Instituto Tecnológico de Tuxtla Gutiérrez

Sobre el tema:

Procesamiento de Textos de Opinión para la Extracción de Expresiones Valorativas

Para obtener el título de:

Ingeniero en Sistemas Computacionales

Presenta:

Viviana del Rocío Hernández Castañón
Carlos Chang Granados

Asesor interno:

M.C. Aida Guillermina Cossío Martínez

Asesores externos:

Dr. Aurelio López López

M.C. Laritza Hernández Rojas
INAOE

Tuxtla Gutiérrez Chiapas, mayo 2012

Agradecimientos

Me permito agradecer y dedicar la presente tesis con todo mi amor y cariño:

A Dios por darme las fuerzas necesarias en los momentos en que más las necesité y bendecirme con la posibilidad de caminar a su lado durante toda mi vida.

A mi querida familia, quienes siempre han creído y confiado en mí, me han apoyado y estimulado en todas les decisiones que he tomado en la vida.

A mis asesores por el apoyo personal y profesional para que lograr este éxito académico.

A mis amigos quienes han estado conmigo en todo este tiempo y con los que he pasado muy buenos momentos.

Y a todos los que de mi memoria no puedo extraer en este momento.

Resumen

La presente investigación se realizó en el laboratorio de Tecnologías del Lenguaje de la Coordinación de Ciencias Computacionales del Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE). Su finalidad es apoyar y ampliar el campo de la Minería de Opiniones *(OpinionMining),* con el procesamiento manual y automático de textos. De ahí que su propósito fuese la elaboración de un una plataforma que permitiera procesar textos, brindando información de utilidad para la extracción de expresiones valorativas en textos de opinión. Esta plataforma se desarrolló en lenguaje Java, y cuenta con cuatro casos de uso. El primero se encarga de la generación de representaciones, permitiendo escoger entre diferentes modelos de representación del contenido textual. El segundo consiste en la gestión de intensidades de clasificación, permitiendo seleccionar la probabilidad de distribución para cada clase, así como seleccionar entre diferentes umbrales de intensidad. El tercero permite generar diferentes estadísticas, cómo cantidad de palabras, categorías gramaticales entre otras, la cuales son especialmente útiles en el análisis de resultados experimentales. Por último, el cuarto caso de uso se encarga de las operaciones de conjuntos, *Unión, Intersección y Diferencia,* que en este caso se realiza con listas de frases. Por otra parte, se trabajó en el desarrollo manual de colecciones textuales que sirvieron de apoyo a la validación de la plataforma desarrollada. Concluyéndose con la satisfacción del objetivo propuesto. Para ello fue necesaria la elaboración del Marco Teórico, estudiando trabajos relacionados desde el punto de vista lingüístico y computacional. El trabajo se justificó porque posee relevancia práctica y social, y por su utilidad metodológica.

Índice general

Capítulo 1

Introducción

1.1. Planteamiento del problema

La capacidad de generar información en la actualidad se ha multiplicado y cada persona contribuye de una manera u otra en la producción de contenidos que se comparten. Las empresas generalmente tienen el problema de no poder procesar toda esta información para encontrar intereses particulares, de lo que la sociedad necesita o sus consumidores piensan acerca de su producto. Esto se puede extender en campañas políticas, por lo que no se limita al sector comercial o de servicios. De este modo lo indicado anteriormente se ha vuelto una problemática al tener relativamente a la mano toda esta basta información y las empresas tengan que invertir por otro lado en estudios de mercado.

Lo anterior ha abierto nuevos retos a la Inteligencia Artificial, dando surgimiento al Análisis del Sentimiento *(Sentimental Analysis)* o la llamada Minería de Opiniones *(Opinión Mining).* Ambas disciplinas tienen abierta una significativa diversidad de aplicaciones, las que van desde el seguimiento y generación automática de resúmenes de las opiniones de los usuarios sobre productos comerciales (películas, artículos domésticos o de entretenimiento, etc.), perfil de individuos y organizaciones, entre otros, expresadas en encuestas abiertas o en foros en línea, hasta la evaluación de las relaciones públicas y del mercado de las empresas [1].

Existen una serie de enfoques de análisis de sentimiento en las expresiones valorativas, entre ellos automáticos y manuales, se debe mencionar que los enfoques manuales para este análisis tienen dificultades en conjuntos de datos grandes, en cuanto al enfoque de aprendizaje automático es la forma más prometedora, pero existe un lento avance en su desarrollo pues ha sido poco investigado para el español y como consecuencia se tiene una carencia de herramientas que permitan el pre-procesamiento de este tipo de expresiones [2], para su posterior procesamiento con algoritmos de aprendizaje automático.

La Coordinación de Ciencias Computacionales del Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE) de México, se encuentra desarrollando una investigación en la que una de las tareas con mayor consumo de tiempo es la extracción de expresiones valorativas su tipo e intensidad en textos en español.

La tarea de identificar expresiones valorativas así como un léxico de expresiones valorativas asociadas con tres categorías de actitud, y dos de polaridad de acuerdo a los planteamientos de la Teoría de la Valoración del lenguaje la cual se emplea como marco teórico lingüístico de la investigación.

A partir de una metodología propuesta por el centro de investigación mediante la que se elaboran recursos textuales para la minería de opinión, existe la necesidad de obtener una colección de expresiones valorativas partiendo del tratamiento de un corpus de oraciones pertenecientes a diversas fuentes (internet, libros, ficticias), las cuales emiten una opinión sobre entidades,ésta es una tarea que requiere conocimientos del manejo de motores de búsqueda de información y del dominio de los basamentos de la Teoría de la Valoración, también es necesaria una herramienta de software para el pre-procesamiento textual, la herramienta requiere funcionalidades adecuadas a los métodos de preparación de la información para su posterior procesamiento, también no menos importante el contar con una interfaz amigable para la tarea de extracción de expresiones valorativas su tipo e intensidad. Es así como se ha determinado que los recursos que se obtengan serán el principal apoyo para la evaluación de la herramienta de software, hay que destacar que para realizar las pruebas en la herramienta se requiere de recursos textuales con características adecuadas para su utilización, de tal manera que se espera que la colección obtenida resuelva el requerimiento para realizar las pruebas en la herramienta, lo cual nos lleva a obtener mejores resultados en pre-procesamiento de textos de opinión.

1.2. Variables

- Categorías valorativas de palabras individuales (actitud y polaridad).
- Intensidad valorativa de palabras individuales (alto, medio, o nulo).
- Otros rasgos lingüísticos de palabras individuales (categoría gramatical, posición dentro de una oración, etc.)
- Tipo e intensidad de expresiones o secuencias de palabras.
- Contexto lingüístico de una palabra (el conjunto de palabras que tiende a ocurrir en la misma oración).

1.3. Hipótesis

- Con la validación y mejora de las colecciones textuales elaboradas de forma manual, realizada bajo ciertos lineamientos que marca la Teoría de la Valoración será posible llevar a cabo la extracción de expresiones valorativas, indicarles un tipo e intensidad, particularmente en textos en español.
- Con el desarrollo de una plataforma de software será posible lograr con mayor eficacia y eficiencia la generación de archivos a través del pre-procesamiento y evaluación automático de textos de opinión.

1.4. Objetivos

Objetivos generales:

- Validar y mejorar las colecciones textuales ya elaboradas, haciendo uso de motores búsqueda como *Google,* así como asimilando y aplicando los basamentos de la Teoría de la Valoración.
- Analizar de forma crítica el estado actual de los trabajos más relacionados.

Objetivos específicos:

- Desarrollar una herramienta en lenguaje de programación *Java* que permita con eficiencia adecuada el pre-procesamiento y evaluación automático de textos de opinión, con vista a lograr mayor eficiencia y eficacia en la tarea la extracción de expresiones valorativas su tipo e intensidad.
- Validar la utilidad de las colecciones y la plataforma elaborada, como vía de comprobación y fiabilidad de la investigación realizada, a partir de los resultados obtenidos.
- Hacer uso del número de todas las Unidades de Procesamiento Centrales para los procesos con mas demanda de cálculos.

1.5. Justificación

Por medio del presente trabajo se contribuye en el avance de investigaciones teóricas y aplicadas en el área de la Minería de Opinión, con vista al desarrollo de colecciones textuales y herramientas automáticas que facilitan el pre- procesamiento de textos de opinión a partir de grandes volúmenes de textos lo cual mejora la eficacia y eficiencia de las tareas y proporciona una fuente de referencias para futuros proyectos.

1.6. Alcances y limitaciones

Alcances:

- Se pretende que el presente proyecto sea viable y de interés en las investigaciones actuales sobre Minería de Opiniones.
- El desarrollo de la plataforma se realizará con un lenguaje multiplataforma para uso local.
- Diseñar un software con interfaz gráfica para su fácil manejo.
- Desarrollar una colección de expresiones valorativas a partir del tratamiento de un corpus de oraciones.

Limitaciones:

- La herramienta está sujeta a los trabajos realizados actualmente por lo que se encuentra susceptible a reemplazos ante los avances que se generen en las diferentes líneas de investigación en la que sea requerida.
- La herramienta trabaja exclusivamente de manera local y bajo el sistema Windows
- La herramienta trabaja únicamente con archivos de texto plano y archivos TTG

1.7. Estado del arte

Desde hace unos pocos años, ha habido un gran aumento del interés en la identificación y extracción automática de las actitudes, opiniones y sentimientos expresados en textos. Esta motivación, se debe a la necesidad de proveer herramientas y soporte para analistas de sistemas de diferentes dominios, los cuales necesitan del seguimiento automatizado de la información que expresa sentimiento.

A continuación se exponen las herramientas más representativas que apoyan al entrenamiento y procesamiento de lenguaje además de manejar un sistema de extracción de información,

dichas aplicaciones están relacionadas con el área en la que se desarrolla el presente trabajo de investigación.

GATE: Una arquitectura para el desarrollo de aplicaciones robustas HLT

GATE es un software libre de código abierto, en la que se ha invertido a la fecha cerca de 5 millones de euros, se inicio en 1990 y la última versión, se lanzo en el año 2009. Fue hecha para desarrollar numerosas aplicaciones para tareas de procesamiento de lenguaje, también para construir y hacer anotaciones de corpus y llevar a cabo las evaluaciones de las aplicaciones generadas, así como para extraer instancias de entrenamiento para algoritmos de aprendizaje automático [3]

UAM Corpus Tool

Es un conjunto de herramientas para la anotación de corpus de texto. Este software permite al usuario anotar un corpus de archivos de texto en distintos niveles lingüísticos previamente definidos por el usuario. Por ejemplo, uno puede anotar los textos a nivel de documento (Ej., tipo de texto, características del escritor, registro, etc.), en niveles semántico-pragmáticos, y en niveles sintácticos (Ej., cláusula, sintagmas, etc.). Utilizando una herramienta gráfica, el usuario define una jerarquía de etiquetas apropiadas para cada nivel de anotación. A continuación, el usuario anota el texto en cada nivel, seleccionando primero el texto para indicar un segmento, y asignándole características elegidas de entre la jerarquía de etiquetas definidas para ese nivel. Este artículo describe también otras funcionalidades añadidas al software, tales como instrumento de búsqueda en el corpus, etiquetador automático basado en la correspondencia de patrones léxicos, y producción de informes estadísticos[4].

TRUE: Una plataforma de pruebas en línea para evaluación multimedia

Es una plataforma en línea desarrollada para crear y realizar pruebas subjetivas orientadas a la evaluación de los estímulos de diferente naturaleza, tales como audio, video, gráficos y texto. Debido a la alta flexibilidad que la plataforma ofrece a investigadores en sus diferentes pruebas, puede llevarse a cabo la identificación de la emoción o evaluación de la calidad de los sistemas de síntesis, entre otros.

Continuando con los trabajos relacionados, uno de los más citados en la literatura vinculado con la clasificación de la polaridad es el de Turney en 2002, en este trabajo se propone un algoritmo no supervisado simple para clasificar opiniones de películas como recomendadas y no recomendadas [5]. Turney aplica un reconocedor de partes de la oración, para identificar frases en el texto que contengan adjetivos y adverbios.

Kennedy e Inkpen, en 2006 clasifican opiniones de películas en positivas, negativas y neutras [6]. Para la clasificación ellos consideran que existen palabras que pueden modificar la polaridad de otra, estos son las negaciones e intensificadores. Así en lugar de reconocer patrones sintácticos ellos, usan el diccionario General *Inquirer* y extraen términos definidos como positivos, y negativos, así como los modificadores de valencia; también extraen los términos positivos y negativos de otras fuentes. Además, utilizan un *parser* para determinar qué modificadores afectan a cada término según la sintaxis del texto. Este tipo de modificadores forma parte del subsistema de Gradación y Compromiso de la Teoría de la Valoración expuesta en el siguiente capítulo.

Además de la polaridad algunos autores han intentado reconocer el tipo de actitud de una expresión apoyándose en la Teoría de la Valoración expuesta en el siguiente capítulo. Whitelaw et al. 2005, intentó en su trabajo clasificar opiniones de películas en positivas y negativas, usando

el tipo de actitud de una expresión [7].

Ellos obtienen estas frases a partir de expresiones semillas de cada tipo de actitud tomadas de (Matthiessen 1995 [8]) y (Martin and White 2005 [9]), y se apoyaron en WordNet y otros tesauros para expandir estas listas con nuevos términos (sinónimos y otros términos relacionados pero con la misma categoría gramatical que los términos semilla).

Por otra parte, Brooke, en 2009, propuso la creación de diccionarios de orientación semántica en español, haciendo una analogía con diccionarios en inglés de adjetivos, sustantivos, verbos, adverbios e intensificadores [10]. Cada diccionario en inglés de adjetivos, sustantivos, verbos, y adverbio se traduce al español por medio del diccionario bilingüe *Spanishdict,* y el traductor automático en línea Google *translator.* La orientación semántica de las palabras en inglés se mantiene luego de su traducción al español. Además del diccionario bilingüe y el traductor, el autor propuso otro método, usando un corpus textual en español formado por 400 críticas sobre hoteles, películas, música, teléfonos, lavadoras, libros, coches, y computadoras.

Sin duda el trabajo más relacionado a este trabajo de tesis es el de estos autores que proponen un sistema para la creación de un corpus de cuentos populares con anotaciones emocionales, al cual llamaron *EmoTales*[11]. Por esta razón se consideró necesario brindarle una mayor atención en este documento.

Estos autores, para el diseño del corpus consideraron los siguientes requisitos:

- El corpus debe contener frases en contexto en lugar de frases aisladas.
- Las etiquetas emocionales asignadas a las frases en el corpus debe basarse en evaluaciones subjetivas humanas.
- El conjunto de descriptores utilizados en la anotación debe ser amplio y flexible.
- El corpus debe ser anotado por un número representativo de anotadores humanos.
- La extensión del corpus debe ser representativa.

La consideración más importante en el diseño de este corpus es tener un corpus emocional grande, desarrollado por un gran número de anotadores. El contenido del corpus y los temas fueron cuidadosamente seleccionados.

EmoTales fue un intento de crear un corpus lo más general posible, que facilite la comparación y evaluación de otros recursos relacionados con el etiquetado de textos con emociones. Para lo que se definieron tres pasos principales.

Anotación: Se propone el uso de dimensiones y categorías emocionales, incluyendo las emociones básicas y específicas. El corpus debe ser anotado por un gran número de personas; en la anotación de *EmoTales* se emplearon 36 anotadores para categorías emocionales y 26 con dimensiones emocionales.

Post-procesamiento: El corpus puede ser post-procesado de diferentes maneras dependiendo de su aplicación.*EmoTales* se puede utilizar sin post-procesamiento teniendo en cuenta las anotaciones de inicio que fueron proporcionados por los evaluadores. De lo contrario puede ser post-procesado para obtener una versión del corpus en el que los valores de referencia se han definido para cada frase, mediante la selección de los valores que se acordaron en la mayoría de los evaluadores. Estos valores de referencia se pueden encontrar en diferentes especificaciones de los niveles en función de una amplia gama de categorías emocionales o solo categorías de base

emocional.

Evaluación: Los anotadores deben estar de acuerdo acerca de cómo evaluar y validar las anotaciones en el corpus. Necesitamos dos métricas diferentes para analizar el acuerdo entre los anotadores, uno para anotación con dimensiones emocionales y otro para la anotación con categorías emocionales.

Debido a que los autores tenían un interés especial en la narrativa, se decidió centrar su interés y esfuerzo en un dominio textual muy especifico: los cuentos de hadas, dado que estos están destinados a ayudar a los niños a comprender mejor sus sentimientos, y por lo general involucran casos de las emociones (felicidad, tristeza, ira o miedo). Por otra parte, los autores consideran que los cuentos son especialmente adecuados para la identificación y estudio de emociones, por que las emociones que representan en ellos son más evidentes y explícitos que los representados en dominios más complejos. Con el fin de crear *EmoTales* se seleccionaron 18 cuentos de diversa longitud, escritos en ingles, 26 anotadores participaron en la creación del corpus, en dimensiones emocionales con entre 6 y 14 anotadores por cada una de las historias y 36 anotadores colaborando con las categorías emocionales, trabajando entre 7 y 12 anotadores por cada una, haciendo un total de 1389 frases y 16816 palabras. Estos cuentos fueron elegidos de acuerdo a las exigencias de las aplicaciones futuras del corpus, cubriendo un amplio espectro de estilos al incluirse cuentos de diferentes tiempos y periodos además de diversos autores.

La identificación y asignación de las emociones de una oración es una tarea subjetiva, por lo que todos los textos del corpus se analizaron por varios anotadores para obtener mejor uniformidad en los resultados. La anotación del corpus se llevo a cabo con la herramienta de evaluación TRUE, una plataforma multimedia en línea, para crear y realizar pruebas subjetivas orientadas a la evaluación de diferentes estímulos naturales como audio, video, gráficos y textos. Puede realizarse pruebas como identificación de emociones, evaluar la calidad de síntesis de sistemas entre otros [12].

Para la representación de las dimensiones emocionales en el corpus se seleccionaron tres dimensiones básicas: valencia, activación y control. Para ayudar a los anotadores en la asignación de valores para cada dimensión utilizaron la norma SAM (Self- Assessment Manikin),la cual es un formato de texto delimitado por tabuladores que consiste en una sección de cabecera que es opcional y una sección de alineación[13], esta norma consta de nueve valores por dimensión. Se les pide a los anotadores que seleccionen la figura o el punto entre las cifras que mejor describen la emoción en la frase y se le asignará un número entero entre 1 y 9. El sistema SAM se ha utilizado en otros trabajos, mostrando en sus resultados una baja desviación estándar y alta concordancia entre evaluadores.

Los autores hicieron frente al desacuerdo entre los anotadores introduciendo una etapa de post-procesamiento del corpus[1] a nivel conceptual. Esto se realizó por medio de una ontología de las emociones, en el caso de las categorías emocionales, el tratamiento post-conceptual con la ayuda de una ontología permite la generación de más de un punto de vista de los recursos anotados, cada una de ellas se basa en las etiquetas a un nivel diferente de granularidad, esto quiere decir que a partir de más de un punto de vista generados a partir de desacuerdos se reduce a acuerdos conceptuales.

Finalmente el corpus anotado se evaluó de acuerdo a cada anotador con el fin de obtener alguna medida de la validez de las anotaciones en el corpus, se uso la estadística de *Fleiss Kappa*

[1]Es un conjunto, habitualmente muy amplio, de ejemplos reales de uso de una lengua. Estos ejemplos pueden ser textos (lo más común) o muestras orales (generalmente transcritas)

para analizar un acuerdo entre evaluadores en el caso del corpus marcado con categorías emocionales, se fusionaron sinónimos en un valor representativo, se fusionaron categorías de emociones según niveles de la ontología, distribución de las frases, palabras y palabras por frases en el corpus anotado con dimensiones emocionales.

Tras la exposición de los trabajos relacionados se determina que el procesamiento de textos de opinión contiene aspectos y dificultades no consideradas en el procesamiento clásico de textos, como son el alto contenido de subjetividad y la poco disponibilidad de herramientas y recursos textuales esencialmente para el español por lo que se hace evidente la necesidad de elaboración de corpus y léxicos adecuados a cada aplicación evitando la perdida de generalidad tanto como sea posible.

1.8. Estructura del documento

Para estructurar este documento de tesis se escogió dividirlo de la siguiente manera:

Introducción: Se centra en la ubicación y esbozo del problema de investigación en las necesidades, conveniencias de resolverlo, en los objetivos del trabajo y el estado del arte.

Capítulo I: Se describe como introducción en la que se hace referencia al planteamiento del problema, las variables manejadas en el proyecto, hipótesis, objetivos, justificación y alcances y limitaciones.

Capítulo II: Se describe el marco teórico o de referenciase, comentándose los puntos fundamentales de la Teoría de la Valoración en el lenguaje, así como el punto de vista computacional en el área de la Minería de Opinión.

Capítulo III: Se describe el proceso de validación y mejora de colecciones textuales a partir de colecciones previas. Este capítulo comentará el proceso de búsqueda y confección de oraciones de opinión, la clasificación manual de expresiones y palabras individuales de a cuerdo a tres categorías de actitud y dos de polaridad.

Capítulo IV: Se expone el proceso de diseño y desarrollo de una herramienta de software para el pre-procesamiento y validación de colecciones textuales del tipo de las descritas en el Capítulo II. Se describe el ciclo de vida del software, su estructura, y aspectos más sobresalientes de la implementación.

Capítulo V: Se presenta las pruebas y los resultados en el funcionamiento de la herramienta.

Capítulo VI: Se exponen las conclusiones que se obtuvieron al término de la tesis, asimismo se mencionan recomendaciones para mejoras en la herramienta.

Referencias bibliohemerográficas: Se relacionan las fuentes bibliohemerográficas consultadas.

Anexos: Se incluyen los datos con información complementaria, con vista a contribuir a la mejor comprensión de este documento de tesis.

Capítulo 2

Marco teórico

Los fundamentos teóricos que sustentan esta tesis, se encuentran en varias áreas; tales como, la lingüística con la Teoría de la Valoración, la Minería de Opiniones, el Análisis del Sentimiento. Estas últimas se refieren a una amplia área del procesamiento del lenguaje natural, la Lingüística Computacional y Minería de Textos. En términos generales, su objetivo es determinar la actitud de un orador o escritor con respecto a algún tema. La actitud puede ser en su juicio o evaluación, su estado afectivo (el estado emocional del autor al escribir) o la comunicación emocional destino (el efecto emocional que el autor desea tener en el lector) [14].

A continuación en este capítulo se comentarán los trabajos más relacionados a la tesis que se presenta, contando con la siguiente estructura. Sección 2.1 Minería de opiniones, en la sección 2.2 Análisis del Sentimiento, 2.3 Teoría de la Valoración y finalmente 2.4 Pre-procesamiento de textos de opinión.

2.1. Minería de opiniones

Se refiere a una amplia área del Procesamiento del Lenguaje Natural, la Lingüística Computacional y la Minería de Textos. Su objetivo no es determinar el tópico del que trata un documento sino la opinión que este expresa, es decir, su objetivo es determinar la actitud (sentimientos, emociones y subjetividades) de un orador o de un escritor con respecto a cierto tópico.

La Minería de Opiniones se puede dividir en varias tareas:

1. **Detección de subjetividad,** que consiste en determinar si una unidad textual tiene una naturaleza objetiva (hecho) o subjetiva (opinión).

2. **Clasificación de la opinión,** determinar su polaridad, es decir, si la opinión es negativa o positiva.

3. **Determinar la fuerza de la opinión,** en qué medida es positiva negativa.

4. **Determinar la fuente de la opinión,** la fuente de una opinión puede ser una persona o una institución, esta tarea requiere frecuentemente resolución de anáforas.

5. **Determinar el objetivo de la opinión,** de quien se habla en la opinión, con quién se está de acuerdo o no.

6. **Resumen de las opiniones y/o visualización gráfica de los resultados,** puede ser agregando votos (índice de 1-5, estrellas), sobresaltando algunas opiniones, representando acuerdo/desacuerdo, etc. [14]

2.2. Análisis del sentimiento

Las emociones han sido estudiadas en muchos campos, por ejemplo, psicología, filosofía, sociología, biología, etc. Sin embargo, todavía no hay un acuerdo entre los investigadores acerca del conjunto de emociones básicas en las personas. Sobre la base [15], la gente tiene 6 tipos de emociones primarias, es decir, amor, alegría, sorpresa, enojo, tristeza y miedo, que pueden ser subdivididos en muchas emociones secundarias y terciarias. Cada emoción también puede tener diferentes intensidades. Los puntos fuertes de las opiniones están estrechamente relacionados con las intensidades de ciertas emociones, por ejemplo, la alegría y la ira. Sin embargo, los conceptos de las emociones y las opiniones no son equivalentes a pesar de que tienen una gran intersección.

Cuando se habla de los sentimientos subjetivos, de las emociones u opiniones, es útil distinguir entre dos diferentes nociones: los estados mentales de las personas (o sentimientos) y expresiones de lenguaje utilizado para describir los estados mentales. Aunque solo hay 6 tipos de emociones, hay un gran número de expresiones lingüísticas que pueden ser utilizadas para su expresión. De manera similar, también hay un gran número de expresiones de opinión que describen los sentimientos positivos o negativos.

Todas las tareas del Análisis del Sentimiento son muy difíciles. Nuestra compresión y el conocimiento del problema y su solución son todavía limitados. La razón principal es que es una tarea de Procesamiento del Lenguaje Natural, y el Procesamiento del Lenguaje Natural no tiene problemas fáciles. Otra razón puede ser debido a las formas populares de hacer la investigación, probablemente confiando demasiado en máquinas con algoritmos de aprendizaje.

Sin embargo se ha tenido progreso significativo durante los últimos años debido, entre otras causas, al gran número de empresas de nueva creación que ofrecen servicios de Análisis del Sentimientos o de Minería de Opiniones. Hay una necesidad real y enorme en la industria de estos servicios por que cada empresa quiere saber cómo los consumidores perciben sus productos y servicios y el de sus competidores. Lo mismo puede decirse de los consumidores, porque cada vez que una persona quiere comprar algo, se interesa por la opinión de aquellos que ya consumieron el producto. Estas necesidades prácticas y los desafíos técnicos mantendrán al campo lleno de vida en los próximos años [16].

2.3. Teoría de la valoración

Uno de los estudios más evolucionados para el desarrollo y comprensión de la evaluación en el lenguaje es la Teoría de la Valoración *(AppraisalTheory,* en inglés), la cual se desarrolló a partir del trabajo de investigación en educación y se enmarca dentro de la corriente de la Lingüística Funcional Sistémica. Esta teoría se ocupa de los recursos lingüísticos por medio de los cuales los textos/hablantes llegan a expresar, negociar y naturalizar determinadas posiciones intersubjetivas y en última instancia, ideológicas. Dentro de esta amplia área de interés, se ocupa particularmente del lenguaje (la expresión lingüística) de la valoración, la actitud y la emoción, y del conjunto de recursos que explícitamente posicionan de manera interpersonal las propuestas y posiciones textuales. Es decir, se ocupa de los significados que hacen variar o modifican los términos del compromiso del hablante con sus emisiones y que modifican lo que está en juego en la relación interpersonal, tanto en las emisiones individuales como en lo que se va acumulando a medida que el texto se va desplegando [17].

La Valoración *(Appraisal)* un término de amplio alcance, ha sido considerada para incluir todos los usos evaluativos del lenguaje, mediante los cuales los hablantes y escritores no sólo adoptan posturas de valores particulares, sino que, además, negocian dichas posiciones con sus

interlocutores reales o potenciales [17].

Los recursos evaluativos, según la Teoría de la Valoración, pueden dividirse en tres grandes dominios semánticos: la actitud, el compromiso, y la gradación [7], [17].

La actitud incluye los significados por los cuales los textos/hablantes atribuyen un valor o una evaluación intersubjetiva a los participantes y los procesos, ya sea relacionado con respuestas emocionales o con sistemas de valores culturalmente determinados. La **actitud** puede ser positiva o negativa, además se divide tres sus sistemas, los cuales no son completamente independientes; o sea, es posible encontrar expresiones de que pertenecen a más de un tipo de actitud. El **afecto** refiriéndose a la evaluación mediante la cual el hablante indica su disposición emocional o reporta las respuestas emocionales de terceros hacia personas, cosas, situaciones o eventos. El **juicio** que puede entenderse como la institucionalización de las emociones en el contexto de las propuestas: normas sobre cómo deben y no deben comportarse las personas.

Las normas sociales que se ponen en juego en estas evaluaciones de juicio adoptan la forma de regulaciones, o de expectativas sociales. Y por último la **apreciación** relacionada con la evaluación de objetos, procesos, constructos o textos, en función de principios estéticos y otros sistemas de valor social. Con estos valores se evalúan la forma, la apariencia, la composición, el impacto y la importancia. Este sistema incluye una evaluación estética, así como una categoría de valuación social, no estética.

El **compromiso** corresponde a los recursos lingüísticos que pueden utilizarse para posicionar la voz del hablante o del autor en relación con los enunciados comunicados por un texto. Tiene que ver también con los significados por medio de los cuales los emisores reconocen o ignoran los diversos puntos de vista que sus enunciados ponen en juego. De este modo, negocian un espacio interpersonal para sus propias posturas dentro de dicha diversidad. Para decirlo en términos de funcionalidad comunicativa y de potencial retorico, este trabajo se ocupa de los recursos por medio de los cuales un texto llega a expresar, negociar y naturalizar determinadas posiciones intersubjetivas y en última instancia, ideológicas.

La **gradación** es un espacio semántico de escala que está relacionado con la manera en que los hablantes intensifican o disminuyen la fuerza de sus enunciados y gradúan, desdibujando o agudizando, el foco de sus categorizaciones semánticas. En la gradación, nos preocupa los valores que proporcionan escalas de grado, ya sea en términos de fuerza interpersonal que el hablante adjudica a una emisión como en términos de la precisión con que un elemento pone en foco una relación de valor. Estas dos dimensiones se denominan: fuerza (escala variable de intensidad) y foco (agudizando o desdibujando los límites de la categoría).

2.4. Pre-procesamiento de textos de opinión

La amplia mayoría de los trabajos en el área de la Minería de Opinión se enfocan en la distinción de piezas de textos (palabras, frases, oraciones o documentos) según su polaridad; es decir, intentan distinguir si la opinión o la evaluación es positiva, negativo a o neutra. Para ellos se requiere el pre-procesamiento de los textos con el objetivo de obtener dichas piezas textuales, confeccionar corpus y diccionarios, asignarle una clasificación aprioris (en al caso de una clasificación supervisada), así como llevarlas a un formato útil según los método de extracción de opinión donde se utilizarán posteriormente.

La etapa del pre-procesamiento es la etapa del proceso de Minería de Texto donde se transforman los textos a una representación estructurada o semiestructurada de su contenido.

La representación intermedia de los textos debe ser, por una parte, sencilla para facilitar el análisis de los textos, pero por otra parte, completas para permitir el descubrimiento de patrones interesantes, e incluso de nuevos conocimientos.

Por otra parte, puede notarse como ésta es una tarea compleja tanto desde el punto de vista manual como automático, ya que para realizarla los investigadores y anotadores requieren de mucho tiempo, lo cual resulta ineficiente. Por lo que desarrollar herramientas que apoyen la investigación, mediante el pre- procesamiento automático del texto, facilitando la experimentación y la evaluación de los resultados, es realmente una contribución en recursos para ser más eficientes en esta área de investigación [18].

Capítulo 3

Desarrollo de recursos textuales para la Minería de Opinión

Como se comentó en el capitulo anterior existen escasos recursos textuales para el análisis automatizado de textos de opinión en el idioma español. Además, no se cuenta con colecciones en este idioma anotadas según la taxonomía propuesta por la Teoría de la Valoración. Por lo que los investigadores que trabajan en esta área en el INAOE, se han visto en la necesidad de confeccionar sus propios recursos a partir de cero.

En este capítulo se describe en la sección 3.1 la confección de un léxico de palabras valorativas, en la sección 3.2 cómo se elaboró un corpus de oraciones valorativas y la creación de un pequeño léxico de expresiones extraídas de dicho corpus, en la sección 3.3 se llevo a cabo la evaluación y clasificación de expresiones valorativas

3.1. Léxico de palabras valorativas

La generación de léxicos o diccionarios es de gran utilidad en la extracción automática del tipo y la intensidad de expresiones valorativas, porque pueden servir como palabras semillas para identificar nuevas expresiones del mismo tipo valorativo, o también pueden emplearse como referencia de una buena clasificación *(Gold-standar).*

En esta tesis se desarrolla un nuevo diccionario a partir de dos listas de palabras valorativas clasificadas manualmente. Estas listas fueron creadas asignado a cada palabra una polaridad (positiva o negativa) y un tipo de actitud (afecto, juicio, y apreciación). En cada clase se asignó un valor entero entre 0 y 2, correspondiendo a la pertenencia de la palabra a la clase correspondiente. Así cada lista fue anotada por una persona en particular, sin conocimiento de los resultados entre sí.

El proceso de asignación de un tipo de valoración, ya sea de polaridad o actitud es un proceso complejo no sólo para las máquinas sino para los propios seres humanos, ya que se mueve en el terreno de lo subjetivo. Es decir, los individuos tienen diferentes juicios y sistemas de valores éticos y estéticos entre sí, lo que hace que no siempre coincidan sus opiniones. Esto se complica, más aún cuando se considera el solapamiento entre las clases de actitud de acuerdo a los planteamientos de la teoría de la valoración. Por estas razones fue necesario unir dos listas confeccionadas en una nueva lista tomando un consenso entre las clasificaciones de cada anotador. Debe destacarse que la anotación de las palabras se realizó considerando todos los posibles significados valorativos de las mismas, es decir, sin considerar un contexto específico.

Entonces el proceso desarrollado fue el siguiente:

Respecto de dos listas con las mismas palabras valorativas H1 y H2, ordenadas alfabéticamente y con criterios diferentes en la asignación de valores de pertenencia en cada una de las cinco clases valorativas (positivo, negativo, afecto, juicio y apreciación), se crea un tercer léxico a partir de comparaciones de estas dos y con aportación en criterio propio de una tercera persona H3. En los casos donde coincidían anotaciones de H1 y H2 estos valores se mantenían, en los casos contrarios se consideraba una tercera opinión de H3. A continuación en la figura 3.1 se muestra un ejemplo del proceso de anotación en el que podemos observar que en las dos primeras listas se muestra claramente las variaciones en los valores asignados, lo cual no aporta información consistente para la evaluación de las expresiones, de esta manera surge la necesidad de unificar dichos criterios, derivado de ello surge una tercer lista que está sustentada en la Teoría de la valoración la cual aporta el conocimiento para identificar la valoración, la actitud y la emoción en cada expresión, resultado de esto se obtuvo un conjunto de expresiones valorativas que se consideran aptas para la experimentación.

Fichero H1

Word	|positive|negative|affect|juicio|apreciation
anejo	|2|1|0|0|2|
anorar	|0|1|2|0|0|
anoso	|0|1|0|2|2|

Fichero H2

Word	|positive|negative|affect|juicio|apreciation
anejo	|2|1|2|0|1|
anorar	|1|1|0|0|0|
anoso	|1|2|1|0|1|

Fichero H3

Word	|positive|negative|affect|juicio|apreciation
añejo	|2|1|0|0|2|
añorar	|0|1|2|0|0|
añoso	|0|1|0|2|2|

Figura 3.1: Anotación en los tres léxicos H1, H2 y H3

3.2. Corpus de oraciones valorativas

El contenido obtenido de las opiniones que los usuarios emiten sobre muchas entidades, nos resulta importante en esta investigación, ya que a partir de ello podemos crear un corpus de oraciones amplio extraído de diversas fuentes, a fin de realizar una clasificación sobre lo más valorativo que resulte de esta recopilación.

Haciendo uso de un conjunto de oraciones de diferentes fuentes entre ellas internet, libros, ficticias (inventadas), todas estas con expresiones que aportan alguna opinión o valoración sobre entidades. Ubicadas por la fuente de donde provenían, se tomaron las oraciones con la asignación de ficticias y se hizo de cada una de ellas una búsqueda avanzada en internet a través del buscador de Google, esperando comprobar la correspondencia que existe entre este tipo de oraciones y lo que los usuarios de diversas edades, ideología, raza, entorno social, etc., con sus opiniones expresan en la red, de lo antes mencionado se derivó una correspondencia adecuada para la validación.

Se identificaron las expresiones valorativas que contiene cada oración del total de ella y se le dió una clasificación para poder ubicarla respecto a las demás.

En la figura 3.2 se indica que tipo de clasificación se asignó a la recopilación de oraciones:

Figura 3.2: Clasificación en las oraciones del corpus

La clasificación dada a las oraciones ficticias fue hecha con el siguiente criterio:

Resuelta: Oración encontrada completamente en internet, es posible que varíe de la original por la ausencia de algún intensificador, que el sustantivo esté en plural y la original en singular o viceversa, la conjugación del verbo y el género, detalles que se pueden modificar, pero la oración no puede perder lo esencial que en este caso es la opinión que este emitiendo.

Opcional: Elegida en este clasificación cuando se encuentra la mitad o menos de las expresiones o quizás si se da el caso de encontrar más de la mitad el inconveniente es que ese expresiones no sean lo más valorativas para la oración.

Error: Oración que no contiene expresiones valorativas en relación con la original, se pudo agregar alguna oración si se considerara con algo de valoración para la original, pero son casos especiales únicamente.

En la tabla 3.1 se muestra algunos ejemplos que siguieron el criterio de clasificación anteriormente mencionado:

CLASIFICACION	ORACION ORIGINAL	ORACION DE INTERNET
Resuelta	En los momentos más difíciles actúa con extrema naturalidad.	En los momentos "mas" difíciles actúa con "extrema" suficiencia y naturalidad.
Opcional	Tu esbelto cuerpo me provoca una desorbitante pasión.	Tu cuerpo es mi "desorbitante" pasión me invita me incita me provoca.
Error	Pronunció un paradójico discurso que dejó a todos muy perplejos.	El discurso moderado de Kirchner dejó "muy" perplejos a varios matutinos.

Cuadro 3.1: Clasificación del corpus de oraciones

3.3. Clasificación de Expresiones

Una expresión valorativa es una expresión que contiene al menos una palabra que emite valoración, la evaluación sobre estas expresiones nos permite conocer cuál es el objeto de evaluación de la oración que la compone, de esta manera y haciendo uso del conjunto de oraciones de la clasificación antes realizada, se extrajo expresiones valorativas de cada una, esta selección fue tanto para las Ficticias, las de Internet y de Libros. En el caso de las oraciones Ficticias la mayor parte consideradas fueron las resueltas, analizando mejor oraciones de error se tomaron algunas para este procedimiento.

Las expresiones obtenidas fueron listadas conservando el orden de las oraciones, posteriormente se trabajo con el fichero resultante del listado para elaborar una clasificación basada en el criterio de la Teoría de la Valoración y de los principales enfoques computacionales.

De esta manera fue como se clasificó en uno o más sub sistemas de las clases a lo largo de todas las expresiones valorativas.

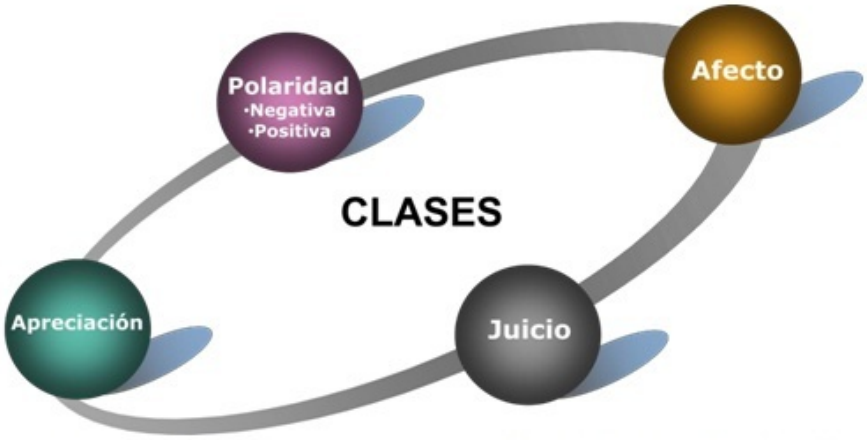

Figura 3.3: Clasificación de expresiones justificada en la Teoría de la Valoración

Como se menciona en la Teoría de la valoración existen dominios de apoyo para poder interpretar lo que expresan los hablantes con sus emisiones textuales. Dentro de estos también hay subsistemas que hacen más minuciosa la clasificación y de la cual nos apoyamos en este procedimiento.

La gradación es el sistema que nos apoyó al manejo en la intensidad de las expresiones y clasificación, proporcionando un valor a la polaridad no solo de positiva o negativa si no de fuerza en la expresión.

La actitud a través de los subsistemas (afecto, juicio, apreciación) facilita la evaluación de las expresiones y a partir del significado se asignó la categoría y valor correspondiente a cada una, para lo anterior se manejó las siguientes categorías:

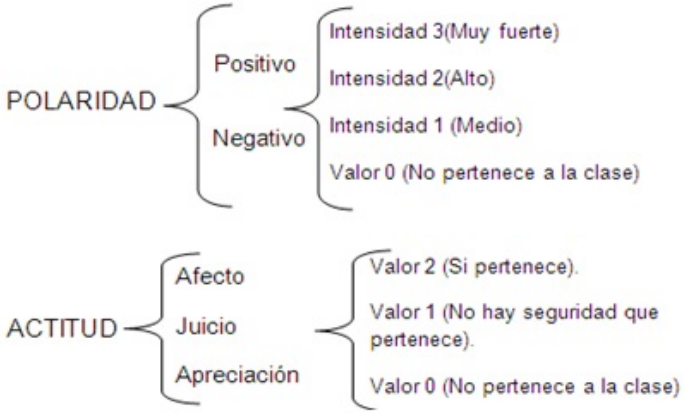

Figura 3.4: Clases y valores otorgados a la clasificación de expresiones valorativas

Es de considerarse que las expresiones deben apoyarse en toda la oración a la cual pertenecen ya que el contexto puede definirla mejor para poder tener una clasificación más precisa, en todas las categorías que se vienen manejando. Con la siguiente tabla se muestra un ejemplo en la elaboración del listado en las expresiones valorativas:

1. Una sola gota lo **colma** y derrama.

2. Apoyan la conclusión de que el **desarrollo de la compasión** y el altruismo tiene un efecto positivo sobre nuestra salud física y emocional.

3. El vicepresidente ha comprobado en primera persona la **complejidad del conflicto.**

Expresiones	CLASES				
	Polaridad		Afecto	Juicio	Apreciacion
	Positiva	Negativa			
Colma	1	1	2	0	0
desarrollo de la compasión	1	0	0	2	0
complejidad del conflicto	0	2	0	0	2

Cuadro 3.2: Clasificación y asignación de valores en expresiones

Se ha obtenido satisfactoriamente los recursos textuales en el idioma en español, desarrollado bajo el fundamento de la Teoría de la Valoración y por medio del trabajo manual, se creó un léxico de palabras valorativas además de la clasificación de un corpus de oraciones y un conjunto de expresiones valorativas extraídas de las oraciones obtenidas, estos recursos elaborados nos sirven como una importante herramienta para la experimentación y validación, con estos recursos se cuenta con el apoyo para continuar con el desarrollo de la investigación, de lo cual se hará referencia en capítulos posteriores.

Capítulo 4

Plataforma para el procesamiento de textos de opinión en español

Para el desarrollo de la plataforma de procesamiento de textos se usó el ciclo de vida en cascada considerando las actividades fundamentales del proceso de especificación, desarrollo, validación, y evolución. En este capítulo se representan las fases de requerimiento, diseño de software, y pruebas, se explicará a detalle cada módulo así como su interfaz de usuario posteriormente a la presentación del ciclo de vida.

4.1. Definición de objetivos

La plataforma a desarrollar se encarga de realizar ciertos procesos en los datos de diversas procedencias en el área de investigación de Minería de Opiniones. Es una ayuda para visualizar el panorama al cual se encamina la investigación y para poder tomar decisiones en cuanto a las variables que participan en las experimentaciones de clasificación con la herramienta de aprendizaje Weka. [1]

4.2. Análisis de los requisitos y su viabilidad

La recopilación y análisis de los requisitos del investigador en este caso ha sido de manera gradual y en conjunto al desarrollo. La viabilidad ah sido posible gracias al contar con el equipo de cómputo necesario para el procesamiento de los datos de que se disponen. Basado en los requerimientos se diseño el diagrama de casos de uso que se observa en la figura 4.4

[1] Entorno para Análisis del Conocimiento de la Universidad de Waikato. http://www.cs.waikato.ac.nz/ml/weka/

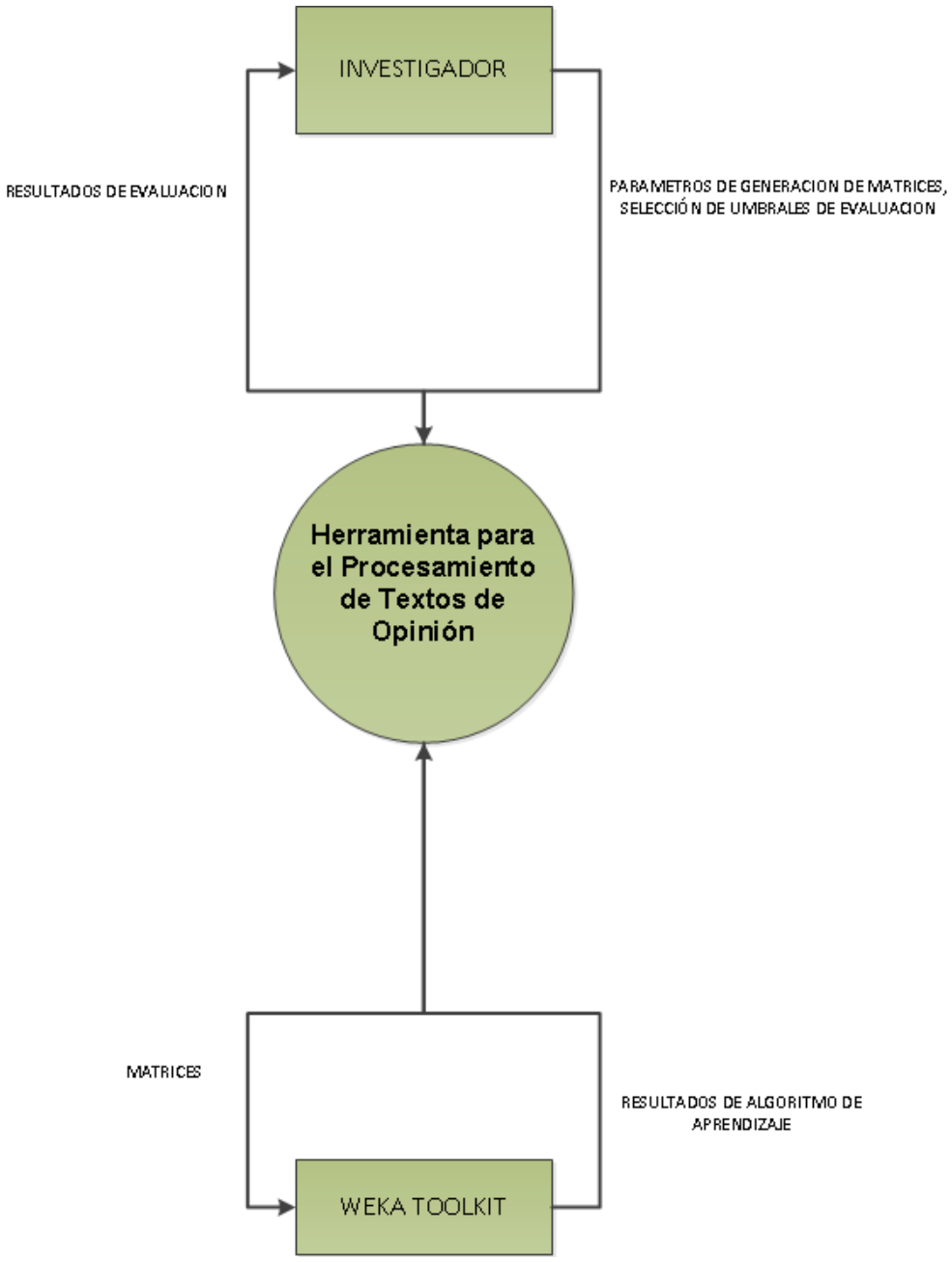

Figura 4.1: Diagrama de contexto: Nivel 0

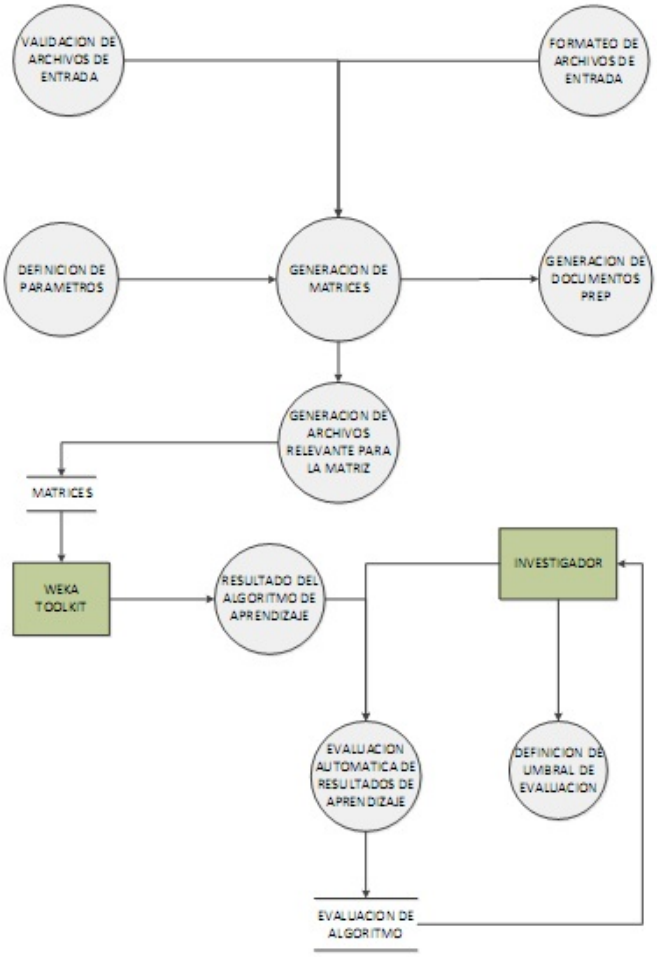

Figura 4.2: Diagrama de nivel superior: Nivel 1

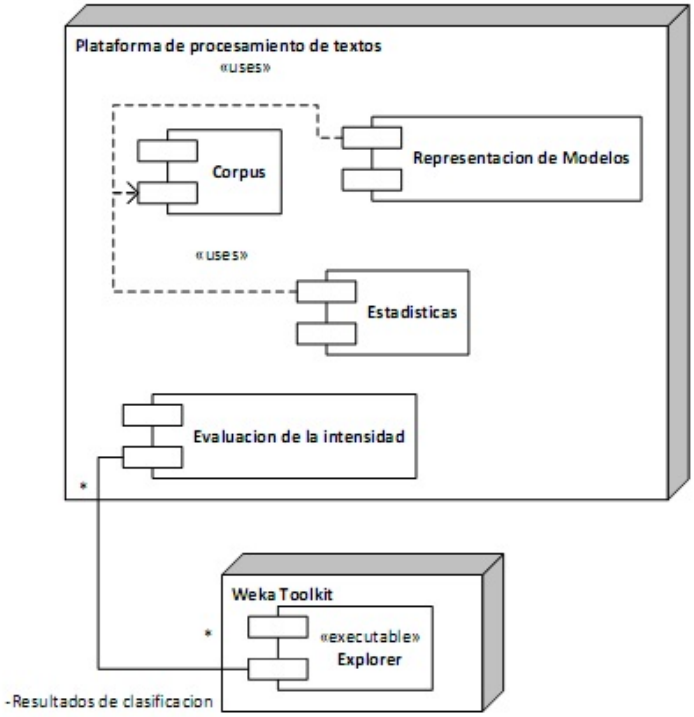

Figura 4.3: Diagrama de componentes

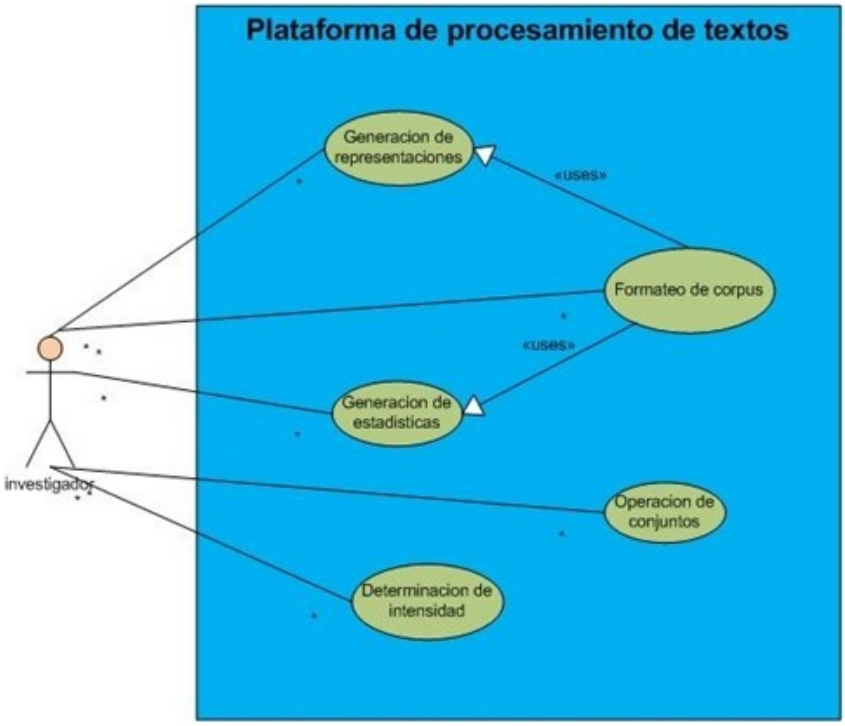

Figura 4.4: Casos de uso de la herramienta

Posteriormente de lo general partimos a lo particular, donde se profundizó en cada caso de uso. Donde se define con certeza las posibilidades y limitaciones que tendrá el usuario con la herramienta. Las tablas que van de la 4.1 a la 4.5 muestran a detalle estas posibilidades y limitaciones del usuario y la herramienta.

Descripción de casos de uso:

Nombre	**Generación de representaciones.**
Actor	**Investigador.**
Descripción	**Describe el proceso de generación de representaciones de los vectores.**

Flujo principal:	Eventos Actor	Eventos Sistema
	1.-Carga corpus, vocabulario y términos clase.	1.-Indica las direcciones de localización en la computadora de los ficheros.
	2.-Se elige/omite opción de formato para el corpus	2.-se formatean los archivos ingresados y se carga el corpus al sistema con la opción seleccionada.
	3.-Se selecciona rango de palabras	3.- Se generan la matriz de vectores
	4.-Seleccionamos ubicación para guardar la matriz	4.- Se generan los datos faltantes de la estructura .arff y se guardan junto con la matriz
		5.- Reinicia el caso de uso
Alternativa 1	1.- Carga corpus	1.- indica direcciones de localización en la computadora de los ficheros.
	2.-Seleccionar generación de ngramas	2.-despliega ventana y espera el parámetro de números de n-gramas
	3.-Se elige rango de gramas a procesar	3.- Procesa y muestra la tabla de n-gramas solicitados.
	4.- Se guarda tabla o la parte seleccionada	4.- Reinicia el caso de uso.
Alternativa 2	1.- Carga corpus, vocabulario, términos clase.	1.- indica direcciones de localización en la computadora de los ficheros.
	2.-Definir parámetros para txt.prep	2.-Valida los parámetros introducidos de acuerdo a los límites definidos.
	3.-Se solicita obtener los archivos txt.prep en una ubicación deseada	3.- Se generan los archivos .txt.prep, se genera un fichero de relación de términos tomados en cuenta.
	4.- Se acepta ventana de finalización exitosa.	4.- Reinicia el caso de uso.
Alternativa 3	1.- Carga corpus, vocabulario, términos clase.	1.- indica direcciones de localización en la computadora de los ficheros.
	2.-Definir parámetros para txt.prep	2.-Valida los parámetros introducidos de acuerdo a los límites definidos.
	3.-Se solicita obtener los archivos txt.prep en una ubicación deseada	3.- Se generan los archivos .txt.prep, se genera un fichero de relación de términos tomados en cuenta.
	4.- Se acepta ventana de finalización exitosa.	4.- Reinicia el caso de uso.

Cuadro 4.1: Caso de uso de generación de representaciones

Nombre	**Generación de estadísticas.**
Actor	**Investigador.**
Descripción	**Describe el proceso de generación de estadísticas del corpus basado en algunos parámetros.**

Flujo principal:	Eventos Actor	Eventos Sistema
	1.-Carga ficheros de frecuencias y polaridades opuestas.	1.-Formateado de listas
	2.-Se elige frecuencia de términos y se especifican los parámetros.	2.-Se genera la tabla de frecuencias
	3.-Se guarda parte o la tabla completa	3.- Guarda los datos en un fichero .txt
		4.- Reinicia el caso de uso.
Alternativa 1	1.- Carga ficheros de frecuencias y polaridades opuestas.	1.- indica direcciones de localización en la computadora de los ficheros.
	2.- Se elige polaridad opuesta y se configuran los parámetros.	2.-despliega ventana con tabla de términos y sus frecuencias.
	3.-Se guarda la tabla o parte de ella.	3.- almacena los datos en un fichero .txt
		4.- Reinicia el caso de uso.
Alternativa 2	1.- Carga ficheros de frecuencias gramaticales.	1.- indica direcciones de localización en la computadora de los ficheros.
	2.- Se guarda la tabla o parte de ella.	2.-Se almacena en un fichero .txt
		4.- Reinicia el caso de uso.

Cuadro 4.2: Caso de uso de generación de estadísticas

Nombre	**Operación de conjuntos.**
Actor	**Investigador.**
Descripción	**Describe el proceso de operación entre listas de términos.**

Flujo principal:	Eventos Actor	Eventos Sistema
	1.-Carga dos listas de términos	1.-Formate la listas
	2.-Se elige el tipo de operación de conjunto a aplicar sobre las 2 listas cargadas	2.-Realiza la operación y finaliza habilitando el botón de guardar los resultados a la vez que despliega el numero de resultados en pantalla.
	3.-Se guarda el fichero en formato .txt	3.- Reinicia el caso de uso.

Cuadro 4.3: Caso de uso de operación de conjuntos

Nombre	**Determinación de intensidad.**
Actor	**Investigador.**
Descripción	**Describe el proceso para la determinación de intensidades a partir de ficheros generados de Weka.**

Flujo principal:	Eventos Actor	Eventos Sistema
	1.-Carga resultados con predicciones generados por Weka con o sin IDs	1.-Analiza y separa las clases con sus probabilidades mostrándolas en una tabla por clase.
	2.-Se selecciona trabajar estadística	2.-Despliega ventana con umbralas de intensidades.
	3.-Se selecciona una umbral de intensidad	3.- Despliega una tabla de intensidades indicando los falsos positivos.
	4.-Guardar la tabla o parte de ella con o sin falsos positivos	4.- almacena los datos en un fichero .txt
		5.- Reinicia el caso de uso.

Alternativa:	Eventos Actor	Eventos Sistema
	1.-Carga listas de términos con sus intensidades	1. Se validan los formatos de las listas.
	2.-Se inicia proceso de evaluación	2.-Despliega una tabla comparativa de las intensidades.
	3.-Se guardan los resultados.	3.- Se genera una tabla de confusión y se guarda con la tabla de intensidades.
		4.- Reinicia el caso de uso

Cuadro 4.4: Caso de uso de determinación de intensidad

Nombre	**Formato de Corpus y unión de documentos.**	
Actor	**Investigador.**	
Descripción	**Describe el proceso para dar un formato o estructura para trabajar con los documentos de texto en la plataforma.**	
Flujo principal:	Eventos Actor	Eventos Sistema
	1.- Carga uno o más documentos de texto en formato texto plano o etiquetados tipo TTG	1.-Verifica si los archivos seleccionados tienen contenido de texto. Analiza el texto o los textos y verifica que cumpla el formato.
	2.-Elige dar formato en la Ventana de incompatibilidad de formato.	2.-El texto ingresado o la unión de todos los archivos de textos se somete a formateo basa en las reglas establecidas.
	3.-Selecciona el botón de aceptar al mostrar texto formateado exitosamente.	3.- El texto formateado exitosamente es cargado en la plataforma para su manipulación de pre-procesamiento o obtención de datos estadísticos.
		4.- Reinicia el caso de uso
Alternativa:	Eventos Actor	Eventos Sistema
	1.-Carga uno o más documentos de texto en formato texto plano o etiquetados tipo TTG	1.-Verifica si los archivos seleccionados tienen contenido de texto. Posteriormente inicia la unión del archivo o archivos en un solo documento. Si el formato es TTG se crea un corpus lematizado y se generan los archivos individuales TTG lematizados en TXT.
	2.-Obtiene la ubicación de los archivos generados y su formato.	2.- Despliega en un cuadro de texto la ubicación de los archivos generados y las advertencias de posibles archivos vacíos si se presenta el caso.
		5.- Reinicia el caso de uso.

Cuadro 4.5: Caso de uso de formato de corpus y unión de documentos

Interfaz de navegación del sistema

La figura 4.5 muestra la interfaz de inicio del sistema la cual contiene un conjunto de pestañas que permiten el acceso a los diferentes módulos.

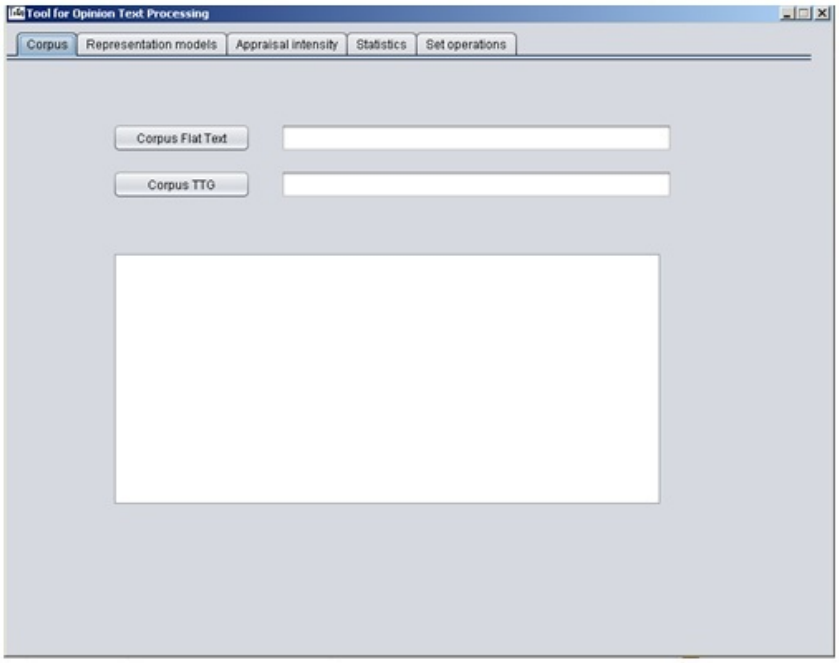

Figura 4.5: Interfaz de navegación del sistema

Módulos

A continuación se exponen los diferentes módulos del sistema.

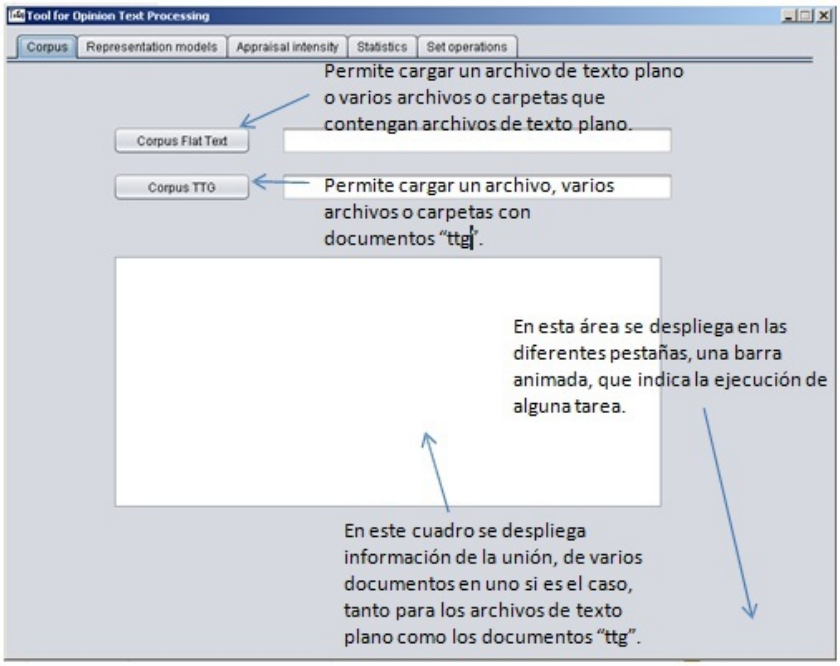

Figura 4.6: Tratamiento del corpus

Esta primera ventana como se observa en la figura 4.6, permite la unión de archivos de texto plano, como de documentos “ttg”. Los documentos de extensión “ttg” son producto de la salida de una herramienta (Tree Tagger) que extrae el lema de una palabra lematizador [19]. Esta unión lleva la validación de cada archivo para posteriormente indicar el orden de los archivos que conforman el documentos final, sea “ttg” o texto plano. También da el formateo de un texto plano si este carece del formato adecuado para la herramienta.

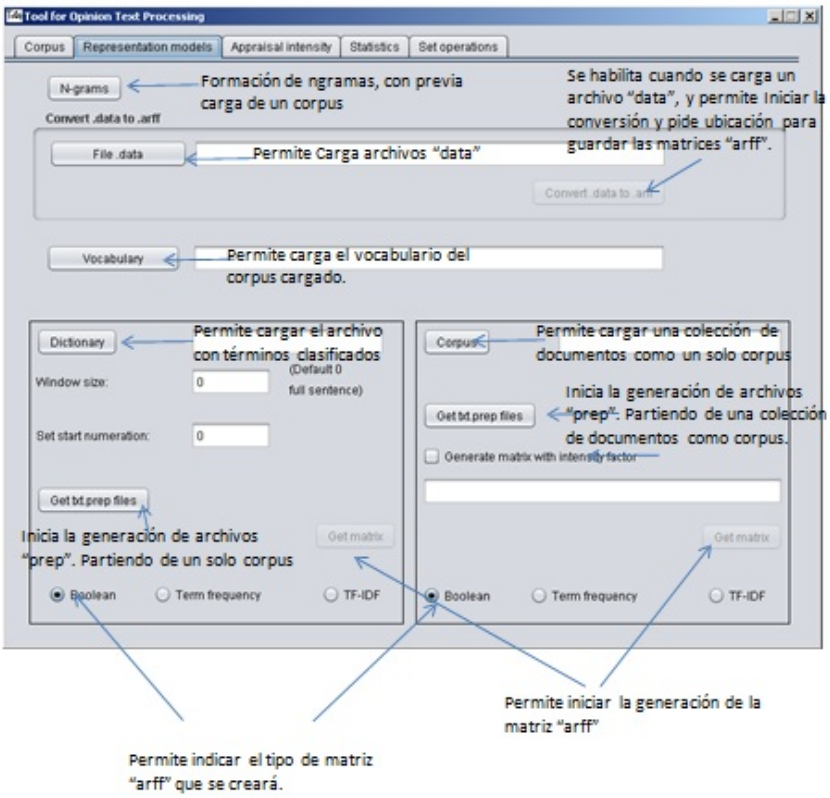

Figura 4.7: Generación de representaciones

La pestaña Representation Models que se observa en la figura 4.7, contiene la generación del formato Weka que es generación de vectores para la clasificación con o sin frecuencias, la generación de los términos (n-gramas), indexación aleatoria que es la creación de documentos de texto por cada termino a clasificar donde dentro se encuentran términos únicos, una opción para generar un formato especifico para el pre-procesamiento de la indexación aleatoria y finalmente la conversión de vectores reducidos a formato ARFF.

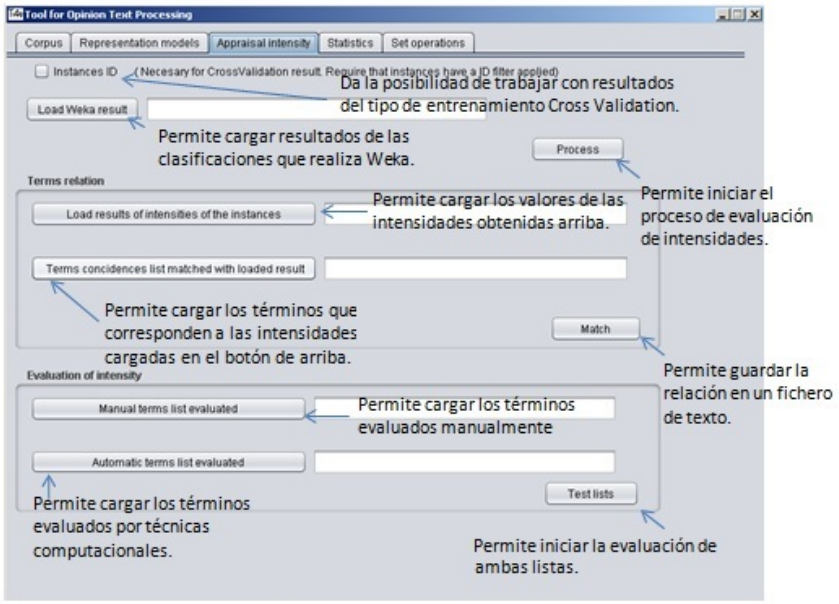

Figura 4.8: Determinación de intensidades

Appraisal Intensity que se muestra en la figura 4.8 es la pestaña que se encarga de la determinación de intensidades, para obtener basado en números y cálculos probabilísticos, una tabla de contingencia, en la cual se ve las intensidades asignadas basado en el resultado de *Weka* del área de predicciones fueron acertados o erróneos en comparación a la asignación manual de intensidades.

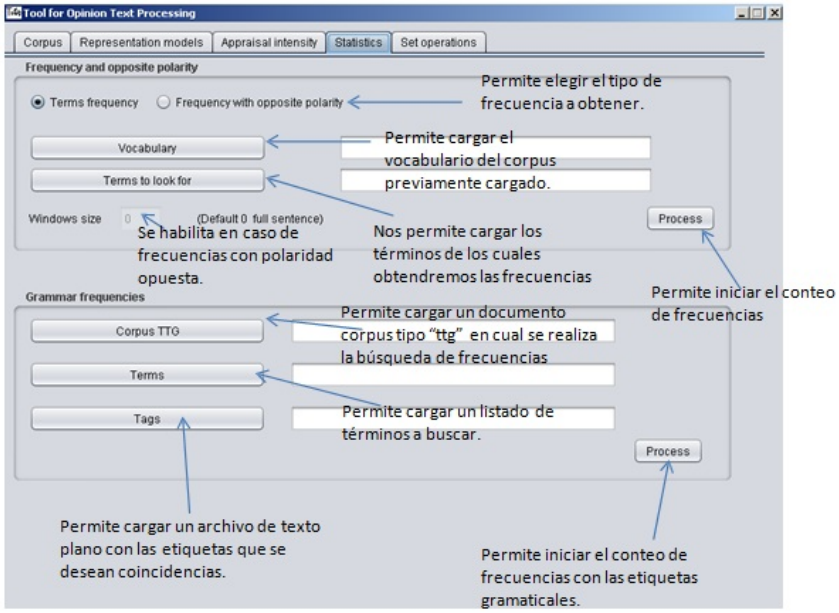

Figura 4.9: Generación de estadísticas

La pestaña Statistics que se muestra en la figura 4.9 consiste en la generación de estadísticas, que es obtener palabras por categorías gramaticales, contar frecuencias en el corpus previamente una lista de términos dados, contar la polaridad invertida dado también unos términos que se

encuentren cercanos a una rango de términos que se especifique, y la frecuencia de co-ocurrencia que es basado en 2 listas de termos dados, buscar la coincidencia en el corpus en una misma oración.

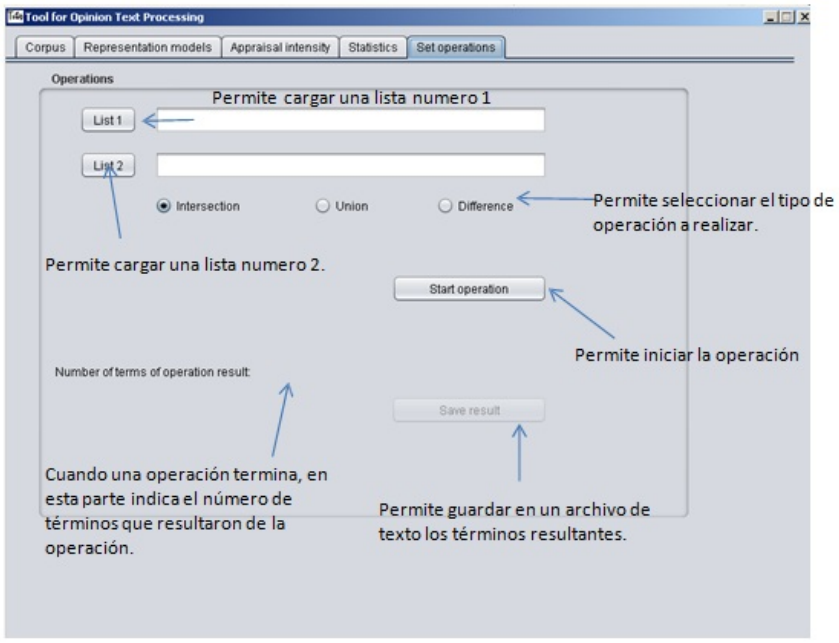

Figura 4.10: Operación de conjuntos

Por último Set Operations que se muestra en la figura 4.10 permite la realización de operaciones de conjuntos que se basa en la conocida teoría de conjuntos, siendo los conjuntos 2 listas de términos con las cuales se trabajarán y que podrán realizar las operaciones de:

- Intersección
- Unión
- Diferencia

Dando como resultado una lista la cual se guarda para luego ser analizada. Hay que tomar en cuenta que no es lo mismo la operación de diferencia con lista 1 diferencia lista 2 a lista 2 con la diferencia de lista 1.

4.2.1. Requerimientos de software del sistema

- JRE (Java Runtime Environment) 1.6.0 o versión superior
- Windows Vista, Windows 7

4.2.2. Requisitos de hardware mínimos

- 1 Gigabyte de memoria Ram (se recomiendan 4 Gigabytes)
- Procesador AMD o Intel a 2.0 Ghz (se recomiendan múltiples núcleos)
- 10 Megabytes en disco duro.

4.3. Diseño general

La aplicación debe ser eficiente en el uso de memoria, por que trabajará con fichero de datos que contengan gran cantidad de datos textuales, y que a partir de ficheros textuales generará datos con tamaños en memoria considerables e información, por tal motivo es necesario implementar el mejor algoritmo que haga uso eficiente de memoria.

4.4. Diseño en detalle

Generación de las representaciones

En la sección de generación de las representaciones se incluye lo que es, la generación del formato Weka que es generación de vectores con valores booleanos, de frecuencias y TF*IDF, la generación del vocabulario (n-gramas) con su frecuencia, indexación aleatoria, una opción para generar un formato especifico para el pre-procesamiento de la indexación aleatoria y la generación de vectores a partir de un formato especial que se genera cuando se reducen los vectores para mejorar aspectos como el uso excesivo de memoria en la etapa de procesamiento.

El corpus sobre el que se trabaja puede ser seleccionado directamente como un único archivo de texto o un conjunto de archivos de texto que se realiza por multi-selección y que la plataforma realiza la unión de los archivos seleccionados posteriormente a la unión desglosa en pantalla en un cuadro de texto, la ubicación del archivo como resultado de la unión, junto con un archivo de texto que muestra el listado en el que se conformó el corpus final. Teniendo en cuenta lo anterior finalmente se procesa un solo corpus con la unión de varios archivos o solo uno y se valida si está libre de errores, si se llega encontrar errores se proponen 3 opciones: limpiar corpus, omitir corpus y cancelar.

Limpiar corpus se realiza bajo las especificaciones siguientes:

- Todas las oraciones deben estar libres de signos o símbolos excepto los signos de admiración, interrogación y coma
- Todo debe estar en minúsculas
- No deben haber espacios dobles o tabulaciones

Omitir corpus como lo expresa el término, deja el corpus con cualquier tipo de error con la posibilidad de que la plataforma falle al procesarla. Cancelar, anula el proceso de cargar el corpus.

Ya que contamos con el corpus depurado y las oraciones bien delimitadas unas de otras, queda listo para ser utilizado en varios módulos de la plataforma. Las listas de términos o expresiones de diferentes tipos sea vocabulario, términos clase u otros son depurados de posibles símbolos como el corpus además de pasar todas a minúsculas y esto funciona así para toda la plataforma a excepción de las listas de términos con intensidades que pertenece al caso de uso Determinación de Intensidades.

Indexación aleatoria con extensión .txt

La generación de indexación aleatoria con salida en archivo con extensión txt, se inicia y se lleva a cabo cuando se acepta en la ventana de dialogo donde indica este proceso después de hacer clic en “Generar documentos .prep”

Este procesamiento requiere un corpus, vocabulario, término a clasificar y si lo requiere parámetros como rango e inicio de numeración. Teniendo lo anterior el proceso es el siguiente:

1. Se genera un fichero por cada término a clasificar y se nombra al fichero según el nombre de la clase a la que pertenece el término a clasificar más 6 dígitos que van en incremento conforme se generan los ficheros.

2. Dentro de cada fichero habrá términos del vocabulario por cada oración y separados por salto de línea delimitando términos del vocabulario de una oración y otra.

3. Los términos que se obtiene en los archivos txt, son términos del vocabulario que se encuentran en una oración con el nombre del archivo txt en el que se encuentra el término del vocabulario, es posible manejar un rango partiendo del término-clase a un número determinado de palabras antes y después del término-clase

4. Finalmente se obtendrá N archivos basados en los términos a clasificar que tuvieron coincidencia de esa lista.

Indexación aleatoria y el formato de .prep

Los requerimientos para este módulo son los mismos que la indexación aleatoria con formato .txt, este formato siempre se obtendrá después de iniciar el proceso con el botón "Generar documentos .prep", requiere que los términos del vocabulario dentro de cada archivos estén junto a un diagonal seguida de 2 jotas, igual a esto: "término/JJ" y finalizando con un salto de línea, lo que se obtiene una lista de términos sin distinción entre oraciones, el nombre del fichero que anteriormente era el termino a clasificar, se sustituye por el nombre de la clase a la que pertenece el término a clasificar más 6 dígitos que van en incremento con forme se generan los archivos y con una extensión ".txt.prep".

Generación de vectores

Se genera el formato con extensión ARFF por sus siglas en ingles **(Attribute- Relation File Format),** que es una relación de atributos con las instancias que se pretenden clasificar. Para generar esto se requiere de un corpus, una lista del vocabulario o n-gramas (conjunto de n términos) y una lista de términos a clasificar.

Se genera un archivo con formato para ser procesado por algunos clasificadores de Weka, a parte del fichero .arff que se genera un archivo que es la relación de términos clase que fueron tomados en cuenta, es decir el número de términos-clase que aparezcan en la lista será el número de vectores creados en la matriz y en ese mismo orden para saber la relación de los vectores a que término clase representan. Existen tres opciones partiendo de las entradas que se mencionan anteriormente, matriz booleana, matriz de frecuencias y matriz TF*IDF. Estas variantes de matrices pueden hacer uso del parámetro rango, la cual sirve para indexación aleatoria y también para la generación de matrices, donde se puede limitar a un número de palabras partiendo del término clase como sucede en indexación aleatoria. La generación de matriz booleana consiste en que cuando un término clase es detectado y en esa oración un término del vocabulario aparece, se marca en el vector el valor de uno, esto nos indica si a cierto rango de palabras o en la oración según como se haya definido existe una coincidencia de un término clase-termino único. En la estructura de un archivo .arff consta de "@relation" que indica el nombre que identifica a la matriz mas los "@attribute" seguido de un identificador que representan todo el vocabulario. El término "numeric" que va seguido de cada identificador de atributo es cambiado por nominal para el caso de booleano que se expresa así: "0,1" indicando que únicamente esos valores serán representados en los vectores, la segunda opción es obtener una matriz de frecuencias que tiene

un formato que comparten estas tres opciones con pocas variantes:

```
1  @relation H1afectopositivo
2
3  @attribute atrib1 numeric
4  ..
5  .
6  .
7  .
8  .
9  @attribute atrib4096 numeric
10 @attribute acctitud {-1,-2}
11
12 @data
13
14 0,0,1,0,1,9,0,0,0,0,1,4,0,0,-1
15 ..
16 .
17 .
18 .
19 .
20 0,0,0,0,0,3,0,0,0,0,1,0,0,0,-1
```

Figura 4.11: Estructura de un archivo ARFF matriz de frecuencia

Siendo H1afectopositivo el nombre de la clase de los términos a clasificar. En la línea 3 "atrib1" indica un término único, por lo que esta línea se escribirá tantas veces cambiando el número que esta junto a "atrib" como términos el vocabulario tenga. En el ejemplo indico que habían 4096 términos únicos y el término "numeric" indica que el vector contendrá valores numéricos diferentes a booleano. Finaliza esta parte con "@attributeacctitud -1,-2" seguido de un salto de línea y la etiqueta "@data", seguida de un salto de línea y aquí es donde comienza la matriz de vectores.

Estos vectores están conformados de la siguiente manera: las columnas son el vocabulario y los renglones los términos a clasificar, lo que quiere decir es que en la oración donde haya coincidencia del término del vocabulario con el término a clasificar será contabilizado en esta matriz.

Término único X

Término a clasificar Y 7

Lo anterior muestra que dentro del corpus conformado por centenas de oraciones, hubo coincidencia del término Y y el término X en siete oraciones distintas.

Al finalizar las frecuencias de coincidencias se asigna "-1" indicando la finalización de este vector de coincidencias.

Existe la tercera opción donde cambia los valores de la matriz de vectores por valores TF*IDF.El TF*IDF es un peso frecuentemente usado en la recuperación de información o minería de texto. Este peso es una medida estadística para evaluar que tan importante es una palabra de un documento en una colección o corpus. La importancia incrementa proporcionalmente al número de veces que aparece la palabra en el documento pero es compensada por la

frecuencia de la palabra dentro de la colección o corpus.

Para obtener el TF*IDF

Se obtienen las frecuencias de los términos únicos por cada oración y se divide cada frecuencia entre el total de coincidencia de todos los términos únicos de esa oración obteniendo las frecuencias pesadas (Ver tabla 4.6). Posteriormente se multiplica por el logaritmo natural del número total de oraciones dividió por la cantidad de oraciones en la que aparece el término.

$$\frac{f}{\Sigma f0} \cdot \log_{10} \frac{n0}{\Sigma fT}$$

Donde:

f = frecuencia de un término
n0 = Número de oraciones totales

$$\Sigma f0 =$$

Sumatoria de las frecuencias de todos los términos únicos en una oración

$$\Sigma fT =$$

Sumatoria del número de oraciones en los que aparece el término único

	TerminoUnico1	TerminoUnico2	TerminoUnico3	Terminounico4
Oración 1	5/19	0	6/19	8/19
Oración 2	0	1	0	0
Oración 3	10/11	1/11	0	0

Cuadro 4.6: Matriz de frecuencias pesadas

El TF*IDF de los términos de la oración1 son los siguientes:

Terminounico1

$$\frac{5}{19} \cdot \log_{10} \frac{3}{2} \therefore = 0.26315$$

Terminounico2

$$0 \cdot \log_{10} \frac{3}{2} \therefore = 0$$

Terminounico3

$$\frac{6}{19} \cdot \log_{10} \frac{3}{1} \therefore = 0.31578$$

Terminounico4

$$\frac{8}{19} \cdot \log_{10} \frac{3}{1} \therefore = 0.42105$$

Por lo que para la matriz de vectores TF*IDF obtenemos los valores de la tabla 4.7

	TerminoUnico1	TerminoUnico2	TerminoUnico3	Terminounico4
Oración 1		0		
Oración 2	0	0.40546	0	0
Oración 3	0.36860	0.03686	0	0

Cuadro 4.7: Matriz de TF*IDF

Las secciones anteriores se refirieron a corpus como un archivo de texto, existe un módulo que maneja la generación de vectores partiendo de un conjunto de archivos de textos que se le considera como un corpus pero de documentos de texto y no de oraciones como se explicó anteriormente.

Indexación aleatoria y formato .prep para colección de documentos

La finalidad de este módulo es la misma que la otra, sin embargo los datos de entrada cambian debido que partimos de una colección de documentos de texto y un vocabulario. El vocabulario se aplicará como un tipo de filtrado donde las palabras que se carguen en esta lista serán las que se buscarán dentro de la colección de documentos, por cada coincidencia de la lista del vocabulario se escribirá en un archivo nuevo que lleva parte del nombre del archivo donde existió coincidencia y el nombre de la carpeta del archivo original; esto se realiza debido a que los archivos pueden llevar el mismo nombre pero pertenecer a diferentes carpetas. Al final tendremos únicamente archivos con extensión .prep con el formato que manejamos anteriormente "termino /JJ" y finalizado con salto de línea. En este módulo no despliega la posibilidad de crear archivos .txt como lo hacia el modulo anterior.

Generación de vectores partiendo de una colección de documentos

Para generación de los vectores ahora requeriremos dos entradas en lugar de las tres que necesitábamos anteriormente. Tendremos una colección de documentos y el vocabulario. Como en el módulo anterior comparte una lista de relación que para este caso es documento-vector, donde encontraremos en el orden de los vectores los nombres de los archivos. La identificación

es importante en la investigación si no se podría evaluar estas matrices de vectores.

La colección de documentos y el vocabulario indica que no podremos fijar un rango como parámetro como se explicó en el módulo anterior, dado que prescindimos de la lista de términos a clasificar y esta era el punto de referencia del rango. Ahora obtendremos coincidencias donde un término del vocabulario aparezca dentro de un documento. Esto quiere decir que tendremos de filas al número total de documentos y de columnas al número total de términos en el vocabulario.

Generación de vectores booleanos

La finalidad de esta matriz de vectores es ser procesado por el clasificador Weka lo cual como las anteriores matrices las generamos con la estructura de un archivo ARFF. Se crean estos vectores con valores de 0 o 1, como el módulo anterior, indicando el número uno la coincidencia de un término del vocabulario en un documento o de lo contrario el valor de cero.

Ejemplificaremos con un vocabulario de siete términos y una colección de tres documentos que se muestran en la tabla 4.8

	T1	T2	T3	T4	T5	T6	T7
Doc1	0	0	0	1	1	0	1
Doc2	1	1	1	1	1	0	1
Doc3	0	0	0	0	0	0	1

Cuadro 4.8: Matriz de frecuencias

La matriz de vectores se conforma por presencias y ausencias de los términos del vocabulario que en este ejemplo son T1 hasta T7 para la colección de doc1 a doc3. Esta matriz va posterior a los parámetros de la estructura del archivo ARFF.

Generación de vectores de frecuencia

La generación de vectores de frecuencia es muy parecida a la de booleanos en su procesamiento, solo que esta cuenta el número de coincidencias en vez del booleano que solo toma en cuenta una coincidencia en un documento. Por lo que la matriz contendrá valores de 0 a un número X veces que aparezca el término del vocabulario en un documento.

Generación de vectores TF*IDF

Se obtienen las frecuencias de los términos del vocabulario por cada documento y se divide cada frecuencia entre el total de coincidencia de todos los términos del vocabulario de ese documento. Posteriormente se multiplica por el logaritmo natural del número total de documentos en las que aparece ese término dividido por la cantidad de documentos en la que aparece el término único.

$$\frac{f}{\sum f0} \cdot \log_{10} \frac{n0}{\sum fT}$$

Donde:

f = frecuencia de un término
n0 = Número de documentos totales

$$\Sigma f0 =$$

Sumatoria de las frecuencias de todos los términos únicos según documento

$$\Sigma fT =$$

Sumatoria del número de documentos en los que aparece el término único

Tomaremos el ejemplo TF*IDF del módulo en el que se utilizaba un fichero como corpus adaptándolo a la colección de documentos en vez de oraciones.

Tenemos que la matriz de frecuencias pesadas es la tabla 4.9

	TerminoUnico1	TerminoUnico2	TerminoUnico3	Terminounico4
Doc. 1	5/19	0	6/19	8/19
Doc. 2	0	1	0	0
Doc. 3	10/11	1/11	0	0

Cuadro 4.9: Matriz de frecuencias pesadas

El TF*IDF de los términos del Doc. 1 son los siguientes:

Terminounico1

$$\frac{5}{19} \cdot \log_{10} \frac{3}{2} \therefore = 0.26315$$

Terminounico2

$$0 \cdot \log_{10} \frac{3}{2} \therefore = 0$$

Terminounico3

$$\frac{6}{19} \cdot \log_{10} \frac{3}{1} \therefore = 0.31578$$

Terminounico4

$$\frac{8}{19} \cdot \log_{10} \frac{3}{1} \therefore = 0.42105$$

	TerminoUnico1	TerminoUnico2	TerminoUnico3	Terminounico4
Doc. 1		0		
Doc. 2	0	0.40546	0	0
Doc. 3	0.36860	0.03686	0	0

Cuadro 4.10: Matriz de TF*IDF de colección de documentos

Finalmente obtenemos la matriz TF*IDF de la colección de documentos que se observa en la tabla 4.10

Generación de matriz con factor de intensidad

Esta representación de vectores es único de este módulo y consiste en obtener la matriz de cualquiera de las tres variantes: Booleano, Frecuencia, TF*ID. Tomando como entrada

un vocabulario con intensidades que es un fichero de texto con la estructura “término único”+”espacio”+”intensidad”+”salto de línea” y una colección de documentos. Posteriormente multiplicar cada término único de cada vector por su intensidad. La intensidad es un rango de valor numérico que es estipulada por el criterio de quien investiga y que le asigna a cada término del vocabulario.

Ejemplificaremos con la matriz TF*IDF que hemos elaborado en la explicación anterior. Tomaremos como hecho que se trabajo con una lista de intensidades siguientes:

TerminoUnicoA 1

TerminoUnicoB 2

TerminoUnicoC 2

TerminoUnicoD 3

Se han cambiado el identificador del término único por una letra para evitar confusión en esta parte de las intensidades.

Como hemos mencionado se ejemplifica con una matriz TF*IDF, esta matriz se multiplicará por las intensidades de su respectivo término como se muestra la matriz en la tabla 4.11

	TerminoUnicoA	TerminoUnicoB	TerminoUnicoC	TerminounicoD
Doc. 1	x1	0 x 2	x2	x3
Doc. 2	0 x 1	0.40546 x 2	0 x 2	0 x 3
Doc. 3	0.36860 x 1	0.03686 x 2	0 x 2	0 x 3

Cuadro 4.11: Matriz de TF*IDF de colección de documentos mostrando la multiplicación por la intensidad

La matriz TF*IDF con factor de intensidad quedaría con lo valores finales mostrados en la tabla 4.12

	TerminoUnicoA	TerminoUnicoB	TerminoUnicoC	TerminounicoD
Doc. 1	0.26315	0	0.63156	1.26315
Doc. 2	0	0.81092	0	0
Doc. 3	0.36860	0.07372	0	0

Figura 4.12: Matriz TF*IDF con factor de intensidad

Basado en lo anterior se puede decir que el rango que manejaron para determinar las intensidades era entre 1 y 3

Generación de n-gramas

1. Dado un corpus

2. Dado el número de n-gramas que se desea conformar.

En cada oración del corpus se obtienen el número de términos que se indica actuando de la siguiente manera:

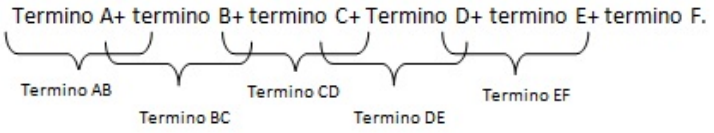

Figura 4.13: Formación de n-gramas partiendo de una oración

Dado el número 2 de n-gramas a formar. Oración i:

Si un n-grama se repite, se va contabilizando el número de veces para al final mostrar en una tabla, los n-gramas encontrados y sus frecuencias.

Permitiendo seleccionar al investigador, los n-gramas que desea guardar en un fichero en forma de lista sin o con las frecuencias para conformar lo que es una lista de términos únicos o vocabulario.

Dentro de la plataforma se limita a la generación de n-gramas a 200 términos, con un valor mínimo de 1 (unigramas).

Conversión de un archivo "data" a .arff

El archivo tipo "data" es el producto de el procesamiento de la indexación aleatoria para reducir los vectores.

Tomando como entrada un archivo tipo data el cual parte de la indexación aleatoria con formato de pre procesado, el archivo tipo data contiene los vectores de todas las clases que se hayan indexado con el formato de pre procesado. Esto quiere decir que si tenemos las 5 clases que hemos manejado en los capítulos anteriores que son afecto, apreciación, juicio, negativo y positivo. Obtendremos vectores que representan la parte positiva y negativa de cada clase. Podemos decir que son diez tipos de vectores que corresponden a las clases antes mencionadas que deben separarse y crear la estructura del archivo tipo arff. Los valores encontrados dentro del vector comprenden valores positivos y negativos normalmente. Como un dato de importante dentro de la conversión, los vectores dentro del archivo tipo data contienen dos elementos, una es el nombre de la clase más su numeración que se generó en la indexación aleatoria y el segundo elemento es el vector. Aquí lo de interés y con el que hay que ser cuidadoso es la asignación de nombres dado que el algoritmo para identificar un nombre de una clase de otra, identifica la cadena previa a cualquier número como la siguiente: "afectopositivo000001" aquí la clase la identifica como "afectopositivo" y como dato de ordenación toma la numeración para crear la matriz en orden ascendente. Es importante mencionar que si llegará a asignarse de nombre a varias clases en algún formato como este:

H1afectopositivo000001
H1afectonegativo000005

La conversión ".data a .arff" solo estaría entregando para esta clase con sus dos polaridades un archivo porque identificaría "H" como nombre de la clase y el resto como identificador para ordenar.

Generación de estadísticas

En la sección de generación de estadísticas consta de obtener palabras por categorías gra-

maticales, obtener frecuencias en el corpus previamente una lista de términos dados, obtener la frecuencia de polaridad invertida dado unos términos que se encuentren cercanos o a una rango de términos que se especifique, y la frecuencia de co-ocurrencia que es basado en dos listas de términos dados, buscar la coincidencia en el corpus en una misma oración.

Frecuencia y polaridad invertida

Teniendo como entrada:

- Un corpus
- Una lista de vocabulario
- Una lista de términos que se desean buscar.
- Especificación del rango para polaridad invertida.

Se obtiene una tabla de los términos a buscar con sus respectivas frecuencias donde se toma como frecuencia la cantidad de veces que el término a buscar coincide en una misma oración con los términos únicos. Existe una opción de permite realizar la búsqueda detrás de los términos de polaridad inversa y que solo se activa con esta sección, donde el rango solo funcionaria para los términos que se encuentran detrás de la polaridad inversa.

La tabla que muestra los términos a buscar con sus frecuencias, es posible seleccionar los términos y guardar solo los términos o si se elige la opción también con las frecuencias respectivas de cada término.

En el módulo siguiente usa un formato de fichero que originalmente se genera por la herramienta *TreeTagger* que mantiene el término original y proporciona el lema. El lema es la abstracción de la palabra que en ingles este procedimiento se le conoce como *stemming*. Estos ficheros llevan como extensión .ttg que pueden ser procesados y lematizados en un solo archivo por la plataforma. La lematización se lleva acabo tomando el lema del término original y volviendo a reformular las oraciones con estos términos. Se puede generar un corpus lematizado partiendo de varias carpetas que contenga colecciones de ficheros .ttg. A la vez que realiza la conversión de cada fichero .ttg a .ttg.txt que indica que fue lematizado y que vuelve a tener la estructura de oraciones dentro del fichero.

Frecuencias Gramaticales

Teniendo como entrada:

Se obtienen las frecuencias de los términos que coinciden con las etiquetas en la lista dada, de lo contrario no serán contabilizados en el corpus.

Determinación de intensidades

En la sección determinación de intensidades, es obtener basado en números y cálculos probabilísticos una tabla de contingencia, para ver las intensidades asignadas basado en el resultado de Weka del área de predicciones fueron acertados o erróneos en comparación a la asignación manual de intensidades.

En esta sección se recibe como entrada de datos un archivo que se obtiene de clasificar una matriz de vectores con la herramienta Weka, que dependiendo del algoritmo y las opciones

seleccionadas incluirá el reporte en el resultado pero para nuestro caso solo es de interés que dentro del archivo se encuentre una sección que contenga lo siguiente:

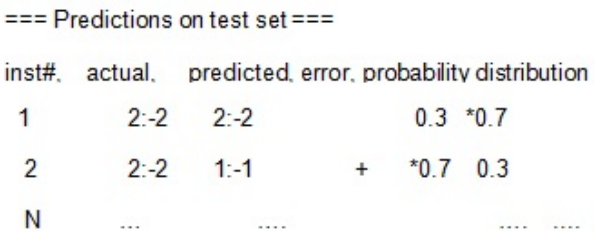

Figura 4.14: Conjunto de prueba de predicciones

Detectando el formato anterior, se tomará en cuenta dos datos, la clase actual y la probabilidad que está señalada por un asterisco.

Con los dos datos que tenemos se divide en dos listas clasificadas por la clase actual, si el reporte que se introdujo contiene dos clases como salida tendremos dos tablas de probabilidades. Si solo existiera una clase como 2:-2 como se observa en el fragmento anterior obtendríamos una tabla con probabilidades.

Después de obtener una tabla o varias tablas de las clases, se procede a trabajar con los umbrales con el botón "trabajar estadísticas", donde despliega una ventana para que se pueda seleccionar 10 umbrales de intensidad y se obtenga las intensidades a partir del umbral seleccionado en una nueva ventana. En la nueva ventana donde se enlista las intensidades, se guarda la tabla completa preservando los falsos positivos para mantener el orden de las instancias que inicialmente estaban en las predicciones del reporte de Weka. Terminando este procedimiento y con la lista guardada de intensidades, procedemos al modulo "Relacionar términos", donde se cargan las lista intensidades y los términos relacionados a los que se obtuvieron en el reporte de Weka, esto permitirá saber que intensidad le corresponde a cada término, para posteriormente que sea guardada la tabla o una selección de ella.

Existe una casilla en este módulo que permite trabajar con resultados entrenados bajo el esquema Cross Validation de la herramienta Weka. Donde las instancias clasificadas son mezcladas y para su identificación requerimos de este parámetro llamado "ID".

Evaluación de intensidad

Tomando como entrada de datos la tabla anterior de términos con sus intensidades, y una segunda tabla realizada manualmente por el investigador y que se toma como correcta para realizar la comparación con la generada automáticamente en el paso anterior. Antes de iniciar el procedimiento de relación de términos con intensidades, se valida la estructura de las listas, que deben llevar el formato de "termino"+"espacio"+"intensidad" seguido de un salto de línea. Si en algún renglón hiciera falta alguna intensidad o fuera un valor no numérico despliega un mensaje de error indicando el número de renglón donde fue hallado dicho error.

Posteriormente se realiza una comparación de ambas tablas y se genera una matriz de confusión donde se indican, con que intensidad fue valorada automáticamente y a que intensidad pertenecía realmente (valorada manualmente).

Para el caso de 2 valores de valoración de intensidad quedaría una matriz igual a la tabla 4.12

V1	V2	
VM	VB	V2
VB	VM	V1

Cuadro 4.12: Valoración de intensidad

Entonces se tiene Valoración 1 **V1** y Valoración 2 **V2**, VM se refiere a *valoración mala* y VB *valoración buena,* tomando en cuenta lo anterior se puede decir que la celda con intersección del valor horizontal con el valor vertical del mismo tipo es el correcto y el que le precede sobre el mismo renglón o que procede son malas clasificaciones.

La matriz de confusión es opcional al realizar la evaluación, puede generarla para guardar la tabla, a demás como toda tabla le permite seleccionar renglones específicos para guardar.

Junto con la generación de la tabla se obtienen tres unidades de medición útiles en el área de recuperación de información que son:

- Precisión: índice de términos correctamente clasificados como positivos entre el total de términos clasificados como positivos
- Recuerdo: índice de términos correctamente clasificados como positivos entre el total de términos positivos
- F-measure: media armónica de la Precisión y el Recuerdo 2xPresicióxRecuedo/(Precisión + Recuerdo)

Operación de conjuntos

La sección operación de conjuntos se basa en la misma teoría de conjuntos, siendo los conjuntos dos listas de términos o expresiones con las cuales se trabajaran y que podrán realizar las operaciones de:

- Intersección
- Unión
- Diferencia

Dando como salida una lista la cual se guarda para luego ser analizada, además de que arroja el número de elementos como resultado de la operación seleccionada.

Codificación (codificación)

Para la implementación, se buscó un lenguaje multiplataforma y de fácil uso, además de que la herramienta experimental de aprendizaje y clasificador que se usa en la investigación es basada en java y se utiliza tanto en Linux como en Windows, obteniendo mejor eficiencia con respecto al manejo eficiente de memoria en Linux. Por lo que se decidió la implementación en Java.

Prueba

En este capítulo reportamos diversas pruebas de la herramienta que se desarrollo para determinar si cumple con las condiciones impuestas al comienzo de esta tesis. Realizaremos actividades

donde se espera que el la herramienta se ejecute en circunstancias previamente especificadas, donde posteriormente se observarán, registrarán y se realizará la evaluación de los resultados. Los casos de prueba a realizar serán en circunstancias normales y extremas donde se variará los tipos de equipos computacionales como las entradas de datos. Esto nos permitirá encontrar defectos y el cual será el éxito de las pruebas. Además se realizará una comparación de mejora de rendimiento que se realizó a la herramienta con base a una versión previa de ésta.

4.4.1. Diseño de las pruebas

Las siguientes pruebas se llevarán a cabo debido a que se consideran de interés para el área que se trabaja:

- Prueba funcional

  Es de importancia saber y comprobar que se llevan a cabo los procesos que inicialmente se definieron por el usuario final. Resultados erróneos afectarían mucho a la investigación de procesamiento de lenguaje que use la herramienta, por lo que son recursos de tiempo, económicos que se pierden y no se recuperan. Se realizará una serie de pruebas que cubra la mayoría de las posibilidades y se utilicen todas las clases del software.

- Prueba de stress

  Nos permitirá conocer las capacidades que tiene la herramienta para realizar tareas con ciertos volúmenes de información. Además de poder saber la mejor configuración de hardware para su mejor desempeño.

4.4.2. Reporte de pruebas

Características del sistema:

- Sistema operativo Windows 7 Ultimate
- Procesador Intel Core 2 Duo T5750 a 2.00 Ghz.
- Memoria RAM 2 GB
- Sistema operativo de 32 bits

4.4.3. Pruebas funcionales

1. **Generando n-gramas**

   Dato de entrada:

   - Corpus de 8,945 kb de tamaño en disco duro, con un total de 56, 970 renglones.

   Parámetros de inicio de la máquina virtual de java:

   - java–Xmx1400m –Dfile.encoding=ISO-8859-1 –jar NumberAddition.jar
     La herramienta nos mostró un ventana de "warning" al ingresar el corpus, donde indica que contiene un formato incorrecto, por lo que se procedió al formateo del corpus. Posteriormente a la carga exitosa y el formato correcto, pasamos a la ventana de generación de n-gramas. Donde generamos unigramas y los ordenamos de mayor frecuencia a menor donde seleccionamos 2 unigramas al azar, uno con la frecuencia más alta y otro con la frecuencia más cercana a mil, como se observa en la figura 4.9 Para corroborar estos 2 unigramas se utilizaron los siguientes editores de texto:

El editor de texto Notepad++ v5.9.8
El editor de texto EditPad Pro Versión 6.6.4

Software / Unigrama	Herramienta desarrollada	Notepad++	EditPad Pro
El	164,640	164,638	164,641
Entonces	1,000	999	999

Cuadro 4.13: Comparativa de frecuencias de palabras en un texto común

Mediante el uso de la opción de contar en el menú búsqueda que presentan ambos editores de texto, se realizó el conteo donde se ve en la Tabla 4.13 que difieren en los resultados pero no varían significativamente. Esto nos indica que los dos editores y la herramienta manejan distintos algoritmos de búsqueda.

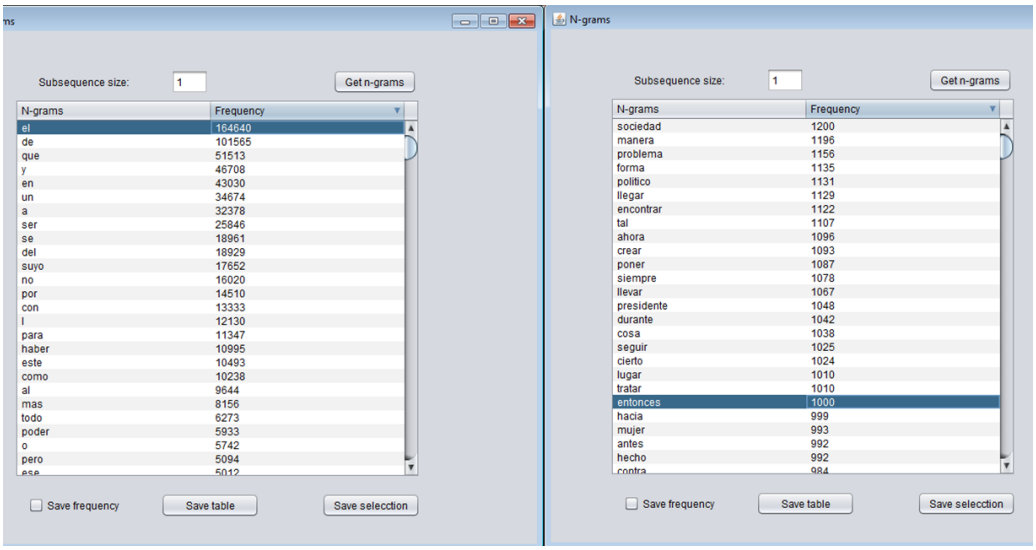

Figura 4.15: Generación de unigramas

Ahora de manera arbitraria generaremos tri-gramas para verificar que no exista diferencia significativa entre los resultados anteriores.

Usando un criterio similar al anterior, tomamos el tri-grama con mayor frecuencia y el que su frecuencia sea 200 como se observa en la figura 4.15. Para corroborar se usarán los editores anteriormente mencionados.

Software / Trigrama	Herramienta desarrollada	Notepad++	EditPad Pro
en el que	967	967	967
de el partido	200	200	200

Cuadro 4.14: Comparativa de frecuencias de palabras en un texto común

Se observa en la Tabla 4.14 que no existió diferencia alguna en esta prueba

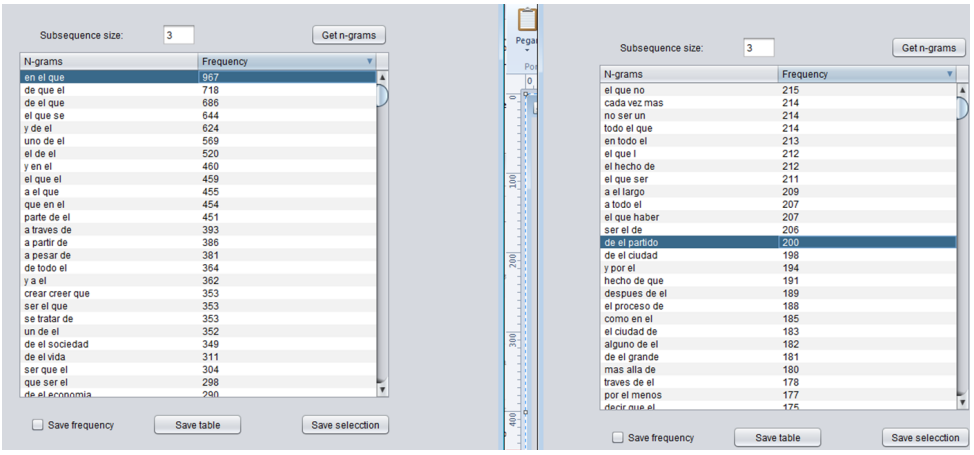

Figura 4.16: Generación de tri-gramas

2. **Generando matriz tipo "arrf" partiendo de un documento como corpus**

   Dato de entrada:

   - Corpus de 8,945 kb de tamaño en disco duro, con un total de 56, 970 renglones.
   - Vocabulario de 191 kb de tamaño en disco duro, con un total de 19,068 palabras. Una palabra por renglón.
   - Términos-clasificados (afecto positivo) de 7 kb de tamaño en disco duro, con un total de 672 palabras. Una palabra por renglón.

   Parámetros de inicio de la máquina virtual de java:

   - java–Xmx1400m –Dfile.encoding=ISO-8859-1 –jar NumberAddition.jar

   Parámetros de generación de matriz:

   - Tamaño de ventana: oración completa.
   - Tipo de matriz: frecuencia.

   La matriz creada con 672 vectores de frecuencias, con los 672 términos-clase que aparecen en el archivo de coincidencias que representan cada vector resultó como se esperaba. El archivo de términos-clase repetidos en una misma oración se obtuvo varios elementos que se dicen repetidos en una misma oración un cierto número de veces. Para comprobar que es funcional la sección que lo genera, tomaremos tres términos-clase repetidos para verificar en el corpus. Tomaremos un término-clase que sea poco común para probar con facilidad que se repite el número de veces que se indica en el archivo. El término-clase "tensar", se obtuvo del corpus 18 veces, por lo que es posible corroborar manualmente que exista en una oración repetida este término-clase como se indica en el archivo de términos-clase repetidos.

   El número 18 se obtuvo con el ambos editores antes mencionados. Y con la opción de "Buscar siguiente" dentro del menú buscar, se recorrió el corpus en busca de más de una coincidencia en una misma oración la cual se encontró como se esperaba. Además fue introducida la matriz "arff" en la herramienta Weka, la cual reconoce la estructura para comenzar a trabajar en varios de sus módulos.

Para este escenario se utilizó el parámetro, matriz de frecuencia porque es del cual se toma base para la matriz booleana y la matriz TF-IDF.

3. **Generando archivos tipo "prep" y archivos "txt" sin el formato "prep" (indexación)**

   Dato de entrada:

   - Corpus de 8,945 kb de tamaño en disco duro, con un total de 56, 970 renglones.
   - Vocabulario de 191 kb de tamaño en disco duro, con un total de 19,068 palabras. Una palabra por renglón.
   - Términos-clasificados (Afecto positivo) de 7 kb de tamaño en disco duro, con un total de 672 palabras. Una palabra por renglón.

   Parámetros de inicio de la máquina virtual de java:

   - java–Xmx1400m –Dfile.encoding=ISO-8859-1 –jar NumberAddition.jar

   Parámetros de generación de matriz:

   - Tamaño de ventana: 4.
   - Tipo de matriz: frecuencia.
   - Inicio de la numeración de los archivos.

Los resultados que se obtuvieron fue de 671 archivos tipo "prep" lo que nos indica que de las 672 palabras, una palabra no tuvo coincidencias con el vocabulario con el tamaño de ventana 4. Esto lo corrobora el archivo de coincidencias usando la pestaña de la herramienta de texto con la operación de diferencia de conjuntos, obtenemos que la palabra "estimarlo" no coincidió fijando el parámetro de tamaño de ventana a 4. Se realizó la búsqueda en el corpus que manejamos y encontramos una única vez la palabra "estimarlo" y se revisaron las 4 palabras que acompañaban a cada lado de este término y no se encontró alguna de estas en el vocabulario.

Las pruebas funcionales se pueden extender a otros módulos, pero se descartan por los motivos que a continuación se describen. Primeramente sería muy extenso y hasta cierto punto confuso detallar las pruebas de todas las secciones, tenemos la conformidad de la persona que realiza la investigación y que funge como usuario final. Los 3 puntos que se probaron sirvieron de referencia en los algoritmos de búsqueda para crear otras secciones del software.

4.4.4. Pruebas de stress

1. **Generación de corpus partiendo de colección de documentos**

   Dato de entrada:

   - Colección de 56,471 archivos de texto, donde se reportan noticias de diversos temas.

   Parámetros de inicio de la máquina virtual de java:

   - java–Xmx1400m –Dfile.encoding=ISO-8859-1 –jar NumberAddition.jar

   Tardó no más de un minuto en la formación del corpus y otro minuto más en darle formato debido a que no tenía el formato esperado. El peso final del corpus fue de 139,468 kb, considerablemente más grande que el corpus con el cual se trabajo a lo largo de la tesis de 8,945 kb. El consumo de memoria finalizando la creación del corpus fue de 850 MB aproximadamente y posterior al formateo del corpus se incrementó a 1,240 MB.

2. **Generando matriz tipo "arrf" partiendo de un documento como corpus**

   Dato de entrada:

   - Corpus de 139,468 kb, de tamaño en disco duro, con un total de 627, 648 renglones.
   - Vocabulario de 409 kb de tamaño en disco duro, con un total de 48,482. Una palabra por renglón.
   - Términos-clasificados (términos seleccionados al azar) de 10.8 kb de tamaño en disco duro, con un total de mil palabras. Una palabra por renglón.

   Parámetros de inicio de la máquina virtual de java:

   - java–Xmx1400m –Dfile.encoding=ISO-8859-1 –jar NumberAddition.jar

   Parámetros de generación de matriz:

   - Tamaño de ventana: oración completa.
   - Tipo de matriz: frecuencia

   La matriz fue creada exitosamente aun cuando la herramienta indicaba trabajar en los límites de memoria principal asignados por los parámetros de inicialización de la máquina virtual de java.

   El archivo tipo "arff" con un tamaño total de 93.8 MB, de dimensión 1000 x 48482, donde cada celda puede ser un valor de cero o mayor a cero.

3. **Generando matriz tipo "arrf" partiendo de una colección de documentos.**

   Dato de entrada:

   - Colección de 56,471 archivos de texto, donde se reportan noticias de diversos temas.
   - Vocabulario de 409 kb de tamaño en disco duro, con un total de 48,482. Una palabra por renglón.

   Parámetros de inicio de la máquina virtual de java:

   - java–Xmx1400m –Dfile.encoding=ISO-8859-1 –jar NumberAddition.jar

   Parámetros de generación de matriz:

   - Tipo de matriz: TF-IDF

La matriz para esta prueba no se logró generar por la falta de memoria para ambos hilos que creaban la matriz, únicamente se guardó y creo la primer parte del archivo "arff", que son el número y tipo de atributos. Se creó el archivo de correspondencia vectores-documento, aún cuando está incompleta debido a que al dejar de generar los vectores, el arreglo de los documentos que tenía en ese momento fue escrito en el archivo. Esto sucedió debido a que la cantidad de archivos es relativamente grande para los parámetros de inicio de la máquina virtual, y también a que la matriz generada es creada en un arreglo de tipo doble precisión, por lo que sencillamente es cuestión de hacer número donde sabemos que un valor de doble precisión es almacenado por 8 bytes. Con esto se nos da la idea de que sencillamente supera un equipo de cómputo con 2 GB o 4 GB de memoria RAM.

4.5. Comparación de rendimiento

A continuación mostraremos una tabla de los rendimientos en varios procesos de la plataforma, estos procesos fueron acelerados por procesamiento en paralelo en las secciones de mayor demanda de procesamiento.

Las características del sistema en el que se ejecutó la herramienta son:

- Sistema operativo Windows 7 Ultimate
- Procesador Intel Core 2 Duo T5750 a 2.00 Ghz.
- Memoria RAM 2 GB
- Sistema operativo de 32 bits

	Archivos de prueba: Corpus de 8,945 kb. Términos-clase de 7kb Vocabulario de 191 kb.		Archivos de prueba: Archivo formato DATA con 177 MB.		Archivos de prueba: Mil documentos de texto con un tamaño total de 5.24 MB. Vocabulario de 412 kb.	
	Prueba 1: Generación de matriz de frecuencias partiendo de un corpus.		Prueba 2: Conversión de archivo "data" a "arff".		Prueba 3: Generación de matriz de frecuencias partiendo de una colección de documentos.	
Primera versión de la herramienta	19 minutos 40 segundos	488 MB	33 segundos	724 MB	27 minutos	302 MB
Segunda versión de la herramienta (Múltiples hilos. Suma de los tiempos de ambos procesadores).	22 minutos 20 segundos	449 MB	35 segundos	643 MB	20 minutos	303 MB

Cuadro 4.15: Comparativa de tiempos de ejecución de tareas de la plataforma

Los tiempos mostrados en la tabla 4.15, a simple vista logra ser evidente que no hay diferencia en los tiempos y ligera diferencia en el uso de memoria para algunos casos. Pero la realidad llega a ser distinta, debido a que la implementación de paralelismo, nos permite usar las capacidades de los procesadores multi-núcleo. Los tiempos de ejecución de la tabla 4.16 fueron obtenidos del administrador de tareas de Windows 7, donde nos permite censar el "tiempo de uso de la CPU" donde el tiempo es relativo al número de núcleos de la CPU. Para el caso de la tabla mencionada, como se menciona previamente, se usa un CPU de doble núcleo. Donde el tiempo realmente de la "Segunda versión de la herramienta" los segundos se contabilizan al doble. Cabe mencionar que no todo el tiempo se ejecuta al doble. Debido a que hay procesos previos y finales que usan solo un núcleo. En otras palabras decimos que los tiempos mostrados en la tabla de la

segunda versión realmente son menores en la vida real por lo antes mencionado.

Para garantizar que el software cumple con las especificaciones originales, se realizaron pruebas individuales de cada sección, y sus módulos, donde se revisaron los resultados con archivos de que se preparaban con resultados específicos evaluados manualmente, que se corroboraban con los resultados que la plataforma en sus diferentes etapas iba entregando, como también se realizaron pruebas con datos reales aun cuando alguna anomalía fuese mas difícil de encontrar por el tamaño de los datos reales. Posteriormente se realizó la integración de las secciones y se lograron reducir variables globales del sistema y mantener la eficiencia de memoria. Se obtuvieron resultados por parte del usuario final, algunas secciones les faltaban de validación de los datos de entrada para que no arrojara resultados inesperados o error de ejecución. Además de que la primer sección de generación de representaciones tenía ausencia de un regla en la generación de las matrices, se debía eliminar aquellas matrices que fueran en su totalidad cero, y obtener como salida paralela a la generación de la matriz de vectores una lista de los términos a clasificar que correspondiera a los vectores generados y aquellos eliminados por la totalidad de ceros, debía ser eliminada también de la lista de términos a clasificar.

En esta etapa se detectó variación de resultados con respecto a las pruebas que se realizaron usando el IDE para desarrollar la plataforma, por lo que se hizo una revisión minuciosa hasta encontrar que la variación consistía en que en modo de depuración y ejecución a través del IDE de desarrollo, se trabajaba con una codificación uft-8 y cuando se realizaba la ejecución desde el archivo final con extensión JAR, se obtenía codificación del sistema operativo y no tomaba en cuenta caracteres extendidos del código ASCII por lo que los resultados variaban. Finalmente se realizó un análisis completo de la importancia de la codificación y resultó definitiva el uso de la codificación ISO-8859-1 y esto se debe a que en términos computacionales es menos carga de procesamiento que el estándar UTF-8, ya que en la codificación en java se realiza la lectura de los corpus o colecciones de datos carácter por carácter ya que se analiza algunos caracteres para tomar ciertas decisiones y se manejo el carácter como un byte, por lo que UTF-8 para los caracteres con tildes o símbolos comienza a utilizar dos bytes en adelante, lo que serían algunas operaciones más, que al final de cuenta son operaciones que se realizan con ficheros o corpus grandes que comienzan a tener un peso significativo en el rendimiento.

Ahora con la solución nos enfrentamos con el problema de ejecución de la plataforma bajo esa codificación, la cual no se podía realizar dentro del código de Java, porque este parámetro de codificación es establecido por la máquina virtual de java al iniciar el programa. Por lo que la solución fue la implementación de un lote de instrucciones que se integró en un archivo ejecutable junto al archivo JAR de la plataforma para que la máquina virtual de java iniciará con los parámetros de codificación ISO-8859-1. Sin embargo es posible pasar los parámetros de inicialización a través de la línea de comandos sin necesidad del uso del fichero .bat que es un script facilitador del procedimiento de pasar los parámetros de inicialización de la máquina virtual con la plataforma de experimentación. Además en el archivo .bat también se puede especificar la cantidad de memoria que se desea asignar a la plataforma porque hay ciertos procesos que requieren más memoria que otros y esto también va relacionado al tamaño de los ficheros. El punto clave para hacer uso intensivo de todo los núcleos del sistema fue mediante la clase *Runtime* de Java la cual a través de la expresión:

```
Int Nprocesadores=Runtime.getRuntime().availableProcessors();
```

Obtenemos en la variable tipo entero “Nprocesadores” el número de procesadores del sistema que posteriormente les asignamos bloques de datos para procesar y se inicializan con las directivas correspondientes a la clase “Thread”.

Mantenimiento

El mantenimiento de la plataforma es crucial para mantenerse en actividad y siga siendo útil para las experimentaciones. Por lo que el código fuente del proyecto fue en su mayoría comentado en los métodos o líneas importantes donde el razonamiento a simple vista no consiga obtener entendimiento. Además de que se crearon clases con nombres de fácil intuición y entendimiento así como los métodos creados para cada clase. De esta manera se logra mayor facilidad para la modificación, implementación de alguna mejora o solución a algún imprevisto que se muestre cuando este en plena operación. Esta abierta la plataforma a posibles extensiones de operaciones como también el trasladarlo a otro sistema operativo.

El enfoque de este capítulo, fue la creación de la plataforma y las etapas que conlleva desarrollar un software, por ser un software para el área de investigación su interfaz no fue un aspecto a resaltar, el enfoque se realizó sobre el correcto funcionamiento que este debía tener y hacer un buen uso de los recursos computacionales. Aún cuando la plataforma esta desarrollado en java, no asegura su total funcionalidad en el sistema operativo Linux, por lo que se realizaran mejoras o modificaciones para una versión en dicho sistema operativo. Como toda primera versión de un sistema, siempre es posible mejorar diferentes aspectos de él, por lo que muchas partes del proyecto se comentaron para un entendimiento a futuras modificaciones.

Capítulo 5

Experimentación y resultados

Los resultados obtenidos con base en las hipótesis que se plantearon al inicio de esta tesis son las que se describen a continuación.

Evaluación de hipótesis 1

La primera hipótesis nos dice lo siguiente:

Hipótesis 1:

"Con la validación y mejora de las colecciones textuales elaboradas de forma manual, realizada bajo ciertos lineamientos que marca la Teoría de la Valoración será posible llevar a cabo la extracción de expresiones valorativas, indicarles un tipo e intensidad, particularmente en textos en español".

Considerando la insuficiencia en recursos textuales para el análisis automatizado de textos en el idioma en español, argumento que nos lleva a la evidencia contenida en la hipótesis uno se define en la presente tesis la validación en textos, oraciones y expresiones valorativas de forma manual y bajo los lineamientos señalados en la Teoría de la valoración, favoreciendo de esta manera el desarrollo de recursos textuales para la Minería de Opinión, partiendo de lo antes expuesto y planteado en el trabajo de investigación, se realizó la extracción de expresiones valorativas y por consiguiente la asignación de un tipo de valoración ya sea de polaridad o actitud, de esta manera es como se desarrolló un nuevo diccionario. Derivado de ello obtuvimos los siguientes resultados:

Para el caso de la **creación del léxico de palabras valorativas,** con el registro que se tenía en los ficheros H1 y H2, basado en ellos y el criterio personal de alguien se obtuvo el fichero H3 con un resultado de 3035 palabras individuales, de las cuales 143 fueron eliminadas por aparecer repetidas o carecer de valoración. En las 2892 palabras restantes, se encontró variación al comparar la asignación de los valores de pertenecía entre las listas H1 y H2, por lo que se hizo una reasignación de dichos valores considerando como se mencionó anteriormente el criterio de una tercera persona, obteniendo de esta manera un fichero integral.

Del **corpus de oraciones valorativas,** la principal utilidad en la clasificación elaborada fue discriminar oraciones no relevantes e integrar lo de mayor utilidad a procedimientos posteriores, de ello, se obtuvo de 2099 oraciones totales, 370 oraciones de internet, 305 de libros y ficticias con búsqueda en internet de 1424, 42 consideradas como Resueltas, 130 Opcionales y 1252 de Error. A partir del resultado obtenido se muestra la siguiente figura.

Figura 5.1: Resultados obtenidos de la clasificación de oraciones valorativas

En el proceso de **clasificación de expresiones** valorativas las cuales se obtuvieran a partir de las oraciones seleccionadas, de ahí que se asignara valores según correspondiese para obtener un conjunto de expresiones clasificadas, así fue como se obtuvo de 423 oraciones en sus distintas fuentes, 572 expresiones valorativas que fueron extraídas de estas oraciones. Se exponen tablas de resultados obtenidos en las intensidades de Polaridad así como los valores asignados al Afecto, Juicio y Apreciación.

	Polaridad	
	Positiva	Negativa
Intensidad 3	14	30
Intensidad 2	32	47
Intensidad 1	165	287
Total	**211**	**364**

	Afecto	Juicio	Apreciación
Valor 2	111	234	215
Valor 1	2	2	4
Total	**113**	**236**	**219**

Cuadro 5.1: Resultados obtenidos en la clasificación de expresiones valorativas

A partir de los resultados, se obtuvo un diccionario de datos propio de la mejora realizada en colecciones textuales, cabe indicar que dicho léxico elaborado se aplicó de manera particular para la experimentación en la plataforma de pre- procesamiento y en el trabajo de investigación de tesis doctoral, por lo tanto se acepta la hipótesis como verdadera.

Evaluación de hipótesis 2

La segunda hipótesis nos dice lo siguiente:

Hipótesis 2:

"Con el desarrollo de una plataforma de software será posible lograr con mayor eficacia y eficiencia la generación de archivos a través del pre-procesamiento y evaluación automático de textos de opinión".

La eficiencia se comprueba al comparar la herramienta desarrollada por el investigador para

ayudar a realizar las manipulaciones o generaciones a partir de los corpus de textos.

La siguiente tabla comparativa se realizó bajo las siguientes circunstancias:

Hardware de equipo:

Procesador: AMD A4-3305M APU with Radeon HD Graphics 1.90Ghz
Memoria RAM: 4 GB.
Sistema: Windows 7 Home Basic a 64 bits.

Características de Archivos utilizados:

Corpus para prueba 1 y 2: Archivo de texto plano con 56,790 renglones y un tamaño de 8.73 MB

Corpus para prueba 3: Archivo de texto plano con 10,000 renglones y un tamaño de 1.54 MB
Vocabulario para prueba 3: Con un total de 19,068 palabras y un tamaño de 191 KB.

Términos clasificados para prueba 3: Con un total de 10 palabras y un tamaño de 1 KB.

	Herramienta desarrollada por investigador		Plataforma de Pre-procesamiento de texto	
	Tiempo en cargar corpus.	Uso de memoria principal	Tiempo en cargar corpus.	Uso de memoria principal
Prueba 1.-Cargar archivos de texto para formateo.	2 minutos 6 segundos	140 MB	2 segundos	170 MB
Prueba 2.-Generación de unigramas	4 minutos 15 segundos	180 MB	3 segundos	250 MB
Prueba 3.-Matriz Binaria	1 minuto 30 segundos	75 MB	3 segundos	130 MB

Cuadro 5.2: Comparativa de eficiencia en tiempo de tareas en común

Las nuevas tareas implementadas son:

- Formateado del corpus
  - Formateo de corpus automático
  - Unión de varios documentos tipo “txt” y “TTG”
- Representación de modelos
  - Generación de n-gramas
- Operación de conjuntos
  - Unión
  - Substracción
  - Intersección

- Obtención de datos Estadísticos
  - Frecuencia con polaridad inversa y sin ella.
  - Frecuencia de elementos etiquetados.
- Evaluación de los resultados de los algoritmos de aprendizaje automático
  - Selección de un rango de umbrales para asignar intensidades
  - Generar matriz de confusión con medidas de precisión, recuerdo y media armónica de las dos anteriores.

La eficacia se obtiene al haber programado correctamente los algoritmos y procesos que fueron verificados en la etapa de pruebas de software donde se hicieron comparaciones tanto con otras aplicaciones como con procedimientos manuales que antes se realizaban para ciertas tareas.A continuación se evaluarán los resultados con versiones parciales de los archivos con los que se trabajaron.

Primero se formateo el corpus con la herramienta, y posteriormente se crearon los términos únicos como se observa en la figura 5.2.

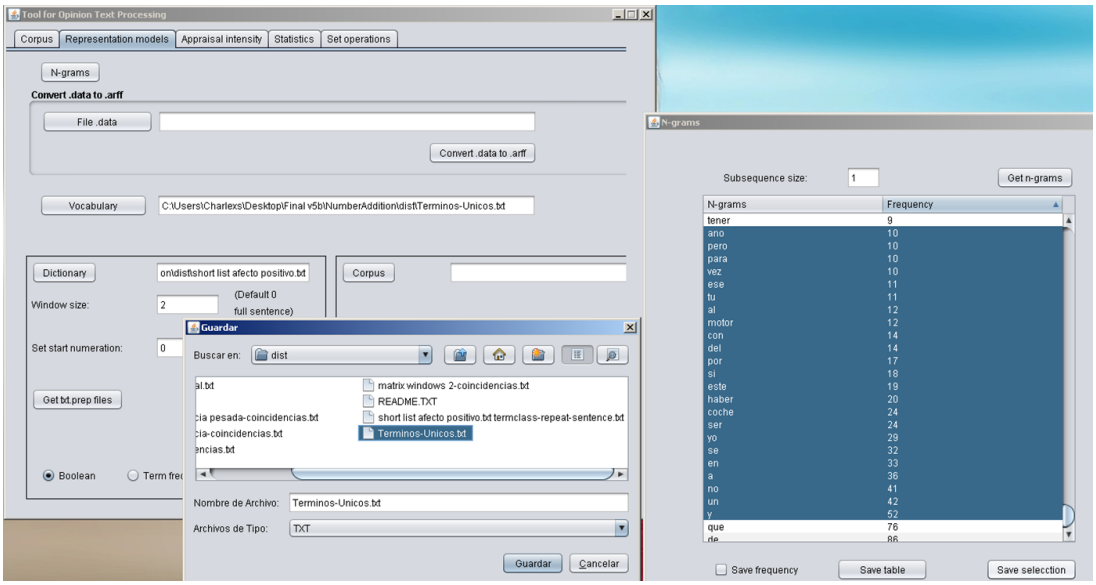

Figura 5.2: Generación de uní-gramas

Por razones demostrativas, tomamos el rango de frecuencias de palabras de 52 a 10 apariciones en el “Documento A” ver Anexo. Tenemos que la lista siguiente son los veinte términos únicos.

1.- vez	11.- haber
2.- ese	12.- coche
3.- tu	13.- ser
4.- al	14.- yo
5.- motor	15.- se
6.- con	16.- en
7.- del	17.- a
8.- por	18.- no
9.- si	19.- un
10.- este	20.- y

Cuadro 5.3: Vocabulario de prueba para verificación de eficacia

A continuación se presentan una lista parcial de términos clasificados como afecto positivo:

1.- abochornado	12.- entretenido
2.- abominar	13.- resentir
3.- cicatriz	14.- resignación
4.- clamar	15.- resignar
5.- colapso	16.- respiro
6.- desenfadado	17.- resplandeciente
7.- deseo	18.- restringir
8.- entranable	19.- vencer
9.- entrega	20.- venerar
10.- entregar	21.- vida
11.- entregarse	22.- vigilar

Cuadro 5.4: Lista parcial de términos.clase de afecto postivo

Teniendo en cuenta estas dos listas, se genera una matriz de frecuencias de acuerdo a las coincidencias de las listas anteriores, tal como se explicó en capítulos anteriores.

De la generación de la matriz, obtuvimos solo un vector que se muestra a continuación:

1,1,0,0,1,1,0,4,0,1,0,1,0,0,3,2,0,0,0,2,-1

Se observa solo un vector, tal vez se hubiese esperado más por el tamaño del "Documento A", pero al final cada vector representa un término clasificado, por lo que yendo al archivo generado de coincidencias cuando se generó la matriz, tenemos que el término "vida" fue la única coincidencia. Teniendo en cuenta esto, podemos observar que en la oración coincidieron 10 términos únicos, de los cuales solo uno de ellos se repitió 4 veces como el de mayor frecuencia en esa oración. De acuerdo al orden que se obtiene en el vector el valor número cuatro está en la octava posición el cual podemos vincular con el archivo de términos únicos que en esta demostración es el término único "por" ocupado la posición 8 en la lista del archivo. Esto es fácilmente comprobable yendo al "Documento A" buscando la oración donde se encuentra el término clasificado

“vida” y verificar que existan 4 términos únicos “por”. Teniendo en cuenta que la base para generar las matrices binarias y de matrices TF-IDF es la generación de matriz de frecuencia, con esto tenemos garantizada que la mayor parte del proceso para generar los valores de las matrices es correcto. En la sección en la que se trabaja con colección de documentos sigue siendo el mismo caso, por lo que al final el algoritmo esencial de la representación de matrices vectoriales es la generación de matrices de frecuencia de un solo documento.

El vector de la matriz de frecuencia con un tamaño de ventana dos, se mostrara a continuación con los mismos archivos de entrada:

0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,-1

Como se observa se reduce a una coincidencia de las muchas que se mostraban en el vector anterior sin límite en el tamaño de ventana.

Ahora se modificará la lista de términos clasificados por la siguiente:

1.- ser	12.- entretenido
2.- coche	13.- resentir
3.- cicatriz	14.- resignación
4.- clamar	15.- resignar
5.- colapso	16.- respiro
6.- desenfadado	17.- resplandeciente
7.- deseo	18.- restringir
8.- entranable	19.- vencer
9.- entrega	20.- venerar
10.- entregar	21.- vida
11.- entregarse	22.- vigilar

Cuadro 5.5: Términos-clase de la modificación en la tabla 5.4

El cambio realizado en la lista de términos clasificados, fueron los primeros dos términos, los cuales se consideraron por la frecuencia de aparición en el texto, por lo que se tiene que ver al menos dos vectores mas en la generación de matriz con frecuencias más altas de las que anteriormente se muestran.

3,5,5,5,8,8,12,9,9,13,14,15,24,14,20,16,20,25,22,28,-1
4,6,7,6,7,9,8,13,11,17,11,24,15,9,17,17,18,21,17,28,-1
1,1,0,0,1,1,0,4,0,1,0,1,0,0,3,2,0,0,0,2,-1

Se observa que ahora como se esperaba se obtuvieron tres vectores, los cuales los dos primeros son de los nuevos términos clasificados que se cambiaron y como se predijo sabiendo la alta frecuencia compara con la del término clasificado “vida” se ven incrementado los valores de incidencia con los términos únicos.

Ahora siguiendo con el “Documento A”, se demostrará el funcionamiento con más de un grama o palabra. Se realizaraá con bi-gramas para esta demostración, comenzamos generando los términos únicos en bi-gramas. Como se observa en la figura figura 5.3.

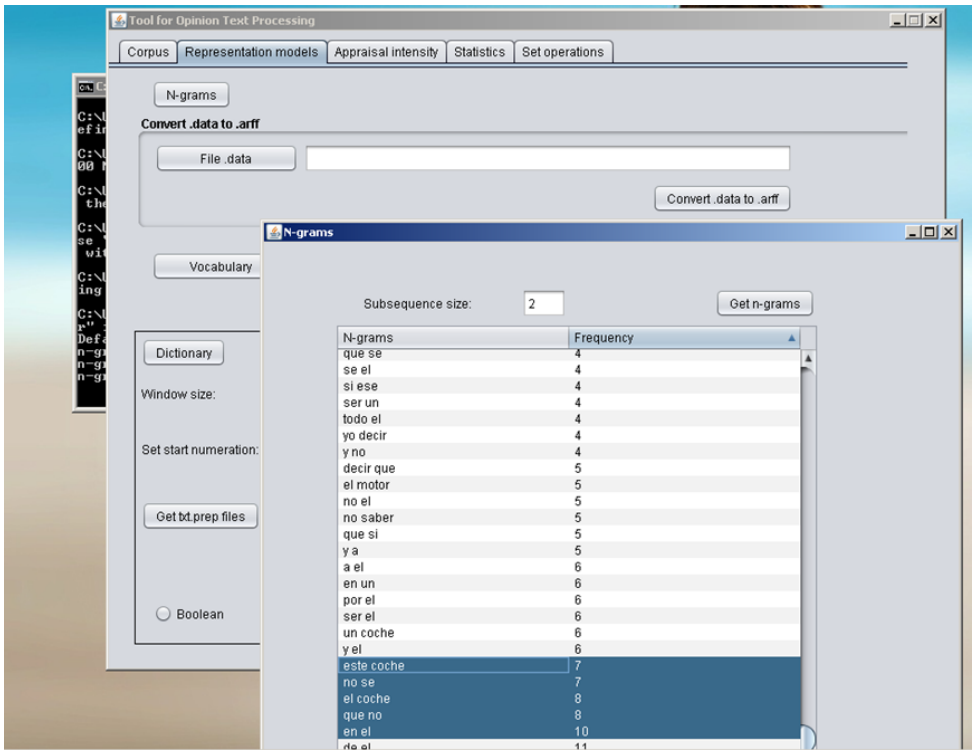

Figura 5.3: Generación de uní-gramas

Ahora partiendo de la lista de términos únicos como se muestra en la siguiente tabla:

1.- este coche	4.- que no
2.- no se	5.- en el
3.- el coche	

Generaremos la matriz de frecuencia sin límite de tamaño de ventana, dándonos como resultado el siguiente vector:

5,6,4,6,6,-1
7,6,8,5,7,-1
1,0,0,0,1,-1

Como era de esperarse tenemos los tres vectores, resultado de las coincidencias con “ser”, ”coche”, ”vida”. Se observa una reducción de también en el tamaño del vector debido a que ahora contamos con cinco términos únicos. De los cuales se logran aún ver coincidencias y con escasas coincidencias para el término clasificado “vida”, donde las únicas 2 coincidencias con frecuencia de uno son los términos únicos ”este coche” y “en el”.

Con lo anterior, se puede observar que la eficacia o certeza del algoritmo de la herramienta es la esperada, lo anterior es aán práctico para poder ser analizado, sin embargo la herramienta fue diseñada para archivos de texto en el orden de cientos de megabytes, o de varios documentos de texto.

En la eficiencia se habla del ahorro de tiempo que las nuevas tareas que anteriormente se realizaban de forma manual como son la unión de varios conjuntos de archivos de texto plano o en formato “TTG”, como también al darle formato a los corpus quitando símbolos y espacios, en la evaluación de los resultados se realiza de forma más rápida y se obtienen las medidas y la

matriz de confusión de forma automática.

Gracias a la implementación de paralelizar ciertos procesamientos de datos, los cuales consumían la mayor parte del tiempo en ciertos procesos que a lo largo de este documento se mencionó. Se logró reducir considerablemente los tiempos y que son más apreciables en grandes tamaños de datos. Sin embargo cierta parte del código no se trabaja en paralelo, por lo que no se puede hablar de disminuir tiempos totales, solamente parte de este tiempo es optimizable al incrementar el número de procesadores, como bien lo menciona la ley de Amdahl [20]. Por lo mencionado anteriormente se puede decir que la segunda hipótesis es aceptada.

Capítulo 6

Conclusiones y recomendaciones

Conclusiones:

En el trayecto de esta investigación y desarrollo de una herramienta para el área de procesamiento de lenguaje natural, se reforzaron y adquirieron conocimientos nuevos. Es importante hacer notar que es necesario obtener los requerimientos del cliente de manera clara, que en este caso fueron dos especialistas del áreas de Minería de opiniones del Laboratorio de Tecnologías del Lenguaje del INAOE y mantener una estrecha comunicación para evitar errores en las entregas parciales de los módulos.

Se observó que los editores de texto manejados, difieren en algoritmos de búsqueda, además de que muchos editores previos a los mencionados, no lograban manejar cantidades de memoria de cientos de megabytes en texto plano.

Hay muchas aplicaciones por realizar en áreas de investigación, que faciliten a los investigadores realizar sus tareas. Algunas herramientas permiten ahorrar el recurso del tiempo, que es lo que se pretendió con la herramienta desarrollada, eficiencia en tiempo y por supuesto eficacia con la información entregada.

Recomendaciones:

Es posible mejorar la robustez en el módulo de generación de matriz, tanto para optimizar las operaciones como para hacer más eficiente el uso de memoria principal. Esto aplica para la matriz de tipo booleana y la matriz TF-IDF, donde una puede ser implementada como valores booleanos y la otra con una técnica diferente de manejo de memoria para evitar usar matrices de grandes tamaños del tipo doble precisión, el cual requiere de más memoria que otros tipos de datos.

Sería de utilidad implementar el cálculo automático de los recursos que requieren la información a pre-procesar en cuanto a tiempo, memoria principal y de cierta forma re-ajustar los valores de memoria principal de inicio de la máquina virtual para evitar pérdida de tiempo en pre-procesamientos que no se concluyan por falta de memoria.

Queda la posibilidad de realizar alguna implementación para la verificación del contexto de las oraciones donde se tomen en cuenta algunos signos de expresión o puntuación y así incrementar nuevas variables a tomar en cuenta para mejorar los resultados de clasificación de términos.

Fuentes Bibliohemerográficas

[1] Hernández, L. Extracción de expresiones valorativas, su tipo e intensidad a partir de textos en español. Documento de propuesta de investigación doctoral. Coordinación de ciencias computacionales, INAOE (2009).

[2] R.R White Peter, Translated by Ghio Elsa. Un recorrido por la Teoría de la Valoración, English Language Research, Department de English. http://www.grammatics.com/appraisal/SpanishTranslation-AppraisalOutline.pdf

[3] Cunningham H., Maynard D., Bontcheva K., Tablan V. GATE: an architecture for development of robust HLT applications Proceedings of the 40th Annual Meeting on Association for Computational Linguistics. Paginas 168-175.

[4] O'Donnell, M. 2008. "The UAM CorpusTool: Software for corpus annotation and Exploration "XXVI Congreso de AESLA", Almeria, Spain, 3-5 April 2008.

[5] Turney, P.D. Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. Procs.of the 40th Annual Meeting of the Association for Computational Linguistics,(2002).pp. 417—424

[6] Kennedy, A. and Inkpen, D. Sentiment Classification of Movie Reviews Using Contextual Valence Shifters. Computational Intelligence, (2006).vol. 22, no. 2, pp. 110–125.

[7] Whitelaw, C., Navendu, G. and Argamon, S. Using Appraisal Groups for Sentiment Analysis. Procs.of the 14th ACM International Conference on Information and Knowledge Management,(2005). pp. 625–631

[8] Matthiessen, C. Lexico-grammatical cartography: English systems. International Language Sciences Publishers.(1995)

[9] Martin, J. R. and White, P.R.R. The Language of Evaluation: Appraisal in English. Palgrave, London. (2005)

[10] Brooke, J., Tofiloski, M. and Taboada, M. Cross-Linguistic Sentiment Analysis: From English to Spanish. In Procs.of RANLP 2009, Recent Advances in Natural Language Processing, (2009). pp. 50-54.

[11] Francisco V., Herva's R., Peinado F., and Gerva's P. EmoTales: creating a corpus of folk tales with emotional annotations. In Lang Resources and Evaluation.(2011).

[12] Planet, S., Iriondo I., Martínez E., Montero J. A. (2008). Automatic refinement of an expressive speech corpus assembling subjective perception and automatic classification (in press). Speech Communication. Elsevier. Volumen 51, Issue 9, Septiembre 2009, Paginas 744–758.

[13] M. Margaret, J. Lang Bradley and Peter, Measuring Emotion: The Self- Assessment Manikin and the semantic differential. University of Florida (1994)
http://www.cnbc.pt/jpmatos/29.%20Bradley.pdf

[14] Wanton, T. Martín. Método para la determinación de la polaridad de las opiniones. Tesis de Maestría. Universidad del Oriente. Cuba, Santiago de Cuba(2009).

[15] W. Parrott. Emotions in Social Psychology, Psychology Press, Philadelphia, 2001

[16] Editores: Indurkhya N. y Damerau F.J.),Opinion Mining, Sentiment Analysis, and Opinion Spam Detection, Manual de Procesamiento Del Lenguaje Natural, Handbook of Natural Language Processing, Segunda Edición 2010. Universidad de Illinois en Chicago.

[17] Kaplan Nora . Nuevos desarrollos en el estudio de la evaluación en el lenguaje: La Teoría de la Valoración. Boletín de lingüística, julio-diciembre 2004.
http://redalyc.uaemex.mx/pdf/347/34702203.pdf

[18] Montes y Gómez, M. Minería de texto empleando la semejanza entre estructuras semánticas. Tesis doctoral. Instituto Politécnico Nacional. México D.F (2002).

[19] Helmut Schmid: Probabilistic part-of-speech tagging using decision trees. Proc. of International Conference on New Methods in Language Processing, pp.44-49 (1994)

[20] Amdahl, Gene. "Validity of the Single Processor Approach to Achieving
Large-Scale Computing Capabilities"(PDF). AFIPS Conference Proceedings (30): 483–485 (1967).

[21] Schmid. TreeTagger desarrollado por el Instituto de Linguistica Computacional de la Universidad de Stuttgart(1994a).

Anexos

Tool for Opinion Text Processing

Herramienta para el Procesamiento de Textos de Opinión

(Manual del Usuario)

INAOE

Febrero 16, 2012

Índice general

Capítulo 1

Herramienta para el Procesamiento de Textos de Opinión ToOp

En este trabajo se presenta el Manual de Usuario para la Herramienta para el Procesamiento de Textos de Opinión, ToOp. El desarrollo de esta herramienta está impulsado por el Laboratorio de Tecnologías del Lenguaje de la Coordinación de Ciencias Computacionales del Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE).Su finalidad es apoyar y ampliar el campo de la Minería de Opiniones (Opinion Mining), con el procesamiento manual y automático de textos. De ahí que su propósito fuese la elaboración de un una herramienta que permitiera procesar textos, brindando información de utilidad para la extracción de expresiones valorativas en textos de opinión. Esta herramienta pretende optimizar el recurso del tiempo en las experimentaciones para los investigadores de este campo.

Capítulo 2

Requisitos

A continuación se exponen los las condiciones requeridas para el manejo de la herramienta.

2.1. Requisitos de operatividad por parte del usuario

Para el manejo de la herramienta debe contar con conocimiento de exploración de carpetas, y manejo de editores de texto.

Esta herramienta fue desarrollada por el Laboratorio de Tecnologías del Lenguaje del Instituto Nacional de Astrofísica, Óptica y Electrónica.

2.2. Requisitos de Software

- JRE (Java RuntimeEnvironment) 1.6.0_ 26 o versión superior.
- Windows Vista, Windows 7

2.3. Requisitos de Hardware Mínimos

- 1 Gigabyte de memoria RAM (se recomiendan 4 Gigabytes)
- Procesador AMD o Intel a 2.0 Ghz (se recomiendan múltiples núcleos)
- 10 Megabytes en disco duro.

Capítulo 3

Ejecución de la Herramienta

Para abrir la herramienta se ejecuta el archivo “herramienta Java.bat” el cual es el archivo de configuración que contiene tres parámetros de inicialización de la Máquina Virtual de Java, tamaño máximo de memoria RAM a utilizar, tipo de codificación que usará la herramienta y el archivo JAR de la herramienta. El contenido del archivo tipo “bat” es el siguiente:

REM The parameter -Xmx define the max size of heap that the platform can use of RAM memory.

REM We need more than 1400 Megabytes when we worked with large data incoming. REM the character ”mïn the end of Xmx1400m”define the previous number in Megabytes.

REM The file encoding; use ÏSO-8859-1ïn the expression Dfile.encoding=ISO- 8859-1”, this is for work with latin characters in the REM GUI. For other encoding only we change ÏSO-8859-1”for other encoding that the system allows.

REM After of commandjarïs the jar file to will be execute.

@echo off

java -Xmx1400m -Dfile.encoding=ISO-8859-1 -jar NumberAddition.jar

Los parámetros son para inicializar la máquina virtual de java con una capacidad máxima de memoria de 1400 Megabytes pudiendo ser modificado cuando la cantidad de datos a procesar superen la memoria especificada, la cual afectará el rendimiento de la herramienta.

El ejecutar el archivo JAR sin el archivo de configuraciones no garantiza el mejor rendimiento de la herramienta por lo que las primeras tareas se realizarán más rápido que otras posteriores.

3.1. Pestaña Corpus

Al ejecutar la herramienta se presenta la ventana mostrada en la figura 3.1 con la pestaña Corpus.

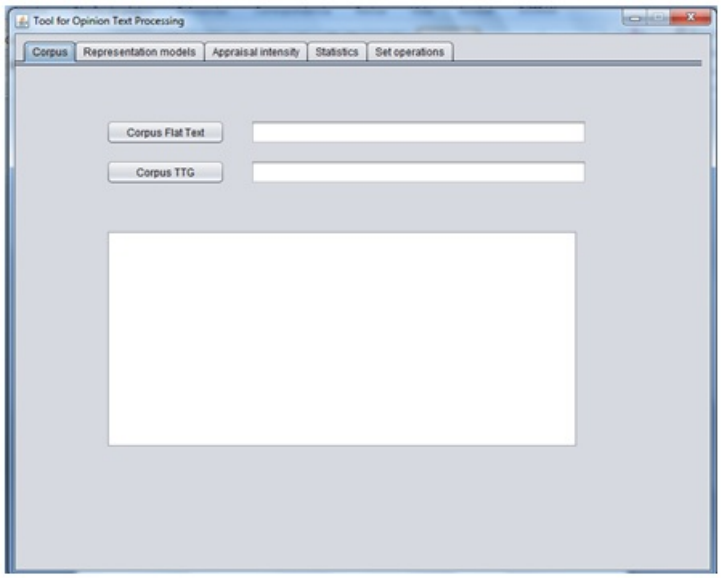

Figura 3.1: Ventana de inicio de la herramienta y vista de la pestaña

El usuario en esta ventana tiene varias funciones que se relacionan a continuación.

3.1.1. Botón "Corpus flat text"

- Ingresar un directorio o directorios de documentos con extensión "txt", los cuales se unirán para formar un solo corpus.
- Ingresar un documento con extensión "txt" que será tomado como el corpus.

De lo anterior por cada corpus cargado la herramienta examina que no haya anomalías en el corpus, ya sea este de un solo documento o de varios. Anomalías como letras en mayúsculas, símbolos como $, \%, &,(,),=,\,+, etc.[1] Si existieran, entonces la herramienta despliega una ventana de advertencia indicando incompatibilidad de formato dando la posibilidad de formatear el corpus, omitir el formato, o cancelar la operación. Esto se muestra en la figura 3.2.

El recuadro blanco del centro de la ventana desplegará la ubicación de todos los archivos creados exitósamente, así como los archivos creados o posibles archivos vacíos. De los archivos creados se resultará el "CORPUS" creado como se observa en la figura 3.2

Indicando que ese archivo es el corpus cuando se debe a la unión de varios documentos. De lo contrario únicamente indicará que el archivo se ha cargado exitósamente.

[1]Los signos de admiración y de interrogación son admitidos.

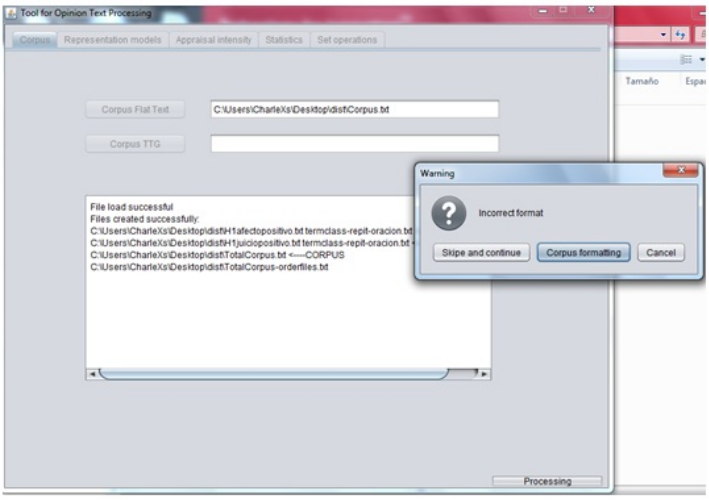

Figura 3.2: Despliegue de información al cargar y procesar un Corpus

3.1.2. Botón "Corpus TTG"

Esta pestaña es para el manejo de archivos con extensión "ttg" que es unas salida del Tree Tagger [21].

- Ingresar un directorio o directorios de documentos con extensión "ttg", los cuales se unirán para formar un solo corpus con el botón "Corpus TTG".
- Ingresar un documento con extensión "ttg" que será tomado como el corpus.

Un archivo "ttg" es un archivo creado con la estructura siguiente:

1. $< NO\_2\_6 >$
2. hola NP hola
3. voy VLfin ir
4. a PREP a
5. hablaros VLinf hablar
6. de PREP de
7. un ART un
8. teléfono NC teléfono
9. $< * >$
10. por PREP por
11. lo ART el
12. que CQUEque
13. a PREP a
14. mi PPO mío

15. respecta VLfin respectar
16. me PPX yo
17. lo PPO él
18. compre VLfin comprar
19. negro NC negro
20. < . >

En la primer línea se observa el nombre del archivo que debe ser “no_2_6.ttg”, posteriormente de la línea 2 a la 8 se ve lo que conforma una oración con sus respectivas etiquetas gramaticales en la columna del medio y la columna de la izquierda el término lematizado, en la línea 9 ésta oración indica el punto final y posteriormente en la línea 11 inicia otra oración y finaliza en la línea 20. Los términos “< . >” y “< ∗ >” son tomados como puntos finales por la herramienta.

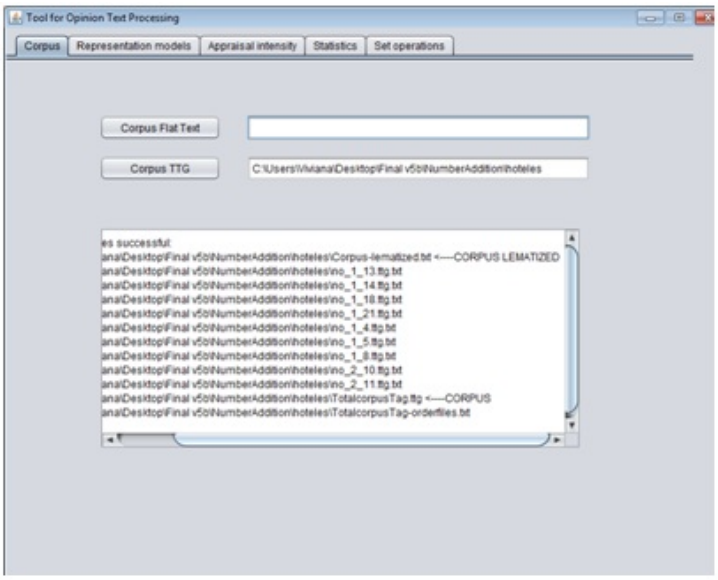

Figura 3.3: Despliegue de información al cargar y procesar documentos “ttg”

Similar a la parte de “Corpus flat text”, con la función “Corpus TTG”, desplegará avisos en el recuadro del centro (ver figura 3.3). Cuando se carga un documento “ttg” informara que un corpus se ha tematizado exitosamente e indicara la ubicación del archivo que lo contiene con una pequeña flecha como en el caso anterior para el “CORPUS” donde su nombre siempre será “TotalcorpusTag.ttg” y con otra flecha para el “CORPUS LEMATIZADO” donde su nombre es Corpus-lematized y el archivo que contiene los nombres de los archivos en el orden que fueron tomados se llama “TotalcorpusTag-orderfiles.txt”.

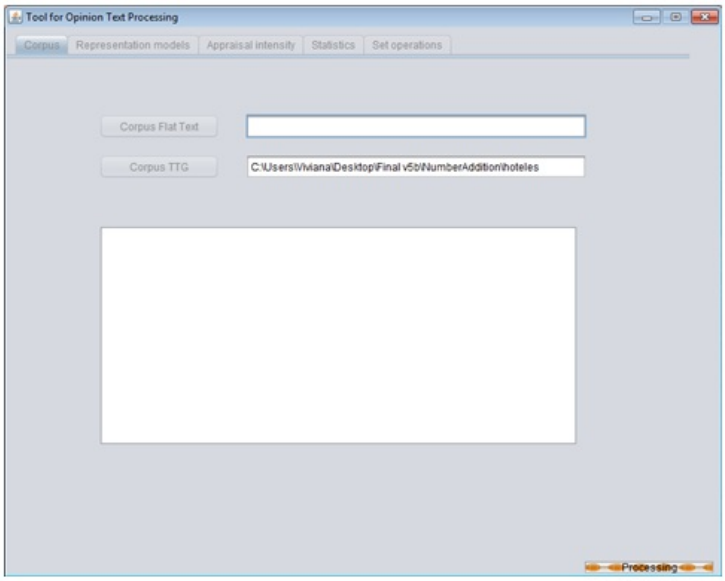

Figura 3.4: Procesamiento de un conjunto de documentos “ttg”

Cuando se realizan uniones de cantidades considerables de archivos para la generación de un corpus, lematización de un documento “ttg” o dar formato a un corpus, dependiendo de los recursos en hardware de la computadora tardará y mientras se realizan estos procesamientos y otros posteriores que se mostrarán a lo largo del manual, se habilita del lado inferior derecho de la ventana de la herramienta un pequeño rectángulo con una etiqueta que indica que se está procesando y que dentro del rectángulo se observa una animación que se estará moviendo mientras se realice algún proceso en la herramienta (ver figura 3.4). Esta barra de procesamiento como se le llamará posteriormente, indicará que la herramienta está trabajando en una tarea específica de lo contrario si la herramienta está congelada y no se observa la animación dará indicación de que la herramienta se colapso en alguna tarea.

3.2. Pestaña Representación de Modelos

En la pestaña de representación de modelos se mostrará una variedad de funciones como se observa en la figura 3.5.

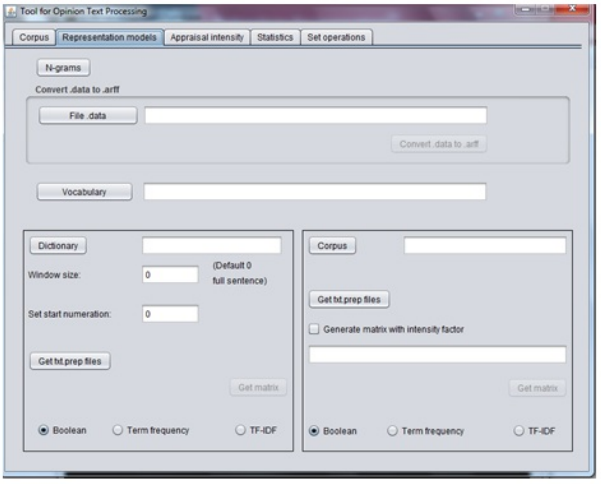

Figura 3.5: Pestaña de representación de modelos.

3.2.1. Generación de n-gramas

Requisitos de otra pestaña:

- Cargar un corpus

Estando en la pestaña actual se da clic en el botón "N-grams", esto nos mostrará una nueva ventana como se observa en la figura 3.6.

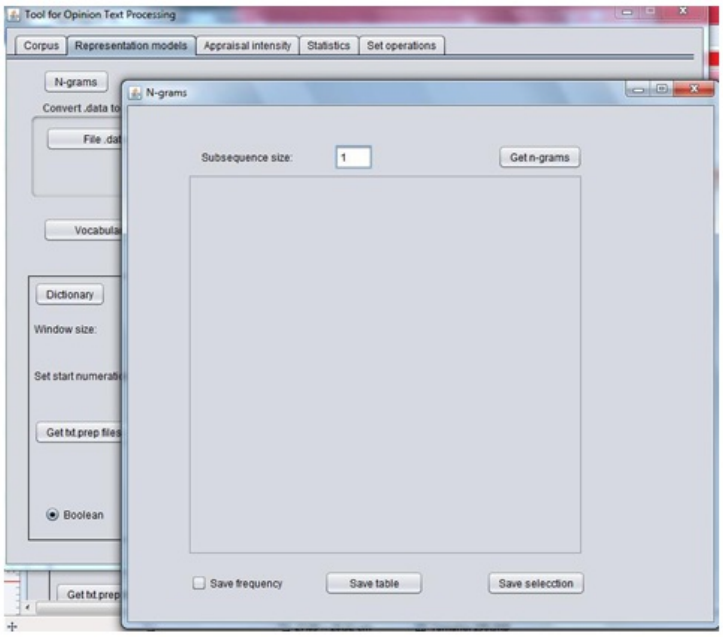

Figura 3.6: Generacion de n-gramas

En el campo "Subsequence size" se puede definir el tamaño de los n-gramas a formar para el corpus cargado. Al oprimir el botón "Get n-grams" se obtiene una lista de n-gramas con su respectiva frecuencia en el corpus. Como ejemplo se muestra en la figura 3.7, los uni-gramas de un corpus cargado para la demostración visual de cómo se conforma esta lista. Esta lista se puede ordenar por orden alfabético o por el tamaño de frecuencias.

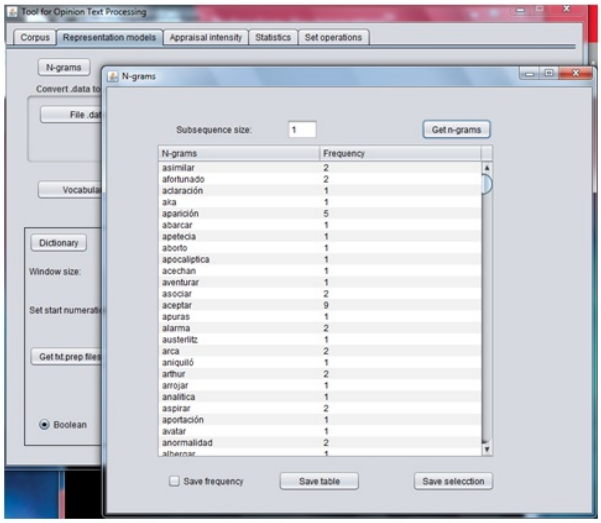

Figura 3.7: Despliegue de unigramas y sus frecuencias

Existen dos formas de guardar información de la lista.

- Seleccionando de segmento o múltiples segmentos de la lista; se logra usando el ratón o mediante el teclado con apoyo del botón "ctrl". Teniendo la selección uno debe seleccionar en la casilla de verificación "Savefrequency" para que el o los segmentos sean guardados con su frecuencia correspondiente o de lo contrario dejar vacía la casilla para guardar únicamente los n-gramas.
- Guardar la tabla completa, con o sin frecuencias como se comenta en el punto anterior. Guardar la tabla almacena la lista tal y como esta ordenada, por lo que se puede ordenar por alfabeto o por frecuencias para que se almacene con el orden deseado.

En cualquiera de estas dos formas de guardar la lista de n-gramas se almacenará en un archivo tipo "txt".

El orden de una tabla en un archivo "txt" guardado es en el del momento previo al almacenaje, por lo que si se utilizó ordenamiento por valores o términos estos nuevos ordenamientos son transferidos al guardarse como archivo "txt".

3.2.2. Convertir de .data a .arff

En esta sección se utilizará un archivo con extensión "data", este archivo tiene la estructura de un objeto de la clase HashMap en java, donde es un mapeo de llaves con sus respectivos valores. En este objeto las llaves será el nombre de un clase o categoría con una numeración que indica el elemento de esta categoría. Como ejemplo:

key="positivoneg02400" Esto indica que existe una categoría o clase "positivoneg" y que el elemento es el elemento 2400, el vector que representa este elemento está contenido en el campo de valor, como un arreglo de valores tipo flotante.

Por lo que el valor del campo "value" puede ser visto como value=Float[ ] dentro de un entorno de programación.

1. Con la estructura que se menciona anteriormente, el archivo tipo "data" se carga mediante el botón "File .data" que desplegará una ventana de exploración de archivos para su selección.

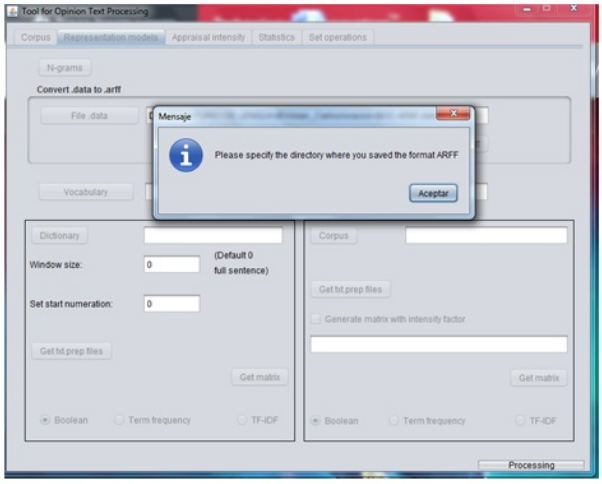

Figura 3.8: Conversión de .data a .arff

2. Una vez cargado se habilitará el botón "Convert .data to .arff que se observa deshabilitado en un inicio, ahora se procede a hacer clic sobre este botón, el cual desplegara inmediatamente un mensaje como se muestra en la figura 3.8, indicando que debe especificar donde desea guardar los archivos con formato tipo "arff". Posterior a la indicación del directorio la barra de procesamiento estará indicando que la herramienta trabaja en la conversión y al finalizar se detendrá y se desplegara una ventana indicando que se realizó una conversión exitosa y los archivos fueron almacenados.

   Los archivos tipo "arff" generados están bajo las especificaciones del software de minería de datos Weka.

   El encabezado de un archivo "arff" contiene el nombre de la relación, una lista de atributos y su tipo. Un ejemplo de encabezado de un conjunto de datos llamados IRIS es el siguiente:

   @RELATION iris
   @ATTRIBUTE atrib1 NUMERIC
   @ATTRIBUTE atrib2 NUMERIC
   @ATTRIBUTE atrib3 NUMERIC
   @ATTRIBUTE atrib4 NUMERIC
   @ATTRIBUTE clase {-1,-2}

   Los datos del archive "arff" sería como el siguiente:

   @DATA
   5.1,3.5,1.4,0.2,-1
   4.9,3.0,1.4,0.2,-1
   4.7,3.2,1.3,0.2,-2
   4.6,3.1,1.5,0.2,-1
   5.0,3.6,1.4,0.2,-1
   5.4,3.9,1.7,0.4,-1
   4.6,3.4,1.4,0.3,-1
   5.0,3.4,1.5,0.2,-2
   4.4,2.9,1.4,0.2,-1

El nombre de la relación que para ejemplo fue iris, la herramienta obtiene el nombre del campo "key" del objeto HashMap, pero elimina el número del elemento obteniendo el nombre de la clase. Retomando el ejemplo anterior donde key="positivoneg02400"se tendrá un archivo tipo "arff" llamado vectorspositivoneg.arff como es de notarse, se antepone el término "vectors" para nombrar el archivo "arff" pero para nombrar la relación dentro del formato "arff" solo se utiliza el nombre de la clase, que para este caso es "positivoneg". Por lo que la primera línea del archivo "arff" se vería así:

@relationpositivoneg
@attribute atrib1 numeric
...
...
..
.

3.2.3. Generación de Documentos Tipo "prep" para un Corpus

Requisitos de otra pestaña:

- Cargar un corpus

Se procede a seguir los siguientes pasos:

1. En la pestaña actual, se usará el botón "Vocabulary", donde mediante una ventana de exploración se selecciona un archivo tipo "txt" donde se encontrará el vocabulario.

2. Con el botón "Dictonary" se realiza la selección de un archivo tipo "txt" donde se enlista términos de alguna categoría o clase.

3. (Opcional) Hay dos parámetros posibles que definir, sería el tamaño de ventana que de manera predeterminada viene indicado cero, para tomar una oración completa. El otro parámetro es definir el inicio de la numeración de los archivos tipo "prep", por defecto inician con la el número cero, pero puede iniciarse con otro valor. Ambos campos de estos parámetros están validados en ciertos rangos que no pueden ser sobre pasados.

4. Finalmente se da clic sobre el botón "Gettxt.prep files" que se encuentra dentro del cuadro izquierdo en la herramienta, esto abrirá una ventana indicando si se desea generar archivos de texto plano sin el formato "prep"; posteriormente independiente de la opción elegida se abrirá una ventana de exploración para indicar el directorio en donde se crearán estos archivos "prep" y "txt" si así se indico; posteriormente a la selección, la herramienta creara los archivos y al finalizar lo indicara en una ventana como se muestra en la figura 3.9. En la carpeta donde se ubican los archivos "prep" se genera un archivo "txt" con el nombre del archivo cargado en "Dictionary" más el término "MATCHES" donde contiene los términos cargados en "Dictionary" que coincidieron en el corpus. Este archivo relaciona cada término con los archivos creados. El primer archivo generado tendrá un numero que predeterminadamente viene cero, este primer archivo representa el primer termino de este archivo "MATCHES". Por lo que el número de términos contenidos en este archivo son el número de archivos "prep" que fueron generados.

Existe un segundo archivo que inicia por el nombre del archivo cargado en "Dictionary" seguido de "termclass-repeat-sentence.txt" en el cual se enlista los términos-clase repetidos en una misma oración, o si están dentro del tamaño de ventana definido.

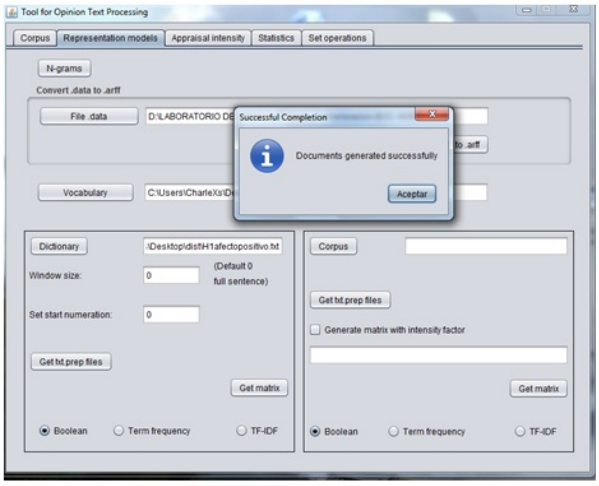

Figura 3.9: Generación de documentos “prep”

Un archivo tipo “prep” tiene la siguiente estructura:

Termino-de-diccionario1/JJ
Termino-de-diccionario2/JJ
Termino-de-diccionario1/JJ
Termino-de-diccionario2/JJ
Termino-de-diccionario3/JJ
Termino-de-diccionario4/JJ
Termino-de-diccionario2/JJ
Termino-de-diccionario2/JJ

Al final del listado queda un renglón vacío, al final de cada término existe un delimitador que es la diagonal seguida de dos jotas “/JJ”.

3.2.4. Generación de Documentos Tipo “prep” para Colección de Documentos

Se procede a seguir los siguientes pasos:

1. En la pestaña actual, se usa el botón “Vocabulary”, donde mediante una ventana de exploración se seleccionará un archivo tipo “txt” donde su contenido será el vocabulario de todos los documentos seleccionados.

2. Con el botón “Corpus” que se ubica en el cuadro derecho, se seleccionará los archivos tipo “txt” o carpetas que contengan los archivos “txt” que se desea conjuntar en un corpus.

3. Finalmente se realiza clic sobre el botón “Gettxt.prep files” que se encuentra dentro del cuadro derecho en la herramienta, se abrirá una ventana de exploración para indicar el directorio en donde se crearan estos arhivos “prep”; posteriormente a la selección, la herramienta creará los archivos y al finalizar lo indicará en una ventana como se muestra en la figura 3.10. Dentro de la carpeta que se encuentran los archivos tipo “prep” se tendrá un archivo llamado “orderfileDocumentsprep.txt” el cual contendrá el nombre de los archivos que fueron utilizados y la carpeta a la que pertenecen.

Si algún archivo de los cuales fueron cargados en “Corpus” no contiene ningún término del archivo cargado en “Vocabulary”, este será descartado y no aparecerá como archivo tipo “prep” y por consiguiente no aparecerá en “orderfileDocumentsprep.txt”.

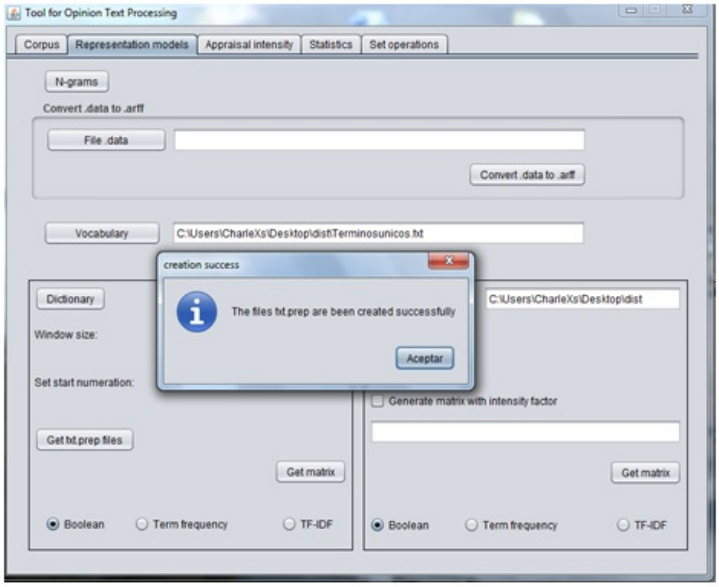

Figura 3.10: Generación de documentos "prep" partiendo de una Colección de documentos de texto

3.2.5. Generación de Matrices (archivo tipo "arff") por un Corpus

Requisitos de otra pestaña:

- Cargar un corpus

Se procede con los siguientes pasos:

1. En la pestaña actual, se usará el botón "Vocabulary", donde mediante una ventana de exploración se seleccionará un archivo tipo "txt" donde su contenido será un listado de los términos del corpus.
2. Con el botón "Dictonary" se realiza la selección de un archivo tipo "txt" donde se enlista términos de alguna categoría o clase.
3. (Opcional) Hay dos parámetros a modificar, el tamaño de ventana que por defecto viene indicado cero, para tomar una oración completa. El segundo parámetros es el tipo de matriz a generar, hay tres posibilidades, matriz booleana, matriz de frecuencia y de TF-IDF.

   Matriz booleana:

   Un valor del vector es igual a "uno" cuando hay coincidencia de un término-clase en una rango determinado por el parámetro tamaño de ventana o en su defecto de una oración completa. De lo contrario si no existió coincidencias en el corpus de este término-clase entonces el valor es "cero".

   Matriz de frecuencias:

   Retomando la matriz booleana, aquí el vector tendrá valores enteros y se irá incrementando los valores del vector como tantas coincidencias exista en el corpus y que este dentro del tamaño de ventana por cada término-clase con cada término del vocabulario.

Matriz TF-IDF:

Se usa la siguiente fórmula

$$\text{TF IDF} = \frac{\text{f}}{\sum \text{fO}} \cdot \log_{10} \frac{\text{nO}}{\sum \text{fT}}$$

Donde:

f = frecuencia de un término del vocabulario
n0 = Número de oraciones totales

$$\Sigma f0 =$$

Sumatoria de las frecuencias de todos los términos únicos según documento

$$\Sigma fT =$$

Sumatoria del número de oraciones en las que aparece un término del vocabulario

4. Esta fórmula sirve para obtener todo los valores del vector término-clase. Finalmente se da clic sobre el botón "Getmatrix" que se encuentra dentro del cuadro izquierdo en la herramienta, esto abrirá una ventana de exploración para indicar donde desea guardar y con que nombre, y al finalizar lo indicará en una ventana como se muestra en la figura 3.11. En la carpeta se encontrara el archivo como inicialmente se nombro pero también estará dos archivos más, uno con el orden de los términos-clase utilizados para la generación de la matriz y estos corresponden directamente a cada vector de la matriz en el archivo "arff". Esto quiere decir que el primer términos-clase en el archivo llamado "filenameDictionary"-coincidencias.txt" es el primer vector de la matriz en el archivo"arff". Donde "filename-Dictionary" es sustituido por el nombre del archivo cargado en el botón "Dictionary". El segundo archivo inicia por el nombre del archivo cargado en "Dictionary" seguido de "termclass- repeat-sentence.txt" en el cual se enlista los términos-clase repetidos en una misma oración, o si están dentro del tamaño de ventana definido.

Los vectores con valor igual a cero son descartados y en el archivo "filenameDictionary"-coincidencias.txt" mantiene esa relación por lo que es posible observar menos términos-clase de los que tiene el archivo cargado.

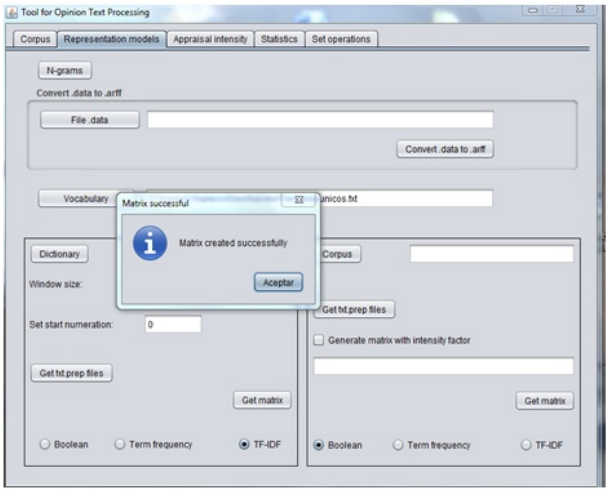

Figura 3.11: Generación de matriz TF-IDF con un Corpus(archivo .arff)

3.2.6. Generación de Matrices (archivo tipo "arff") por Colección de Documentos

Se procede a seguir los siguientes pasos:

1. En la pestaña actual, se usará el botón "Vocabulary", donde mediante una ventana de exploración se seleccionará un archivo tipo "txt" donde su contenido será un listado de los términos de la colección de documentos.

2. Con el botón "Corpus" que se ubica en el cuadro derecho, se seleccionará los archivos tipo "txt" o carpetas que contengan los archivos "txt" que se desea conjuntar en un corpus.

3. Se puede elegir 3 tipos de matrices, matriz booleana, matriz de frecuencia y de TF-IDF. Por configuración predeterminada esta matriz booleana.

4. Finalmente se realiza un clic sobre el botón "Getmatrix" que se encuentra dentro del cuadro izquierdo en la herramienta, esto abrirá una ventana de exploración para indicar donde desea guardar y con que nombre, y al finalizar lo indicará en una ventana como se muestra en la figura 3.12. En la carpeta se encontrara el archivo como inicialmente se nombró pero también estará un archivo con el orden de los archivos utilizados para la generación de la matriz y estos corresponden directamente a cada vector de la matriz en el archivo "arff". Esto quiere decir que el nombre del primer archivo que aparece en "nombre de la matriz.arff+OrderFilesMatrix.txt" es el primer vector de la matriz en el archivo "arff".

Los vectores con valor igual a cero son descartados y en el archivo"OrderFilesMatrix.txt" mantiene esa relación por lo que es posible observar menos archivos de los que formaron el corpus inicial.

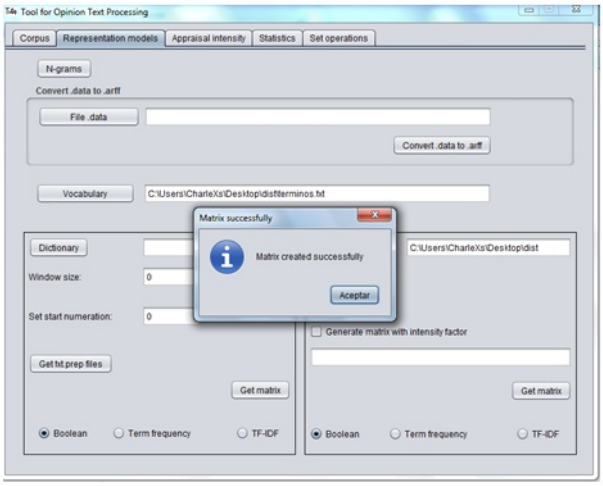

Figura 3.12: Generación de una matriz booleana (archivo .arff) por una colección de documentos.

3.2.7. Generación de Matrices (archivo tipo "arff") por Colección de Documentos con Factor de Intensidad

Se procede a seguir los siguientes pasos:

1. Con el botón "Corpus" que se ubica en el cuadro derecho, se seleccionará los archivos tipo "txt" o carpetas que contengan los archivos "txt" que se desea conjuntar en un corpus.

2. Se selecciona la casilla "Generatematrixwithintensity factor", esto nos desplegará una ventana de exploración para seleccionar el archivo tipo "txt" que deberá llevar la siguiente estructura. Debe ser un listado de términos con un espacio seguido del factor de intensidad, como ejemplo se muestra:

Comprar 3
Jugar 2
Descuidado 1
Limpio siempre 2

   De no seleccionar el archivo se deshabilitará la casilla de "Generatematrixwithintensity". Al habilitar este 2° paso con un archivo como se indica en el ejemplo se debe habilitar el botón "Getmatrix" como se observa en la figura 3.13.

3. Se puede elegir 3 tipos de matrices, matriz booleana, matriz de frecuencia y de TF-IDF. Por configuración predeterminada es matriz booleana.

4. Finalmente se realiza clic sobre el botón "Getmatrix" que se encuentra dentro del cuadro izquierdo en la herramienta, esto abrirá una ventana de exploración para indicar donde desea guardar y con que nombre, y al finalizar lo indicara en una ventana como se muestra en la figura 3.13. En la carpeta encontrará el archivo como inicialmente se nombro pero también estará un archivo con el orden de los archivos utilizados para la generación de la matriz y estos corresponden directamente a cada vector de la matriz en el archivo "arff". Esto quiere decir que el nombre del primer archivo que aparece en "nombre de la matriz+OrderFilesMatrix.txt" es el primer vector de la matriz en el archivo"arff".

El factor de intensidad multiplicará cada valor de los vectores de la matriz, sea esta booleana, de frecuencia o tf-idf por su correspondiente término del vocabulario, por ese motivo no es

necesario seleccionar el vocabulario debido a que el archivo que se selecciona para factor de intensidad es el mismo vocabulario con sus respectivos valores de intensidad. Los vectores con valor igual a cero son descartados y en el archivo nombre de la matriz+OrderFilesMatrix.txt" mantiene esa relación por lo que es posible observar menos archivos de los que formaron el corpus inicial.

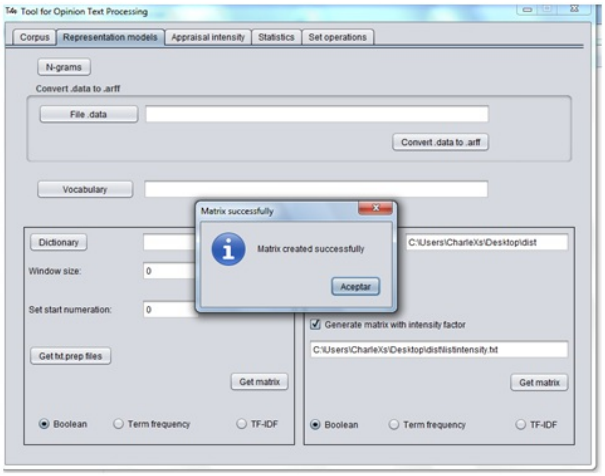

Figura 3.13: Generación de una matriz booleana (archivo .arff) por una colección de documentos con factor de intensidad

3.3. Pestaña Intensidad de Valoración

Este módulo es para el procesamiento de la intensidad de valoración con la que se clasifica una lista de palabras.

3.3.1. Tabulación de Probabilidades de Instancia con Umbrales de una Sola Clase

Requisito previo:

El archivo de resultados para el paso 1 de esta sección, se crea por medio de la sección Explorador dentro de la pestaña Clasificar en Weka; la estructura del archivo de resultados es como se muestra en el sección A.

El dato de interés del resultado es la línea 13:

"inst#, actual, predicted, error, probability distribution"

Donde en la siguiente línea comienzan las instancias clasificadas con sus respectivos valores. En el paso 2 de no encontrar la línea 13, desplegará la herramienta en una pequeña ventana el mensaje de no haber encontrado instancias.

Se procede a seguir los siguientes pasos:

1. Se carga un archivo de resultados de la herramienta Weka con el botón "Load Wekaresult".
2. Al cargar correctamente el archivo en el paso anterior, se habilitará el botón "Process" en el cual se dará clic para trabajar el archivo. En la ventana principal de la herramienta se observará debajo del cuadro de la dirección del archivo cargado un pequeño texto

indicando si existe alguna clase o no en el archivo cargado. Para este ejemplo se observa en la figura 3.14, que hay una clase en el archivo cargado, al mismo tiempo la herramienta desplegará una ventana como el de la figura 3.14 del lado derecho, donde tomara los valores acertados de la columna "probability" del archivo cargado, y se sustituyen con cero las clasificaciones erróneas. En esta ventana esta la posibilidad de guardar parte de la tabla de probabilidades o la tabla completa en un archivo "txt".

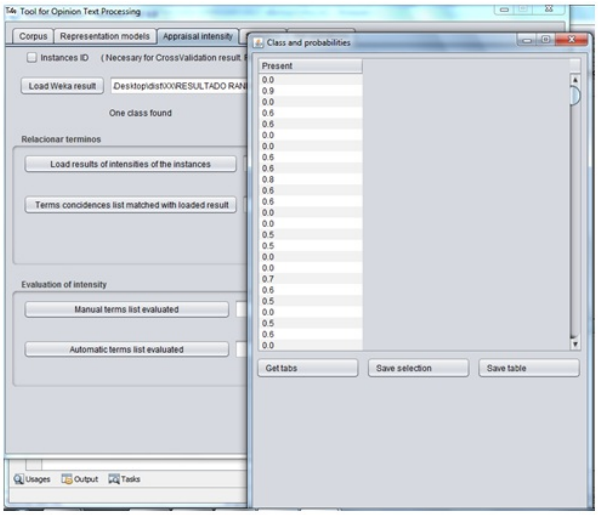

Figura 3.14: Extracción de las probabilidades de una clase de un archivo de resultados de Weka

Para poder tabular las probabilidades de la ventana "class and probabilities" esta la opción "Gettabs" que nos desplegará una ventana llamada " GeneratetabsfromThreshold" con diferentes umbrales dentro de los cuales están:

a. La media $\bar{X}$
b. Media menos desviación estándar $\bar{X}$ - α
c. Media menos el doble de la desviación estándar $\bar{X}$ - 2α
d. Media menos el triple de la desviación estándar $\bar{X}$ - 3α
e. Media más desviación estándar $\bar{X}$ + α
f. Media más el doble de la desviación estándar $\bar{X}$ + 2α
g. Media más el triple de la desviación estándar $\bar{X}$ + 3α
h. Media del rango en el umbral de [$\bar{X}$ - α,$\bar{X}$ + α]
i. Media del rango en el umbral de [$\bar{X}$ - 2α,$\bar{X}$ + 2α]
j. Media del rango en el umbral de [$\bar{X}$ - 3α,$\bar{X}$ + 3α]

Para obtener la tabulación con base en la medida seleccionada, se realiza clic sobre "Obtainintensityfromtheselectedthreshold". Como se muestra en la figura 3.15.

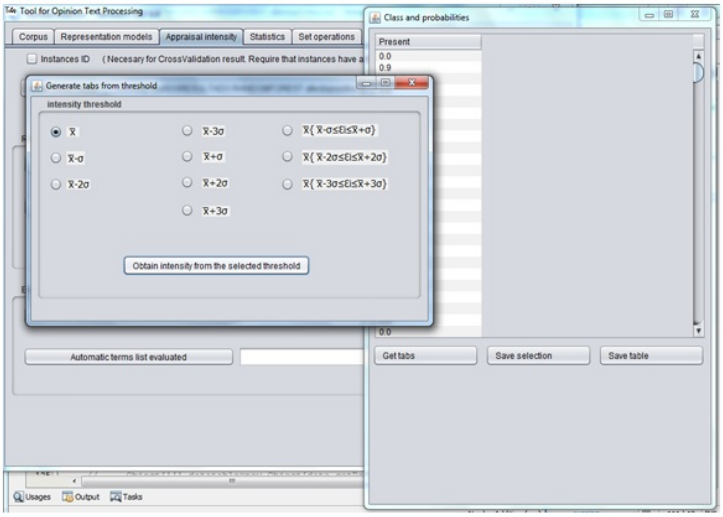

Figura 3.15: Eligiendo un umbral para la tabulación de las probabilidades y obtención de intensidades

3. El paso anterior desplegará una nueva ventana llamada "Valuation of instances", desplegando valores tabulados referentes al umbral o medida seleccionada en el paso 3. Esta ventana como se muestra en la figura 3.16, despliega información bajo la tabla de valores, con datos como el número de instancias **fuertes, promedios y falsos positivos.** Con la posibilidad de guardar por selección o guardar la tabla completa en un archivo "txt".

Es recomendable guardar la tabla completa con los falsos positivos (es posible mediante la opción de una casilla de habilitación no guardar los falsos positivos), para preservar el orden de los valores de las instancias para luego que estos valores posteriormente concuerde la relación con sus respectivos términos.

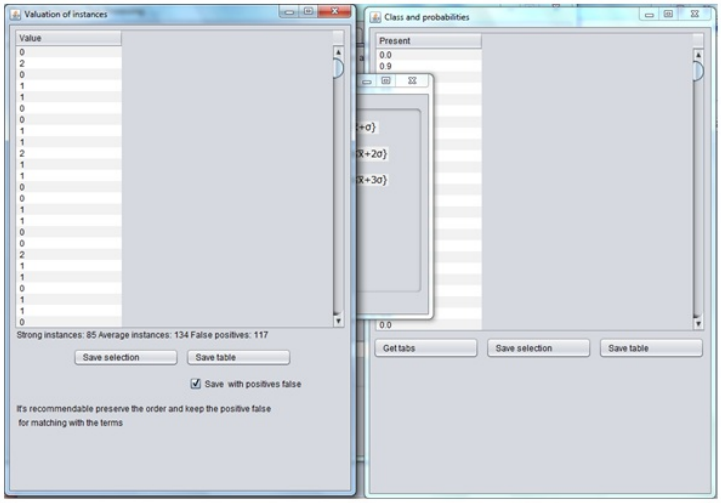

Figura 3.16: Intensidades obtenidas del umbral previamente seleccionado

El orden de una tabla en un archivo "txt" guardado es en el del momento previo al almacenaje, por lo que si se utilizó ordenamiento por valores o términos estos nuevos ordenamientos son transferidos al guardarse como archivo "txt".

3.3.2. Tabulación de Probabilidades de Instancia con Umbrales de dos Clases

Requisito previo:

El archivo de resultados de Weka que se cargará debe contar con el parámetro de instancias con numero ID. Este ejemplo se muestra en la sección B. Este parámetro permitirá no perder la relación de las instancias. Por lo que la línea marcada como número 13 en la sección B debe estar presente en el archivo de resultado.

Los pasos a seguir para realizar esta tarea son:

1. Se habilita la casilla de verificación "Instances ID" donde se hace mención que es necesario para resultados con modo de prueba de validación cruzada.

2. Se carga el archivo de resultado como se resalta en la sección B.

3. Habilitado el botón "Process" se realiza un clic y desplegará dos ventanas sobre puestas, para observar ambas se tiene que arrastras la ventana de la clase 2 hacia algún área disponible en la pantalla. En la herramienta se han definido únicamente clase 1 y clase 2. Se observa estas dos ventanas en la figura 3.17. Cada con las opciones de guardado por selección o tabla completa en archivos "txt". Los siguientes pasos son similares a la sección anterior de una clase.

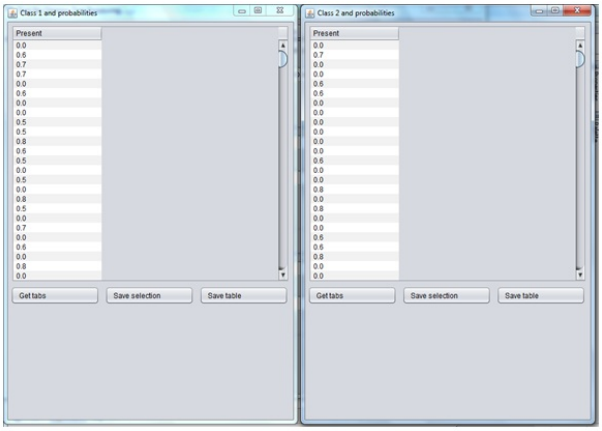

Figura 3.17: Extracción de las probabilidades de dos clases de un archivo de resultados de Weka

4. Para poder tabular las probabilidades de cualquiera de las dos ventanas tiene la opción "Gettabs" que nos desplegara una ventana llamada " GeneratetabsfromThreshold" con diferentes umbrales dentro de los cuales están:

   a. La media $\bar{X}$
   b. Media menos desviación estándar $\bar{X}$ - α
   c. Media menos el doble de la desviación estándar $\bar{X}$ - 2α
   d. Media menos el triple de la desviación estándar $\bar{X}$ - 3α
   e. Media más desviación estándar $\bar{X}$ + α
   f. Media más el doble de la desviación estándar $\bar{X}$ + 2α
   g. Media más el triple de la desviación estándar $\bar{X}$ + 3α
   h. Media del rango en el umbral de [$\bar{X}$ - α,$\bar{X}$ + α]
   i. Media del rango en el umbral de [$\bar{X}$ - 2α,$\bar{X}$ + 2α]
   j. Media del rango en el umbral de [$\bar{X}$ - 3α,$\bar{X}$ + 3α]

Para obtener la tabulación con base en la medida seleccionada, se realiza clic sobre “Obtain intensity from the selected threshold”. Como se muestra en la figura 3.18.

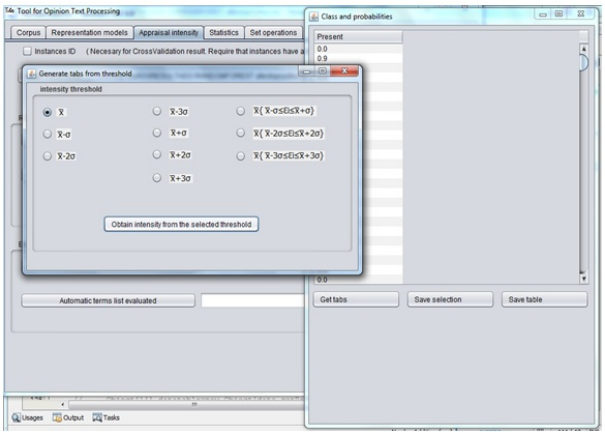

Figura 3.18: Eligiendo un umbral para la tabulación de las probabilidades y obtención de intensidades

5. El paso anterior desplegará una nueva ventana llamada “Valuation of instances”, desplegando valores tabulados referentes al umbral o medida seleccionada en el paso 3. Esta ventana como se muestra en la figura 3.19, despliega información bajo la tabla de valores, con datos como el número de instancias **fuertes, promedios y falsos positivos.** Con la posibilidad de guardar por selección o guardar la tabla completa.

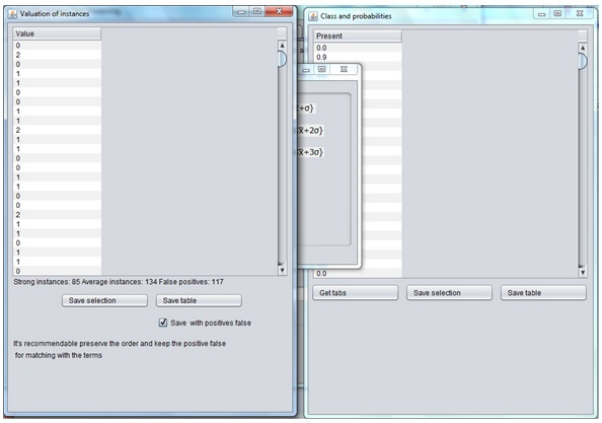

Figura 3.19: Intensidades obtenidas del umbral previamente seleccionado

Es recomendable guardar la tabla completa con los falsos positivos (es posible mediante la opción de una casilla de habilitación no guardar los falsos positivos), para preservar el orden de los valores de las instancias para luego que estos valores posteriormente concuerde la relación con sus respectivos términos.

Nota: El orden de una tabla en un archivo “txt” guardado es en el del momento previo al almacenaje, por lo que si se utilizo ordenamiento por valores o términos estos nuevos ordenamientos son transferidos al guardarse como archivo “txt”.

3.3.3. Relación de Términos

En esta sección se relacionan dos archivos de los cuales uno contiene valores de intensidad obtenidos de la sección anterior a esta y el segundo archivo son los términos que mantienen

relación con las instancias clasificadas, por lo que este archivo seguramente será el archivo de coincidencias de la generación de matriz o los archivos usados para generación de matriz por colección de documentos.

Se procede con los siguientes pasos para conjuntar ambas partes:

1. Se carga un archivo con valores de intensidad.
2. Se carga el archivo con los términos correspondientes a los valores cargados y se realiza un clic en "Match".
3. Se obtendrá una ventana llamada "List of terms and intensities", en la cual mostrará una tabla con los términos y sus respectivas intensidades. Aquí se dispone de opciones conocidas como guardar tabla completa, guardar selección en un archivo "txt" y como parámetro que afecta a los dos modos anteriores de guardar la posibilidad de guardar con intensidad y sin la intensidad mediante una casilla de verificación. Todo esto se observa en la figura 3.20.

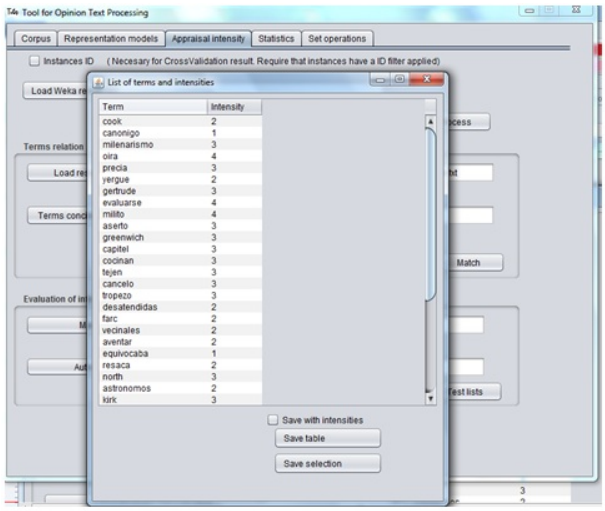

Figura 3.20: Unión de Términos relacionados con su respectiva intensidad

En esta sección existe la validación donde se debe cumplir que el mismo número de términos debe ser igual al número de valores de intensidad. Si algún archivo fue cargado y al realizar la relación indica la herramienta que no fue cargado existe la posibilidad de que no cumpla el formato esperado. El orden de una tabla en un archivo "txt" guardado es en el del momento previo al almacenaje, por lo que si se utilizo ordenamiento por valores o términos estos nuevos ordenamientos son transferidos al guardarse como archivo "txt".

3.3.4. Evaluación de la Intensidad

En esta sección se evalúa los resultados obtenidos de métodos computacionales como puede ser Weka y esta herramienta, y tomando como referencia el mismo conjunto de términos evaluados manualmente o por otro método para obtener 3 tipos de medidas: Precisión, Recuerdo, Fmeasure.

Se procede con los siguientes pasos para la evaluación:

1. Se carga mediante el botón "Manual termslistevaluated" los términos evaluados manualmente con sus intensidades. La estructura es un listado de los términos con un espacio seguido de un número que representa la intensidad y un salto de línea en un archivo "txt".

2. Se realiza lo mismo que en el paso 1 pero con el botón “Automatictermslistevaluated” y con los términos evaluados mediante métodos computacionales.

3. Posteriormente se da clic sobre “Test lists” y esto desplegará una ventana como la mostrada en la figura 3.21. Esta ventana muestra una tabla de relación entre las intensidades de un mismo término. En esta ventana se puede guardar de las siguientes formas:

    a. Generar y guardar el resultado; esto genera una tabla de confusión entre los valores de intensidad correctos (realizados manualmente) contra los evaluados computacionalmente. Entrega las medidas Precisión, Recuerdo y Fmeasure. Con la posibilidad de guardar la tabla de relaciones completa, esto define la estructura del archivo de texto en el orden donde primero aparece la tabla con los valores y finalmente los resultados de la tabla de confusión con las medidas antes mencionadas.

    b. Guardar selección en un archivo “txt”, esto como en otras secciones de despliegue de tablas, permite guardar la selección de ciertos términos de la tabla con sus valores o sin ellos, esto se logra mediante la casilla de verificación que se observa en la figura 3.21.

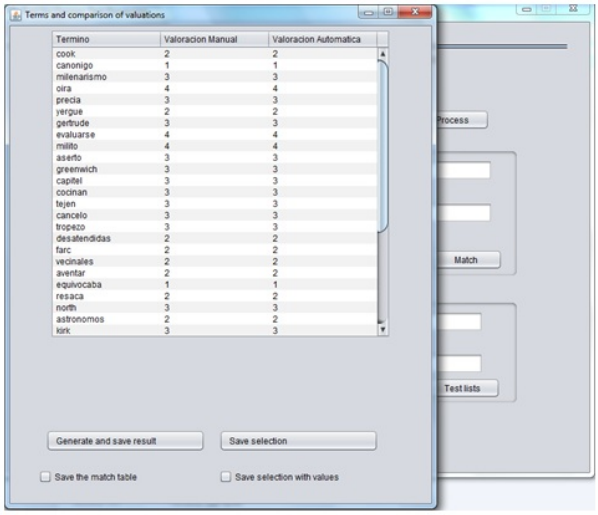

Figura 3.21: Ventana de evaluación de intensidades manuales e intensidades obtenidas por métodos computacionales

Se valida el tamaño de las listas y si no coinciden el mismo número de términos en ambas, se notifica por medio de una ventana, además de que cuando algunos términos que no coinciden en ambas listas este número de términos no coincidentes se notifica por una ventana antes de desplegar los tabla con los términos que si coincidieron. En caso de que un archivo no esté correcto en una línea del listado lo indicará como se muestra la figura 3.22. El orden de una tabla en un archivo “txt” guardado es en el del momento previo al almacenaje, por lo que si se utilizó ordenamiento por valores o términos estos nuevos ordenamientos son transferidos al guardarse como archivo “txt”.

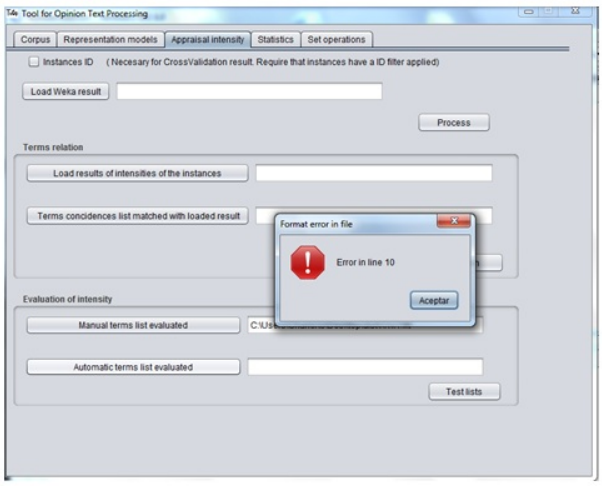

Figura 3.22: Validación de documento al cargar en la sección “Evaluación de intensidades”

3.4. Pestaña Estadísticas

Este módulo está dedicado a recolectar estadísticas sobre la composición de un corpus y listas de palabras.

3.4.1. Frecuencia y polaridad opuesta

Requisitos de otra pestaña:

- Cargar un corpus

Para obtener las frecuencias de una lista de términos donde coinciden en una misma oración con los términos de un vocabulario se procede a los siguientes pasos:

1. Se debe verificar que este seleccionada la opción “Termsfrequency” que predeterminadamente viene en la herramienta seleccionada.
2. Se carga el vocabulario con el botón ”Vocabulary”.
3. Se carga la lista de términos a buscar con el botón“ Termsto look for”.
4. Se realiza un clic sobre el botón “Process” el cual desplegará una nueva ventana donde muestra una tabla de los términos buscados que coincidieron cierto número de veces con el vocabulario estos términos pueden ser guardados por medio de selección o guardando la tabla completa con o sin sus frecuencias en un archivo “txt”. Un ejemplo de los pasos y el despliegue de la nueva ventana lo muestra la figura 3.23.

El orden de una tabla en un archivo “txt” guardado es en el del momento previo al almacenaje, por lo que si se utilizó ordenamiento por valores o términos estos nuevos ordenamientos son transferidos al guardarse como archivo “txt”. En el caso de expresiones, una expresión a buscar que coincida de inicio con una expresión del vocabulario este término o expresión del vocabulario se descarta si existe coincidencia.

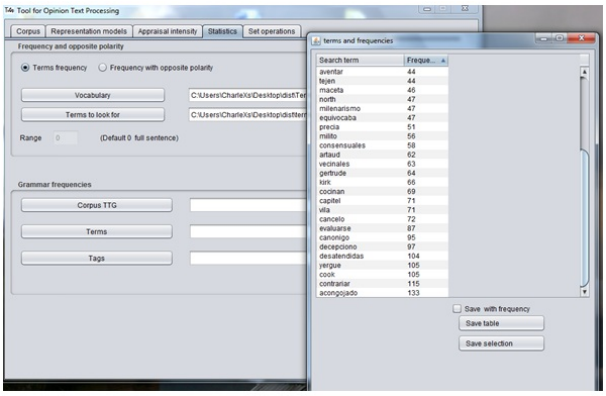

Figura 3.23: Frecuencia de términos que coinciden con el vocabulario en una oración

Para obtener frecuencias de términos con polaridad inversa se habilitan unas opciones más y se procede a los siguientes pasos:

1. Seleccionar la opción “Frequencywithoppositepolarity”
2. Cargar una lista de términos de polaridad inversa.
3. Cargar la lista de los términos a buscar su polaridad inversa.
4. (Opcional) Es posible elegir el rango o tamaño de ventana donde exista coincidencias con los términos de polaridad inversa, de manera predeterminada está configurado el valor “uno” que indica que se tomarán como coincidencias a una distancia de una palabra. Además de esto cuenta con el parámetro de búsqueda detrás del término sin tomar lo que esta adelante del término que se busca su polaridad; para tomar esta opción únicamente se selecciona la casilla “ Serachuniquebehindtotheterm”.

Finalmente se realiza clic sobre el botón “Process” el cual desplegará la ventana como se muestra en la figura 3.24. En la cual se puede observar que la búsqueda se realizo en un tamaño de ventana de 10 y con búsqueda hacia atrás. Con la tabla obtenida se puede almacenar por selección o la tabla completa en un fichero “txt”, con o sin las frecuencias haciendo uso de la casilla de verificación “ Savewithfrequency”.

Nota: Para la búsqueda detrás de un término o expresión (Terms to look for) la herramienta toma en cuenta el primer “término a buscar“ que encuentra en la oración de derecha a izquierda, por lo que si existe el mismo “término a buscar” en la misma oración solo se considera el primero para la obtención de frecuencias.

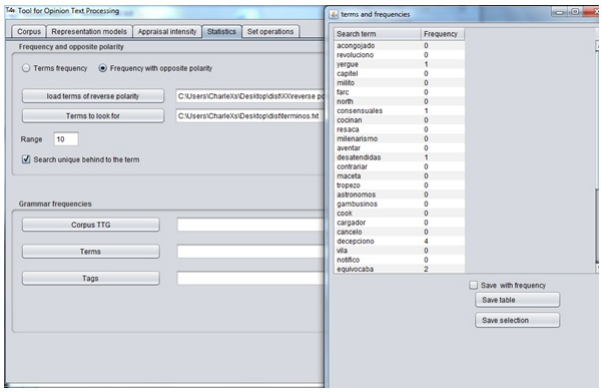

Figura 3.24: Frecuencia de términos que coinciden con el vocabulario en una oración

El orden de una tabla en un archivo "txt" guardado es en el del momento previo al almacenaje, por lo que si se utilizo ordenamiento por valores o términos estos nuevos ordenamientos son transferidos al guardarse como archivo "txt".

3.4.2. Frecuencias Gramaticales

En esta sección se obtienen frecuencias de términos asociados a etiquetas, de las cuales formulan en un corpus tipo "ttg".

Para obtener estas frecuencias se procede con los siguientes pasos:

1. Cargar un archivo "ttg" con el boton "Corpus TTG".
2. Cargar con el botón "Terms" de un archivo "txt" una lista de términos de los cuales se desea obtener la frecuencia con que aparecen con las etiquetas del siguiente paso.
3. Cargar una lista de etiquetas en un archivo "txt" con el botón "Tags".
4. Finalmente al hacer clic en el botón "Process" se desplegará una ventana con una tabla de términos y frecuencias como en la figura 3.25. La cual puede ser almacenada en un archivo "txt". Se puede guardar la tabla completa o por selección con o sin las frecuencias por medio de una casilla de verificación llamada "Savewithfrequency".

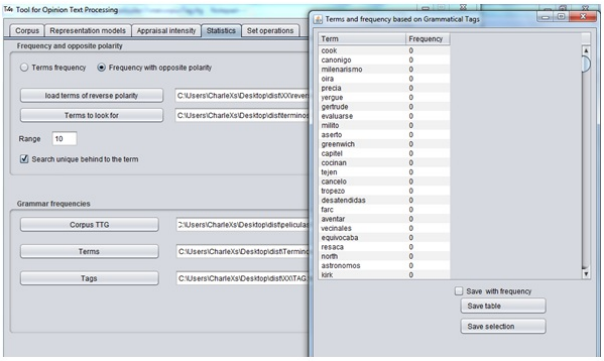

Figura 3.25: Frecuencia de términos vinculados con una lista de etiquetas en un corpus de formato "ttg"

El orden de una tabla en un archivo "txt" guardado es en el del momento previo al almacenaje, por lo que si se utilizó ordenamiento por valores o términos estos nuevos ordenamientos son transferidos al guardarse como archivo "txt".

3.4.3. Operaciones de Conjuntos

En esta sección se realizan 3 tipos de operaciones: Intersección, Unión y Diferencia. Estas operaciones se realizan con listas de términos en archivos de "txt".

Para realizar las operaciones se procede con los siguientes pasos:

1. Se carga con los botones "List 1" y "List 2" dos archivos de texto "txt" con una lista de términos cada una con los cuales se realizaran las operaciones.

2. Se selecciona la operación deseada donde por configuración predeterminada esta "Intersection" y puede ser elegida "Union" o "Difference" y se finaliza haciendo clic sobre "Start operation".

Lo cual hará que realice la operación e indique seguido de la etiqueta "Number of terms of operation result" el número de términos que resultaron. Finalmente estos términos pueden ser almacenados con el botón "Save result" en formato "txt". Se puede observar un pequeño ejemplo en la figura 3.26.

Para la operación de diferencia; de la "lista 1" se obtiene la diferencia con respecto a la "lista 2" por lo que para esta operación el orden de los archivos cargados si afecta al resultado.

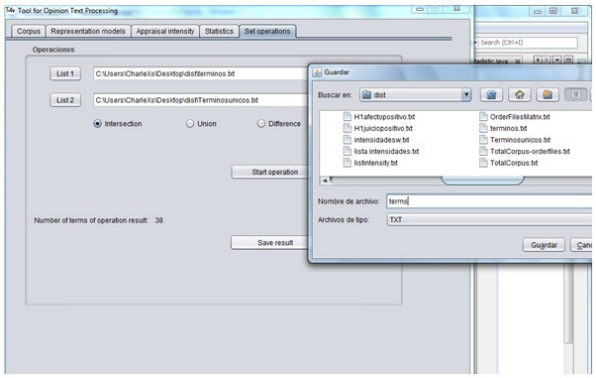

Figura 3.26: Operación de conjuntos con dos listas de términos

Apéndice A

Para fines informativos se muestra a continuación el archivo de resultados "recortado de instancias" de una clasificación de Weka, se observará que en la línea 4 indica el número de instancias y al ver la lista que llega hasta la instancia número 10 en la línea 23 se hace recordar el término "recortado de instancias" por motivos prácticos de demostración.

1. === Run information ===
2. Scheme: weka.classifiers.trees.RandomForest -I 10 -K 0 -S 1
3. Relation: afectopositivo
4. Instances: 672
5. Attributes: 19069
6. [$listofattributesomitted$]
7. Test mode: user supplied test set: size unknown (reading incrementally)
8. === Classifier model (full training set) ===
9. Random forest of 10 trees, each constructed while considering 15 random features.
10. Out of bag error: 0.4583
11. Time taken to build model: 13.94 seconds
12. === Predictions on test set ===
13. inst#, actual, predicted, error, probability distribution
14. 1 1:-1 2:-2 + 0.4 * 0.6
15. 2 1:-1 1:-1 * 0.9 0.1
16. 3 1:-1 2:-2 + 0.4 * 0.6
17. 4 1:-1 1:-1 * 0.6 0.4
18. 5 1:-1 1:-1 * 0.6 0.4
19. 6 1:-1 2:-2 + 0.4 * 0.6
20. 7 1:-1 2:-2 + 0.4 * 0.6
21. 8 1:-1 1:-1 * 0.6 0.4
22. 9 1:-1 1:-1 * 0.6 0.4

23. 10 1:-1 1:-1 * 0.8 0.2
24. === Evaluation on test set ===
25. === Summary ===
26. Correctly Classified Instances 219 65.1786 %
27. Incorrectly Classified Instances 117 34.8214 %
28. Kappa statistic 0
29. K&B Relative Info Score 2144.8651 %
30. K&B Information Score 21.4487 bits 0.0638 bits/instance
31. Class complexity | order 0 336 bits 1 bits/instance
32. Class complexity | scheme 343.5893 bits 1.0226 bits/instance
33. Complexity improvement (Sf) -7.5893 bits -0.0226 bits/instance
34. Mean absolute error 0.472
35. Root mean squared error 0.5052
36. Relative absolute error 94.4048 %
37. Root relative squared error 101.0304 %
38. Total Number of Instances 336
39. === Detailed Accuracy By Class ===
40. TP Rate FP Rate Precision Recall F-Measure ROC Area Class
41. 0.652 0 1 0.652 0.789 ? -1
42. 0 0.348 0 0 0 ? -2
43. Weighted Avg. 0.652 0 1 0.652 0.789 0
44. === Confusion Matrix ===
45. a b – classified as
46. 219 117 $|a = -1$
47. 0 0 $|b = -2$

Apéndice B

Para fines informativos se muestra a continuación el archive de resultados "recortado de instancias" de una clasificación de Weka; se observará que en la línea 4 indica el número de instancias y al ver la lista que llega hasta la instancia número 10 en la línea 23 se hace recordar el término "recortado de instancias" por motivos prácticos de demostración. A diferencia de la sección A, este archivo de resultados maneja un parámetro más en la línea 13 "(ID), este es un número de identificación que se asigna previamente a una matriz "arff" mediante un filtro en la herramienta Weka donde agrega a cada vector de la matriz este ID, para preservar el orden al realizar clasificaciones con el modo de prueba "cross- validation".

1. === Run information ===
2. Scheme: weka.classifiers.trees.RandomForest -I 10 -K 0 -S 1
3. Relation: H1afectopositivo-weka.filters.unsupervised.attribute.AddID-Cfirst-NID
4. Instances: 1342
5. Attributes: 4098
6. [$listofattributesomitted$]
7. Test mode: 5-fold cross-validation
8. === Classifier model (full training set) ===
9. Random forest of 10 trees, each constructed while considering 13 random features
10. Out of bag error: 0.4747
11. Time taken to build model: 2.09 seconds
12. === Predictions on test data ===
13. inst#, actual, predicted, error, probability distribution (ID)
14. 1 2:-2 1:-1 + *1 0 (604)
15. 2 2:-2 2:-2 0.4 *0.6 (1120)
16. 3 2:-2 1:-1 + *0.7 0.3 (657)
17. 4 2:-2 2:-2 0.3 *0.7 (1206)
18. 5 2:-2 2:-2 0.2 *0.8 (1212)
19. 6 2:-2 2:-2 0.1 *0.9 (503)
20. 7 2:-2 1:-1 + *0.8 0.2 (444)

21. 8 2:-2 2:-2 0.4 *0.6 (596)
22. 9 2:-2 2:-2 0.4 *0.6 (1044)
23. 10 2:-2 1:-1 + *0.7 0.3 (381)
24. === Stratified cross-validation ===
25. === Summary ===
26. Correctly Classified Instances 721 53.7258 %
27. Incorrectly Classified Instances 621 46.2742 %
28. Kappa statistic 0.0745
29. K&B Relative Info Score 7218.5465 %
30. K&B Information Score 72.1855 bits 0.0538 bits/instance
31. Class complexity | order 0 1342.003 bits 1 bits/instance
32. Class complexity | scheme 9950.9037 bits 7.415 bits/instance
33. Complexity improvement (Sf) -8608.9007 bits -6.415 bits/instance
34. Mean absolute error 0.4763
35. Root mean squared error 0.5092
36. Relative absolute error 95.2607 %
37. Root relative squared error 101.8486 %
38. Total Number of Instances 1342
39. === Detailed Accuracy By Class ===
40. TP Rate FP Rate Precision Recall F-Measure ROC Area Class
41. 0.636 0.562 0.531 0.636 0.579 0.571 -1
42. 0.438 0.364 0.546 0.438 0.486 0.571 -2
43. Weighted Avg 0.537 0.463 0.539 0.537 0.533 0.571
44. === Confusion Matrix ===
45. a b < −− classified as
46. 427 244 | a = -1
47. 377 294 | b = -2

Glosario

Fmeasure:
Media harmónica de la precisión y el recuerdo.

n-gramas:
Un n-grama es una sub-secuencia de n palabras de una secuencia sea corpus u oraciones.

Vocabulario:
Son los términos o palabras que tienen una participación activa dentro de un corpus, texto o colección de documentos.

Precisión:
Índice de términos correctamente clasificados como positivos entre el total de términos clasificados como positivos.

Recuerdo:
Índice de términos correctamente clasificados como positivos entre el total de términos positivos.

Tamaño de ventana:
El tamaño de ventana, son los limites de búsqueda donde un término sirve como referencia fija en una oración y se realiza la búsqueda un numero de palabras hacia adelante y hacia atrás.

Termino-clase:
Son términos que pertenecen a una clase que fue clasificado con ciertos criterios que le otorgan la pertenencia a esa clase.

Weka:
Waikato Environment for Knowledge Analysis. Entorno para Análisis del Conocimiento de la Universidad de Waikato. Es una plataforma de software para aprendizaje automático y minería de datos escrito en Java y desarrollado en la Universidad de Waikato.
Página oficial de Weka: http://www.cs.waikato.ac.nz/ml/weka/