Clúster# 3 36 (72%)
	Standarize	2	50%	EM	Clúster# 0 22 (44%) Clúster# 1 28 (56%)
	Standarize	3	50%	EM	Clúster# 0 15 (30%) Clúster# 1 34 (68%) Clúster# 2 1 (2%)
	Standarize	4	50%	EM	Clúster# 0 15 (30%) Clúster# 1 5 (20%) Clúster# 2 15 (40%) Clúster# 3 22 (10%)
	RandomProjection	2	50%	EM	Clúster# 0 37 (74%) Clúster# 1 13 (26%)
	RandomProjection	3	50%	EM	Clúster# 0 18 (36%) Clúster# 1 9 (18%) Clúster# 2 23 (46%)
	StringtoNominal	2	66%	FarthestFirst	Clúster# 0 32 (94%) Clúster# 1 2 (6%)
	StringtoNominal	3	66%	FarthestFirst	Clúster# 0 31 (91%) Clúster# 1 2 (6%) Clúster# 2 1 (3%)
	StringtoNominal	4	66%	FarthestFirst	Clúster# 0 28 (82%) Clúster# 1 2 (6%)

					Clúster# 2 1 (3%) Clúster# 3 3 (9%)
	StringtoNominal	10	66%	FarthestFirst	Clúster# 0 16 (47%) Clúster# 1 1 (3%) Clúster# 2 1 (3%) Clúster# 3 2 (6%) Clúster# 4 2 (6%) Clúster# 7 7 (21%) Clúster# 8 1 (3%) Clúster# 9 4 (12%)

3.5 Fase V: Evaluación de resultados

Para realizar el análisis de la categorización de documentos como ya se mencionó en la fase anterior se estableció algunos parámetros de evaluación de los algoritmos con el propósito de contrastar y poder seleccionar el más adecuado para este tipo de trabajos, especialmente cuando se trabaja con datos matemáticos (Jiménez María G).

3.5.1 Parametros de evaluación.

En la tabla 13 podemos verificar cada una de las fórmulas para calcular los parámetros de evaluación de los algoritmos utilizados en este trabajo.

- ✓ **Precisión:** Esta métrica es denominada como la probabilidad de que un documento cualquiera sea clasificado bajo una categoría.
- ✓ **Recall:** Es una medida de capacidad de un modelo de predicción para seleccionar instancias de una clase determinada de un conjunto de datos. Esta medida también se la conoce como sensibilidad.
- ✓ **Accuary:** Es la exactitud global del modelo y se calcula como la suma de clasificaciones correctas dividido por el número total de clasificaciones.
- ✓ **Error:** Es un término que describe el grado de errores encontrados durante la clasificación de los datos, cuando el porcentaje de error es mayor que la precisión podemos decir que el algoritmo no es el correcto para dicha actividad.
- ✓ **Matriz de confusión**: Está formada por filas y columnas, donde el número de instancias correctamente clasificadas es la suma de la diagonal principal de la matriz y el resto están clasificadas de forma incorrecta (Corso C. L., 2009).
  La tabla 14 ilustra la estructura de la matriz de confusión.

Tabla 13. Fórmulas de los parámetros de evaluación de algoritmos.

FORMULAS EVALUACION ALGORITMOS	
PRECISION	*TPi/(TPi+FPi)*
RECALL	*TPi/(TPi+FNi)*
ACCUARY	*(TPi+TNi)/Ni*
ERROR	*(FPi+FNi) = 1 - Accuracyi* *Ni*

Tabla 14. Estructura de la Matriz de Confusión.

Matriz de Confusion		Clases
(VP) Verdaderos Positivos	(FN) Falsos Negativos	Positivos
(FP) Falsos Positivos	(VN) Verdaderos Negativos	Negativos.

3.5.2 Configuraciones de algoritmos.

Use training set: Con esta opción Weka entrenará el método con todos los datos disponibles y luego lo aplicará otra vez sobre los mismos.

Cross-validation: Seleccionando esta opción se realizará una validación cruzada estratificada del número de particiones dado (Folds). Esta validación cruzada consiste en dado un número n se dividen los datos en n partes, donde por cada parte se construye el clasificador con las n-1 partes restantes con la cual se realiza la prueba. Así se realiza por cada una de las partes (Corso C. L., 2009).

Porcentaje Split: Es configuración es en la cual se evalúa la calidad del clasificador según lo bien que clasifique un porcentaje de los datos que se reserva para test.

Para este trabajo en cuanto a las configuraciones de los algoritmos en weka para el análisis de los modelos de datos tanto en la clasificación como clusterizacion trabajaremos con paquete weka.filters para el preprocesado de datos, así como Cross-validation y Porcentaje Split para la configuración de datos a procesar.

3.5.3 Análisis de los resultados aplicando filtros de weka en los datos.

El paquete weka.filters tiene que ver con las clases que transforman los datos en conjuntos eliminado o añadiendo atributos de tal manera que brinda opciones para especificar el conjunto de datos de entrada. Este paquete llamado weka.filters se organiza o se dividen en dos grupos supervisadas y no supervisadas, estos a su vez se subdividen en instancia y filtrado respectivamente.

De los 12 algoritmos de Weka con los cuales hemos realizado las respectivas pruebas podemos visualizar en la tabla 11 que el “CfsSubsetEval” y aplicando como parámetro “RankSearch”, donde se eligen automáticamente los datos, tenemos como instancias correctamente clasificadas 43 que corresponden al 86%, y 7 incorrectamente clasificadas que representan el 14%. En la matriz de confusión los resultados nos

muestran que se han clasificado 26 instancias como categoría Lógica y en dicha fila existe (1) instancia como categoría Circuitos. En la segunda fila de la matriz aparecen 16 instancias clasificadas como Circuitos y (7) como Lógica. Todos estos son resultados obtenidos con 100 instancias y con un número de filas de manera automática.

Por otro lado dentro de las instancias con más alto porcentaje de clasificación de manera correcta se encuentra el algoritmo “SVMAttributeEval” y con parámetros “Ranker” o (Búsqueda) definido con 10 atributos, nos muestra de igual forma 43 instancias correctamente clasificadas que corresponden al 86%, y 7 instancias incorrectamente clasificadas que representan al 14% de los datos. Dentro de este experimento como podemos observar en la tabla 3.4.9 existe similitud con “CfsSubsetEval”en lo que se refiere a instancias clasificadas, pero en la matriz de confusión existe una leve diferencia ya que en la primera fila de la matriz 25 instancias se encuentran clasificados como Lógica y en la segunda fila existen 5 instancias categorizadas como Circuitos.

En lo referente a la clusterización los algoritmos de agrupamiento buscan instancias con características similares según un criterio de comparación entre de atributos de instancias definidos. Para la preparación de los datos tomando en cuenta la clusterizacion se ha utilizado el algoritmo “StringtoNominal” para la experimentación, tal como se muestra en la Tabla 3.4.10 en el cual se han manejado 100 instancias y 200 atributos, además se ha realizado las pruebas con 1, 2, 3 y 4 clúster. En este contexto aplicando el filtro “StringtoNominal”, con Split de un 50% y con la ayuda del algoritmo de clusterizacion “SimpleKMeans”.

El algoritmo K-Medidas es el método más utilizado en aplicaciones científicas e industriales. Este método es uno de los más veloces y eficientes, pero también tiene algunas limitantes ya que precisa únicamente las categorías que son similares. Cuando se realiza la aplicación de algoritmo “SimpleKMeans” con 2 clústeres tenemos como resultados que en el clúster 1, se han agrupado 49 instancias las cuales corresponden al 98% y en el segundo clúster tenemos una instancia que representa el 2% de los datos, dentro de esta clusterizacion se debe mencionar que es el clúster con mayor porcentaje de instancias agrupadas.

En el experimento con 3 clústeres tenemos como salida que en el tercer clúster se representa 36 instancias que corresponden al 72% de los datos. Por otra parte también se ha utilizado “Standarize” para la preparación de los datos y para la clusterización el algoritmo EM, el mismo que cuando se realiza agrupación con tres

clústeres podemos observar que en el clúster uno 34 instancias son el 68% de los datos.

Para realizar de reducción de la dimencionalidad WEKA presenta “RandomProjection”, el mismo que esta experimentación hemos utilizado el algoritmo de clusterizacion “EM”, aplicado el 50% de Split y con la reducción de datos nos quedan 13 atributos más significativos, como resultados con mayor agrupamiento tenemos en el clúster tres con 23 instancias que representan el 46% de los datos como se muestra en la tabla 12.

3.5.4 Análisis de los resultados utilizando la matriz símbolos.

Luego del experimento con la matriz símbolos la misma que contiene palabras números, símbolos matemáticos y categorías como Lógica y Circuitos se puede observar según la tabla 5 que existen dos algoritmos de clasificación de weka que presentan mayor número de instancias correctamente clasificadas como son el algoritmo DMNtext-I1 y NavieBayesMultinominalUpdateable, estos algoritmos según los parámetros de evaluación muestran 28 instancias que corresponden al 82% que son correctamente clasificadas y 6 que son el 17% de instancias incorrectamente clasificadas, estas de un total de 34 instancias. En cuanto a la Precisión en este experimento nos presenta un valor de 0.847, 0,824 de Recall y por otra parte 0.823 de Accuary, por lo tanto se tiene un Error de 0.177

Por otra parte haciendo uso de esta misma matriz y aplicando la configuración de Cross-Validation con un valor de 10, podemos constatar en la tabla 7 que en este caso el algoritmo ComplementNaiveBayes clasifica con mejor efectividad con respecto a los dos mejores algoritmos mencionados anteriormente.
Este último algoritmo si bien es cierto clasifica esta matriz en un 93% de manera correcta, pero está utilizando todos los datos tanto para prueba como para entrenamiento, lo cual no sería lo más óptimo.

3.5.5 Análisis de los resultados utilizando la matriz expresiones.

Dentro del análisis realizado utilizando la matriz de expresiones el cual contiene palabras, números signos matemáticos, expresiones algebraicas y de la misma manera en este modelo trabajamos con dos categorías como son Lógica Y circuitos nos presenta algunos resultados.
En los resultados obtenidos según la tabla 6 también existen dos algoritmos que presentan mayor número de instancias correctamente clasificadas estamos hablando

del algoritmo NavieBayes con 29 que corresponden al 85% de instancias correctamente clasificadas y 5 instancias incorrectamente clasificadas que representan el 14.5%, con una precisión de 0.867 y un Accuary 0.852 lo cual nos da un Error de 0.148. Así mismo el algoritmo que se encuentra entre los dos mejores junto con el anterior es el NavieBayesMultinominalUpdateable con 29 instancias correctamente clasificadas que corresponde al 85%, 14 que son el 17.6% de instancias incorrectamente clasificadas con una precisión de 0.867, un Accuary 0.852 quedando como error un valor de 0.148.

Según los resultados obtenidos luego de cada experimento con los diferentes modelos de datos de acuerdo a la tabla 5 y 6 y basados en los parámetros de evaluación de cada uno de los algoritmos, considerando la eficiencia para la clasificación nos hemos dado cuenta que para trabajar con información matemática se puede utilizar los algoritmos DMNtext-I1, NavieBayes y NavieBayesMultinominalUpdateable ya que son algoritmos que pueden trabajar con símbolos y caracteres especiales a más del texto, estos algoritmos se ajusta de la mejor manera distintos modelos de datos presentado valores de error mínimos en nuestro caso tenemos un valor promedio de 0.1585.

La tabla 7 y tabla 8 muestran desde el punto de vista de la efectividad para la clasificación de preguntas, que el algoritmo ComplementNaiveBayes con la configuración Cross-Validation con un valor de 10, tanto con la matriz Símbolos, como con la matriz Expresiones todos con 100 instancias. Este algoritmo encuentra entre los mejores clasificadores con un resultado de 92% instancias correctamente clasificadas con un error del 8%.

Como ya hemos mencionado utilizando todos los datos tanto para pruebas y entrenamiento. Esta configuración no es muy recomendable ya que para una mejor clasificación es recomendable utilizar un parte de los datos.

Se ha realizado varios experimentos con diferentes modelos de datos, algoritmos y configuraciones lo cual hemos obtenidos varios resultados muy importantes, pero se ha elegido solo los mejores para contrastar resultados y mejorar el análisis como se aprecia en la tabla 15.

Tabla 15. Resumen mejores algoritmos de clasificación.

CORPUS	CONTENIDO	CLASIFICADOR	CONFIGURACIONES	EFECTIVIDAD	ERROR
Matriz Símbolos Compuesta 0 y 1	Texto, operadores y símbolos matemáticos	DMNtext-I1	Split 66% Entrenamiento 66 Prueba 34	82%	0,177
Matriz Símbolos Compuesta 0 y 1	Texto, operadores y símbolos matemáticos	NavieBayesMultinominalUpdateable	Split 66% Entrenamiento 66 Prueba 34	82%	0,177
Matriz Símbolos Compuesta 0 y 1	Texto, operadores y símbolos matemáticos	ComplementNaiveBayes	Cross-Validation 10 Instancias 100	93%	0,07
Matriz Expresiones Compuesta 0 y 1	Texto, operadores, símbolos y expresiones matemáticas	NavieBayes	Split 66% Entrenamiento 66 Prueba 34	85%	0,14
Matriz Expresiones Compuesta 0 y 1	Texto, operadores, símbolos y expresiones matemáticas	NavieBayesMultinominalUpdateable	Split 66% Entrenamiento 66 Prueba 34	85%	0,14
Matriz Expresiones Compuesta 0 y 1	Texto, operadores, símbolos y expresiones matemáticas	ComplementNaiveBayes	Cross-Validation 10 Instancias 100	92%	0,08

3.5.6 Análisis de los resultados obtenidos con la matriz tf-idf.

Luego del realizar los cálculos con el fin de obtener los pesos de cada término de la matriz TF-IDF como se puede apreciar en el Anexo 7 es una representación gráfica de los datos en la cual se observan que existen 3 términos (and, falso, variable) que sobresalen en cuestión de frecuencias lo que hace que podamos interpretar que se trata de un documento matemático como muestra a breves rasgos la tabla 16.

Aplicando la formula TF-IDF los términos antes mencionados bajan su valor, por tanto no se aconseja trabajar con este tipo de fórmula. Cómo mencionan (Martineau & Patel, 2009) los pesos IDF son una mala elección cuando el corpus es de un dominio específico porque este tipo de fórmulas tienen preferencias por características raras o poco comunes. Es decir en un documento con un dominio especifico las características más descriptivas serán más frecuentes y por tanto aplicando estas formular a las mejores características les dará un puntaje IDF bajo.

Por lo antes mencionado se decidió trabajar solo con valores de 0 y 1 para representar la ausencia o presencia de un término por cada pregunta del cuestionario.

Tabla 16. Resumen de los términos con más frecuencia en las preguntas.

CORPUS	TERMINO	TF	TF-IDF
Matriz con 201 términos, y compuesta por la frecuencia de las palabras.	and	6	0,07
	falso	4	0,05
	variable	10	0,09

3.5.7 Análisis de los resultados obtenidos con la matriz símbolos aplicando clusterización.

Para este experimento hemos aplicado algoritmos de Clusterización los más utilizados en temas clasificación de texto (SimpleKMeans, EM, FarthestFirst). En nuestro caso el texto incluye algunos caracteres especiales por tratarse de un cuestionario matemático, por esta razón nos hemos centrado en algoritmos de clasificación ya que tenemos mejores resultados.

Según los resultados obtenidos que nos muestra la tabla 9 utilizando el algoritmo SimpleKMeans, tomado desde la matriz símbolos tenemos dos resultados importantes, estos resultados con la configuración en la cual se utiliza 66% de Split, con 5 interacciones, 66 datos de prueba, 34 de entrenamiento y con 4 clústeres, obtenemos como resultado que los datos se dividen en el clúster 1 con 8 instancias que corresponden al 24% y en el clúster 2 se agrupan 25 instancias que corresponden al 74%.

El siguiente resultado importante es producto de la configuración Split de 66% con 5 interacciones, 66 datos de prueba, 34 de entrenamiento y con 3 clústeres, donde los resultados nos muestran que existen dos grupos con mayor número de elementos clusterizados, hablamos del clúster 1 con 8 instancias representa el 24% y en el clúster 2 existen 26% que corresponde al 76%.

Utilizando la matriz expresiones y aplicando en modo de prueba la evaluación con todos los datos de entrenamiento, con un valor de 4 interacciones y no conservando el orden de las instancias tal como se muestra en la tabla 10 se puede observar que no existe una buena distribución de los datos.

Por otro lado se ha configurado un valor de Split de 66%, con 4 interacciones y no conservando el orden de las instancias, observamos que de la misma manera existen clústeres que tienen 0% de datos los cual son valores que no aportan con este análisis.

3.6 Fase VI: Desarrollo

En esta fase donde los modelos y procesos han pasado con éxito se puede implementar el despliegue en cuanto a nuestro proyecto es la única fase que no desarrollaremos debido a que no creará una herramienta específica para la minería de texto y/o entrega de reportes.

CONCLUSIONES

Una vez que se ha llevado a cabo cada una de las fases necesarias para la clasificación de los cuestionarios digitales se puede concluir que:

- ✓ Este trabajo representa una guía que implementa ciertos elementos que formarían parte de un sistema completo para la clasificación de documentos, donde el texto es separado en tokens, diferenciando palabras, números, símbolos matemáticos y caracteres especiales, además se representan los datos en dos matrices diferentes TDM, TF-IDF en la primera matriz se anotan las ocurrencias de las palabras tomando en cuenta que si existe el término la pregunta se coloca el número 1 y si no existe se representa por un 0.En la segunda se coloca los peso a los términos de acuerdo a la frecuencia de ocurrencia de cada termino en la pregunta del cuestionario.

- ✓ La información de los cuestionarios digitales está formada por letras, números, símbolos matemáticos y algunos otros caracteres especiales, por lo cual no es posible generar un modelo de datos directamente desde los cuestionarios, para ello se debe aplicar el preprocesamiento de los datos. Para mitigar esto hemos utilizado la herramienta RapidMiner que nos ha permitido realizar actividades como; tokenizing, stemming, stopwords entre otras lo cual nos ha dado como resultado que se han eliminado símbolos matemáticos y caracteres especiales quedando como datos solo lo que es texto. Al contar con este tipo de información es difícil encontrar una herramienta que permita realizar esta actividad antes de generar el modelo, ya que sin querer se puede eliminar información valiosa para el análisis de datos, por lo cual se ha decidido realizar esta tarea de forma manual.

- ✓ De las dos Matrices construidas luego de realizar el análisis se puede concluir que utilizando la matriz TDM para la representación de información matemática es más beneficiosa que cualquier otra representación ya que al procesar estos datos los resultados son bastante efectivos.

- ✓ Se realizó la experimentación con algoritmos de clasificación supervisada con el propósito de contrastar los resultados obtenidos por cada algoritmo y en base a los parámetros (Presicion, Recall, Accuary, Error) de evaluación que hemos definido para este trabajo. Basados en los parámetros de evaluación y

en la eficiencia para la clasificación podemos decir que al trabajar con información matemática se puede utilizar los mejores algoritmos de clasificación de Weka como son; DMNtext-I1, NavieBayes y NavieBayesMultinominalUpdateable ya que son algoritmos que pueden trabajar con texto, símbolos, caracteres especiales y expresiones matemáticas, y además se ajustan de mejor manera a los distintos modelos de datos que desarrollemos presentado valores de Error mínimos, en nuestro caso tenemos un valor promedio de Error 0.1585.

- ✓ Este proyecto ha contribuido de manera muy importante para identificar y resaltar algunos puntos que hay que considerar para llevar a cabo una implementación de una herramienta o sistema de clasificación y categorización de documentos digitales. Nos deja muchas cosas importantes para una buena implementación de este tema.

RECOMENDACIONES

Como recomendaciones se sugiere lo siguiente:

- ✓ Cuando se realiza categorización de documentos es mucho más sencillo trabajar con datos que de tipo texto, ya que se puede aplicar cualquier herramienta para el reprocesamiento datos, ya que los resultados van hacer muy exitosos de cara a la construcción del modelo para el análisis, clasificación o categorización, mientras que si se requiere trabajar con datos que estén formados por caracteres especiales se torna en cierta parte complicado al momento de realizar el preprocesamiento y construcción del modelos de datos.

- ✓ Para realizar el preprocesamiento de los datos y limpieza en nuestro proyecto al no contar con una herramienta para realizar estas tareas ya que se contaba con datos netamente matemáticos y las herramientas utilizadas nos excluían la mayoría de expresiones que eran muy importantes, por lo cual se procedió a realizarlo de manera manual, como recomendación para el desarrollo de proyectos similares es de vital de importancia que contemos con una herramienta que permita en primer lugar realizar limpieza, reconocimiento de texto, eliminación de fijos y subfijos en entre otras actividades de tal manera que faciliten la construcción del modelo de datos.

- ✓ Tener presente cuales son los datos con los que vamos a trabajar y que queremos obtener como resultado, ya que al trabajar con información que es en su totalidad texto es mucho más fácil la clasificación de preguntas en los documentos, mientras que si se trabaja con datos que contienen como en nuestro caso expresiones algebraicas y caracteres especiales en algunas de las fases del proyecto se las debe realizar de manera manual.

- ✓ Para trabajos futuros sobre temas de clasificación de preguntas en cuestionarios digitales o temas a fines, se recomienda trabajar con el máximo de datos posibles para evitar que se esté ignorando información que sea de aporte importante y por ende los resultados no serán los mejores.

BIBLIOGRAFÍA

Basilio, S. A. (2006). *Aprendizaje automático conceptos básicos y avanzados: aspectos prácticos utilizando el software weka.*

Bassi, A. A. (2012). *Lematizacion basada en analisis no supervisado de corpus.* Obtenido de http://users.dcc.uchile.cl/~abassi/ecos/lema.html

Brun, R., & Senso, J. (2004). Minería textual.

Cervantes, C. V. (1992). *Congreso de Sevilla.* Obtenido de http://cvc.cervantes.es/obref/congresos/sevilla/unidad/ponenc_ordoez.htm

Chapman, P. K., & Colin, S. (2000). CRISP-DM 1.0. *Step-by-step data mining guide*.

Cobo, A. R., & Martínez, M. (2009). Descubrimiento de conocimiento en repositorios documentales mediante técnicas de Minería de Texto y Swarm Intelligence. *Revista Electrónica de Comunicaciones y Trabajos de ASEPUMA, ISSN-e 1575-605X, Nº. 10, 2009, págs. 105-124*.

Compass, B. T. (2013). *SinameFore Analytics Made Accessible.* Obtenido de RapidMiner for text mining: http://www.simafore.com/blog/bid/118417/R-vs-RapidMiner-for-text-mining-Part-1-make-friends-with-regex

Contreras, H., & Davila, J. (2001). Procesamiento del lenguaje natural basado en una gramática de estilos para el idioma español.

Corso, C. L. (2007). Aplicación de algoritmos de clasificación supervisada usando Weka.

Corso, C. L. (2009). Aplicación de algoritmos de clasificación supervisada usando Weka.

CRISP-DM. (2000). *Cross Industry Standard Process for Data Mining.*

Eleazar, B. F., & Cabrera, J. (2007). *Minería de textos*. Obtenido de una herramienta útil para mejorar la gestión del bibliotecario en el entorno digital: https://svpn.utpl.edu.ec/+CSCO+00756767633A2F2F6F69662E6679712E7068++/revistas/aci/vol16_4_07/aci051007.html

Feinerer, K. H. (2008). *Text Mining Infrastructure in R.* Obtenido de http://www.jstatsoft.org/v25/i05

Fernández, P. L. (2009). REVISTA INSTITUCIONAL DE LA UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA. *En el 38 aniversario de la Universidad y los 33 años de la Modalidad de Educacion Abierta y a Distancia*.

Gallardo, J. A. (2000). Metodología para el Desarrollo de Proyectos en Minería de Datos CRISP-DM. *EPB 603 Sistemas del Conocimiento*.

Gary Miner, D. D., & Nisbet, R. (2012). *Practical Text Mining and Statistical Analysis for Non-structured Text Data Applications.*

Gelbukh, A. (2010). *Procesamiento automático del español con enfoque en recursos léxicos grandes.*

Gomez, L. (2002). *Analisis morfologico, Teoria y Practica.*

Gonzalez, J. M. (1997). *Metodololgia de la investigacion social: Tecnicas de recoleccion de informacion.* AGUACLARA.

González, J., Castellón, & M, C. M. (2009). TÉCNICAS DE CLASIFICACIÓN EN EL ENTORNO DE WEKA PARA LA DETERMINACIÓN DE CULTIVOS DE REGADÍO.

Guelboukh, k. (2010). Procesamiento de Lenguaje Natural y sus Aplicaciones.

Hall, M., & Eibe, F. (2011). *Data Mining: Practical Machine Learning Tools and Techniques.*

Hearst, M. (1999). Untangling Text Data Mining. *The 37th Annual Meeting of the Association for Computational Linguistics*.

Hernandez, E. A. (2009). Naive Bayes Multinomial para Clasificacion de exto Usando un Esquema de Pesado por Clases.

Hernández, J. y., María, J., & Aránzazu, S. (s.f.). Análisis de Datos en WEKA – Pruebas de Selectividad. *Introducción al Weka. Curso de Doctorado Extracción Automática de. Conocimiento en Bases de Datos e Ingeniería del. Software*.

Ian H. Witten, E. F. (2011). *Data Mining Practical Machine Learning Tools and Techniques.*

Ian, W., & Eibe, F. (2005). *DATA MINING PRACTICAL MACHINE LEARNING TOOLS & TECHNIQUES 2ND EDITION.*

IMB, C. (2012). Manual CRISP-DM de IBM SPSS.

Jiménez María G, S. A. (s.f.). *Análisis de Datos en Weka: Pruebas de selectividad.* Obtenido de http://www.it.uc3m.es/jvillena/irc/practicas/06-07/28.pdf

Jiménez, A. (2008). *Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society,.* Obtenido de ccc.inaoep.mx/~emorales/Cursos/NvoAprend/node81.html

Laredo, L. T. (2005). *Procesamiento del Lenguaje Natural 6.*

López, R., & Baeza, Y. (2001). Un Método de Agrupamiento. *Procesamiento de Lenguaje Natural*.

Louise, F., & Matt, F. (2010). Text Mining Handbook.

Manuel, M., & Gómez. (2005). Obtenido de Minería de Texto empleando la Semejanza entre Estructuras Semánticas: http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S1405-55462005000300008

Marti, M., & Llisterri, J. (2002). *Tratamineto del Lenguaje Natural.*

Martineau, J. F., & Patel, S. (2009). Improving Binary Classification on Text Problems using Differential Word Features.

Méndez, E., & José, A. (1999). *Lenguaje natural e indización automatizada.* Obtenido de http://eprints.rclis.org/12685/

microsystem. (2010). *Informacion inteligente*. Obtenido de http://www.microsystem.cl/plataformas/rapidminer/

Miner, G. (2011). *Practical Text Mining and Statistical Analysis for Non-structured Text Data Applications.*

Miner, G. D., & Nisbet, R. (2012). *Practical Text Mining and Statistical Analysis for Non-structured Text Data Applications.*

Molina, L. C. (Noviembre de 2002). *Data mining.* Obtenido de http://www.uoc.edu/web/esp/art/uoc/molina1102/molina1102.html

Montes, M. (1999). *Minería de texto.* Obtenido de Un nuevo reto computacional.

Oberreuter, G., & Velásquez, J. D. (2013). Text mining applied to plagiarism detection:.

P, C., & Stutz, J. (1996). *Data Mining and Knowledge Discovery Handbook.* ODED MAIMON LIOR ROKACH.

Perez, C., & Moreno, A. (2009). *Lingüística Computacional y Lingüística de Corpus.Potencialidades para la investigacion textual*.

Perez, C., & Ortiz, A. (2009). Linguistica Computacional y Linguistica de Corpus.

Pérez, S., & Carranza, R. (2008). Clasificación de e-mails Detección de Spam.

Pete, C., Khabaza, T., & Shearer, C. (2000). Metodologia CRISSP-DM . *CRISSP-DM 1.0*.

Pinar, J. (2007). Identificación de autores en bases de datos bibliográficas.

Pincay, J. (2013). *Clasificación automática de respuestas a foros de discusión de acuerdo al dominio cognitivo de la taxonomía de bloom en sidweb 4, empleando minería de texto y algoritmos de aprendizaje.* Obtenido de www.cib.espol.edu.ec/digipath/d_tesis_pdf/d-83179.pdf

Pino, R., & Nicolas, M. (2009). *Introduccion a la Inteligecia Artificial.*

Preisach, C.-T., & Decker, R. (2007). *Data Analysis Machine Learning and Applications.*

Rocha, B. R. (2009). *Dialnet.* Obtenido de Utilización de técnicas de swarm intelligence para la gestión documental en el ámbito de la empresa: https://svpn.utpl.edu.ec/+CSCO+00756767633A2F2F71766E796172672E686176657662776E2E7266++/servlet/tesis?codigo=20905

Rodríguez, S., & Benavides, A. (2007). Procesamiento del lenguaje natural en la recuperación de información.

Salton, G., & Lesk, M. (s.f.). *Computer evaluation of indexing and text processing.*

Sánchez, M., & Antonieta, M. (2011). Aplicación de los algoritmos genéticos en la categorización automática de documentos.

Santana, P. C., & Daniela, M. (2014). Aplicacion de Algoritmos de clasificacion de Mineria de Textos para el Reconocimiento de Habilidades de E-tutores Colaborativos. *INTELIGENCIA ARTIFICIAL*.

Sunikka, A., & Bragge, J. (2012). Expert Systems with Applications. *Applying text-mining to personalization and customization research*.

Ur-Rahman, & Harding, J. (2011). Expert Systems with Applications. *Textual data mining for industrial knowledge management and text classification: A business oriented approach*.

UTPL. (s.f.). *Informacion General*. Obtenido de Entorno Virtual de Aprendizaje: http://www.utpl.edu.ec/utpl/informacion-general/historia

UTPL. (s.f.). *Modalidad abierta y a Distancia*. Obtenido de Entorno Vitual de Aprendizaje: http://www.utpl.edu.ec/academia/pregrado/modalidad-abierta-y-distancia/

Valiente, A., & Cebrián, J. (s.f.). Inteligencia en redes de comunicaciones Practica WEKA Diagnóstico cardiología.

Vanrell, J. A. (2011). Un modelo de procesos para proyectos de explotacion de informacion. *Tesis de maestria en Ingenieria en Sistemas de Informacion*.

Varela, R. (2006). *Minería de datos y descubrimiento de conocimiento en bases de datos aplicados al ámbito bibliotecario.* Obtenido de http://lemi.uc3m.es/est/forinf@/index.php/Forinfa/article/view/122/127

Vickers, P. (1985). A holistic approach to the management of information. *ASLIB Proceedings*, 19-30.

Wakil, M. M. (1999). Introducing Text Mining. *Information Systems Dept.*

Wen-der, Y., & Jia-yang, H. (2013). Content-based text mining technique for retrieval of CAD documents.

Wu, H., & Kwok, K. (2008). Interpreting TF-IDF Term Weights as Making Relevance Decisions.

Zhang, J. L., & Guo, G. (2013). Knowledge-Based Systems. *Projected-prototype based classifier for text categorization*.

Zu G., O. W., & Kimura, F. (2012). *Accuracy improvement of automatic text classification based on feature transformation.* Obtenido de http://miuse.mie-u.ac.jp/bitstream/10076/11098/3/40A12191.pdf

ANEXOS

1. Identifique cuál de las siguientes oraciones es una proposición. a. Los niños de ecuador. b. 4 es número primo. c. Me gustan las margaritas.

2. Cuál de las siguientes oraciones es una proposición. a.2 = 4 b. 2 + 5 c.7 / 5

3. Identifique cuál de las siguientes oraciones es una proposición simple o atómica. a. Cuatro no es un número primo. b. Cinco es un número primo. c. Seis es divide exactamente para 2 y para 3.

4. Identifique cuál de las siguientes oraciones es una proposición compuesta o molecular. a. Cuatro es mayor de 6. b. Cuatro no es mayor o igual a 6 c. Cuatro es un número par.

5. Una importante condición para que una oración pueda ser considerada proposición es: a. La oración debe tener conectores. b. La oración se le puede asignar valores de verdadero o falso pero no ambos vez. c. La oración puede tener signos de interrogación.

6. Utilizando conectivas lógicas es posible construir enunciados compuestos de cualquier longitud. (V)
7. Una proposición es verdadera o falsa, pero no ambas a la vez. (V)
8. La negación de una proposición p se escribe ¬p tal que si p es verdadera, entonces ¬p es falsa y viceversa. (V)
9. La proposición "No es verdad que: No iré a la fiesta", se la puede representar como: ¬P (F)
10. La proposición "No es el caso que Carmen apruebe los exámenes de admisión y no ingrese a la universidad", se simboliza de la siguiente forma: a)~(p Λ ~q) b)(~p Λ ~q) c)~(p Λ q)

Anexo 2. Estructura Matriz TDM Simbolos

verdad	verdadero	vez	viaje	viceversa	xyz	+	-	*	/	=	categorías
0	0	0	0	0	0	0	0	0	0	0	logica
0	0	0	0	0	0	1	0	0	1	1	logica
0	0	0	0	0	0	0	0	0	0	0	logica
0	0	0	0	0	0	0	0	0	0	0	logica
0	1	0	0	0	0	0	0	0	0	0	logica
0	0	0	0	0	0	0	0	0	0	0	logica
0	1	1	0	0	0	0	0	0	0	0	logica
1	0	0	0	1	0	0	0	0	0	0	logica
1	0	0	0	0	0	0	0	0	0	0	logica
0	0	0	0	0	0	0	0	0	0	0	logica
1	1	0	0	0	0	0	0	0	0	0	logica
0	1	0	0	0	0	0	0	0	0	0	logica
0	1	0	0	0	0	0	0	0	0	0	logica
0	0	0	0	0	0	0	0	0	0	0	logica
0	1	0	0	0	0	0	0	0	0	0	logica
0	0	0	0	0	0	0	0	0	0	0	logica
0	0	0	0	0	0	0	0	0	0	0	logica
0	0	0	0	0	0	0	0	0	0	0	logica
0	0	0	0	0	0	0	0	0	0	0	logica
0	0	0	0	0	0	0	0	0	0	0	logica
1	0	0	0	0	0	0	0	0	0	0	logica
0	0	0	0	0	0	0	0	0	0	0	logica
0	0	0	0	0	0	0	0	0	0	0	logica
0	0	0	0	0	0	0	0	0	0	0	logica
0	0	0	0	0	0	0	0	0	0	0	logica

Anexo 3. Estructura Matriz TDM Expresiones

Λx (h(x) -> c(x))	**vx (h(x) v - c(x))**	**- Vx (h(x) -> - c(x))**	**(x1 v x2) v x3**	**x b.y c.x+y**	**(x1 Λ x2) v (x3 Λ x1)**	**p**	**q**	**a**
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	1	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	1	0
0	0	0	0	0	0	1	1	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

Anexo 4. Términos eliminados por poeeer una caracateristica de poco aporte sobre los datos.

TÉRMINOS MENOS IMPORTANTES	
Utpl	podria
Clorofil	por
Com	sean
Con	seleccion
Cual	siempre
Cualquiera	siguiente
Dienro	sol
Donde	son
Esta	ser
Escribe	sus
Gran	tal
Hombre	tenga
Las	trabaja
Los	tiene
Margarita	tipo
Millonario	todo
Mortal	una
Niño	utiliza
Per	vez

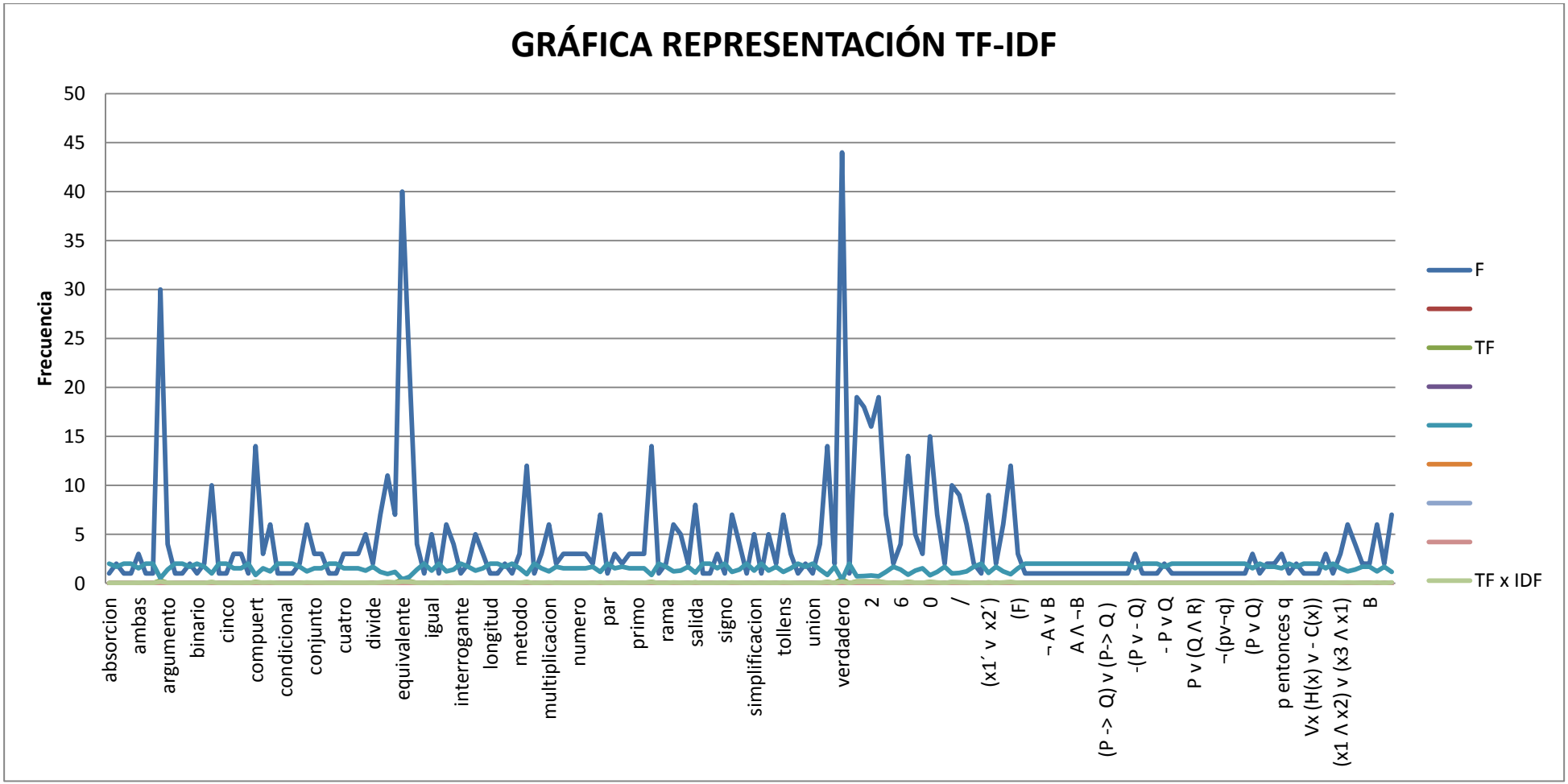
Anexo 5. Estructura de la Matriz TF-IDF

utiliz	Utpl	valor	variabl	Verd	verdader	vez	viaj	vicevers	xyz
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0	0	0
1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	0	0	0
0	0	0	0	0	1	0	0	1	0
6	1	15	2	31	11	2	2	1	1
0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	1	0	1	3	0	0	0	0
0	0	1	0	1	3	0	0	0	0
0	0	0	0	0	2	0	0	0	0
0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	2	0	0	0	0

Anexo 6. Cálculos realizados para obtener la Matriz TF-IDF

	absorcion	acotado	admision	algebra	Algún	ambas	ambos	and	anterior	aplicado
F	1	2	1	1	13	3	1	1	6	5
TF	0,01	0,02	0,01	0,01	0,13	0,03	0,01	0,01	0,06	0,05
	2	1,698970004	2	2	0,88605665	1,52287875	2	2	1,22184875	1,30103
IDF	0,02	0,0339794	0,02	0,02	0,11518736	0,04568636	0,02	0,02	0,073310925	0,0650515

Anexo 7. Resultados del cálculo Matriz TF-IDF



Anexo 8. Articlo basado en el trabajo realizado