Universidad de las Ciencias Informáticas

Facultad 3

Transformación de documentos legales en ficheros de entrada para algoritmos de minería de texto

Trabajo de Diploma para optar por el Título de Ingeniero en Ciencias Informáticas

Autor: Manuel Antonio Rodríguez Aguirre

Tutores: Msc. Julio Cesar Díaz Vera

Msc. Alexander Rodríguez Torres

Ciudad de La Habana, 15 de Mayo de 2012

“Año 54 de la Revolución”

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor del trabajo titulado **Transformación de documentos legales en ficheros de entrada para algoritmos de minería de texto** y autorizo a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los días ___ del mes _____ del año 2012.

_____

Manuel Antonio Rodríguez Aguirre

Firma del Autor

_____ _____

Msc. Julio Cesar Díaz Vera **Msc. Alexander Rodríguez Torres**

Firma del Tutor Firma del Tutor

DATOS DE CONTACTO

Tutores:

Julio Cesar Diaz Vera graduado de Ingeniero en telecomunicaciones y Master en Ciencias en Gestión de Proyecto y Jefe del departamento de Ingeniería de Software de la facultad 3

Msc. Julio Cesar Diaz Vera correo electrónico (jcdiaz@uci.cu**)**

Alexander Rodriguez Torres graduado de Ingeniero en Ciencia Informáticas, Master en Ciencias en Informática Aplicada y Jefe de departamento de Programación de la facultad 3.

Msc. Alexander Rodríguez Torres correo electrónico (atorres@uci.cu**)**

DEDICATORIA

A mi mamá, por su incondicionalidad, su amor, su entrega y su confianza.

A mi papá, por ser mi ejemplo, mi guía, mi amigo y enseñarme que con esfuerzo todo se puede

gracias por haberme convertido en lo que soy hoy.

A mis hermanos Koki, José y Neni, por ser la luz de mis ojos y mi mayor inspiración.

A la memoria de mi abuelo Manuel que siempre es un ejemplo para la familia.

A mis abuelos, Juana, Irene y Canuto, por estar siempre cada vez que los he necesitado.

AGRADECIMIENTOS

A la Revolución, por haberme dado la oportunidad de formarme como profesional en un centro de altos estudios como la Universidad de las Ciencias Informáticas.

A mis tutores por haberme ayudado tanto y por soportarme todo este tiempo, Gracias Julio y Alexander.

A mi hermanito de la UCI a Nani por ser mi cómplice estos 5 años, gracias de corazón.

Al piquete del aula a Amalia, Deysi, Lisbeth, Yadiani, a los que se incorporaron después Lisandra, Aimé.

A mi gente del 9105 Adrián, Rene, Alejandro, El chino, El Chardo, Camejo, Julito, Eddy.

A mi equipo de la FEU Yidian, Pepe Víctor, Olía, Baby.

A Felo por su amistad y su apoyo en todo.

A Nelly y Arle por aguantarme en estos últimos meses donde más necesité de mis amistades.

A todos los que de una forma u otra me han brindado su mano y un poco de su tiempo en mi paso por la UCI.

A todos muchas gracias.

RESUMEN

Este trabajo aborda la temática de preparación de los datos para aplicarle algoritmos de minería de texto, con vista a su utilización en el aprendizaje de ontologías dentro del dominio legal. Se proponen e implementan algoritmos que responden a las tareas de pre procesamiento más importantes: tokenización y lematización de textos. Los resultados del trabajo fueron validados a partir de la implementación de los algoritmos y su utilización para pre procesar dos resoluciones del marco regulatorio de la aduana. Estos resultados obtenidos fueron comparados con los alcanzados por expertos humanos comprobándose la validez de los mismos.

Palabras claves:

Aprendizaje de ontologías, minería de texto, técnicas de pre procesamiento de texto, tokenización, lematización.

ÍNDICE

INDICE DE TABLAS Y FIGURAS

TABLAS

FIGURAS

INTRODUCCIÓN

El interés manifiesto de la sociedad cubana en la implementación de estándares para el gobierno electrónico ha posibilitado la aparición de varios proyectos de informatización en sectores asociados al mismo. En la Universidad de las Ciencias Informáticas se trabaja en la construcción de un conjunto de sistemas de información para los tribunales, las fiscalías y se crean las bases tecnológicas para la implementación de la infraestructura de interoperabilidad entre las distintitas entidades identificadas como partes importantes para el éxito de la tarea. Todo lo anterior conforma la plataforma del país para el gobierno electrónico. Una de las necesidades establecidas como básicas para lograr la implementación de la referida plataforma pasa por la capacidad de intercambiar y utilizar documentos que contienen en sí mismos información sobre el procedimiento que debe seguir el documento y los niveles de acceso a su contenido. Además en la plataforma debe ser posible describir con precisión la semántica de los documentos, facilitando la tarea de recuperación de información sobre los mismos y garantizando los niveles de acceso a su contenido. El uso de ontologías[1] ha sido identificado como un mecanismo importante para satisfacer estas necesidades.

La construcción de ontologías legales para la plataforma de gobierno electrónico en Cuba se ve dificultada por la poca disponibilidad de especialistas funcionales y el limitado entendimiento que los desarrolladores del sistema tienen sobre el dominio legal. Es por ello que en lugar de utilizar metodologías de desarrollo de ontologías ***Ascendentes*** que son aquellas que surgen desde cero donde no existen previo conocimiento ni información contenida de sus relaciones, es recomendable la utilización de enfoques ***Descendentes*** las cuales surgen de una previa información ya sea contenida en bases de datos o en los corpus de documentos como el corpus legal de la Gaceta Oficial de la República de Cuba y particularmente se consideran significativas aquellas que utilizan técnicas de minería de texto como uno de sus componentes. Para lograr aplicar estas técnicas en el dominio jurídico es necesario superar la dificultad que deriva de la no estructuración de los documentos jurídicos.

Problema a resolver:

La no estructuración de los documentos jurídicos dificulta la aplicación de técnicas de minería de datos.

[1] Ontología: Representación formal de un conjunto de conceptos dentro de un dominio específico y de las relaciones entre dichos conceptos.

Objetivos Generales:

Proponer un algoritmo para la transformación de documentos legales en ficheros de entrada para algoritmos de minería de texto.

Objeto de Estudio:

Minería de texto para el aprendizaje de ontologías.

Campo de Acción:

Etapa de pre procesamiento de la minería de texto.

Posibles resultados:

Un algoritmo que permita transformar documentos legales en ficheros de entrada a algoritmos de minería de texto.

Objetivos Específicos:

1. Establecer el marco conceptual de referencia.
2. Definir los requisitos especiales del idioma español, para la extracción de raíces en las palabras.
3. Ajustar los algoritmos del inglés al español.
4. Definir el modelo de componentes.
5. Probar la validez del resultado.

Tareas a cumplir:

1. Recopilación, selección y análisis de la bibliografía referente al tema.
2. Definición de las reglas lingüísticas significativas del español.
3. Selección del mecanismo de transformación.
4. Definición de un marco arquitectónico.
5. Implementación de las funciones asociadas al algoritmo.
6. Validación del resultado obtenido.

7. Presentación de los resultados.

Contenido de los capítulos.

En el **Capítulo 1** se desarrolla el marco conceptual de la investigación, se presentan los principales conceptos de minería de textos, pre procesamiento, extracción de información, creación de ontologías se aclara el uso indistinto de estos conceptos por varios autores. Se abordan los principales objetivos de la extracción de información y el pre procesamiento para la construcción de ontologías, definiendo el objetivo en que se enmarca el pre procesamiento. Se define este último así como algunas variantes importantes que se han presentado para resolver ciertos problemas específicos.

El **Capítulo 2**, a través de las resoluciones 320/11 y 321/11 existentes en la Ley de regulación Aduanal de la República de Cuba, se describe todo el proceso que conllevará a la propuesta de solución esclareciendo los principales inconvenientes que se pretenden resolver con la investigación. Se proponen los ajustes a los algoritmos de pre procesamiento con los que se pretende mejorar la complejidad computacional de este proceso. Se presenta la solución implementada definiendo las entradas correspondientes al algoritmo, los procesos básicos del mismo así como sus salidas.

El **Capítulo 3**, presenta los diferentes mecanismos utilizados para validar los resultados de una investigación. Entre ellos se encuentran los experimentos, modelos matemáticos, el razonamiento lógico, y la demostración. Este último es el seleccionado para validar la solución propuesta en esta investigación por adaptarse a las características de la misma. Se describen los recursos computacionales utilizados para desarrollar la demostración. Se incluye además la descripción de todos los elementos que conforman el dominio de la demostración. Por último se presentan los resultados obtenidos y se realiza una detallada discusión de los mismos, demostrando la validez de la presente investigación.

CAPÍTULO 1. **FUNDAMENTACIÓN DEL TEMA**

1.1. Sistema de Gobierno Electrónico

El aumento de información y la variedad de servicios que brindan las Administraciones Públicas y las nuevas posibilidades que nos ofrecen las Tecnologías de la Información, como nuevas vías de interactuar con organizaciones y personas han posibilitado el surgimiento del Gobierno Electrónico (e-government). Los Gobiernos Electrónicos representan la rápida difusión de las TIC asociadas a la agenda de reforma en la gestión de las Administraciones Públicas en la actualidad (1). Esta definición es un fenómeno relativamente reciente que no ha sido claramente definido aún.

Dichos Gobiernos no hacen referencias solamente a información digitalizada y en línea para el ciudadano, sino que tienen múltiples dimensiones e implican una serie de hitos a lograr, entre los que se destacan:

- La integración de la información y de la comunicación intergubernamental.
- La promoción del desarrollo económico.
- La Democracia Electrónica (e-democracy).
- Las Comunidades Electrónicas (e-communities).
- La política ambiental.

Los Gobiernos electrónicos surgen con el objetivo de alcanzar la eficacia y eficiencia de la gestión pública e incrementar sustantivamente la transparencia del sector público y la participación de los ciudadanos. Además de ser una propuesta rápida y de fácil acceso debido a la comodidad de consulta y atención en línea que proponen a sus usuarios, economizando así el tiempo de gestión y respuesta en el proceso legal. Todo ello, sin perjuicio de las denominaciones establecidas en las legislaciones nacionales. (2)

1.2. Ontología en Sistemas Legales de Gobierno Electrónico

En la actualidad el desarrollo de ontologías ha salido de los centros de investigación y se ha convertido en patrimonio de los expertos en los dominios más diversos, en el marco de muchas disciplinas se desarrollan ontologías estandarizadas. Las mismas juegan un rol significativo en la recuperación de información y en los procesos que comparten, reutilizan y adquieren conocimiento. Las ontologías hacen posible compartir una comprensión común de la estructura de la información entre personas o agentes de software, rehusar y

analizar el conocimiento de dominio, así como separarlo del conocimiento operacional y hacer explícitos presupuestos de dominios.

Una ontología es una descripción formal explícita de conceptos en un dominio de discurso - las clases - las propiedades de cada concepto que describen las funcionalidades y los atributos del concepto – los slots, algunas veces llamados roles o propiedades - y las restricciones sobre los slots - las facetas, algunas veces llamadas restricciones sobre los roles. Para desarrollarlas es necesario definir las clases necesarias y organizar las mismas en una taxonomía jerárquica estableciendo las relaciones subclase-superclase; definir los "*slots*" y describir los valores permitidos para las instancias.

Las ontologías estudian las formas existentes y la relación entre ellas, por tanto permiten actuar sobre las carencias semánticas que hoy hacen difícil y dispendioso el acceso a la información; generalmente se usan para especificar y comunicar el conocimiento del dominio de una manera genérica y son muy útiles para estructurar y definir el significado de los términos. (3)

1.3. Complejidad de la construcción de ontologías

La creación de ontologías es una labor enormemente compleja y costosa en recursos técnicos, humanos y en tiempo. El principal problema con la construcción de ontologías es el esfuerzo que conlleva. Los intentos de generar ontologías de forma automática o semi-automática han dado siempre escasos frutos, imponiéndose la realización manual de las mismas. Actualmente existen una serie de herramientas que facilitan la construcción manual de las ontologías. Estas herramientas se encargan de comprobar la consistencia de la información introducida y de generar su código necesario a partir de la información que el operador introduce mediante la interfaz gráfica. De todos modos, se trata de una ardua labor, porque supone la estructuración del conocimiento humano partiendo no desde un dominio determinado sino desde el nivel más alto del conocimiento.

Algunas de las dificultades a superar para la construcción de ontologías son las siguientes:

- La Objeción de la Redundancia: es posible resolver ambigüedades semánticas mediante mecanismos no semánticos (sintácticos, morfológicos, etc.). Esta suposición ha llevado a la implementación de sistemas donde el reconocimiento ha suplantado a la interpretación, por lo que no consigue ninguna representación del significado del texto.

- La Objeción de la Dependencia de la Lengua: es quizá la objeción más comúnmente citada y se refiere al hecho de que una ontología reflejará la concepción lingüística de sus creadores y por tanto su estructura y contenidos estarán influenciados por la concepción de una lengua determinada, con lo que se revoca la afirmación de que una ontología es independiente de la lengua.
- La Objeción de la Implementación: las dificultades y costes que supone la creación de una ontología no compensan sus beneficios. (4)

La más seria de estas objeciones, critica la inexistencia de principios subyacentes a la creación de ontologías. El problema básico, bien conocido por los investigadores en el campo, es la imposibilidad de exponer un procedimiento algorítmico para la adquisición de ontologías, un procedimiento que marque exactamente la inclusión o no de un determinado concepto, así como sus propiedades y conexiones con otros conceptos. Un problema añadido es la inexistencia de una metodología definida para la creación y el mantenimiento de ontologías, es decir, no existe un algoritmo que permita la adquisición de conceptos. Lo que sí existen son una serie de líneas maestras desarrolladas a partir de la experiencia acumulada por los investigadores en el campo. Una ontología se crea desde cero o se adquiere de forma incremental, a través de una interacción continua con otras fuentes de conocimiento.

1.4. Aprendizaje de ontologías

Las ontologías pueden ser aprendidas de diversas fuentes, ya sea de bases de datos, estructurada y no estructurada, documentos o preliminares, incluso ya existentes, como diccionarios, taxonomías y directorios. La atención se centra en la adquisición de ontologías a partir de un texto no estructurado, la mayoría de los enfoques utilizan sólo los nombres como punto de partida para la construcción de las ontologías y desprecian las relaciones ontológicas entre las clases de palabras y otros. Las técnicas de aprendizaje de ontología se pueden dividir en la construcción de ontologías a partir de cero y en la extensión de ontologías existentes. El primero se compone en su mayoría por los métodos de agrupamiento y la última es una tarea de clasificación, que es la distinción entre métodos supervisados y métodos sin supervisión, aunque algunos de los enfoques de agrupamiento implican supervisión en los pasos intermedios.

En la agrupación jerárquica, los conjuntos de términos están organizados en una jerarquía que puede ser transformada directamente en una ontología basada en prototipos. Para la agrupación en los documentos no estructurados, a una distancia medida, en los términos tiene que ser definido qué servirá como criterio para la fusión de los términos o grupos de términos. La misma medida se puede utilizar para calcular los

ejemplos más típicos de un concepto como los más cercanos al centro de gravedad (el hipotético "medio" en la instancia de un conjunto). Para garantizar el éxito de esta metodología es fundamental la selección de una medida apropiada de distancia semántica y un algoritmo de agrupamiento adecuado. Una visión general de los métodos de agrupamiento para la obtención de ontologías a partir de diferentes fuentes incluyendo textos libres, se encuentran los tipos de métodos de agrupamiento ya sean aglomerativo o divisivo se pueden aplicar a todo tipo de representaciones, ya sea de espacio vectorial, las redes asociativas o a la teoría de conjuntos.

Dada una ontología existente, su extensión puede ser vista como una tarea de clasificación, las características de los datos existentes se utilizan como un conjunto de entrenamiento para el aprendizaje de la máquina, que produce un clasificador para los casos previamente desconocidos. Una posibilidad es utilizar la estructura jerárquica en un árbol de decisión.

Al introducir nuevos conceptos, se comprueba si encajan mejor para el nodo actual o uno de los nodos hijos. El árbol se recorre de arriba hacia abajo, desde la raíz hasta la posición adecuada. El mayor problema aquí es la naturaleza general de los conceptos de nivel superior que lleva a tomar el camino equivocado en el comienzo del proceso, esto puede eliminarse mediante la propagación de las firmas de nivel inferior de los conceptos. Un pequeño sub-árbol de una ontología ya existente y su jerarquía se utiliza como la formación y los datos de prueba. Se propagan las descripciones semánticas iterativamente hacia la raíz, y se nivela el enfoque hasta poner nuevas palabras en los sub-árboles más grandes. (5)

1.5. Minería de texto para el aprendizaje de ontologías

La Minería de Texto, también conocida como minería de información en documentos (document information mining), minería de datos texto, o descubrimiento de conocimiento en bases de datos textuales es una fusión de tecnologías para analizar colecciones grandes de documentos sin estructura con el propósito de extraer modelos (patrones) interesantes y no triviales o conocimiento. El objetivo es extraer la información hasta el momento no descubierta de colecciones grandes de texto. También se ha definido como: “La extracción no trivial de información implícita, previamente desconocida y potencialmente útil de grandes cantidades de datos texto”. (6)

La categorización de documentos de texto es una aplicación de la minería de texto que asigna a los documentos una o más categorías, etiquetas o clases, basadas en el contenido. Es un componente importante de muchas tareas de organización y gestión de la información. (7)

Los sistemas inteligentes de acceso a la información están integrando de manera creciente técnicas de minería de texto y de análisis del contenido, y recursos semánticos como las ontologías. Las técnicas de minería de texto permiten explorar y extraer conocimiento de colecciones de documentos textuales. Los tres problemas básicos que pueden abordarse con técnicas de minería de texto son:

- Recuperación de información relevante, es decir, extraer de manera automática aquellos documentos que puedan resultar interesantes para el usuario a partir de una consulta realizada por éste.
- Categorización de documentos, consiste en asignar a cada documento una o varias categorías temáticas de un conjunto de categorías preestablecidas.
- Agrupamiento, consiste en la generación automática de grupos de documentos relacionados, por ejemplo, documentos que traten un mismo tema o asunto. A diferencia de lo que ocurre en la categorización, en los procesos de agrupamiento no existe un conjunto de categorías preestablecido, sino que el propio algoritmo a utilizar debe generar automáticamente esas categorías, contribuyendo de esta forma a generar un nuevo conocimiento. (8)

Tanto la categorización como el agrupamiento pueden verse como un proceso de clasificación, en el primer caso se habla de clasificación supervisada mientras que en el segundo se utiliza el concepto de clasificación no supervisada. El objetivo principal de la clasificación de documentos, como concepto general, es reducir la diversidad de datos y la sobrecarga de información mediante el agrupamiento de documentos similares. Con respecto a la gestión del conocimiento, la clasificación de documentos puede ser vista como una herramienta que permite simplificar el acceso y procesamiento del conocimiento explícito, facilitando la recuperación, organización, visualización, desarrollo e intercambio de conocimientos.

Un primer aspecto a la hora de afrontar la integración de técnicas de minería de texto en el aprendizaje de ontologías es la necesidad de disponer de modelos de representación de documentos que permitan la aplicación de técnicas numéricas sobre ellos. El modelo vectorial permite representar los documentos a partir de un vector de pesos asociados a una serie de rasgos seleccionados del documento. Habitualmente estos rasgos se obtienen a partir de las palabras presentes en el texto tras realizar diferentes operaciones

de filtrado, eliminación de palabras con poco valor discriminante y transformaciones morfológicas, reduciendo el tamaño del diccionario total de palabras utilizado. En el modelo vectorial cada rasgo, representa una dimensión del espacio. En el caso de colecciones de documentos escritos en diferentes idiomas resulta interesante la utilización de diferentes recursos lingüísticos (glosarios, tesauros, taxonomías) para representar los documentos mediante rasgos independientes del idioma y de esta forma poder aplicar las técnicas de minería de texto con independencia del idioma en el que se encuentren escritos los documentos.

Con el objeto de facilitar las labores de extracción de conocimiento en repositorios documentales se han venido confeccionando diferentes elementos de naturaleza lingüística que tratan de representar el conocimiento compartido y común sobre áreas temáticas específicas. Entre ellos destacan los glosarios especializados, taxonomías, tesauros y ontologías.

- Se podría definir un glosario como un repertorio de términos pertenecientes a un área de conocimiento o disciplina, añadiendo por lo general definiciones o explicaciones necesarias para su comprensión.
- La taxonomía, entendida como ciencia, se ocupa de los principios, métodos y fines de la clasificación. Desde el punto de vista de la lingüística computacional, se puede ver una taxonomía como una lista estructurada en árbol, organizada jerárquicamente desde los términos más generales hasta los términos más específicos.
- Los tesauros, entendidos de alguna manera como “taxonomías con extras”, facilitan un almacenamiento adecuado de la información, así como implementan un nexo de comunicación, identidad conceptual o interface entre el lenguaje natural y el lenguaje en que se hayan escrito los documentos contenidos en un sistema de gestión documental. Básicamente los tesauros son listas estructuradas que pretenden representar de forma unívoca el contenido conceptual de los documentos asociados a un área temática determinada y que pueden ser fácilmente integrados en los sistemas de gestión documental. (9)

1.6. Pre-procesamiento de texto

En minería de texto, como en minería de datos, se presentan algunos problemas iniciales antes de que el trabajo de análisis de la información pueda empezarse. Primeramente, se necesita algún pre-procesamiento de los datos. La idea principal es transformar los datos de tal forma que puedan ser procesados, podría ser

eliminando las imágenes, tablas o formaciones complejas de texto, o podría consistir en el reemplazo de símbolos matemáticos, números, URL (Localizador de Recursos Uniforme), y direcciones de correo electrónico por símbolos especiales. Si se usan las técnicas OCR (Reconocimiento Óptico de Caracteres) el paso de pre-procesamiento puede consistir en la revisión ortográfica (spelling checking). Además el pre procesamiento podría ser también un proceso de reconocimiento del idioma. En idiomas como el alemán, finlandés, ruso o español podría necesitarse la tokenización del documento y además el uso de la lematización (un método ampliamente usado en procesamiento del lenguaje natural para agrupar diferentes palabras que tengan una raíz común). La información del idioma actual de los textos es muy útil, debido a que el procesamiento de una colección monolingüe es más directo que el procesamiento de una colección de idioma cruzado o multilingüe. (10)

1.7. Técnicas de pre procesamiento

Esta dimensión responde a la pregunta: "¿Existe algún pre-procesamiento para convertir documentos en la entrada adecuada para los algoritmos de minería de textos?". El pre-procesamiento más popular utilizado en el aprendizaje de ontologías a partir de textos es el pre-procesamiento lingüístico. Esto responde a que la comprensión completa de los documentos proporcionaría las relaciones específicas entre los conceptos, mientras que las técnicas de pre procesamiento podrían proporcionar el conocimiento genérico acerca de los conceptos. Dicho conocimiento genérico por lo general disminuye la velocidad del proceso de construcción de la ontología, la mayoría de los sistemas existentes utilizan técnicas de procesamiento de texto de poca profundidad como tokenización, que forma parte del etiquetado y del análisis sintáctico, para extraer las estructuras esenciales de los textos de entrada.

Por ejemplo, la técnica de Text-To sobre el uso de dichos métodos de procesamiento de texto desarrollados en las PYME (Sistema de Extracción de Saarbrücken) para procesar textos en alemán e identificar los pares lingüísticos relacionados con las palabras, las cuales se asignan a los conceptos que utilizan el léxico de dominio. InfoSleuth utiliza un simple etiquetador para llevar a cabo el análisis sintáctico superficial y ASIUM utiliza Sylex para procesar textos en francés. Otras herramienta como SYNDIKATE utiliza técnicas más avanzadas como la lematización para extraer el conocimiento ontológico del texto y Hasti es un sistema de procesamiento de texto persa para extraer estructuras de frases, que indican los roles temáticos, a partir del texto.

Existen también otros módulos de pre procesamiento para la extracción de estructuras especiales donde el aprendizaje de módulos es capaz de aprender elementos ontológicos como las categorías y la extracción de la terminología para ayudar a un experto a construir una ontología en cualquier campo existente. (8)

1.8. Conclusiones Parciales.

El presente capítulo recoge los principales conceptos asociados a Gobierno electrónico, a la construcción de ontologías y a las técnicas de pre procesamiento en la minería de texto como etapa de mayor volumen e importancia dentro del proceso de extracción de conocimiento, específicamente centra la atención en el estudio de las técnicas de pre procesamiento en la construcción de ontologías legales. La extracción de información mediante el pre procesamiento de texto puede ser utilizada para la construcción de ontologías mediante los algoritmos de minería de texto. Se definen algunas técnicas para la estructuración de los documentos, concluyendo que las técnicas de pre procesamientos más utilizadas son la tokenización y la lematización, aunque los resultados obtenidos con el uso de estas son subjetivos pues están condicionados por el intelecto humano.

CAPÍTULO 2. **ANÁLISIS Y DISEÑO DEL ALGORITMO**

2.1. Introducción

La construcción de ontologías legales es una tarea ardua de realizar manualmente debido a que la extracción de conceptos y las relaciones existentes entre ellas se ve dificultada por la falta de especialistas y por el gran tamaño del corpus legal existente en la Gaceta Oficial de la República de Cuba, estos elementos hacen que el trabajo de construcción sea prácticamente interminable.

Para agilizar el engorroso proceso de construcción de ontologías se ha propuesto el uso de mecanismo de aprendizaje automatizado que use técnicas de minería de texto, pero para lograr construir una ontología de forma automatizada se deben preparar los documentos de entrada mediantes técnicas de pre procesamiento de minería de textos pues los algoritmos de minado de texto no pueden trabajar los documentos en su forma original debido a sus dificultades para procesar el lenguaje natural sin restricciones, ya una vez que se le apliquen técnicas de pre procesamiento de minería de textos se facilitará la extracción de la información necesaria para la construcción de ontologías legales.

Con vistas a clarificar la magnitud del problema planteado se presenta un ejemplo de la tarea que favorecerá un mejor entendimiento de la necesidad de pre procesar los documentos legales cuando van a ser utilizados en la construcción automática de ontologías legales.

La extracción de la información se realizó manualmente, vale aclarar que parte de la información en la Ley de Regulación Aduanal, la escogida para desarrollar el ejemplo, está en tablas lo que facilita su entendimiento, esto no ocurre en la mayoría del corpus legal.

Considerando que no es posible incluir el universo de artículos y productos objeto de importación sin carácter comercial, se recoge una gama limitada de estos atendiendo a su uso, empleo y funciones propias evitando con ello innumerables denominaciones específicas que simplifiquen la clasificación como lo mostrado en la siguientes tablas. Vea (Tabla 1).

ARTÍCULOS	CLASIFICACIÓN
Puré de tomate en conserva	Capítulo 01.1 Preparaciones Alimenticias. Capítulo 01.2 Conservas de todo tipo.
Atún, sardinas en conserva	
Jugos, frutas, mermeladas, salsas en conserva	
Binoculares	Capítulo 04.10 Aparatos de medición y precisión.
Reguladores de gas	
Multímetros, volt amperímetros, etc.	
Cafetera eléctrica	Capítulo 10.21 Los demás equipos electrodomésticos de cocina y del hogar.
Olla arrocera, procesador de alimentos	
Batidoras, etc.	
Ojetes para calzado	Capítulo 06.4 Partes y accesorios empleados en talabartería y calzado.
otros remaches, aros	
Broches, suelas y pieles	
Cuerdas, llaves para guitarras	Capítulo 12.06 Partes piezas y accesorios
Parches para tambores	
Accesorios de instrumentos musicales	
Cámaras para computadoras	Capítulo 10.43 Partes, piezas y accesorios de computadoras.
Impresoras, fax	
Demás periféricos de computadoras	
Material para ponche	Capítulo 14.35 Los demás accesorios y piezas de vehículos y motos.

Tabla 1. Clasificación de los artículos y su ubicación en los capítulos que corresponden. (11)

La tabla anterior permite agilizar la ubicación y la extracción de información en la Ley de Regulación Aduanal para la construcción de su ontología.

A continuación mostramos cinco juegos de tablas donde son ubicados los artículos de carácter no comercial dependiendo de su clasificación y uso, la tabla superior pertenece a la resolución 320/11 y la tabla posterior a la resolución 321/11 de la Ley de Regulación Aduanal respectivamente donde cada juego de tabla contiene la información de un capitulo especifico de dicha resolución, para un mejor entendimiento vea los juegos de tablas mencionados. De la tabla (2 a la 6).

CAPÍTULO 01 - COMESTIBLES, BEBIDAS, TABACOS Y CIGARROS			
Artículos	U/M	Cantidad	Observaciones
1.Preparaciones alimenticias (alimentos deshidratados)	U	30*	*la cantidad se refiere al total de los artículos, siempre que la cantidad total de todas las unidades no exceda de 5 kg.
2.Conservas de todo tipo	U	50*	*la cantidad se refiere al total de los artículos, siempre que la cantidad total de todas las unidades no exceda de 50 kg.
3.Productos lácteos cualquier tipo	U	30*	*la cantidad se refiere al total de los artículos, siempre que la cantidad total de todas las unidades no exceda de 15 kg.
4.Mantequillas, quesos	kg	5*	*hasta el límite de 5 kg de cada artículo contenido en la descripción.
5.Miel natural	L	5	

CAPÍTULO 01 - COMESTIBLES, BEBIDAS, TABACOS Y CIGARROS			
Artículos	U/M	Valor	Observaciones
1.Preparaciones alimenticias (alimentos deshidratados)	U	1.50	
2.Conservas de todo tipo	U	1.50	
3.Productos lácteos cualquier tipo	U	1.00	
4.Mantequillas, quesos	kg	0.50	
5.Miel natural	L	0.50	

Tabla 2. Muestra de los elementos y sus atributos relacionados en el Capítulo 1 “Comestibles, bebidas, tabacos y cigarros”. (11)

CAPÍTULO 02 - COSMÉTICOS, PERFUMERÍA Y ARTÍCULOS DE LIMPIEZA			
Artículos	U/M	Cantidad	Observaciones
1. Perfumes	U	10*	*la cantidad total del contenido de todas las unidades no exceda de 1 litro.
2. Agua de tocador (agua de colonia, de toilet, de Lavanda, Portugal, etc.)	U	10*	*la cantidad total del contenido de todas las unidades no exceda de 10 litro.
3. Preparaciones capilares	U	10*	*la cantidad total del contenido de todas las unidades no exceda de 10 litro.
4. Preparaciones para maquillaje y cuidado de la piel, cremas, etc.	U	48*	*la cantidad se refiere al total de los artículos.
5. Preparaciones para higiene bucal	U	15*	*la cantidad total del contenido de todas las unidades no exceda de 1 litro.

CAPÍTULO 02 - COSMÉTICOS, PERFUMERÍA Y ARTÍCULOS DE LIMPIEZA			
Artículos	U/M	Valor	Observaciones
1. Perfumes	U	5.00	
2. Agua de tocador (agua de colonia, de toilet, de Lavanda, Portugal, etc.)	U	1.00	
3. Preparaciones capilares	U	1.00	
4. Preparaciones para maquillaje y cuidado de la piel, cremas, etc.	U	3.00	
5. Preparaciones para higiene bucal	U	1.00	

Tabla 3. Muestra de los elementos y sus atributos relacionados en el Capítulo 2 “Cosméticos, perfumería y artículos de limpieza”. (11)

CAPÍTULO 03 - PRODUCTOS FOTOGRÁFICOS Y CINEMATOGRÁFICOS			
Artículos	U/M	Cantidad	Observaciones
1. Cámaras fotográficas digitales	U	2	
2. Cámaras de vídeo	U	2	
3. Memorias de estas cámaras	U	10	
4. Películas fotográficas en rollos sensibilizados para fotografiar	U	20	
5. Datashow y similares	U	2	

CAPÍTULO 03 - PRODUCTOS FOTOGRÁFICOS Y CINEMATOGRÁFICOS			
Artículos	U/M	Valor	Observaciones
1. Cámaras fotográficas digitales	U	60.00	
2. Cámaras de vídeo	U	100.00	
3. Memorias de estas cámaras	U	1.00	
4. Películas fotográficas en rollos sensibilizados para fotografiar	U	3.00	
5. Datashow y similares	U	1.00	

Tabla 4. Muestra de los elementos y sus atributos relacionados en el Capítulo 3 “Productos fotográficos y cinematográficos”. (11)

CAPÍTULO 04 - PINTURAS, BARNICES, ARTÍCULOS DE FERRETERÍA Y HERRAMIENTAS

Artículos	U/M	Cantidad	Observaciones
1. Pinturas y barnices	L	20*	*La cantidad se refiere al total de los artículos.
2. Pigmentos y colorantes	U	10*	*La cantidad se refiere al total de los artículos.
3. Velas	U	30	
4. Pastas para modelar	kg	5*	*La cantidad se refiere al total de los artículos.
5. Juego de herramientas de mano	U	2	

CAPÍTULO 04 - PINTURAS, BARNICES, ARTÍCULOS DE FERRETERÍA Y HERRAMIENTAS

Artículos	U/M	Valor	Observaciones
1. Pinturas y barnices	L	2.00	
2. Pigmentos y colorantes	U	1.00	
3. Velas	U	0.50	
4. Pastas para modelar	kg	0.50	
5. Juego de herramientas de mano	U	7.00	

Tabla 5. Muestra de los elementos y sus atributos relacionados en el Capítulo 4 “Pinturas, barnices, artículos de ferretería y herramientas”. (11)

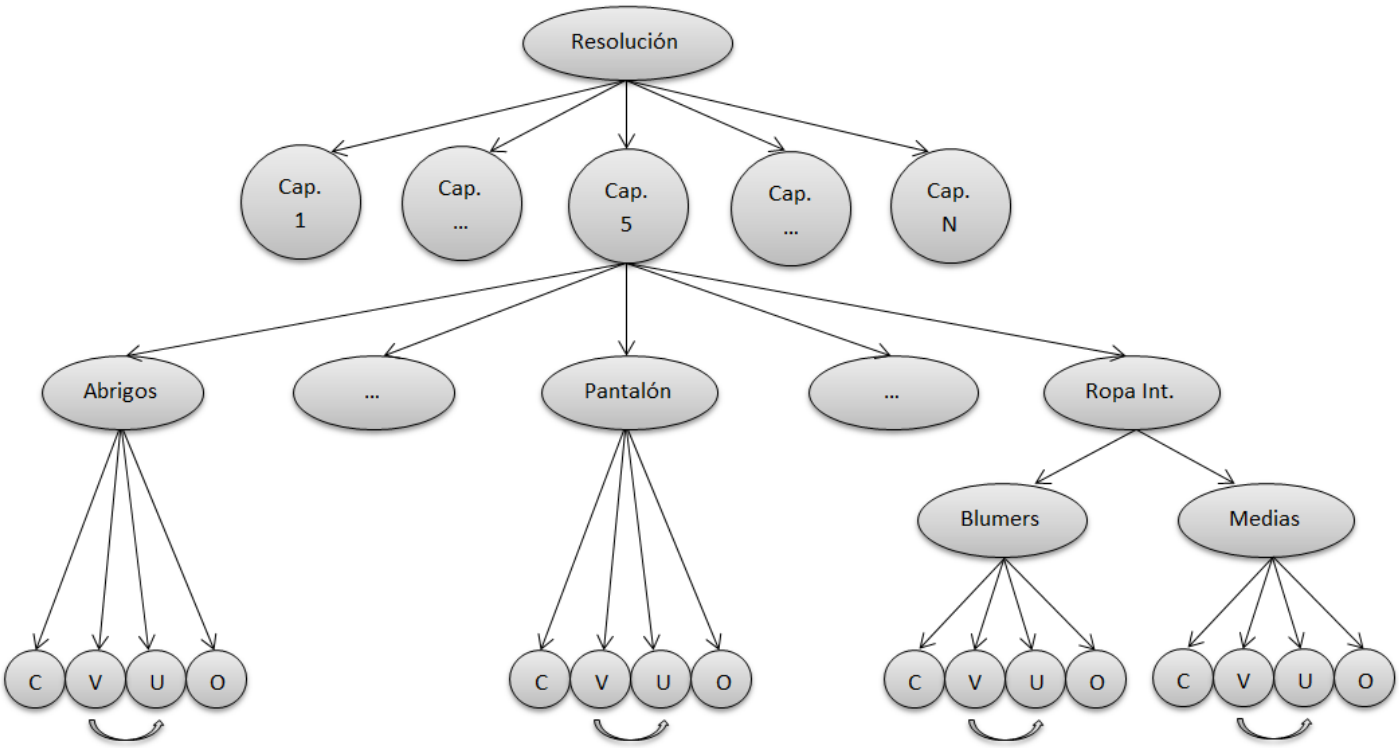
CAPÍTULO 05 - CONFECCIONES			
Artículos	U/M	Cantidad	Observaciones
1. Abrigos, sobretodos, chaquetas, sacos, chalecos	U	5	
2. Ropa interior femenina (adulto, joven, niña)			
- Blúmeres, Ajustadores, Medias	Docena	4	
3. Vestuario masculino (adulto, joven, niño)			
- Camisas, Pullover, Pantalones	U	40*	*la cantidad se refiere al total de los artículos.

CAPÍTULO 05 - CONFECCIONES			
Artículos	U/M	Valor	Observaciones
1. Abrigos, sobretodos, chaquetas, sacos, chalecos	U	6.00	
2. Ropa interior femenina (adulto, joven, niña)			
- Blúmeres, Ajustadores, Medias	Docena	6.00	
3. Vestuario masculino (adulto, joven, niño)			
- Camisas, Pullover, Pantalones	U	5.00	

Tabla 6. Muestra de los elementos y sus atributos relacionados en el Capítulo 5 “Confecciones”. (11)

La información contenida en los juegos de tablas presentados anteriormente arrojan la información necesaria para comenzar la construcción de una ontología para los capítulos a las que pertenecen, en ellas se regulan la cantidad de artículos permitidos y su valor aduanal.

La presente ontología fue realizada a través de la información obtenida de las tablas pertenecientes a las resoluciones No. 320/11 y No. 321/11, esta ontología legal es solo para el capítulo 5 que aborda sobre las Confecciones de vestir en las personas, debido a que es muy complicado obtener la relevancia de las palabras y sus relaciones de forma manual, para eso se necesita realizar un pre procesamiento automatizado.



Figura

1. Propuesta de una Ontología para el Capítulo 5 “Confecciones”.

2.2. Propuesta de Solución

La necesidad de tener un identificador único en cada palabra para el reconocimiento y el pre procesamiento en la minería de texto es fundamental, la relevancia de tener un token para cada palabra es que actúa como identificador en grandes cantidades de textos, facilitando la localización y el manejo de dicha palabra a la hora de aplicar técnicas de minería de texto.

Por lo anteriormente explicado se propone aplicar como técnicas de pre procesamiento para los documentos legales la tokenización y lematización, el criterio para la selección de estas técnicas para el cumplimiento de los objetivos planteados fue la rapidez y el correcto procesamiento que proponen en el manejo de grandes cantidades de texto, con lo que se propone un algoritmo para cada técnica, para lograr la correcta estructuración y limpieza de los documentos legales para la construcción de ontologías. Luego de seleccionadas las técnicas a aplicar, el primer paso del análisis morfológico es la tokenización, cuyo objetivo

es la exploración de las palabras en un documento. La tokenización que se propone asigna a cada palabra un tipo de token *(tipo_token)* y un *identificador de* token *(token _id*) en el orden de aparición en los documentos de entrada del algoritmo, así como a cada carácter distinto de una letra como los signos de puntuación, números, etc., almacenándolos en un lista con la siguiente estructura ***<tipo_token, "palabra o carácter", token_id>***.

El siguiente paso llamado lematización utiliza como principal identificador los token de cada palabra siendo así una forma ágil y eficaz de manejar las palabras existentes en los documentos legales, la lematización es una técnica para la reducción de las palabras en su raíz. El algoritmo propuesto para realizar esta técnica propone la reducción a su raíz de todas las palabras recibidas en la lista de tokens donde *(tipo_token)* sea igual a ***id*** e incrementar en 1 el contador de dicha raíz por cada ocurrencia de la palabra en la lista recibida. El resto de los caracteres no se tienen en cuenta pues no son relevantes para establecer las relaciones existentes entre los conceptos. Este método, es llamado el método de la *fuerza bruta*, uno de los más usados y simples. A pesar del costo computacional, las soluciones de fuerza bruta se utilizan para las lenguas de gran complejidad gramatical, idiomas como las lenguas romances (francés, español, italiano, portugués, etc.) que tienen inflexiones diferentes para la raíz de una palabra en dependencia de la conjugación entre la persona y el tiempo verbal. (6)

El algoritmo de lematización estándar, con fuerza bruta, requiere de un diccionario que contiene las inflexiones de las palabras, el diccionario se utiliza como mecanismo de reducción de elementos partiendo de que el universo de palabras posibles es el que está contenido en el diccionario. (10) Este trabajo propone una mejora a la utilización de diccionarios, sustituyendo los mismos por un grupo de reglas, las cuales definen los sufijos existentes en el idioma español pertenecientes a los morfemas de cada palabra, se determinaron cuatro reglas que clasifican los tipos de sufijos:

- Sufijos estándares.
- Sufijos especiales.
- Sufijos residuales.
- Terminaciones léxicas.

Estas reglas proponen eliminar o remplazar los sufijos de cada palabra mediante cuatro pasos para llegar a las raíces correctas y así no perder información necesaria para la construcción de las ontologías, estas

reglas permiten llevar el control de la variable ocurrencia perteneciente a cada raíz, para determinar la relevación de cada palabra en los documentos.

El algoritmo devuelve un vector espacial organizado de forma descendente teniendo en cuanta la variable de ocurrencia de cada raíz **"cant_ocurrencia"**, el vector presenta la siguiente estructura **<token_id, "raiz", cant_ocurrencia>**.

A continuación se muestra como es generada la lista de token y el vector espacial mediante el pre procesamiento propuesto en la solución. Vea figura (2 y 3).

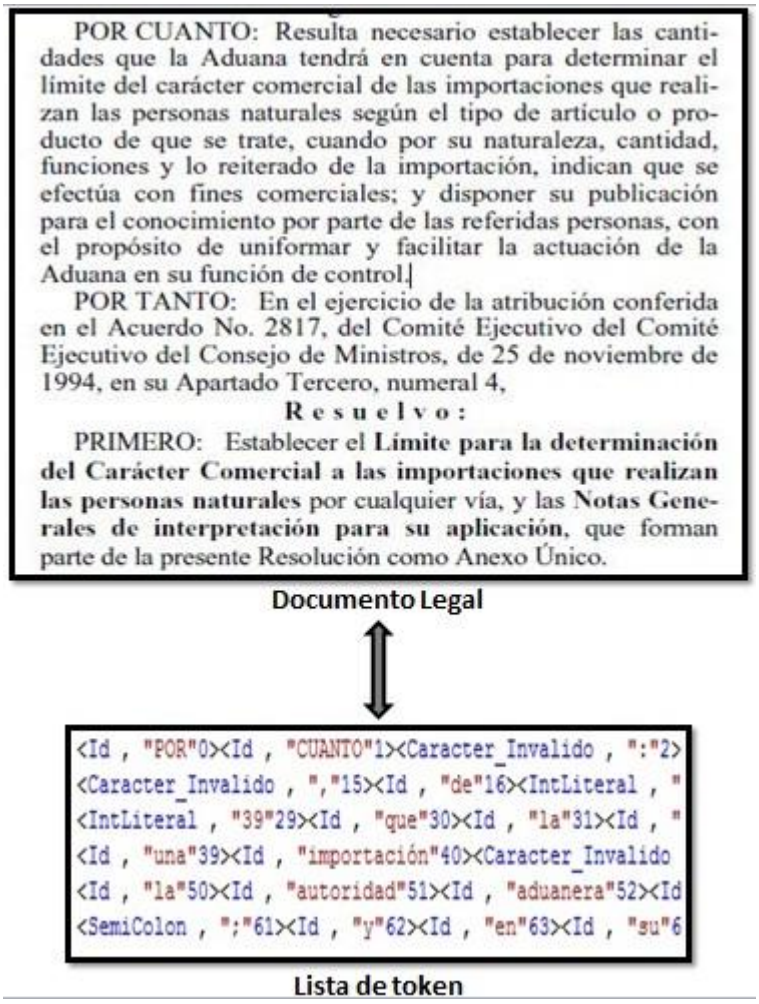


Figura 2. Lista generada por el algoritmo de Tokenización.

Una vez tokenizado el documento recibido como entrada y teniendo la lista de token se procede a lematizar las palabras almacenadas en la lista de token por el algoritmo de lematización.

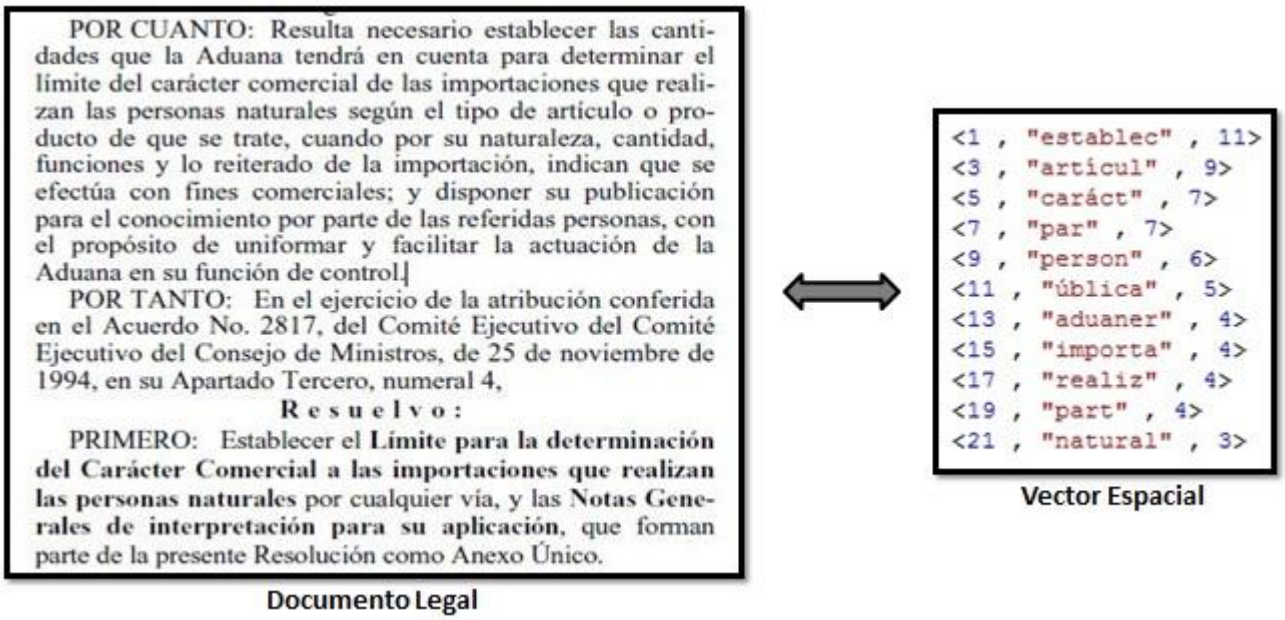


Figura 3. Vector generado por el algoritmo de Lematización.

2.3. Propuesta de la Solución

Se proponen dos algoritmos principales para el pre procesamiento de los documentos legales, el primero ***Tokenizar*** se encarga de analizar cada uno de los caracteres que obtiene llamando al algoritmo auxiliar ***leer_caracter***, clasificándolos en letras, números o caracteres especiales, en esta última clasificación se incluyen los operadores matemáticos, los signos de puntuación y los desplazamiento de línea y palabra. Finalmente estos caracteres se agrupan hasta formar varios tokens que posteriormente son almacenados en el orden de aparición en una lista creada para este fin. La secuencia de ejecución de este algoritmo está dirigida a obtener la lista de token. De manera general el algoritmo ***Tokenizar*** quedaría de la siguiente forma:

Tokenizar

Entrada: T

Salida: Lista de token.

Algoritmo:

1. Obtiene un carácter de la lista de entrada.

- Clasifica el carácter obtenido en letra número o carácter especial.
- De ser una letra o un número reinicia el método con el próximo carácter repitiendo este paso hasta encontrar un carácter especial.
- En caso de encontrar un carácter especial crea un token con los caracteres anteriores a este y otro token con este carácter.
- Clasifica los tokens según su contenido.

2. Almacena los token creados en una lista.

El algoritmo de tokenización recibe como entrada un texto (T), compuesto por palabras y caracteres, donde los caracteres pueden ser las letras que componen a cada palabra o ser signos de puntuación, números etc.

La solución cuenta con una estructura **Token** que contiene las propiedades de los token tales como su tipo almacenado en un enumerativo, y cada palabra del documento de entrada y su posición.

El algoritmo encargado de recorrer los caracteres existentes en los documentos de entrada es ***leer_caracter***, a continuación se describe el funcionamiento de este algoritmo**.**

El Algoritmo leer_caracter:

leer_caracter

Entrada: C

Salida: Lista de caracteres.

Algoritmo:

1. Obtiene un carácter de la lista de entrada.

- Lee el carácter obtenido sea letra, número o carácter especial.
- De ser una letra o un número reinicia el método con el próximo carácter. repitiendo este paso hasta encontrar un carácter especial.
- En caso de encontrar un carácter especial hace un salto de palabra o de línea.

Este algoritmo recibe como parámetro las cadenas de textos que contienen los documentos de entrada, donde va formando la palabra con los caracteres que va leyendo mientras tanto no exista un espacio o un salto de página, devolviendo finalmente una palabra o caracteres especiales, existentes en el documento.

El algoritmo de lematización recibe como entrada una lista de token, estructurada de la siguiente manera ***<tipo_token, "palabra o carácter", token_id>***. Este algoritmo solo toma las palabras en las cuales el tipo de token es ***id*****,** obviando los demás tipos de token como los pertenecientes a los operadores matemáticos, los signos de puntuación, guiones etc. La secuencia de ejecución de este algoritmo está dirigida a generar el vector espacial.

Se definieron además seis algoritmos que responden a sub procesos utilizados por el algoritmo ***Lematizar***. A continuación se presentan de manera detallada.

La mayoría de los lematizadores hacen uso de al menos una de las regiones en las que se puede dividir una palabra R1, R2 o RV.

- R1: Es la región después de la primera no-vocal después de una vocal, o es la región nulo al final de la palabra si no hay tal falta de vocal.
- R2: Es la región después de la primera no-vocal después de una vocal en R1, o es la región nulo al final de la palabra si no hay tal falta de vocal.
- RV: Si la segunda letra es una consonante, RV es la región después de la vocal siguiente, o si las dos primeras letras son vocales, RV es la región después de la consonante siguiente, y de otro tipo (consonante-vocal caso) RV es la región después de la tercera letra. RV es el final de la palabra, si estas posiciones no se puede encontrar.

El algoritmo ***Lematizar*** propuesto hace uso de esas regiones en distintos momentos y son definidas en tres variables que facilitan la lematización.

Para la construcción de estas tres regiones se utilizaron tres algoritmos donde son utilizados para recorrer e identificar cada vocal y cada consonante en las palabras, fundamentalmente identificando las posiciones siguientes a cada letra permitiendo así delimitar las regiones antes mencionadas, los algoritmos utilizados fueron.

El algoritmo es_vocal.

Es_vocal

Entrada: C

Salida: Verdadero o falso en dependencia del carácter.

Algoritmo:

1. Obtiene un carácter de la lista de entrada.

- Lee el carácter obtenido y devuelve verdadero si es letra sino devuelve falso.
- El algoritmo se repite mientras existan caracteres en el documento de entrada. En caso de encontrar un carácter especial hace un salto de palabra o de línea.

El algoritmo posicion_siguiente_vocal.

posición_siguiente_vocal

Entrada: T

Salida: Devuelve la posición de una vocal.

Algoritmo:

1. Obtiene un carácter y su posición de la lista de entrada.

- Utiliza el algoritmo es_vocal, lee el carácter obtenido y si es vocal lo devuelve y devuelve su posición.
- El algoritmo se repite mientras existan caracteres en el documento de entrada. En caso de encontrar un carácter especial hace un salto de palabra o de línea.

El algoritmo posición_siguiente_consonante.

posición_siguiente_consonante

Entrada: T

Salida: Devuelve la posición de una carácter.

Algoritmo:

1. Obtiene un carácter y su posición de la lista de entrada.

- Llama al algoritmo es_vocal lee el carácter obtenido y si no es vocal lo devuelve y devuelve su posición.
- El algoritmo se repite mientras existan caracteres en el documento de entrada. En caso de encontrar un carácter especial hace un salto de palabra o de línea.

Además de estos algoritmos utilizados en la construcción de las regiones R1, R2 y RV se utilizaron dos algoritmos más para la búsqueda de las terminaciones existentes en las palabras de entrada como en los arreglos de sufijos para su comparación y posterior eliminación.

El algoritmo buscar_terminacion_palabra.

buscar_terminacion_palabra

Entrada: Cadena de textos, sufijos.

Salida: Devuelve verdadero o falso si las palabras contienen algunos de los sufijos existentes.

Algoritmo:

1. Obtiene una palabra de la cadena de entrada.

- Llama al algoritmo Substring y lee la palabra de atrás hacia adelante buscando el sufijos contenido en la palabra y lo compara con los sufijos existentes en la cadena de sufijos.

El algoritmo buscar_terminacion_arreglo.

buscar_terminacion_arreglo

Entrada: Cadena de textos, arreglo de sufijos.

Salida: Devuelve en una variable el sufijo existente en la apalabra luego de compararlo con los existente en el arreglo y haber coincidido.

Algoritmo:

1. Obtiene una palabra de la cadena de entrada.
- Llama al algoritmo buscar_terminacion-palabra y si los sufijos son iguales devuelve el sufijo en la variable de salida.

Por último el algoritmo de lematización utiliza un algoritmo ***Vectorizar*** que es el que crea el vector espacial que es la salida del algoritmo ***Lematizar***, que consiste en organizar las raíces obtenidas luego de la lematización teniendo en cuenta el nivel de ocurrencia de cada raíz devolviéndolas de forma descendente.

El algoritmo vectorizar:

Vectorizar

Entrada: token, lista de vectores, variable contador.

Salida: Lista de vectores.

Algoritmo:

1. Obtiene un vector de la lista de vectores.

- Compara los lexemas contenidos en la lista de vectores y si son iguales incrementa en uno su variable contador y lo adiciona a la lista de vectores de salida, así si encuentra un lexema distinto crea un nuevo vector con esos datos y lo adiciona a la lista de salida con su variable contador y su identificador.
- Ordena la lista de vectores de salida descendentemente.

Luego de haber explicado los algoritmos que intervienen en el funcionamiento del algoritmo ***Lematizar***, se presenta como quedaría este último:

Lematizar

Entrada: Lista de token.

Salida: Vector espacial.

Algoritmo:

1. Crear las regiones R1, R2 y RV para lograr identificar los lexemas y morfemas de las palabras.
2. Eliminar los bloques de sufijos definidos.
3. Crear el vector espacial.

El algoritmo ***Lematizar*** se encarga de convertir cada palabra recibida en su raíz natural incrementando su contador para controlar el nivel de aparición de dicha raíz y actualizando su Token _ ID, este proceso de convertir las palabras en su raíz se realiza a través de cuatro pasos fundamentales:

Paso 1: Eliminar los sufijos.

Busca el más largo de los siguientes sufijos existente en la palabra

me	***se***	***sela***	***selo***	***selas***	***selos***	***la***	***le***	***lo***	***las***	***les***	***los***	***nos***

y lo elimina si se produce después de las siguientes terminaciones:

iéndo	***ándo***	***ár***	***ér***	***ír***
iendo	***ando***	***ar***	***er***	***ir***

yendo si lo precede la ***u***

Paso 2: Eliminación de sufijos estándar.

Busca el más largo entre los siguientes sufijos, y lleva a cabo la acción indicada

anza	***anzas***	***ico***	***ica***	***icos***	***icas***	***ismo***	***ismos***	***able***	***ables***	***imiento***

ible	***ibles***	***ista***	***istas***	***oso***	***osa***	***osos***	***osas***	***amiento***	***amientos***	***imientos***

lo elimina de no tener esta terminación busca otro bloque de sufijos tales como:

adora	***ador***	***ación***	***adoras***	***adores***	***aciones***	***ante***	***antes***	***ancia***	***ancias***

elimina estas terminaciones si son precedidas por ***ic.***

De no encontrar ninguno de los sufijos antes mencionado se procede a remplazar terminaciones tales como:

logia, logias por ***log***

ucion, uciones por ***u***

encia, encías por ***ente***

amente se elimina en R1 si es precedido por ***iv, at, os, ic, adse*** se elimina ***en R2***

mente se elimina en R2 si va precedido de ***ante, able, o ible***

idad, idades se elimina en R2 se va precedida de ***abil, ic, o iv***

iva, ivo, ivas, Ivos se elimina en R2 se va precedida ***de at***

Si no apareciera ninguno de los sufijos y terminaciones antes mencionados se procede a buscar el más largo entre los siguientes sufijos en RV, y si lo encuentra, se elimina si es precedida por u.

ya	***ye***	***yan***	***yen***	***yeron***	***yendo***	***yo***	***yó***	***yas***	***yes***	***yais***	***yamo***

Paso 3: Eliminación de sufijos especiales

Buscar el más largo entre los siguientes sufijos en RV, y llevar a cabo la acción indicada.

en, es, eis, emos se elimina y si es precedido por ***gu*** eliminar la u (el gu no tiene por qué estar en RV), luego se procede a la eliminación de estas terminaciones:

arían	***arías***	***arán***	***arás***	***aríais***	***aría***	***aréis***	***aríamos***	***aremos***	***ará***
aré	***erían***	***erías***	***erán***	***erás***	***eríais***	***ería***	***eréis***	***eríamos***	***eremos***
erá	***eré***	***irían***	***irías***	***irán***	***irás***	***iríais***	***iría***	***iréis***	***iríamos***

iremos	***irá***	***iré***	***aba***	***ada***	***ida***	***ía***	***ara***	***iera***	***ad***
ed	***id***	***ase***	***iese***	***aste***	***iste***	***an***	***aban***	***ían***	***aran***
ieran	***asen***	***iesen***	***aron***	***ieron***	***ado***	***ido***	***ando***	***iendo***	***ió***
ar	***er***	***ir***	***as***	***abas***	***adas***	***idas***	***ías***	***aras***	***ieras***
ases	***ieses***	***ís***	***áis***	***abais***	***íais***	***arais***	***ierais***	***aseis***	***ieseis***
asteis	***isteis***	***ados***	***idos***	***amos***	***ábamos***	***íamos***	***imos***	***áramos***	***iéramos***
iésemos	***ásemos***								

Paso 4: Eliminación de sufijos residuales.

Buscar el más largo entre los siguientes sufijos en RV, ***os, a, o, á, i, ó*** y eliminarlos en RV.

Eliminar ***e, è*** en RV y si es precedida por ***gu*** eliminar la ***u.***

Por último se procede a quitar los acentos agudos.

Posteriormente procede a organizarlos en un vector de forma descendente teniendo en cuenta la variable contador estructurándolo de forma jerárquica en el vector espacial. Este algoritmo solo toma las palabras que su tipo de token es ***id*****,** obviando los demás tipos de token como los pertenecientes a los operadores matemáticos, los signos de puntuación, guiones etc.

2.4. Pruebas Funcionales.

Estos resultados deben obtenerse a partir del conjunto de datos escritos en el formato establecido anteriormente. La muestra con la que se prueban los algoritmos fue extraída de la Ley de Regulación Aduanal de la República de Cuba en sus resoluciones 320/11 y 321/11 respectivamente. Una vez procesados los documentos y escritos de forma correcta, de acuerdo a los estándares de entrada de los algoritmos, se puede contabilizar un total de 704 token asignados a la resolución 320/11 distribuidos en los tipos de token declarados, luego de ejecutar el algoritmo ***Tokenizar***, al ejecutar el algoritmo de lematizacion se obtienen 310 raíces de las palabras de tipo de token “Id”, luego se muestra el vector espacial de forma descendente teniendo en cuenta la variable ocurrencia.

La prueba fue desarrollada en una computadora con 1 Gb de memoria RAM y un procesador intelCore2Duo a 2.8 GHz. Los resultados obtenidos se muestran a continuación:

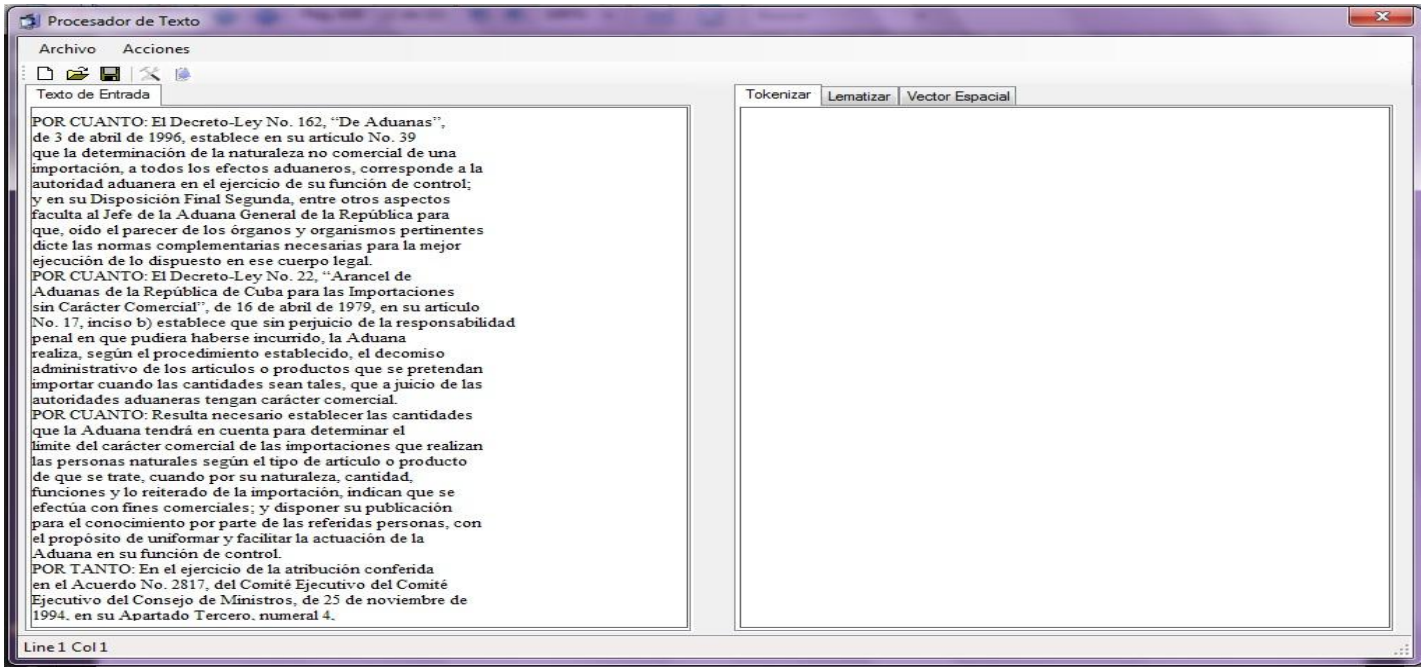


Figura 4. Cargando la Resolución 320/11 de la Ley de Regulación Aduanal.

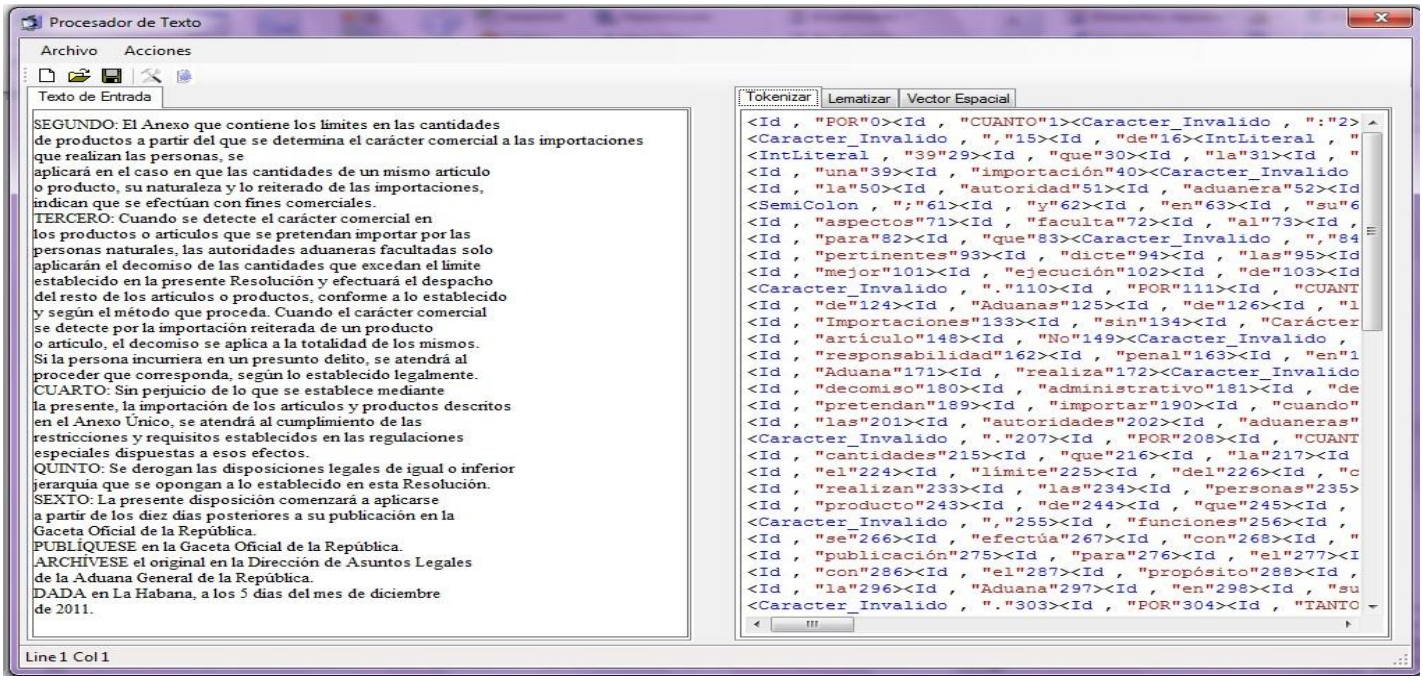


Figura 5. Resultados obtenidos al ejecutar el algoritmo ***Tokenizar***.

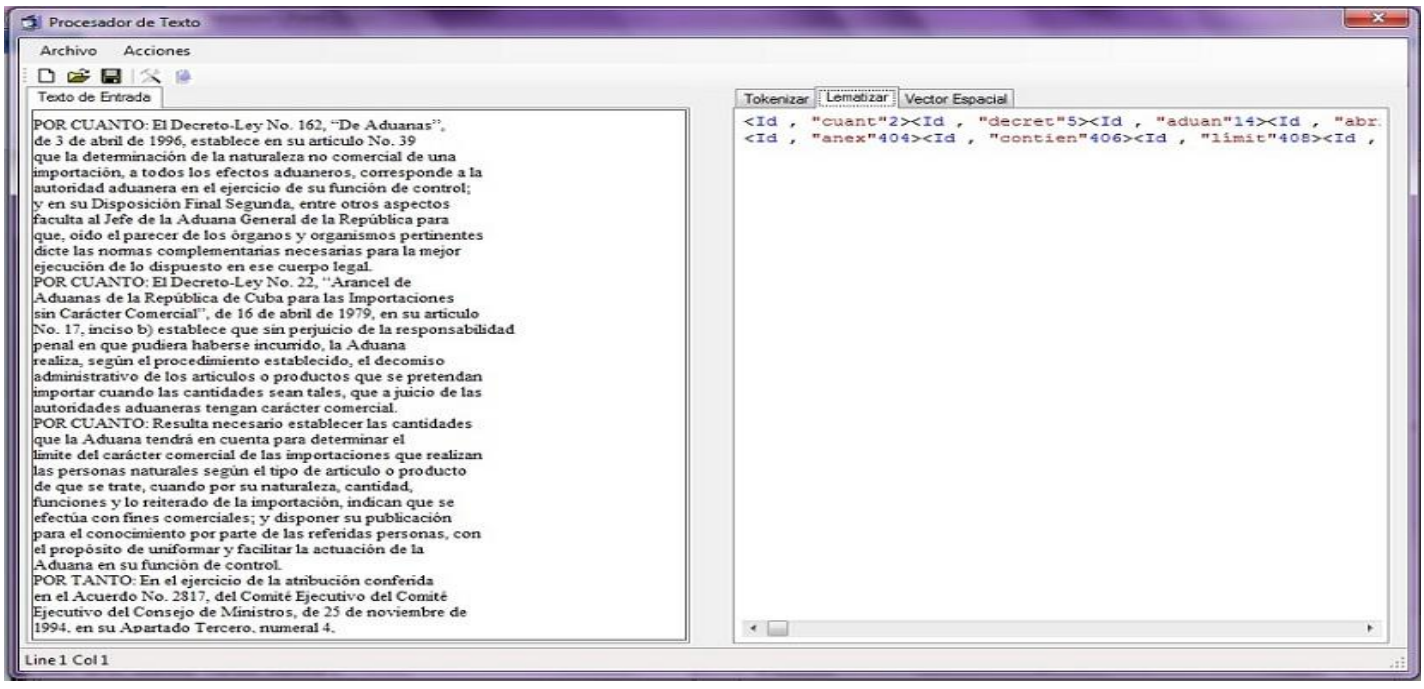


Figura 6. Resultados obtenidos al ejecutar el algoritmo ***Lematizar***.

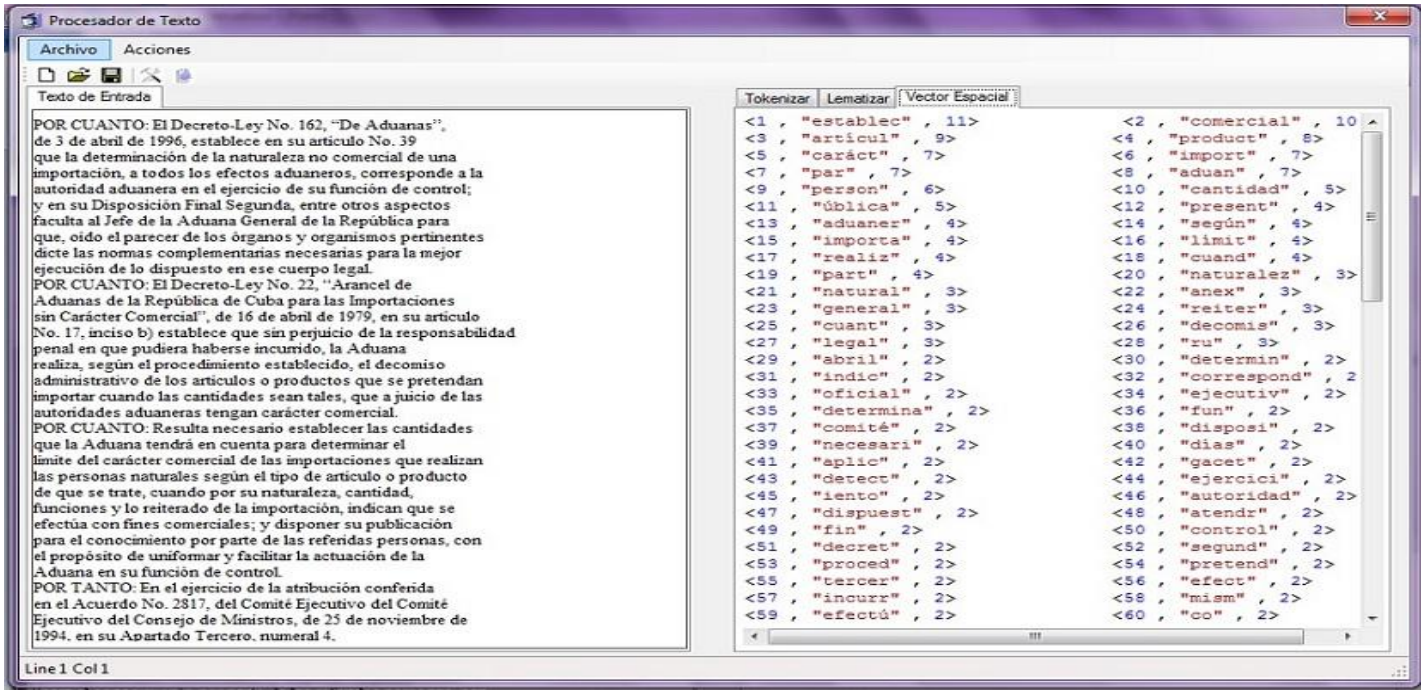


Figura 7. Muestra del vector espacial que devuelve el pre procesamiento.

2.5. Conclusiones Parciales.

En el presente capítulo se muestra un ejemplo con el que se pretende clarificar el problema que representa el pre procesamiento de documentos legales para la construcción de ontologías. Se presenta una propuesta de solución que pretende modificar el algoritmo de lematización propuesto en (6) bajo el nombre de Fuerza Bruta. El algoritmo de Fuerza Bruta, está implementado para el idioma inglés con la utilización de diccionarios para la conversión de sus raíces. Debido a esto, los principales ajusten están enmarcados en la sustitución de los diccionarios por reglas de eliminación de sufijos para el idioma español por su complejo tratamiento en las palabras, con el objetivo de obtener resultados correctos de las raíces en dichos documentos legales. Se presenta la solución propuesta, describiendo cada uno de los algoritmos desarrollados.

CAPÍTULO 3. VALIDACIÓN DE LA SOLUCIÓN

3.1. Introducción

Existen una serie de patrones o métodos que permiten evaluar y validar los resultados alcanzados en una investigación. Entre ellos se encuentran la demostración, experimentación, simulación, uso de métricas, evaluación comparativa, razonamiento lógico y los modelos matemáticos (12). A continuación se describen algunos de estos métodos que a criterio de los investigadores de este trabajo tienen mayor relevancia en las ciencias de la computación.

- **Demostración**

  Intenta demostrar que la solución es factible y válida para una o varias situaciones predefinidas. Es especialmente relevante cuando la demostración de una solución en sí misma se considera una contribución. Consta de dos momentos importantes, construir la solución o prototipo de solución que demuestre que esta es factible y demostrar que la solución construida es razonable para un conjunto predefinido de situaciones. Estas situaciones deben estar predefinidas y no se ha creado para adaptarse a la solución. (12)

  La demostración puede mostrar las deficiencias de la solución. Por otra parte, puede mostrar que la solución es viable y aceptable. Si las situaciones de prueba están diseñadas apropiadamente, entonces la construcción de la solución y sus pruebas para estas situaciones pueden demostrar su validez a pesar de que teóricamente constituyen el patrón de validación menos formal o débil de validación ha sido utilizado en innumerables artículos científicos de la especialidad hasta el punto de ser el tipo de validación que más se utiliza desde el año 1999. (13)

- **Experimentación**

  Intenta validar o rechazar un conjunto de hipótesis asociada a las afirmaciones acerca de la solución. Según (14) un experimento es un estudio de investigación en el que se manipulan deliberadamente una o más variables independientes para analizar las consecuencias de esa manipulación sobre una o más variables dependientes, dentro de una situación de control para el investigador. La experimentación procede a establecer resultados asociados con la solución del

problema de investigación en situaciones en que la recogida y el análisis de los datos son el único método factible de validación.

- **Simulación**

  Intenta validar la solución propuesta para el problema de investigación a través de un software. Consta de cinco momentos importantes, el primero de ellos es desarrollar el modelo conceptual del problema y su solución para que sea simulado en una computadora. Luego se desarrolla el conjunto inicial de datos de prueba, se selecciona la simulación diseñada específicamente para el dominio del problema. Se ejecuta dicha simulación para el conjunto de prueba elaborado con anterioridad y por último se demuestra la validez de la solución argumentando que las pruebas realizadas representan situaciones de la vida real. (12)

- **Razonamiento Lógico**

  Utiliza la argumentación como forma de validación de la solución. Es una forma más débil de validación que el modelo matemático o el uso de experimentos. Consta de tres momentos importantes, identificación de las suposiciones (axiomas) relacionadas con el problema, identificación de las reglas (reglas de deducción) relacionadas con el problema o la solución y construcción un camino lógico de las hipótesis (axiomas) a los planteamientos de la solución, utilizando las reglas de deducción que se identifiquen.

  Cuando los axiomas, reglas de deducción y las afirmaciones acerca de la solución se pueden afirmar con precisión; esta técnica constituye un modelo matemático para validar cualquier investigación siempre que no exista vaguedad en demostrar que las afirmaciones son consecuencia lógica de los axiomas. (12)

- **Modelos Matemáticos**

Intenta demostrar matemáticamente las afirmaciones acerca de la solución desarrollada. Consta de cuatro momentos importantes, expresar las afirmaciones acerca de la hipótesis de forma cuantitativa y precisa, convertir dichas afirmaciones para ser probadas como un teorema bien definido, demostrar los resultados auxiliares (lemas) que pueden ayudar a demostrar el teorema y por último demostrar el teorema de las

afirmaciones, que pueden utilizar los lemas ya probados. Este modelo ofrece la forma más segura de validación de la solución. Esta validación es incluso más certera que la validación experimental. (12)

La aplicación de estos patrones varía de acuerdo a su idoneidad y la seguridad con la que se puede establecer la validez de una solución. La demostración provee la forma más débil de la validación. Puede, sin embargo, ser adecuado si la solución es novedosa y resuelve un problema para el cual no existe ninguna solución previa. Por otro lado, las pruebas matemáticas constituyen la forma más certera de validación. La certeza del razonamiento lógico depende en gran medida de la precisión de sus argumentos y suposiciones. La experimentación y simulación son útiles cuando el problema es complejo y es inviable efectuar una demostración matemática. El uso de métricas y la evaluación comparativa, mecanismos para cuantificar afirmaciones respecto a una solución, son generalmente útiles cuando se utiliza en la experimentación y la simulación.

El uso de uno o varios métodos de validación depende en gran medida de las características del problema estudiado y de las convenciones aceptadas por la comunidad de investigadores como alternativas válidas de corroboración en la temática. La presente investigación utiliza el método de "Demostración" como mecanismo de validación. Este patrón se adapta correctamente a las características del trabajo siendo ideal para corroborar la validez del mismo pues no existen soluciones anteriores para este problema. La demostración en sí misma tiene valor práctico al estar compuesta por un prototipo funcional que permite construir modelos descriptivos aun cuando es necesario perfeccionar algunos detalles de usabilidad en dicho prototipo. Por último, la "Demostración" constituye el patrón de validación más utilizado en las publicaciones científicas para el área de las ciencias de la computación de acuerdo a los trabajos de (13), (15), (16).

3.2. Recursos computacionales utilizados.

Los recursos computacionales en esta investigación refieren las características de hardware y software de la computadora utilizada en el proceso de validación de la solución. Para efectuar el proceso de "Demostración" se utilizó una computadora ACPI Multiprocessor PC. Esta máquina cuenta con una motherboard Intel Rogers City DG965RY que incorpora un procesador DualCore Intel Core 2 Duo E4500 a 2.20GHz y una memoria RAM DDR2 SDRAM Kingston 9905320-007.A00LF con 1GB de capacidad.

Los algoritmos propuestos fueron probados en una plataforma Microsoft Windows XP Professional publicada en el año 2002, a la que se le incorpora el paquete de corrección de errores Service Pack 3.

3.3. Conjunto de información utilizada para la demostración.

El conjunto de datos utilizados para efectuar la demostración fue recolectado en la ley de Regulación Aduanal General de la República de Cuba en sus resoluciones No. 320/11 y No. 321/11. Específicamente, fueron seleccionados los Artículos referentes a la "Determinación de la naturaleza no comercial de una importación", a todos los efectos aduaneros, y el "Arancel de Aduanas de la República de Cuba para las Importaciones sin Carácter Comercial". Estos contienen elementos significativos para la obtención de información contenida en las palabras que permitan determinar las relaciones existentes entre dichos conceptos de palabras así como obtener la relevancia de cada raíz existente en los documentos de entrada. Los documentos fueron transformados en una lista de token con la siguiente estructura ***<tipo_token, "palabra o carácter", token_id>*** y posteriormente se convierte en el vector espacial estructurado de esta manera ***<token_id, "raiz", cant_ocurrencia>*** cumpliendo con el formato definido para el pre procesamiento propuesto por los algoritmos ***Tokenizar*** y ***Lematizar*** (epígrafe 2.3). A partir de estos se evaluaron las palabras contenidas en la ley de Regulación Aduanal de la República de Cuba definiendo un ***tipo_token*** y un ***token_id*** para cada una así como llevar cada palabra a su raíz monitoreándola por su variable de ocurrencia definiendo que toda variable de ocurrencia mayor que 5 es una raíz relevante en el documento teniendo en cuenta el volumen general de palabras en las resoluciones, y también partiendo del criterio de los especialistas en el tema.

El vector espacial de salida del pre procesamiento es de fácil comprensión ya que cada palabra luego del pre procesamiento está compuesta por tres elementos, esta cantidad puede variar por lo que puede servir como mecanismo de optimización en trabajos futuros.

Primeramente este pre procesamiento se realizó manualmente en la resolución 320/11 convirtiéndose en una tarea muy difícil debido al tiempo empleado en ella y las dificultades en la extracción de información. En el proceso de tokenización en la resolución se asignaron 663 token correspondientes a las palabras, incluyendo artículos, pronombres, etc., ya que a simple vista no es posible definir cuales palabras serán relevantes, en la lematización se obtuvieron todas las raíces de cada palabra pero sin desechar ningún

tipo de información, luego en la construcción del vector espacial se trató de ser lo más exacto posible para que estuviera estructurado correctamente en dependencia de la ocurrencia de dichas raíces, la información arrojada por la aplicación del pre procesamiento manualmente se representa en la siguiente tabla.

	Resultados del Pre procesamiento manualmente				
Resolución	Cant. Token	Cant. Raíces	Raíces en Vector E.	Max. Ocurrencia	Min. Ocurrencia
320/11	663	315	165	13	1
Cantidad de token por Tipos de Token					
Inliteral	Id	Car. Invalid	Rigth Paren	Lefth Paren	Semi Colon
12	240	57	1	1	2

Tabla 7. Resultados arrojados luego de aplicar manualmente el pre procesamiento en la Resolución 320/11.

La muestra de datos obtenida a partir de las resoluciones 320/11 y 321/11 constan con 710 y 769 elementos entre palabras y caracteres especiales respectivamente, con este volumen de datos se puede determinar el buen funcionamiento del algoritmo y obtener las raíces de cada palabra.

Teniendo en cuenta la heurística para la cual fueron diseñados los algoritmos ***Tokenizar*** y ***Lematizar*** y el formato (epígrafe 2.3) requerido por dichos algoritmos, es necesario transformar cada una de las palabras en su raíz asignándole un ***tipo_token*** y un ***token_id***, obteniendo como resultado 710 y 769 token respectivamente. Por otro lado, al transformar las palabras en su raíz obtenemos 167 y 181 raíces con información respectivamente cada una con su token y su nivel de aparición.

Las siguientes tablas muestra la información que se obtiene luego de realizar el pre procesamiento propuesto automáticamente. Vea tablas (9 y 10).

	Resultados del Pre procesamiento automáticamente				
Resolución	Cant. Token	Cant. Raíces	Raíces en Vector E.	Max. Ocurrencia	Min. Ocurrencia
320/11	710	315	167	14	1
Cantidad de token por Tipos de Token					
Inliteral	Id	Car. Invalid	Rigth Paren	Lefth Paren	Semi Colon
13	245	57	1	1	2

Tabla 8. Resultados arrojados luego de aplicar el pre procesamiento a través de la tokenización y lematización propuesta en el Capítulo 2 en la Resolución 320/11.

	Resultados del Pre procesamiento automáticamente				
Resolución	Cant. Token	Cant. Raíces	Raíces en Vector E.	Max. Ocurrencia	Min. Ocurrencia
321/11	769	345	181	14	1
Cantidad de token por Tipos de Token					
Inliteral	Id	Car. Invalid	Rigth Paren	Lefth Paren	Semi Colon
15	266	58	1	1	2

Tabla 9. Resultados arrojados luego de aplicar el pre procesamiento a través de la tokenización y lematización propuesta en el Capítulo 2 en la Resolución 321/11.

Al aplicar los algoritmos propuesto ***Tokenizar*** y ***Lematizar*** utilizando los datos descritos anteriormente se obtuvo un grupo de información que no se consiguió al realizar este proceso manualmente. A partir de los datos generados para cada resolución se puede determinar que los algoritmos propuestos devuelven resultados adecuados para el pre procesamiento.

3.4. Conclusiones Parciales

El presente capítulo presentó una serie de patrones o métodos utilizados para validar los resultados de una investigación científica. Se determinó que el patrón “Demostración” es el adecuado para realizar las pruebas de validación para la solución propuesta en el presente trabajo. Se definieron los elementos fundamentales de la demostración propuesta así como los recursos de hardware y software con que se cuenta para efectuar las pruebas.

Para corroborar el correcto funcionamiento de los algoritmos se varió la muestra de las resoluciones existentes en la Ley de Regulación Aduanal de la República de Cuba. Los resultados obtenidos demuestran, primeramente, que los algoritmos propuestos asignan un token a cada palabra o carácter existente en las resoluciones y obtienen las raíces de las palabras significativas asignándole una variable de ocurrencia a cada raíz, esta información según el criterio de los especialistas en el tema, brinda patrones relevantes para la clasificación de documentos y la creación de ontologías. En la mayoría de

los casos de prueba, los algoritmos devolvieron resultados adecuados en cuanto a la cantidad de raíces que generan, de acuerdo a el corpus del documento que se pre procesa.

CONCLUSIONES

En el presente trabajo se propusieron dos algoritmos para el pre procesamiento de minería de texto en corpus de documentos legales, con vistas a su utilización en la creación de una ontología. Los referidos algoritmos siguen los supuestos teóricos establecidos en el diseño de los algoritmos ***Tokenizar*** y ***Lematizar*** y realizan un grupo de mejoras enfocadas a la tipología específica del idioma español y su uso en los documentos legales, dentro de ellas destacan las siguientes:

- Se ajustó el algoritmo para que pueda trabajar con documentos legales.
- Se modificó la utilización de diccionarios en el algoritmo de lematización por reglas de eliminación de sufijos, proporcionando un mejor funcionamiento en el pre procesamiento en el idioma español.
- Se ajustó el mecanismo de construcción del vector espacial donde solo son relevantes las palabras que tienen una relación taxonómica.

Los resultados alcanzados permiten concluir que:

1. La aplicación de las técnicas de pre procesamiento mediante los algoritmos propuestos permite obtener el fichero de entrada para los algoritmos de minería de textos en la construcción de ontologías.
2. Es posible obtener un conjunto de información contenida en las raíces de las palabras y sus relaciones existentes en el vector espacial a partir de la aplicación de los algoritmos diseñados.

RECOMENDACIONES

Para el mejoramiento de este trabajo es importante optimizar un grupo de elementos que no fueron tomados en cuenta por cuestiones de tiempo, dentro de ellos señalamos los siguientes:

1. Establecer nuevas reglas de conversión de formas verbales a su raíz que aún no se han tratado, debido a que su aparición en el corpus estudiado es muy infrecuente.
2. Trabajar en el perfilado de las técnicas utilizadas para ajustar la salida de los algoritmos.

Debido a su relevancia y a que forman parte del trabajo futuro que se desarrollará en la temática.

GLOSARIO DE TÉRMINOS

Minería de Texto: La extracción no trivial de información implícita, previamente desconocida y potencialmente útil de grandes cantidades de datos texto.

Ontología: Representación formal de un conjunto de conceptos dentro de un dominio específico y de las relaciones entre dichos conceptos.

Taxonomía: Tiene su origen en un vocablo griego que significa “ordenamiento”. Se trata de la ciencia de la clasificación en este caso la clasificación de palabras y significados.

TIC: Tecnología de la Informática y las Comunicaciones.

Tokenización: Asignación de un identificador único a través de un token a objetos tales como palabras, caracteres, etc.

Lematización: La extracción de la raíz contenida en las palabras a través de técnicas de minería de texto.

REFERENCIAS BIBLIOGRÁFICAS

1. **Conferencia Iberoamericana de Ministros de Administración Pública y Reforma del Estado.** *Carta Iberoamericana de Gobierno Electrónico.* Pucón, Chile : s.n., 2007. pág. 25, Carta Iberoamericana de Gobierno Electrónico.

2. **Estela Pocoví, Gertrudis María y Farabollini, Gustavo Ricardo.** *Gobierno Electrónico: Un cambio estructural - La integración de la información como requisito.* Caracas, Venezuela : s.n., 2002. pág. 29. XVI Concurso de Ensayos y Monografías del CLAD sobre Reforma del Estado y Modernización de la Administración Pública.

3. **Blaquier Ascaño, Dr. Marta Isabel.** *OnProc: Una Ontología para el Proceso Jurídico.* Ciudad de La Habana : s.n. pág. 7.

4. **Lamarca Lapuente, María Jesús.** Hipertexto. [En línea] Mayo de 2011. [Citado el: 10 de Junio de 2012.] http://www.hipertexto.info/documentos/ontologias.htm.

5. **Biemann, Chris.** *Ontology Learning from Text: A Survey of Methods.* s.l. : LDV-Forum, 2005. 20 (2) – 75-93.

6. **Feldman, Ronen y Sanger, James.** *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data.* Cambridge : s.n.

7. **Perez Abelleira, M. Alicia y Cardoso, Carolina A.** *Minería de texto para la categorización automática de documentos.* 2010.

8. **Dapozo, Gladys, y otros, y otros.** *Técnicas de preprocesamiento para mejorar la calidad de los datos en un estudio de caracterización de ingresantes universitarios.* Corrientes, Argentina : s.n.

9. **Cobo Ortega, Angel, Rocha Blanco, Rocio y Alonso Martínez, Margarita.** *Descubrimiento de conocimiento en repositorios documentales mediante técnicas de Minería de Texto y Swarm Intelligence.* 2009.

10. *Literature Review on Preprocessing for Text Mining.* Montfort University.

11. **Pérez Betancourt, Pedro Miguel.** *Ley de Regulación Aduanal.* Gaceta Oficial de la República de Cuba, Ministerio de Justicia. La Habana, Cuba : s.n., 2011. pág. 32. ISSN 1682-7511.

12. **Vaishnavi , Vijay K. y Kuechler Jr., William.** *Design Science Research Methods and Patterns Innovating Information and Communication Technology.* New York : Auerbach Publications, 2008.

13. **Shaw, M.** *What makes good research in software engineering.* s.l. : FOR TECHNOLOGY TRANSFER (STTT). SPRINGER BERLIN / HEIDELBERG, 2002.

14. **H. Sampieri, C. Roberto, Fernández Collado, Carlos y Baptista Lucio, Pilar.** *Metodología de la investigación.* s.l. : McGRAW - HILL INTERAMERICANA DE MÉXICO, 1991.

15. **Cañete.** *¿Qué se entiende, en España, por Investigación en Ingeniería del Software?* s.l. : MIFISIS, 2002.

16. **Shaw, M.** *Writing good software engineering research papers: minitutorial.* Washington : Proceedings of the 25th International Conference on Software Engineering, ICSE, 2003.

BIBLIOGRAFÍA

1. **Conferencia Iberoamericana de Ministros de Administración Pública y Reforma del Estado.** *Carta Iberoamericana de Gobierno Electrónico.* Pucón, Chile : s.n., 2007. pág. 25, Carta Iberoamericana de Gobierno Electrónico.

2. **Estela Pocoví, Gertrudis María y Farabollini, Gustavo Ricardo.** *Gobierno Electrónico: Un cambio estructural - La integración de la información como requisito.* Caracas, Venezuela : s.n., 2002. pág. 29. XVI Concurso de Ensayos y Monografías del CLAD sobre Reforma del Estado y Modernización de la Administración Pública.

3. **Blaquier Ascaño, Dr. Marta Isabel.** *OnProc: Una Ontología para el Proceso Jurídico.* Ciudad de La Habana : s.n. pág. 7.

4. **Lamarca Lapuente, María Jesús.** Hipertexto. [En línea] Mayo de 2011. [Citado el: 10 de Junio de 2012.] http://www.hipertexto.info/documentos/ontologias.htm.

5. **Biemann, Chris.** *Ontology Learning from Text: A Survey of Methods.* s.l. : LDV-Forum, 2005. 20 (2) – 75-93.

6. **Feldman, Ronen y Sanger, James.** *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data.* Cambridge : s.n.

7. **Perez Abelleira, M. Alicia y Cardoso, Carolina A.** *Minería de texto para la categorización automática de documentos.* 2010.

8. **Dapozo, Gladys, y otros, y otros.** *Técnicas de preprocesamiento para mejorar la calidad de los datos en un estudio de caracterización de ingresantes universitarios.* Corrientes, Argentina : s.n.

9. **Cobo Ortega, Angel, Rocha Blanco, Rocio y Alonso Martínez, Margarita.** *Descubrimiento de conocimiento en repositorios documentales mediante técnicas de Minería de Texto y Swarm Intelligence.* 2009.

10. *Literature Review on Preprocessing for Text Mining.* Montfort University.

11. **Pérez Betancourt, Pedro Miguel.** *Ley de Regulación Aduanal.* Gaceta Oficial de la República de Cuba, Ministerio de Justicia. La Habana, Cuba : s.n., 2011. pág. 32. ISSN 1682-7511.

12. **Vaishnavi , Vijay K. y Kuechler Jr., William.** *Design Science Research Methods and Patterns Innovating Information and Communication Technology.* New York : Auerbach Publications, 2008.

13. **Shaw, M.** *What makes good research in software engineering.* s.l. : FOR TECHNOLOGY TRANSFER (STTT). SPRINGER BERLIN / HEIDELBERG, 2002.

14. **H. Sampieri, C. Roberto, Fernández Collado, Carlos y Baptista Lucio, Pilar.** *Metodología de la investigación.* s.l. : McGRAW - HILL INTERAMERICANA DE MÉXICO, 1991.

15. **Cañete.** *¿Qué se entiende, en España, por Investigación en Ingeniería del Software?* s.l. : MIFISIS, 2002.

16. **Shaw, M.** *Writing good software engineering research papers: minitutorial.* Washington : Proceedings of the 25th International Conference on Software Engineering, ICSE, 2003.

17. **Hernández Ramírez, Haliuska y Saiz Noeda, Maximiliano.** *Ontologías mixtas para la representación conceptual de objetos de aprendizaje.*

18. **Maña, Manuel, y otros, y otros.** *Los proyectos SINAMED e ISIS: Mejoras en el Acceso a la Información Biomédica mediante la integración de Generación de Resúmenes, Categorización Automática de Textos y Ontologías.* pág. 2.

19. **Vañó Vañó, María José.** *Integración de la documentación legal electrónica a través e LEXML.* Valencia, España : s.n., 2009. ISSN 1135-3716.

20. **Tramullas, Dr. Jesús.** *Agentes y ontologías para el tratamiento de información: clasificación t recuperación en Internet.* Zaragoza : s.n.

21. **Ortiz Rodriguez, Fernando, Palma, Raúl y Villazón Terrazas, Boris.** *Aplicaciones para Gobierno Electrónico Semántico en México: una aproximación para el Desarrollo Municipal.*

22. **del Moral, María Esther y Cernea, Doina Ana.** *Diseñando Objetos de Aprendizaje como facilitadores de la construcción del conocimiento.*

23. **Hilera, José R., y otros, y otros.** *Aplicación de técnicas de Ingeniería Lingüística en sistemas de e-learning basados en objetos de aprendizaje.* 2004. Este trabajo está soportado por el Ministerio de Industria a través del proyecto FIT-350101-2004-7: “Plataforma para la gestión y explotación de recursos educativos virtuales”, del Programa Nacional de Tecnologías de Servicios de la Sociedad de la Informa.

24. **Tricas Lamana, Fernando.** *El gobierno electrónico: servicios públicos y participación ciudadana.* 2007. ISBN: 978-84-96653-56-6.

25. **Cardona, Ing. Diego.** *El gobierno electrónico - Una revisión desde la perspectiva de la prestación de servicios.* Barcelona, España : s.n., 2002.

26. **Criado Grande, J. Ignacio, Ramilo Araujo, María Carmen y Salvador Serna, Miguel.** *La Necesidad de Teoría(s) sobre Gobierno Electrónico. Una Propuesta Integradora.* Caracas, Venezuela : s.n., 2002. XVI Concurso de Ensayos y Monografías del CLAD sobre Reforma del Estado y Modernización de la Administración Pública "Gobierno Electrónico".

27. **Santacruz Valencia, L. P., Aedo, I. y Delgado Kloos, C.** *Objetos de aprendizaje: Tendencias dentro de la web semántica.* 2003-2004.

28. **Montes y Gómez, Manuel.** *Minería de texto: Un nuevo reto computacional.* México D.F. : s.n.

29. **Orta Palacios, Claudia Patricia.** *Métodos basados en patrones léxicos para la extracción de información.* s.l. : INAOE, 2008.

30. **Shamsfard, Mehrnoush y Abdollahzadeh Barforoush, Ahmad.** *The State of the Art in Ontology Learning: A Framework for Comparison.*

31. **Agrawal, R, Imielinski, T y Swami, A.** *Mining Associations between sets of items in massive databases.* ACM-SIGMOD International Conference on Data : s.n., 1993.

32. **Agrawal, R y Srikant, R.** *Fast algorithms for mining association rules.* s.l. : Proceedings of the 20th, 1994.

33. **Frawley, W., Piatetsky-Shapiro, G y Matheus.** *Knowledge discovery in databases: An overview.* s.l. : AAA/MIT Press, 1992.

34. **Triantaphyllou, Evangelos.** *Data Mining and Knowledge Discovery via Logic-Based Methods.* Louisiana : Springer, 2010.

35. **Motoda, Hiroshi y Ohara, Kouzou.** *Apriori.* s.l. : Taylor & Francis Group, 2009.

36. **Li, YingJiu, y otros, y otros.** *Discovering calendar-based temporal association rules.* s.l. : Knowledge-Based Systems, 2003.

37. **Kuok, C.M, Fu, A.W. y Wong, M.H.** *Mining fuzzy association rules in databases.* s.l. : SIGMOD Record, 1998.

38. **Jimenez Ruiz, Maria Dolores.** *Modelado formal para la representación y evaluación de reglas de asociación.* Granada : Departamento de ciancias de la computación e inteligencia artificial, 2010.

39. **Gyenesei, A.** *A fuzzy approach for mining quantitative.* s.l. : Acta Cybern, 2001.

40. **Fayyad, U M, y otros, y otros.** *Advances in Knowledge Discovery and Data Mining.* Cambridge, MA : AAAI Press and MIT Press, 1996.

41. **Delgado, Miguel, Ruiz, M. Dolores y Sánchez, Danel.** *Reglas de asociación difusas: Nuevos Retos.* Granada : ESTYLF08, Cuencas Mineras, 2008.

42. **Clark, P. y Boswell, R.** *Data Mining. PracticalMachine Learning Toolsand Techniques.* s.l. : Morgan Kaufmann Publishers, 2000.

43. **Chena, Y y Weng, C.H.** *Mining association rules from imprecise ordinal data. Fuzzy.* 2008.

44. **Chen, G, y otros, y otros.** *Simple association rules (SAR) and the SAR-based.* s.l. : Computers & Industrial Engineering, 2002.

45. **Laleh, Naeimeh y Abdollahi Azgomi, Mohammad.** *A Taxonomy of Frauds and Fraud Detection Techniques.* Tehran : Department of Computer Engineering, 2009.

46. **Orallo Hernández, José, Quintana Ramírez, Ma José y Ferri Ramírez, Cesar.** *Introducción a la Minería de Datos.* Madrid : Pearson Educación S.A., 2004.

47. **Li, Jiye y Cercone, Nick.** *Introducing A Rule Importance Measure.* Canada : s.n., 2009.

48. **Mesa Rodriguez, Alejandro, y otros, y otros.** *Obtencion d conjuntos frecuentes usando computo reconfigurable.* La Habana : CENATAV, 2009.

49. **Delgado, Miguel, y otros, y otros.** *Fuzzy Association Rules: General Model.* s.l. : IEEE TRANSACTIONS ON FUZZY SYSTEMS, 2003.

50. **Sánchez, D.** *Adquisición de relaciones entre atributos en bases de datos.* Granada : Dept. Comput. Sci. Artificial, 1999.

51. **Brin, S, y otros, y otros.** *Dynamic itemset counting and implication rules for market basket data.* s.l. : SIGMOD, 1997.

52. **Silverstein, C, Brin, S y Motwani, R.** *Beyond market baskets: Generalizing association rules to dependence rules.* s.l. : Data Mining Knowl. Disc., 1998.

53. **Hernández León, Raudel, y otros, y otros.** *Descubrimiento de conjuntos frecuentes de items en datos estaticos y dinamicos.* La habana : Dpto. Minería de Datos, Centro de Aplicaciones de Tecnología de Avanzada (CENATAV), 2010.

54. **Medina Pagola, José E., y otros, y otros.** *Generación de conjuntos de items y reglas de asociación.* La Habana : Dpto. Minería de Datos, Centro de Aplicaciones de Tecnología de Avanzada (CENATAV), 2007.

55. **Delgado, M., y otros, y otros.** *Fuzzy association rules: general model and applications.* s.l. : Fuzzy Systems, IEEE Transactions on, vol. 11, no. 2, 2003.

56. **Delgado, M., y otros, y otros.** *Mining fuzzy association rules: an overview.* s.l. : Soft Computing for Information Processing and Analysis, 2005.

57. **Delgado, M., Sánchez, D. y Vila, M. A.** *Fuzzy cardinality based evaluation of quantified sentences.* s.l. : International Journal of Approximate Reasoning, 2000.

58. **Han, Jiawei y Fu, Yongjian.** *Discovery of Multiple-Level Association Rules from Large Databases.* British Columbia : School of Computing Science Simon Fraser University, 1995.

59. **Ramakrishnan, Srikant y Agrawal, Rakesh.** *Mining Generalized Association Rules.* Zurish : IBM Almaden research center, 1995.

60. **Sriphaew, Kritsada y Theeramunkong, Thanaruk.** *A New Method for Finding Generalized Frequent Itemsets in Generalized Association Rule Mining.* Thailand : Information Technology Program Sirindhorn International Institute of Technology, Thammasat University, 2002.

61. **A. Savasere, E. Omiecinski, and S.Navathe.** *An eficient algorithm for mining association rules in large.* Atlanta : Technical Report GIT-CC-95-04, Institute of Technology, 1995.

62. **Chung., J. D. Holt and S. M.** *Multipass algorithms for mining association rules in text databases.* s.l. : Knowledge and Information Systems, Springer-Verlag, 2001.

63. **Agrawal R., Srikant R.** *Fast Algorithms for Mining Association Rules.* 1994.

64. **Medina Pagola, José E., y otros, y otros.** *Generación de conjuntos de items y reglas de asociación.* La Habana : Dpto. Minería de Datos, Centro de Aplicaciones de Tecnología de Avanzada (CENATAV), 2007.

65. **Comisión, económica para américa latina y el caribe.** *Indices de comercio exterior de Cuba.* 2006.

66. **OMA, Organización Mundial de Aduanas.** *Convenio de Kyoto - Directivas de control aduanero.* Kyoto : http://www.wcoomd.org/Kyoto/20Sp/cap6, 1997.

67. **Vaishnavi , Vijay K. y Kuechler Jr., William.** *Design Science Research Methods and Patterns Innovating Information and Communication Technology.* NewYork : Auerbach Publications is an imprint of the , an informa business , 2008.

68. **Yu-Lu, LIU, y otros, y otros.** *An Efficient Algorithm of Mining Association Rules Based on Digital Pure Subset.* Chongqing : College of Math and Computer Science, Chongqing Three Gorges University, 2009.

69. **Yueqin, Zhang, Qingwei, Yan y Lili, Zong.** *An Association Rule Algorithm Based on Quotient Space.* Taiyuan : College of Computer and Software Taiyuan University of Technology, 2009.

ANEXOS

Imágenes de la codificación de los algoritmos principales que utilizan los algoritmos propuestos para la tokenización y lematización en este trabajo.

- Algoritmo leer_caracter:

```
public char leer_caracter()
        {
            int char_leido = textReader.Read();
            posicion_actual++;
            if (char_leido == -1)
                return '\0';
            char c = (char)leer_caracter();
            if (c == '\n')
                linea_actual++;
            return c;
        }
```

Figura 8. Algoritmo que recorre el documento carácter a carácter.

- Algoritmo es_vocal:

```
private bool es_vocal(char c)
        {
        return (c >= 'a' && c <= 'z') ||
               (c >= 'A' && c <= 'Z') ||
                c == '_' ||
               (c == 'Á') || (c == 'á') ||
               (c == 'é') || (c == 'É') ||
               (c == 'í') || (c == 'Í') ||
               (c == 'ó') || (c == 'Ó') ||
               (c == 'ú') || (c == 'Ú');

         }
```

Figura 9. Algoritmo que define si un carácter es vocal

- Algoritmo posicion_siguiente_vocal:

```
public int posicion_siguiente_vocal(string texto)
    {
          start = 0;
           int longitud = texto.Length;
           for (int i = start; i < longitud; i++)
        {
            if (es_vocal(texto[i]) == true)
            {
                return i;
            }
        }
           return longitud;
     }
```

Figura 10. Algoritmo que define si el próximo carácter es vocal.

- Algoritmo posicion_siguiente_consonante:

```
public int posicion_siguiente_consonante(string texto)
        {
         start = 0;
           int longitud = texto.Length;
           for (int i = start; i < longitud; i++)
        {
            if ((es_vocal(texto[i])) == false)
            {
                return i;
            }
        }
           return longitud;
        }
```

Figura 11. Algoritmo que define si el próximo carácter es consonante.

- Algoritmo buscar_terminacion_palabra:

```
public bool buscar_terminacion_palabra(string texto, string sufijo)
     {

     if (texto.Length < sufijo.Length)
         {
             return false;
         }

      if (texto.Substring(texto.Length- sufijo.Length)== sufijo)
          {
              return true;
          }
            return false;
     }
```

Figura 12. Algoritmo de búsqueda para las terminaciones de las palabras.

Algoritmo buscar_terminacion _arreglo:

```
public string buscar_terminacion_arreglo(string texto, string[] arreglo)
        {
            string resultado = "";
            for (int i = 0; i < arreglo.Length - 1; i++)
            {
                if (buscar_terminacion_palabra(texto, arreglo[i]))
                {
                    resultado = arreglo[i];
                }
            }
            return resultado;
        }
```

Figura 13. Algoritmo de búsqueda de sufijos en los arreglos.

- Algoritmo vectorizar:

```
public List<Vectores> vectorizar (Token token, int id,
                                  List<Vectores> listas_de_vectores)
        {
            bool bandera = false;
              for (int i = 0; i < listas_de_vectores.Count; i++)
            {
               if (listas_de_vectores[i].Token.Lexeme1 == token.Lexeme1)
                {
                    listas_de_vectores[i].Cantidad++;
                    bandera = true;
                    break;

                }
            }
              if (bandera == false)
              {
                  Vectores nuevo_vector = new Vectores(token, id);
                  listas_de_vectores.Add(nuevo_vector);
                  nuevo_vector.ordenar(listas_de_vectores);
              }
              return listas_de_vectores;
        }
```

Figura 14. Algoritmo que crea el vector espacial.