INSTITUTO POLITÉCNICO NACIONAL
CENTRO DE INVESTIGACÓN EN COMPUTACIÓN

Sistema de extracción y representación del conocimiento a partir de documentos descriptivos.

TESIS

QUE PARA OBTENER EL GRADO DE
MAESTRO EN CIENCIAS DE LA COMPUTACIÓN

PRESENTA

Gabriela López Yebra

Directores:

Dr. Adolfo Guzmán Arenas
Dra. Alma Delia Cuevas Rasgado

México D.F. **2012**

AGRADECIMIENTOS

Agradezco profundamente a mi director, el Dr. Adolfo Guzmán Arenas, por su orientación y confianza durante la realización de este trabajo, sin duda un gran ejemplo e inspiración durante este periodo y lo seguirá siendo en toda mi vida profesional.

Agradezco a mi co-directora, la Dra. Alma Delia Cuevas Rasgado por su disposición y apoyo durante todo el proceso que me llevó a culminar exitosamente este trabajo.

A mis sinodales, Dr. Jesús Olivares, Dr. Grigori Sidorov, Dr. Alexander Gelbukh y Dr. Gilberto Martínez, por sus exigencias y consejos.

Agradezco al Consejo Nacional de Ciencia y Tecnología (CONACyT) por el apoyo que me brindó, sin el cual no me hubiera sido posible realizar un posgrado.

Agradezco infinitamente a mis padres, Víctor y Yolanda, por su motivación y apoyo incondicional, por su guía y empeño en hacer de mi una buena mujer, personal y profesionalmente. Cada uno de mis éxitos es también suyo.

A mi hermana Adriana por ser mi amiga y darme todo su apoyo y ayuda cada que lo necesito.

Finalmente, gracias a mi amor Gerardo, por estar a mi lado, por cada palabra de aliento, por la confianza y toda la paciencia que me tuviste. Gracias por tu ayuda y consejos que sin duda son un pilar importante de este logro.

ÍNDICE

RESUMEN

Se presenta un método basado en búsqueda de patrones en textos y un sistema de reglas para encontrar relaciones ontológicas en documentos descriptivos de objetos concretos del mundo real, en especial herramientas, escritos en lenguaje natural.

Se desarrolla una adaptación del método de búsqueda de patrones para agregarle un sistema de reglas que permite que las relaciones encontradas se extiendan a relaciones específicas buscadas dentro de un documento.

Se utilizan técnicas de análisis de lenguaje natural para procesar el texto, posteriormente se encuentran las relaciones buscadas en base a un archivo de reglas y se genera una salida en lenguaje OM.

Como parte de esta tesis se desarrolló el Sistema de extracción y representación del conocimiento a partir de documentos descriptivos (SERCDD), el cual implementa el sistema de reglas.

Las pruebas se realizan en el idioma español, sin embargo es posible extender el método a cualquier idioma.

ABSTRACT

This document presents a method based on finding patterns in text and the use of a set of rules to find ontological relations in descriptive documents for specific real-world objects, specially tools, such documents are written in natural language.

We develop an adaptation of pattern search method by adding a rule system to allow the method to find extra relationships.

The use of natural language processing tools enables the system to find relationships based on defined rules. Relations found are represented in OM ontology language.

As part of this work we developed a software called knowledge extraction and representation from descriptive documents system.

The tests are performed in Spanish, nevertheless it is possible to extend our method to any other language.

INTRODUCCIÓN

Extraer el conocimiento directamente del lenguaje natural de manera automática es una tarea interesante que brinda la posibilidad de obtener conocimiento de manera sencilla sin la necesidad de un experto. Porque el proceso requiere un equipo de trabajo y grandes cantidades de tiempo y esfuerzo. El principal problema es que el lenguaje natural tiene como característica inherente la ambigüedad, que los humanos resuelven no sólo con el contexto, sino con la experiencia del mundo real y el sentido común, ninguno de los cuales hasta ahora se ha implementado con amplitud de uso en las computadoras.

La idea de obtener conocimiento automáticamente de cualquier texto y poder transformarlo a una representación que la computadora entienda es un problema abierto que incluye a diversas áreas del conocimiento, principalmente procesamiento de lenguaje natural <<NLP>> (Jurafsky & Martin, 2009) y adquisición de conocimiento <<KA>> (Musen, 1993). La presente investigación se enfoca en el proceso de aprendizaje de ontologías y ha sido delimitada a tratar con textos o fragmentos descriptivos dentro de un texto. Las ontologías son especificaciones formales y explícitas, en forma de conceptos y relaciones, de concepciones compartidas (Gruber, 1993).

Existen diversos trabajos en el área de aprendizaje de ontologías enfocados en aprender ontologías a partir de texto. El lenguaje humano es el medio principal de transmisión del conocimiento, por lo que el proceso de aprendizaje de ontologías a partir de colecciones de texto relevantes es, sin duda, una opción viable, como se ilustra por el gran número de sistemas que están basados en este principio, e.g. ASIUM (Faure & Poibeau, 2000), TextToOnto (Maedche & Staab, Ontology Learning for the Semantic Web, 2001), OntoLearn (Velardi, Navigli, & Missikoff, 2002). Todos estos combinan cierto nivel de análisis lingüístico con algoritmos de aprendizaje automático para encontrar conceptos potencialmente interesantes y relaciones entre ellos.

Se propone un método para aprendizaje de ontologías, el cual parte de un texto en con un previo análisis sintáctico. En él se busca información específica mediante un sistema de reglas en dos partes, 1)patrones a buscar dentro del texto los cuales describen una relación y 2)encontrado el patrón, la regla a seguir para encontrar los elementos relacionados. La idea es saber la información (relaciones) que se espera obtener del texto, e identificarla basándose en la estructura del corpus. Por ejemplo, en un texto que habla de alguna herramienta de carpintería, esperaremos encontrar su definición, usos más comunes y quizá materiales o clasificación. Los patrones a buscar pueden incluir la forma de las palabras, la etiqueta gramatical, el lema o su categoría semántica. La búsqueda de una categoría semántica mejora la extracción de información, al permitir buscar relaciones entre palabras, según la semántica de las mismas. Las categorías semánticas que se tienen son las definidas en la notación de marcos especificada por el grupo de investigación OM, la cual está basada en los marcos de Minsky (Minsky, 1974).

Una vez encontrados los patrones dentro del texto, se utiliza la regla definida para encontrar los elementos de una relación binaria, dependiendo de su clase gramatical, posición en la oración o semántica.

Finalmente se obtiene una representación formal del conocimiento extraído, dicho conocimiento es presentado en una ontología en lenguaje OM (Cuevas Rasgado, 2006).

1. PROBLEMA DE INVESTIGACIÓN

1.1. Planteamiento del problema

Existe información textual en la web que contiene grandes cantidades de conocimiento en distintas áreas, el volumen de información sigue creciendo y no existe una manera definitiva de organizar todo ese conocimiento para aprovecharlo eficientemente. Los textos que abundan en la web están diseñados para que un humano los lea y entienda antes de poder extraer de ellos la información que está buscando. Es necesario contar con un método para que toda esa información sea "entendida" por una computadora y se pueda recopilar el conocimiento de manera estructurada. Actualmente los buscadores son la manera universal de encontrar "respuestas", de manera que se presenta al usuario con un número de documentos para que sea él quien finalmente busque la respuesta que está buscando. Con una fuente de conocimiento estructurado, se provee la oportunidad de entregar una respuesta concreta a una pregunta no trivial.

1.2. Justificación

Las ontologías se han convertido en un medio importante para representar información estructurada y como base de conocimiento para sistemas de información. La razón para ello es que las ontologías tienen la ventaja sobre otras representaciones de no ser ambiguas y pueden utilizarse para realizar deducciones y obtener respuestas a preguntas complejas. Partiendo de una ontología las deducciones pueden ser hechas por un programa de computadora de manera automática. Sin embargo, el problema de crear ontologías grandes y adecuadas en poco tiempo y con un costo reducido sigue siendo una gran limitación en su uso y funcionalidad. Existe mucha investigación al respecto, sin embargo la mayoría de los sistemas derivados de dicha investigación requieren de supervisión humana para la generación o expansión de ontologías, lo cual sigue causando aumento en tiempo y costos para tal proceso. Por lo anterior es relevante lograr que el proceso pueda realizarse sin necesidad de supervisión humana, el presente trabajo proporciona un método no supervisado para la generación de ontologías a partir de textos

en lenguaje natural. La finalidad de tener esta clase de sistemas es lograr dar a las computadoras la capacidad de "leer" y "entender" información textual para posteriormente explotarla y pueda "explicar" la información recopilada, por ejemplo en sistemas que puedan dar respuesta a preguntas no triviales. Esta área conocida como búsqueda de respuestas (QA por sus siglas en inglés) (Lampert, 2004).

1.3. Alcances

La entrada al sistema desarrollado en esta tesis son a textos descriptivos o fragmentos descriptivos de textos en general. Su característica principal es que su contenido está expresado en oraciones del modo indicativo y que carecen de muchas formas lingüísticas que aumentan el grado de ambigüedad y confusión en los textos de lenguaje natural. Existen otras formas lingüísticas que pueden ser exploradas en otros trabajos, como el modo subjuntivo, imperativo e interrogativo.

No se atacan problemas que existen en el área del procesamiento de lenguaje natural como la resolución de anáforas, ambigüedad gramática y desambiguación de sentidos.

Las ontologías generadas no incluirán la extracción de reglas y axiomas, lo cual es el último nivel en el modelo de capas de aprendizaje de ontologías. Este trabajo se limitará a la extracción de términos, sinónimos, conceptos de la taxonomía y relaciones.

1.4. Objetivos

1.4.1. Objetivo general

Construir un sistema de aprendizaje de ontologías que no requiera de la interacción del usuario y que genere ontologías en lenguaje OM transformando oraciones de textos en lenguaje natural que describen objetos concretos. Realizar la transformación utilizando técnicas de extracción del conocimiento.

1.4.2. Objetivos específicos

- Encontrar y extraer conceptos encontrados en textos descriptivos en lenguaje natural.
- Identificar la taxonomía (relaciones de hiperonimia e hiponimia)descrita en documentos descriptivos en lenguaje natural.
- Encontrar y extraer sinónimos encontrados en textos descriptivos en lenguaje natural.
- Encontrar y extraer relaciones entre conceptos encontrados en textos descriptivos en lenguaje natural.
- Representar en el lenguaje de ontologías OM los conceptos, taxonomía y relaciones extraídos de documentos descriptivos en lenguaje natural.

2 MARCO TEÓRICO

2.1 Textos descriptivos

De manera general, un texto descriptivo es definido como aquel que consiste en la representación verbal real de un objeto, persona, paisaje, animal, emoción, y prácticamente todo lo que pueda ser puesto en palabras. Este tipo de texto pretende que el lector obtenga una imagen exacta de la realidad que estamos transmitiendo en palabras, una especie de "pintura verbal".

La clasificación de los textos descriptivos según el tipo de descripción que realizan considera la descripción técnica y la descripción literaria. Los documentos analizados en el presente trabajo se basan en descripciones técnicas. En la descripción técnica es fundamental que la objetividad siempre sea respetada para que la información no sea distorsionada por algún punto de vista u opinión. El lenguaje que se utilizará es frío, con palabras técnicas que sólo apuntan a explicar una característica de lo que se intenta representar. Su fin es dar a conocer un objeto: sus partes, su funcionamiento y su finalidad. (Álvarez, 1988)

Las características que se asume poseen los textos descriptivos a analizar son:

- Predominio de sustantivos y adjetivos
- Las formas verbales más utilizadas son el presente y el pretérito imperfecto de indicativo.
- Predominan los verbos estativos (ser y estar) y los pertenecientes a los campos semánticos de los cinco sentidos.
- El lenguaje usado es de valor denotativo, ausente de connotaciones.
- Abundancia de coordinación y yuxtaposición.
- Atemporalidad
- La Información es presentada de manera precisa y objetiva.

En la Tabla 1 se muestra una comparativa de textos descriptivos útiles para el análisis contra textos en otros modos, de los cuales la extracción de información será poco eficiente.

Texto adecuado para el análisis	Texto inadecuado para el análisis
El martillo es una herramienta utilizada para golpear una pieza, causando su desplazamiento o deformación.	**El martillo es una herramienta básica para cualquier persona que realice trabajos de bricolaje, este no puede faltar en el maletín básico.**
El martillo es una herramienta útil en tareas de albañilería, carpintería, fontanería, electricidad, etc.	**Independientemente de que vaya usted a realizar alguna reparación o no, el martillo, es una herramienta imprescindible en cualquier hogar, pues es sumamente útil para realizar infinidades de tareas.**

Tabla 1 Texto adecuado vs Texto inadecuado para el análisis

2.2 Representación del conocimiento

La representación del conocimiento es un área de la inteligencia artificial cuyo objetivo fundamental es representar el conocimiento de una manera que facilite hacer inferencias a partir del conocimiento.

La idea detrás de buscar una manera de representar el conocimiento es tener una manera de razonar, inferir u obtener conclusiones tal como lo hacen los humanos.

Para comprender el concepto se han listado cinco roles que juega: (Davis, Shrobe, & Szolovits, 1993)

- Es un sustituto para la entidad en sí.
- Es un conjunto de compromisos ontológicos.
- Es una teoría fragmentaria del razonamiento inteligente
- Es un medio para realizar cómputo eficiente.
- Es un medio de la expresión humana.

Existen distintas técnicas para la representación del conocimiento, como son marcos, reglas y redes semánticas.

En el desarrollo de este trabajo no se discutirán en detalle las ventajas y desventajas de cada enfoque de representación del conocimiento, se ha elegido el enfoque de representar el conocimiento en un primer nivel de ontología, por ser una representación no ambigua que da la ventaja de poder ser leída por una computadora y realizar deducciones para contestar preguntas no triviales. El concepto de ontologías está explicado con mayor detalle en la siguiente sección.

2.2.1 Ontologías

La ingeniería del conocimiento, y en particular el procesamiento de ontologías, son uno de los problemas en los que más se está centrando la atención en la informática actual. La razón principal es el surgimiento de un nuevo conjunto de aplicaciones en las que las ontologías juegan un papel fundamental, la mayoría de las cuales se encuentran en dos campos: la última generación de sistemas para la Web (Web 2.0, Web Semántica) y la necesidad de integración de aplicaciones empresariales cada vez más distribuidas.

El término ontología fue usado por primer vez en el área de la filosofía que estudia el ser o existencia, sin embargo el término ha tomado un significado nuevo al haber sido utilizado en la ciencia de la computación con el significado generalmente aceptado de ser una especificación de una conceptualización. Existen distintas definiciones de ontología en el área de la inteligencia artificial, sin embargo la definición con mayor aceptación es:

> *"Una conceptualización es una visión abstracta y simplificada del mundo que deseamos representar con algún propósito. Cada base de conocimiento, sistema basado en conocimiento o agente a nivel del conocimiento está comprometido a cierta conceptualización, explícita o implícitamente. Una ontología es una especificación explicita de una conceptualización. El término se toma de la filosofía, donde una ontología es una explicación sistemática de la Existencia. Para los sistemas de IA, lo que "existe" es lo que puede ser representado"* (Gruber, 1993)

La definición está basada en el concepto de conceptualización propuesto en (Genesereth & Nilsson, 1987) y que en esencia se refiere al conjunto de conceptos y relaciones entre conceptos que son relevantes en un dominio o una aplicación.

La definición utilizada en la tesis de Cuevas Rasgado (Cuevas Rasgado, 2006), en la cual se especifica el lenguaje OM es la siguiente:

Es una representación formal de información de cierto tipo, simbolizada a través de un hipergrafo dirigido donde cada vértice es un concepto que se encuentra enlazado a través de aristas que aquí se llaman relaciones.

Se establece que una ontología por definición debe tener tres características fundamentales: (Studer, Benjamins, & Fensel, 1998)

- Es **formal**, esta característica se refiere a que puede ser entendida por una máquina.
- **Explicita**, es decir que los conceptos y las restricciones en su uso están definidas explícitamente.
- Es una conceptualización **compartida**, contiene conocimiento consensuado aceptado por un grupo.

2.2.2 Lenguaje OM

El lenguaje OM es propuesto en (Cuevas Rasgado, 2006) con la finalidad de diseñar ontologías con conceptos y relaciones que proporcionan más semántica a las operaciones de búsqueda de conocimiento.

En este trabajo, se ha elegido el lenguaje de ontologías OM sobre otros, como OWL (W3C, OWL 2 Web Ontology Language Document Overview, 2009) y RDF (W3C, Resource Description Framework (RDF), 2004) porque en OM se pueden representar con más facilidad relaciones de cualquier número de componentes mediante el uso de hipergrafos. lo cual da la capacidad de escalabilidad al presente trabajo.

La estructura semántica de las ontologías definidas en este lenguaje, es a través de un conjunto de etiquetas que identifican el concepto y sus relaciones, por ejemplo, <concept> que indica el nombre del concepto. Esta etiqueta permite el anidamiento de conceptos, <language> que representa el lenguaje del concepto, <word> donde se encuentran las palabras que definen al concepto y <relation>que representa el tipo de relación que conecta al concepto.

Las relaciones en OM pueden ser implícitas y explícitas:

- Implícitas: son las relaciones expresadas por el anidamiento, el concepto externo se reconoce como antecesor mientras que el interno como sucesor. Estas relaciones son: member, part, part* y subset.
- Explícitas: son las que se encuentran definidas entre las etiquetas <relation></relation>, puede ser una actividad (eats), propiedad (color) o atributo del concepto. No se permiten relaciones anidadas.

Existe otro tipo de relación llamada Partición, esta se diferenciará con la palabra "Partition" y contiene una estructura diferente a las relaciones explícitas. Una partición es un conjunto cuyos subconjuntos están bien definidos por un rango o intervalo. Cada instancia de un subconjunto no puede ser instancia de otro dentro de la misma partición, es decir cada subconjunto de la partición son mutuamente exclusivos y colectivamente exhaustivos.

2.3 Aprendizaje de ontologías (Ontology Learning)

Obtener conocimiento general o de dominio para construir ontologías requiere de mucho tiempo y recursos, ya que la mayoría de las ontologías son construidas a mano según las necesidades específicas de la aplicación que las utilice. Por esta razón se tiene la necesidad de desarrollar métodos y técnicas que permitan reducir el esfuerzo necesario para completar el proceso de adquisición del conocimiento, este precisamente es el objetivo del aprendizaje de ontologías.

El aprendizaje de ontologías es inherentemente multidisciplinario, y diversas técnicas de múltiples campos (como recuperación de la información, minería de datos, aprendizaje automático, adquisición de conocimiento y procesamiento de lenguaje natural) han sido fundamentales dentro del desarrollo de sistemas de aprendizaje de ontologías.

Alexander Maedche y Steffen Staab (Maedche & Staab, Ontology Learning for the Semantic Web, 2001) identifican diversos enfoques del aprendizaje de ontologías según el tipo de entrada :

- Texto en lenguaje natural
- Diccionarios
- Bases de conocimiento
- Datos semi-estructurados
- Esquemas de relación

El presente trabajo se concentra en el primer enfoque, el cual se analiza con mayor detalle a continuación y en la sección 3 Trabajos Previos se discuten y comparan diversos esfuerzos por resolver el problema.

2.3.1 Aprendizaje de ontologías a partir de texto

El aprendizaje de ontologías a partir de texto es el proceso de identificar términos, conceptos, relaciones y opcionalmente axiomas a partir de información textual y utilizar esta información para construir y mantener una ontología. El proceso consiste en aplicar técnicas de procesamiento de lenguaje natural a un corpus para extraer ontologías de éste.

Los métodos más conocidos dentro de este enfoque son:

- Extracción basada en patrones (Morin, 1999) (Hearst, 1992). Se reconoce una relación cuando una secuencia de palabras en un texto, corresponde con un patrón.

- Reglas de asociación .Se parte del concepto de regla de asociación que se maneja en el área de las bases de datos para realizar minería de datos (Agrawal, Imieliński, & Swami, 1993), y siguiendo la misma idea se buscan relaciones no taxonómicas en textos donde ciertos ítems (conceptos) tienen a aparecer juntos, se utiliza una taxonomía de conceptos como conocimiento previo (Maedche & Staab, Discovering Conceptual Relations from Text, 2000).
- Agrupamiento conceptual (Faure & Poibeau, 2000). Se agrupan conceptos de acuerdo a la distancia semántica que existe entre ellos para hacer jerarquías.
- Poda de ontologías (Kietz, Volz, & Maedche, 2000). Su objetivo es construir una ontología de dominio a partir de fuentes heterogéneas. Consiste en utilizar una ontología genérica como una estructura de nivel superior para la ontología de dominio, posteriormente a partir de un diccionario que define en lenguaje natural conceptos importantes del dominio se extraen los conceptos relevantes al dominio, estos conceptos se clasifican dentro de la ontología genérica. Finalmente se utilizan tanto textos generales como específicos al dominio para eliminar aquellos conceptos que no pertenezcan al dominio de interés. La eliminación de los conceptos se basa en la heurística de que los conceptos de dominio serán más frecuentes en textos específicos al dominio que en los textos generales.
- Aprendizaje de conceptos (Hahn & Schulz, 2000). Dada una taxonomía existente, se actualiza incrementalmente agregado conceptos extraídos de textos tipo tesauro.

2.3.2 Etapas en el aprendizaje de ontologías

El desarrollo de ontologías está particularmente enfocado a la definición de conceptos y las relaciones entre ellos, pero conectado a esto se tiene el conocimiento acerca de los símbolos que son utilizados para referirse a estos. En nuestro caso, esto implica la adquisición de conocimiento lingüístico acerca de términos que son usados para referirse a un concepto específico dentro de un cierto texto y posiblemente los sinónimos de dicho término. Adicionalmente, una ontología consiste en una taxonomía global (relaciones del tipo es-un) y otras relaciones no jerárquicas. Finalmente, para poder derivar hechos que

no están explícitamente codificados dentro de la ontología, pero que pueden derivarse de ella, deben definirse reglas (y de ser posible adquirirlas) que permitan realizar tales derivaciones.

Todos los aspectos anteriores del proceso de desarrollo de una ontología pueden ser organizados como un pastel en capas, cada capa representando una tarea de complejidad mayor a la anterior, como se ilustra en la Figura 1 (Cimiano, 2005)

Figura 1 Modelo de los niveles del aprendizaje de ontologías (Cimiano, 2005).

3 TRABAJOS PREVIOS

La mayoría de los trabajos que intentan obtener una ontología a partir del lenguaje natural consisten en procesos semi-supervisados, en los que un experto indica los errores cometidos al sistema o guía el proceso indicando las secciones del texto que contienen el conocimiento relevante. A continuación se describen brevemente algunos de los trabajos existentes que generan una representación formal del conocimiento a partir de textos en lenguaje natural. Al final de la sección se hace una comparación de las herramientas descritas y algunas otras en una tabla resumen.

3.1 OntoLT

OntoLT (Buitelaar, Olejnik, & Sintek, 2003) es un plug-in para la herramienta de desarrollo de ontologías Protégé el cual soporta la extracción interactiva y/o extensión de ontologías a partir de texto.

El plug-in permite la definición de reglas de mapeo con las cuales los conceptos (clases de Protége) y los atributos (slots de Protégé) pueden ser extraídos automáticamente a partir de colecciones de textos anotadas lingüísticamente.

El proceso de extracción de la ontología que utiliza OntoLT es el siguiente: OntoLT proporciona un lenguaje de precondiciones con el cual el usuario puede definir reglas de mapeo. Las precondiciones son implementadas como expresiones XPATH sobre una anotación basada en XML. Si todas las condiciones se satisfacen las reglas de mapeo se activan y uno o más operadores que describen de qué forma se debe extender la ontología es encontrado.

Las precondiciones seleccionan por ejemplo el predicado de una oración, su sujeto lingüístico o su objeto directo. Las precondiciones también pueden ser utilizadas para verificar ciertas condiciones sobre las entidades lingüísticas, por ejemplo si el sujeto de una oración corresponde a un lema en particular (la raíz morfológica de una palabra). El lenguaje de precondiciones consiste de Términos y Funciones.

Las entidades lingüísticas seleccionadas pueden ser utilizadas para construir o extender una ontología. Para este fin, OntoLT provee operadores para crear clases, slots e instancias. Según las precondiciones que se satisfagan, se activarán los operadores correspondientes para crear un conjunto de clases candidatas y slots que deben ser validados por el usuario. Los candidatos validados se integran a una ontología nueva o bien, a una ya existente.

3.2 Text-To-Onto

Text-To-Onto (Maedche & Volz, 2001) integra un ambiente para construir ontologías de dominio a partir de una ontología núcleo. También descubre estructuras conceptuales de distintas fuentes de conocimiento utilizando técnicas de adquisición del conocimiento y aprendizaje de máquinas. Text-To-Onto ha implementado algunas técnicas para el aprendizaje de ontologías a partir de texto libre y texto semi-estructurado, diccionarios, ontologías y bases de datos. El resultado del proceso de aprendizaje es una ontología de dominio que contiene conceptos específicos al dominio, así como conceptos independientes del dominio. Los conceptos independientes del dominio son retirados para ajustar mejor el vocabulario de la ontología de dominio. El resultado del proceso es una ontología de dominio que contiene conceptos del dominio aprendidos a partir de las fuentes de entrada mencionadas anteriormente. Todo el proceso es supervisado por el encargado de crear la ontología. Este es un proceso cíclico, en el sentido de que es posible refinar y completar la ontología si se repite el proceso.

3.3 Terminae

Terminae (Biébow & Szulman, 1999) (Aussenac-Gilles, Despres, & Szulman, 2008) integra herramientas lingüísticas y de ingeniería del conocimiento. La herramienta lingüística permite definir formas terminológicas a partir del análisis de las ocurrencias de términos en un corpus. El supervisor de la creación de las ontologías (ontologist) analiza los usos del término en el corpus para definir el significado de cada término. La herramienta de ingeniería del conocimiento involucra un editor y un navegador para la ontología. La

herramienta ayuda a representar formas terminológicas como un concepto (llamado concepto terminológico).

Terminae utiliza un método para construir conceptos a partir del estudio del término correspondiente en un corpus. Primero, la herramienta establece una lista de términos, lo cual requiere la constitución de un corpus relevante en el dominio. Utilizando una herramienta de extracción de términos, un conjunto de términos candidatos se propone al supervisor, quien selecciona un subconjunto de éstos. Posteriormente, el supervisor conceptualiza los términos y analiza los usos del término en el corpus para definir todos los significados de dicho término. El supervisor da una definición en lenguaje natural para cada significado, y después traduce la definición a un lenguaje de implementación.

3.4 Sistema de clasificación de términos basado en TFIDF

Este sistema tiene como objetivo detectar términos relevantes en cierto dominio y encontrar las relaciones que se guardan entre ellos (Xu, Kurz, Piskorski, & Schmeier, 2002). Esta herramienta cuenta con los siguientes componentes principales:

- Un clasificador de términos de una sola palabra que es utilizado para extraer palabras sencillas de un corpus.
- Un buscador de patrones léxico-sintácticos que tiene dos sub-módulos; el primero es para aprender patrones basados en un conjunto de relaciones conocidas (utilizando GermaNet o WordNet) e implementa las interfaces necesarias para inter-operar con estos dos sistemas. El segundo es para aprender patrones basados en métodos de co-localización de términos.
- Un extractor de relaciones.

El sistema recibe como entrada un corpus de dominio, que está anotado y con un previo análisis sintáctico (parsing) utilizando una herramienta de "shallow NLP". En este caso la herramienta utilizada es SPPC (Xu, Kurz, Piskorski, & Schmeier, 2002). Esta herramienta de lenguaje natural provee una extracción independiente de dominio para procesar documentos en alemán.

3.5 SOAT

SOAT (Wu & Hsu, 2002) es una herramienta de adquisición de ontologías de dominio semi-automática. El objetivo principal de la herramienta es extraer relaciones a partir de oraciones analizadas sintácticamente (parsed) basadas en aplicar reglas de frases para identificar palabras clave con enlaces semánticos fuertes como hiperónimos o sinónimos. El proceso de adquisición está basado en el uso de InfoMap (Hsu, Wu, & Chen, 2001), un framework de representación del conocimiento que integra conocimiento lingüístico, de sentido común y de dominio. InfoMap ha sido desarrollado para comprender el lenguaje natural y capturar palabras del tema, usualmente pares de verbo y sustantivo o sustantivo y sustantivo en una oración. InfoMap tiene dos relaciones principales entre conceptos: relaciones taxonómicas (categoría y sinónimo) y no taxonómicas (atributo y evento).

El proceso de adquisición que realiza SOAT incluye el obtener palabras claves del dominio y encontrar las relaciones entre ellas. Para realizar esto, un conjunto de reglas se ha definido para extraer las palabras clave de una oración relacionada con los conceptos de InfoMap con una fuerte relación semántica entre ellas. La herramienta recibe como entrada un corpus de dominio etiquetado con la clase gramatical. Una palabra clave, usualmente el nombre del dominio, se selecciona dentro del corpus como la raíz. Después, con estas palabras clave el proceso intenta encontrar una nueva palabra clave relacionada con la anterior mediante la aplicación de las reglas de extracción y agregar la nueva palabra clave a la ontología según las reglas y la estructura fija en InfoMap. Esta nueva palabra clave se toma ahora como la raíz para repetir el proceso durante un determinado número de veces o hasta que sea imposible encontrar una palabra clave relacionada. La intervención del usuario es necesaria para verificar los resultados de la adquisición y para refinar y actualizar las reglas de extracción. Las restricciones de SOAT radican en que el corpus debe tener una calidad muy alta en el sentido en que las oraciones deben ser certeras y suficientes para contener la mayoría de las relaciones importantes que serán extraídas.

3.6 Comparación entre sistemas de aprendizaje de ontologías

En la Tabla 1 se hace un resumen de las características más importantes de las herramientas descritas anteriormente, además de algunas otras que debido a su fecha de publicación o su alta dependencia de interacción con el usuario no se describen en detalle.

Las características que se describen son:

- Nombre de la herramienta
- El objetivo para el cual fue desarrollada y el alcance de la misma
- La técnica de aprendizaje de ontologías que implementa
- El método que está utilizando para realizar el aprendizaje
- Las fuentes a partir de las que se realiza el aprendizaje, estas pueden ser textos en lenguaje natural, textos que ya fueron analizados previamente u ontologías las de cuales se parte para agregar conocimiento
- El nivel de intervención requerido por parte del usuario, el menor nivel es en el que el usuario únicamente evalúa las ontologías obtenidas
- La referencia de los trabajos que describen a mayor detalle la herramienta.

Nombre	Objetivo y alcance	Técnica de aprendizaje	Método seguido para aprender	Fuentes	Intervención del usuario	Referencia
ASIUM	Aprender relaciones taxonómicas	Técnicas de agrupamiento conceptual	Método propio	Texto analizado sintácticamente	Durante todo el proceso	Faure et al 2000, 1999, y 1998
Corporum-Ontobuilder	Extraer una taxonomía inicial	Técnicas semánticas y lingüísticas	Método propio	Texto	No es necesaria	Engels 2001 y 2000
DOE	Ayudar al desarrollador de ontologías durante el proceso de construcción	Semántica diferencial	Método de Bachimont	Texto en Lenguaje Natural	Durante todo el proceso	Bachimont 2000
MO'K Workbench	Aprender una taxonomía de conceptos	Agrupamiento conceptual	Método propio	Texto etiquetado	Durante todo el proceso	Bisson et al. 2000
OntoLearn	Enriquecer una ontología de dominio	Técnicas de Lenguaje Natural. Técnicas de aprendizaje de máquinas	Método de Missikoff y colegas	Texto en lenguaje natural	Evaluación	Velardi et al., 2002 y 2001
OntoLT	Extracción de conceptos y atributos para crear o enriquecer ontologías	Enfoque estadístico Definición de patrones lingüísticos	Método propio	Texto anotado lingüísticamente	Durante todo el proceso	Buitelaar et al., 2004
SOAT	Adquisición de relaciones	Patrones de frases	Método propio	Texto en lenguaje natural	No se especifica	Wu et al 2002
TFIDF	Aprender una jerarquía de conceptos y relaciones entre ellos	Minería de texto Enfoque estadístico	Enfoque de minería de texto para adquirir los términos de dominio	Texto en lenguaje natural	Evaluación	Xu et al., 2002
Terminae	Construir una ontología inicial	Agrupamiento conceptual	Método propio	Texto en lenguaje natural	Validación	Biébow y Szulman 1999
Text-To-Onto	Encontrar relaciones taxonómicas y no taxonómicas	Enfoque estadístico Técnicas de poda Reglas de asociación	Método de Kietz y colegas	Texto en lenguaje natural Diccionarios Ontologías	Validaciones	Maedche y Volz, 2001

Tabla 1 Comparativa de herramientas de aprendizaje de ontologías a partir de tex

4 IMPLEMENTACIÓN

4.1 Arquitectura

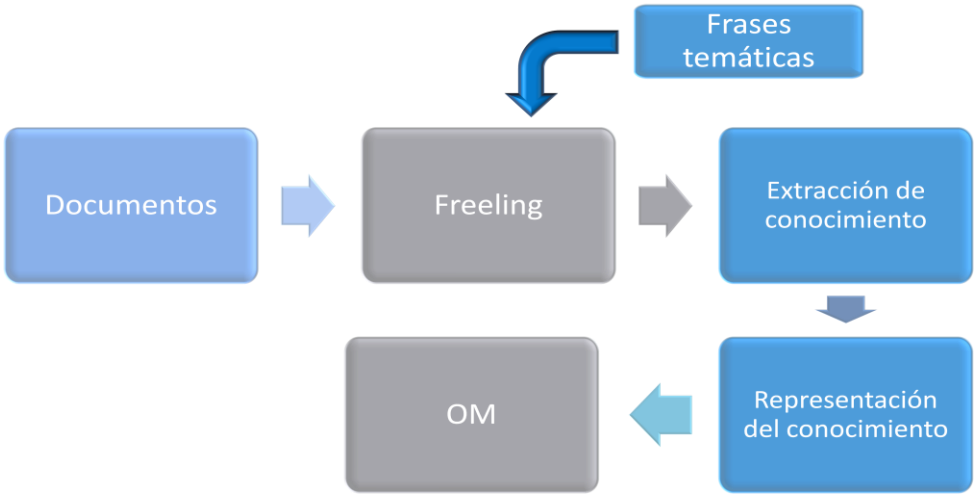

Figura 1 Diagrama general del SERCDD y sus interacciones con otros sistemas

La Figura 2 muestra la arquitectura del Sistema de Extracción y Representación del Conocimiento, SERCDD.Los componentes del sistema son:

1. Documentos: La entrada al sistema son documentos de texto, escritos en lenguaje natural, para la óptima extracción de la información, dichos documentos deberán ser descriptivos, es decir, contar con las características que se definieron en la sección 2.1.
2. Freeling: Librería que cuenta con las funciones de procesamiento de lenguaje natural, las cuales son las encargadas de realizar la segmentación y el etiquetado de los documentos, la salida de este componente son una serie de oraciones divididas en palabras, cada una con una única etiqueta de clase gramatical.
3. Frases temáticas: Este componente es un diccionario de frases temáticas, las cuales son incluidas al diccionario de Freeling para que dentro de su analisis logré identificar mas multi-palabras. Este diccionario es el recopilado en el trabajo de extractor de frases temáticas (Meneses, 2011).

4. Extracción del conocimiento: La entrada a este módulo es la salida de Freeling, la cual es procesada para encontrar los patrones que determinaran las relaciones a extraer (hiponimia, hiperonimia, sinonimia, etc.).
5. Representación del conocimiento: Una vez identificados los componentes del conocimiento necesarios para construir la ontología, estos son representados en el lenguaje de ontologías OM, identificando la taxonomía de conceptos, así como las relaciones extraídas para poder representar los conceptos en notación OM.
6. Sistema OM: OM recibirá como entrada las ontologías de cada documento para fusionarlas en una ontología general.

4.2 Diagrama de componentes

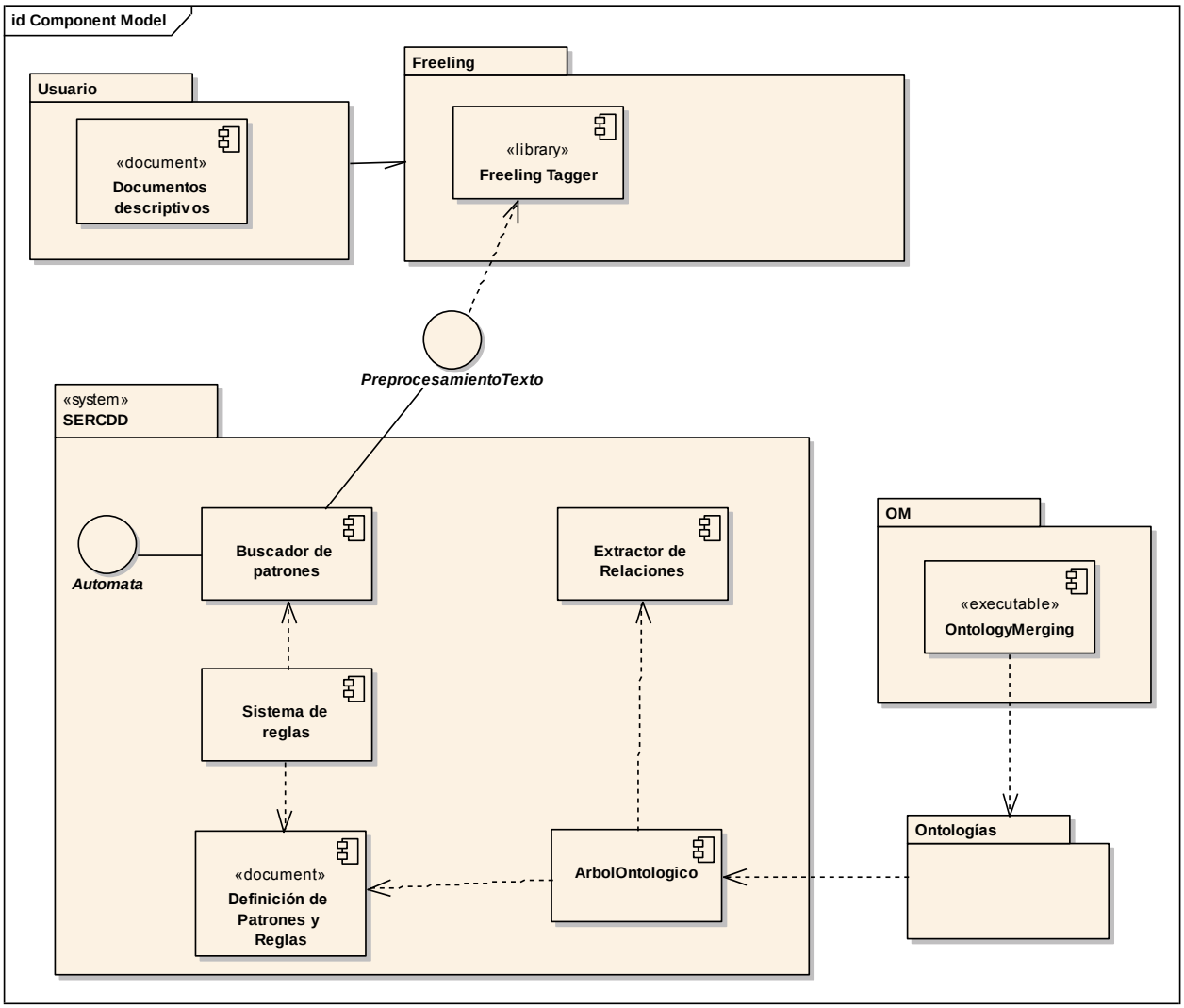

Figura 2 Diagrama de Componentes

4.3 Diagrama de clases

Se presenta el diseño inicial de las clases necesarias para realizar el proceso de la extracción del conocimiento a partir de documentos y plasmarlos en las ontologías en lenguaje OM.

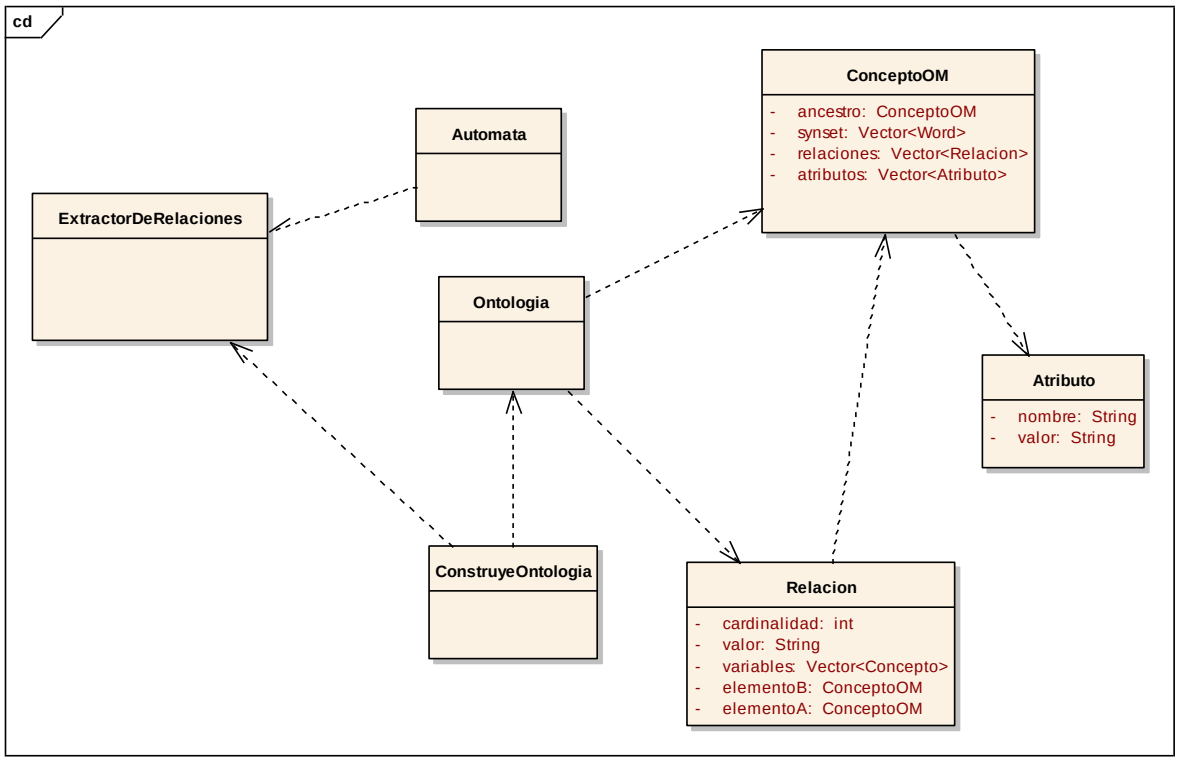

Figura 3 Diagrama de clases

4.4 Frases temáticas

Dado que existen muchos conceptos que son formados por más de una palabra e incluso se pueden tener conceptos complejos dentro de distintos ámbitos o áreas del conocimiento, se considera la posibilidad de anexar un archivo creado por el usuario en el cual se definen los conceptos que se esperan encontrar en un documento. La definición de conceptos dentro del archivo debe tener el siguiente formato:

frase_temática etiqueta_gramatical

donde:

frase_temática: es la cadena que se considerará dentro del análisis sintáctico como una sola palabra y por lo tanto en análisis para la extracción del conocimiento será un solo concepto o nodo en la ontología.

etiqueta_gramatical: etiqueta de clase gramatical acorde al estándar EAGLES que el término deberá llevar en el análisis morfológico.

La Figura 5 muestra un un fragmento de un posible archivo de frases temáticas y su formato correcto.

```
cuchillo_de_cirujano NCMS000
procedimiento_quirúrgico NCMS000
ciudad_de_la_esperanza NPFS000
metro_Balderas NPMS000
conducto_hepático NCMS000
plexo_solar NCMS000
```

Figura 4 Ejemplo de archivo de términos semilla

El uso de un archivo de frases temáticas es opcional y su ruta se puede definir en el archivo de configuraciones del sistema.

4.5 Archivo de configuración

El archivo de configuración consiste principalmente de configuraciones de los parámetros del sistema de PLN Freeling, a continuación de muestra un ejemplo del archivo utilizado con los valores recomendados para el análisis, en la última sección se agregan las configuraciones propias del SERCDD.

El parámetro SynonymsFirst indica que el análisis de búsqueda de patrones se realiza en dos etapas, en la primera se buscan las relaciones de sinonimia del objeto que el documento a analizar describe, para que en la segunda etapa, las ocurrencias de cualquiera de las palabras identificadas como sinónimos, sean consideradas como el mismo objeto que se está describiendo.

```
Path=/usr/local
FreelingPath=/share/freeling
Lang=es
Locale=default
#### Tokenizer options
TokenizerFile=/es/tokenizer.dat
#### Splitter options
SplitterFile=/es/splitter.dat
#### Morfo options
AffixAnalysis=true
MultiwordsDetection=true
NumbersDetection=true
PunctuationDetection=true
DatesDetection=true
QuantitiesDetection=false
DictionarySearch=true
ProbabilityAssignment=true
OrthographicCorrection=false
DecimalPoint=,
ThousandPoint=.
LocutionsFile=/es/locucions.dat
QuantitiesFile=/es/quantities.dat
AffixFile=/es/afixos.dat
ProbabilityFile=/es/probabilitats.dat
DictionaryFile=/es/dicc.src
PunctuationFile=/common/punct.dat
ProbabilityThreshold=0.001
# NER options
NERecognition=true
NPDataFile=/es/np.dat
## comment line above and uncomment that below, if you want
## a better NE recognizer (higer accuracy, lower speed)
#NPDataFile=/es/ner/ner-ab.dat
#Spelling Corrector config file
CorrectorFile=/es/corrector/corrector.dat
#### NEC options
NEClassification=false
NECFile=/es/nec/nec-svm.dat
#### Tagger options
Tagger=hmm
TaggerHMMFile=/es/tagger.dat
TaggerRelaxFile=/es/constr_gram.dat
TaggerRelaxMaxIter=500
TaggerRelaxScaleFactor=670.0
TaggerRelaxEpsilon=0.001
TaggerRetokenize=true
#TaggerForceSelect=tagger
TaggerForceSelect=true
#### Parser options
GrammarFile=/es/grammar-dep.dat
#### Dependence Parser options
DepTxalaFile=/es/dep/dependences.dat
###SERCDD Configuration options
SeedTermsFile=reglas/semillas.txt
OMRulesFile=reglas/reglas.txt
OMSynRulesFile=reglas/reglasSyn.txt
SynonymsFirst=true
```

Figura 5 Archivo de configuración del sistema

4.6 Pre procesamiento de texto

El sistema recibe como entrada inicial documentos de texto, los cuales se encuentran escritos en lenguaje natural. Los textos son procesados utilizando una herramienta de procesamiento de lenguaje natural llamada Freeling [24]. Sobre el procesamiento realizado por Freeling se pasa una segunda etapa de procesamiento dentro de SERCDD para generar la estructura del documento, ya que por la naturaleza de los documentos descriptivos se puede esperar cierta organización del documento en secciones, titulos, listas, enumeraciones, etc.

Las etapas de pre procesamiento a las que se someten los documentos descriptivos son:

- Segmentación (tokenization): Se segmenta el texto en palabras, oraciones, párrafos y secciones Para la segmentación en palabras se utiliza un diccionario de conceptos multi palabra, el cual ha sido aumentado con frases temáticas extraídas de [23].

- Etiquetado: Se etiqueta el texto con la clase gramatical de cada palabra según la especificación del tagset definido por el estándar EAGLES, el etiquetador se configura para asignar una sola etiqueta por palabra.

4.7 Patrones

Las reglas están formadas por dos partes, un patrón y una serie de reglas que determinarán los elementos de la relación, una vez encontrado el patrón. En esta sección se describe el formato de los patrones, los cuales son buscados en el texto por medio de un autómata. El funcionamiento del autómata se describe en la sección 4.8 y la segunda parte de la regla en la 4.9.

Se define un archivo de texto con una lista de los patrones a identificar en el texto, los cuales serán cargados a memoria al iniciar la aplicación y estarán disponibles para cualquier documento que sea cargado.

Los patrones se definen en un archivo de texto el cual puede ser cambiado por medio de un archivo de configuraciones para poder utilizar una búsqueda de patrones distinta según el ámbito de los documentos a analizar e incluso para poder analizar documentos con distintos idiomas con tan solo configurar la aplicación para acceder al documento correcto.

La definición de los patrones se realiza de manera que se puedan identificar los siguientes elementos dentro del texto:

- **Forma**: se identifica la forma de las palabras tal como aparecen en el texto, para indicar que el patrón debe coincidir exactamente se coloca la palabra tal cual desea ser encontrada.
- **Lema**: Se puede indicar el lema de una palabra, la cual se desea encontrar en el texto y que, sin importar su conjugación, genero, número etc., coincidirá la forma canónica que se está definiendo. Para identificar que el patrón está definiendo que se debe buscar por lema se encierra el lema entre <>. Ejemplo: <cantar> coincidirá con cantas, cantamos, cantó, etc.
- **Etiqueta**: Se puede también definir la etiqueta de la palabra que debe coincidir, la etiqueta deberá estar definida según el estándar EAGLES y se identificará por estar encerrada entre llaves {}. Ejemplo: {NCMS000} coincidirá con perro, niño, etc.

- **Clase semántica**: Se considera poder definir una clase semántica a la cual puede corresponder una palabra, la clase semántica corresponde a una categoría dentro de la ontología definida por Euro WordNet (Vossen, et al., 1998). La clase semántica deberá ser asignada previamente en un pre procesamiento al texto y si no se cuenta con ella al momento de buscar los patrones, no se generará un error, simplemente no habrá ninguna coincidencia. La clase semántica se definirá entre corchetes cuadrados []. Ejemplo: [Living] coincidirá con animal, gato, árboles, humano, etc.
- **Concepto definido en el documento:** Si los documentos a analizar serán unidades descriptivas de algún objeto concreto en específico, es posible hacer referencia al objeto del que habla el documento, por ejemplo si se está analizando un documento que describe a los martillos, es posible hacer referencia a este concepto dentro de los patrones, en forma de una variable, cuyo valor dependerá de cada documento individual. El patrón que hace referencia al concepto del documento será definido como: ?concepto, el concepto podrá ser la forma de la palabra o el lema, en el último caso, se deberá indicar que se busca el lema como <?concepto>. Ejemplo: En un documento que trata sobre los martillos, ?concept coincidirá con martillos y <?concept> con martillo, martillos, martillito, etc.

Se pueden definir los patrones que se necesiten y es escalable, ya que la carga se realiza directamente de un archivo el cual puede ser modificado según el análisis lo requiera, de manera que se tiene un conjunto de archivos para análisis de documentos generales, pero el usuario puede definir sus propios patrones a buscar en un documento especializado. La definición de un mismo patrón no debe tener espacios, las palabras dentro de un patrón se separarán con un guión bajo.

La Tabla 3 muestra una serie de ejemplos de patrones y algunos textos que coinciden con el patrón, en el caso de las etiquetas, se tienen sustantivos comunes y signos de puntuación coma {Fc}.

Patrón	Ejemplos de coincidencias
{NCFS000}_es_una_{NCFS000}	manzanilla es una hierba camisa es una prenda
{NCMS000}_es_una_{NCFS000}	martillo es una herramienta pinza es una herramienta
{NCMS000}_{Fc}_es_un_{NCMS000}	bisturí, es un instrumento sillón, es un mueble
se_usa_en	se usa en

?concept_o_{NMS000}	banjo o banyo En un documento cuyo tema o concepto principal sea banjo. maleta o maletín En un documento cuyo tema o concepto principal sean las maletas
?concept_{AQ0CP0}	martillo portátil En un documento cuyo tema o concepto principal sean los martillos México central En un documento cuyo tema o concepto central sea México.

Tabla 1 Ejemplos del sistema de patrones del SERCDD

Como se puede observar por la definición de los patrones, no se tiene una dependencia directa a ningún idioma, sin embargo el idioma que se esté analizando deberá ser definido en el archivo de configuraciones para poder realizar el pre procesamiento del texto correctamente.

La idea básica detrás de la búsqueda de patrones dentro del texto para encontrar relaciones parte de (Hearst, 1992), sin embargo con las adecuaciones de la manera de buscar y el agregar la posibilidad de variables, así como el uso en conjunto con el sistema de reglas definido en la siguiente sección, se amplía el concepto a encontrar relaciones de otros tipos, ya que en (Hearst, 1992) sólo se consideran relaciones de hiponimia.

4.8 Autómata para reconocimiento de patrones

En esta sección se describe el uso de un autómata finito determinista (AFD) que es empleado para realizar el reconocimiento de patrones.

Se define el AFD de reconocimiento de patrones en el texto como $(Q, \Sigma, q_0, \delta, F)$ donde:

- Q es el conjunto de estados del AFD *{edo_prefijo, edo_patrón, edo_alto}*
- Σ es el alfabeto definido para el AFD *{símbolo_pref, símbolo_patrón, símbolo_prefL, símbolo_patrónL, símbolo_prefE, símbolo_patrónE, símbolo_otro}*
- q_0 es el estado inicial: *edo_prefijo*
- δ es la función de transición, la cual define el nuevo estado del AFD, dado un símbolo recibido. (sección 7.3)
- F es el conjunto de estados finales o de aceptación del AFD *{edo_patrón}*

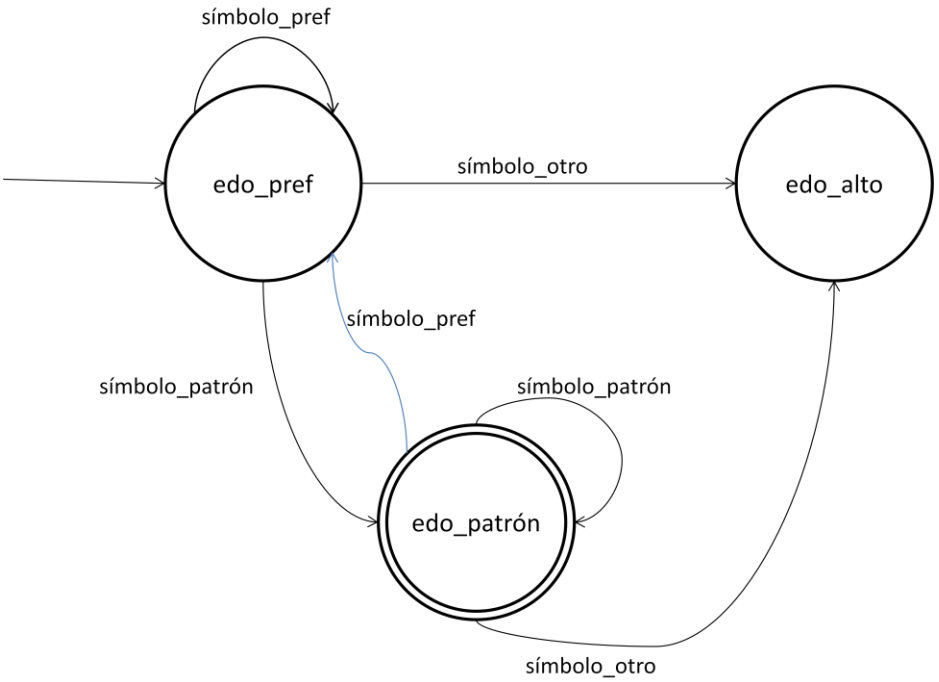

Figura 6 Diagrama AFD

4.8.1 Conjunto de estados Q

Q = {edo_prefijo, edo_patrón, edo_alto}

- ***edo_prefijo***: el AFD se encuentra en este estado cuando el texto procesado es un prefijo de uno o más de los patrones a buscar, con lo que se vuelve un candidato a ser un patrón encontrado. Cabe mencionar, que la cadena vacía es el prefijo de todos los patrones, por lo que el estado inicial q_0 = *edo_prefijo*
  Ejemplo:
  Patrón: {NCMS000}_es_un_{NCMS000}
  Prefijos: {NCMS000}, {NCMS000}_es, {NCMS000}_es_un
- ***edo_patrón:*** el AFD se encuentra en este estado cuando el texto procesado en la iteración actual cumple con un patrón a buscar. Nótese que el conjunto de estados finales F, contiene a edo_patron, ya que cuando una cadena corresponde con un patrón, se considera un estado de terminación válido del AFD.
  Ejemplo:
  Patrón: se_<usar>_para
  Textos que generan un AFD en estado edo_patrón: se usa para, se usaba para.
- ***edo_alto:*** el AFD se encuentra en este estado cuando el texto procesado en la iteración actual, no ha coincidido con ningún patrón, ni algún prefijo de los patrones a buscar.
  Ejemplo:

Patrón: {NCMS000}_es_un_{NCMS000}
Texto que genera un estado edo_alto: alguacil es un **importante**

4.8.2 Alfabeto Σ

El procesamiento del texto se realiza de modo que cada palabra en la oración es equivalente a un símbolo del alfabeto, dependiendo de si la forma, etiqueta o lema de la misma concatenado con el prefijo acumulado, forma un prefijo, un patrón o bien, no cumple con ninguno.

Los símbolos del alfabeto pueden indicar que se recibió un elemento que forma un prefijo de un patrón o un patrón, cada palabra es un símbolo recibido, y puede ser su forma, etiqueta o lema el símbolo que recibe el automata, los posibles símbolos son:

- símbolo_pref: por ejemplo, si el patrón es "un_{NCMS000}_es_un", y en el texto se encuentra "un", se generará este símbolo, ya que es un prefijo del patrón.
- símbolo_patrón: si el patrón es "un_{NCMS000}_es_un", y al momento en el texto se ha encontrado "un ciervo es" y el siguiente elemento es "un", se activa este símbolo, ya que la forma de la palabra "un" corresponde al último elemento del patrón.
  Igual que en los casos anteriores, se tienen símbolos para el lema de la palabra y la etiqueta.
- símbolo_prefL: si el patrón es "un_{NCMS000}_<ser>_un", y se tiene acumulado el prefijo "un perro", si el siguiente elemento tiene como lema <ser>, se activará este símbolo, por ejemplo con: un perro es, un perro era. En este caso se tienen 3 elementos encontrados, de 4 que forma un patrón, por lo que se ha encontrado un prefijo.
- símbolo_patrónL: • si el patrón es "un_{NCMS000}_<ser>", y se tiene acumulado el prefijo "un perro", si el siguiente elemento tiene como lema <ser>, se activará este símbolo, ya que es el último elemento del patrón, por ejemplo con: un perro es, un perro era.
- símbolo_prefE: Si el patrón es "el_{NCMS000}_se_usa" y se tiene en el texto "el martillo", cuando martillo entra al autómata con su etiqueta, que es {NCMS000}, formará el prefijo del patrón a buscar, activando este símbolo.
- símbolo_patrónE: en el patrón "un_{NCMS000}_es_un_{NCMS000}" si se tiene acumulado el prefijo "un piano es un" y se recibe la palabra "instrumento" cuya etiqueta es {NCMS000}, se activa este símbolo, ya que la etiqueta de la palabra completa el patrón a buscar.

- símbolo_otro: símbolo que indica que la palabra procesada en la iteración actual no corresponde con ningún patrón o prefijo de los patrones a buscar, con lo que se rompe la cadena y se reinicia el autómata.

4.8.3 Función de transición δ

La función de transición ($\delta : Q \times \Sigma \rightarrow Q$) está definida por la siguiente tabla de transición de estados.

Estado actual	Símbolo	Estado siguiente
edo_prefijo	símbolo_pref	edo_prefijo
edo_prefijo	símbolo_patrón	edo_patrón
edo_prefijo	símbolo_prefL	edo_prefijo
edo_prefijo	símbolo_patrónL	edo_patrón
edo_prefijo	símbolo_prefE	edo_prefijo
edo_prefijo	símbolo_patrónE	edo_patrón
edo_prefijo	símbolo_otro	edo_alto
edo_patrón	símbolo_pref	edo_prefijo
edo_patrón	símbolo_patrón	edo_patrón
edo_patrón	símbolo_prefL	edo_prefijo
edo_patrón	símbolo_patrónL	edo_patrón
edo_patrón	símbolo_prefE	edo_prefijo
edo_patrón	símbolo_patrónE	edo_patrón
edo_patrón	símbolo_otro	edo_alto

Tabla 2 Tabla de transición de estados AFD

Cualquier transición diferente a las definidas en la Tabla 4, genera un edo_alto.

4.9 Sistema de reglas

Como se menciono anteriormente, una regla está formada por dos partes, un patrón a buscar en el texto y una regla que indica cómo buscar la relación una vez encontrado el patrón.

En las secciones anterior se explicó cómo se ha seleccionado el estándar para la definición de los patrones, y el algoritmo empleado para localizar los patrones definidos, dentro del texto. En esta sección se expone el sistema de reglas que, en conjunto con los patrones permiten la extracción de las relaciones ontológicas básicas y además relaciones adicionales que se desee encontrar en un documento y que se pueda utilizar una cierta plantilla para localizarla.

El lenguaje de reglas se define para relaciones binarias, es decir del tipo:

R(A, B)

donde:

- ***R***: define el nombre de una relación entre dos elementos. ejemplos: hiponimia, hiperonimia.
- ***A, B***: son los elementos relacionados por medio de ***R***.

A pesar de no poder definir relaciones de más elementos, se tiene la definición para poder considerar que de una relación transitiva y simétrica se pueda obtener una relación de más elementos, sin embargo esto debe ser realizado con cautela ya que no todas las relaciones cumplirán con esto.

Una regla completa tendrá la siguiente forma:

patrón R ai/bj at/bt ae/be

donde:

- ***patrón***: es la definición de un patrón a buscar dentro del texto, el patrón deberá tener el formato que se definió en la sección 4.7.
- ***R***: nombre de la relación que se encontrará con el patrón definido.
- ***ai***: índice o variable dentro del texto en el cual se encuentra el primer elemento relacionado
- ***bj***: índice o variable dentro del texto en el cual se encuentra el segundo relacionado
- ***at:*** definición de forma, etiqueta, lema o clase semántica del primer elemento relacionado. Por ejemplo si el elemento debe ser un sustantivo, se validará que cumpla con esta condición, ya que el elemento en el índice ai podría ser un verbo, con lo que no se cumpliría la regla.
- ***bt***: definición de forma, etiqueta, lema o clase semántica del segundo elemento relacionado. Por ejemplo si el elemento debe ser un sustantivo, se validará que cumpla con esta condición, ya que el elemento en el índice bj podría ser un verbo, con lo que no se cumpliría la regla.
- ***ae***: indica si el índice ***ai*** es estricto o no, si ***ae*** es 0, se le da prioridad a la propiedad definida por ***at***, buscando la primer palabra que la cumpla a la derecha o izquierda

del índice ***ai,*** si es 1, indica que forzosamente se deben cumplir tanto el índice como el valor de ***at***. Por ejemplo, si el elemento en el índice ***ai***, debería ser un sustantivo (definido por ***at***) pero es un verbo, este elemento nos indicaría se debemos continuar buscando en el texto hasta encontrar el primer sustantivo, o si es estricto, y al momento de encontrar un verbo en el índice ***ai***, la regla simplemente no se cumple.

- ***be***: indica si el índice ***bj*** es estricto o no, si ***be*** es 0, se le da prioridad a la propiedad definida por ***bt***, buscando la primer palabra que la cumpla a la derecha o izquierda del índice ***bj***, si es 1, indica que forzosamente se deben cumplir tanto el índice como el valor de ***bt***. Por ejemplo, si el elemento en el índice ***bi***, debería ser un sustantivo (definido por ***bt***) pero es un verbo, este elemento nos indicaría se debemos continuar buscando en el texto hasta encontrar el primer sustantivo, o si es estricto, y al momento de encontrar un verbo en el índice ***bi***, la regla simplemente no se cumple.

Los índices son definidos por números enteros que indican la posición en el texto del concepto con respecto al patrón, cada palabra ocupa una posición y la primer palabra en el patrón tiene el índice 0. Si no se tiene un índice, puede tenerse una variable, la cual puede ser:

- ***?c***: se refiere al tema o concepto principal que se describe en el documento.
- ***this***: se refiera a todo el patrón.

Para at y bt se puede tener al igual que en los patrones, la definición de la forma, <lema>, {etiqueta}, [clase], identificando cada una con el respectivo símbolo. Para la etiqueta de la clase gramatical, no es necesario definir, genero, número etc, es posible definir tan solo la categoría de la palabra. Estos parámetros sólo se tomarán en cuenta si los índices ai o bj están fuera de rango de las palabras dentro del patrón, es decir en el caso de que ai y/o bj estén dentro del rango de los índices del patrón definido, se ignorarán los elementos at y/o bt. Esta parte de la regla es opcional y el valor por defecto es buscar conceptos con clase gramatical de sustantivo.

La tabla 5 muestra un fragmento del archivo de reglas, cada una de las reglas está formada por dos partes, un patrón y la regla para encontrar a los elementos de la relación.

En la Tabla 6 se muestran ejemplos de distintas posibilidades de uso del sistema de reglas definido.

```
?concept_es_la_{NCFS000} HIPER 0/3 */* 1/1 -1/-1
?concept_es_el_{NCMS000} HIPER 0/3 */* 1/1 -1/-1
{NCFS000}_es_una_{NCFS000} HIPER 0/3 */* 0/0 -1/-1
{NCMS000}_es_un_{NCMS000} HIPER 0/3 */* 0/0 -1/-1
{NCMS000}_es_una_{NCFS000} HIPER 0/3 */* 0/0 -1/-1
?concept_de_{NCCS000} HIPO ?c/this */* 0/0 -1/-1
?concept_de_{NCFS000} HIPO ?c/this */* 0/0 -1/-1
?concept_{AQ0MS0} HIPO ?c/this */* 0/0 -1/-1
?concept_{AQ0FS0} HIPO ?c/this */* 0/0 -1/-1
?concept_{AQ0CS0} HIPO ?c/this */* 0/0 -1/-1
?concept_{AQ0CP0}_{AQ0CP0} HIPO ?c/this */* 0/0 -1/-1
{NCFS000}_es_una_?concept HIPO ?c/0 */* 0/0 -1/-1
{NCMS000}_es_un_?concept HIPO ?c/0 */* 0/0 -1/-1
?concept_{NCCS000} HIPO ?c/this */* 0/0 -1/-1
?concept_{NCMS000} HIPO ?c/this */* 0/0 -1/-1
?concept_{NCCS000} HIPO ?c/this */* 0/0 -1/-1
<usar>_{RG}_para USO -1/3 {N}/{V} 0/0 -1/-1
se_usa_en USO -1/3 {N}/{N} 0/0 1/-1}
<utilizar>_para USO -1/2 {N}/{V} 0/0 0/-1
<utilizar>_{RG}_en USO -1/3 {N}/{N} 0/0 0/-1
<utilizar>_{RG}_para USO -1/3 {N}/{V} 0/0 0/-1
usado_en USO -1/2 {N}/{N} 0/0 0/-1
usada_para USO -1/2 {N}/{V} 0/0 0/-1
usados_en USO -1/2 {N}/{N} 0/0 0/-1
se_emplean_en USO -1/3 {N}/{N} 0/0 1/-1
empleados_en USO -1/2 {N}/{N} 0/0 0/-1
diseñada_para_{VMN0000} USO -1/2 {N}/* 0/0 0/-1
sirve_{RG}_para_{VMN0000} USO -1/2 {N}/* 0/0 0/-1
?concept_consiste_de_un_{NCMS000} PART ?c/4 */{N} 0/0
?concept_consiste_de_una_{NCFS000} PART ?c/4 */{N} 0/0
consiste_de_un_{NCMS000} PART -1/3 {N}/{N} 0/0 0/-1
consiste_de_una_{NCFS000} PART -1/3 {N}/{N} 0/0 0/-1
contiene_un PART -1/2 */{N} 0/0 0/-1
formado_por PART -1/2 */{N} 0/0 0/-1
formada_por PART -1/2 */{N} 0/0 0/-1
esta_compuesto_por PART -1/3 */{N} 0/0 0/-1
las_partes PART -1/2 */{N} 0/0 -1/-1
sus_partes PART -1/2 */{N} 0/0 -1/-1
consta_de PART -1/2 */{N} 0/0 0/-1
constan_de PART -1/2 */{N} 0/0 0/-1
un_?concept_{RG}_tiene_una_{NCCS000} PART ?c/5 */* 0/0 -1/-1
se_encuentra_en LOC -1/3 */{N} 0/0 1/-1
encontrado_en LOC -1/2 */{N} 0/0 0/-1
encontrada_en LOC -1/2 */{N} 0/0 0/-1
de_[material] MATERIAL -1/1 {N}/* 0/0 -1/-1
procede_de_{NP00000} procedencia -1/2 {N}/* 0/0 0/-1
```

Tabla 3 Fragmento del archivo de reglas

Regla	Coincidencia patrón	Valores de los elementos R(A,B)	Ejemplo de Relación
{NCMS000} _es_una_ {NCFS000} hiperónimo 0/3	servidor es una computadora	R: hiperónimo A: servidor B: computadora	hiperónimo(servidor,computadora)
también_llamado sinónimo -1/2	…puede ser tambien llamado… …es tambien llamado… …, tambien llamado… un cactus, tambien llamado cactacea	R: sinónimo A: primer concepto con categoría gramatical de sustantivo que se encuentre a la izquierda del patrón B: primer concepto con categoría gramatical de sustantivo que se encuentre a la derecha del patrón	sinónimo(cactus, cactacea)
se_usa_para uso ?c/3 N/V	…que se usa para cortar	R:uso A:concepto descrito en el documento B:siguiente verbo que se encuentre a la derecha del patrón localizado	uso(cuchillo, cortar) En este caso, la parte N/V de la regla indica que el primer elemento debe ser un sustantivo y el segundo un verbo.
?concept_de_{NCMP000} HIPO ?c/this	caballo de carreras	R: hiponimia A: concepto del documento b: todo el patrón localizado	hipónimo(caballo, caballo de carreras) En este ejemplo en la regla: ?c/this define que el primer elemento de la relación es el concepto del cual habla el documento, en este caso caballo, y que el segundo elemento será todo el texto que forma el patrón, en este caso, caballo de carreras

Tabla 4 Ejemplos del sistema de reglas de SERCDD

5 Resultados

En esta sección se presentan los resultados obtenidos para un conjunto de pruebas con documentos de distintas fuentes, se analiza la confiabilidad de los resultados, así como el número de relaciones encontradas y no encontradas.

Se inicia por explicar una corrida del algoritmo AFD y del sistema de reglas en funcionamiento, sobre un corpus muy pequeño, con la intención de lograr un mejor entendimiento del proceso de análisis del SERCDD.

Posteriormente se presenta una serie de ejemplos que permiten entender los pasos del análisis realizad, los resultados obtenidos en cada documento y mostrar la aplicación en funcionamiento. En esta parte se muestran algunos de los casos en los que un corpus no es analizado correctamente, tanto para el caso de relaciones no detectadas como en los casos en los que se encuentran relaciones incorrectas. Finalmente se presentan los datos sintetizados de todas las pruebas realizadas, con estadísticas de los aciertos y errores, así como del número de relaciones que se esperaban encontrar contra las que el algoritmo encontró exitosamente.

5.1 Corrida del algoritmo

- Corpus:

> El martillo es una herramienta utilizada para golpear una pieza, causando su desplazamiento o deformación.

5.1.1 Pre-procesamiento del texto (Freeling)

La Tabla 7 muestra el resultado de la primera etapa por la que pasa el corpus, siendo esta la salida de la librería Freeling (Carreras, Chao, Padró, & Padró, 2004) se considera la etapa 0 del algoritmo, en este punto, el texto ya ha sido segmentado y etiquetado.

```
<SENT>
 <WORD forma="el" lemma="el" pos="DA0MS0">
  <ANALYSIS lemma="el" pos="DA0MS0" prob="1.000000"/>
 </WORD>
 <WORD forma="martillo" lemma="martillo" pos="NCMS000">
  <ANALYSIS lemma="martillar" pos="VMIP1S0" prob="0.250000"/>
  <ANALYSIS lemma="martillo" pos="NCMS000" prob="0.750000"/>
 </WORD>
 <WORD forma="es" lemma="ser" pos="VSIP3S0">
  <ANALYSIS lemma="ser" pos="VSIP3S0" prob="1.000000"/>
 </WORD>
 <WORD forma="una" lemma="uno" pos="DI0FS0">
```

```
  <ANALYSIS lemma="uno" pos="PI0FS000" prob="0.027391"/>
  <ANALYSIS lemma="unir" pos="VMM03S0" prob="0.000217"/>
  <ANALYSIS lemma="unir" pos="VMSP1S0" prob="0.000217"/>
  <ANALYSIS lemma="unir" pos="VMSP3S0" prob="0.000217"/>
  <ANALYSIS lemma="uno" pos="DI0FS0" prob="0.971957"/>
 </WORD>
 <WORD forma="herramienta" lemma="herramienta" pos="NCFS000">
  <ANALYSIS lemma="herramienta" pos="NCFS000" prob="1.000000"/>
 </WORD>
 <WORD forma="utilizada" lemma="utilizar" pos="VMP00SF">
  <ANALYSIS lemma="utilizar" pos="VMP00SF" prob="1.000000"/>
 </WORD>
 <WORD forma="para" lemma="para" pos="SPS00">
  <ANALYSIS lemma="parar" pos="VMIP3S0" prob="0.000299"/>
  <ANALYSIS lemma="parar" pos="VMM02S0" prob="0.000299"/>
  <ANALYSIS lemma="parir" pos="VMM03S0" prob="0.000299"/>
  <ANALYSIS lemma="parir" pos="VMSP1S0" prob="0.000299"/>
  <ANALYSIS lemma="parir" pos="VMSP3S0" prob="0.000299"/>
  <ANALYSIS lemma="para" pos="SPS00" prob="0.998507"/>
 </WORD>
 <WORD forma="golpear" lemma="golpear" pos="VMN0000">
  <ANALYSIS lemma="golpear" pos="VMN0000" prob="1.000000"/>
 </WORD>
 <WORD forma="una" lemma="uno" pos="DI0FS0">
  <ANALYSIS lemma="uno" pos="PI0FS000" prob="0.027391"/>
  <ANALYSIS lemma="unir" pos="VMM03S0" prob="0.000217"/>
  <ANALYSIS lemma="unir" pos="VMSP1S0" prob="0.000217"/>
  <ANALYSIS lemma="unir" pos="VMSP3S0" prob="0.000217"/>
  <ANALYSIS lemma="uno" pos="DI0FS0" prob="0.971957"/>
 </WORD>
 <WORD forma="pieza" lemma="pieza" pos="NCFS000">
  <ANALYSIS lemma="pieza" pos="NCFS000" prob="1.000000"/>
 </WORD>
 <WORD forma="," lemma="," pos="Fc">
  <ANALYSIS lemma="," pos="Fc" prob="1.000000"/>
 </WORD>
 <WORD forma="causando" lemma="causar" pos="VMG0000">
  <ANALYSIS lemma="causar" pos="VMG0000" prob="1.000000"/>
 </WORD>
 <WORD forma="su" lemma="su" pos="DP3CS0">
  <ANALYSIS lemma="su" pos="DP3CS0" prob="1.000000"/>
 </WORD>
 <WORD forma="desplazamiento" lemma="desplazamiento" pos="NCMS000">
  <ANALYSIS lemma="desplazamiento" pos="NCMS000" prob="1.000000"/>
 </WORD>
 <WORD forma="o" lemma="o" pos="CC">
  <ANALYSIS lemma="o" pos="NCFS000" prob="0.001155"/>
  <ANALYSIS lemma="o" pos="CC" prob="0.998845"/>
```

```
        </WORD>
        <WORD forma="deformación" lemma="deformación" pos="NCFS000">
         <ANALYSIS lemma="deformación" pos="NCFS000" prob="1.000000"/>
        </WORD>
        <WORD forma="." lemma="." pos="Fp">
         <ANALYSIS lemma="." pos="Fp" prob="1.000000"/>
        </WORD>
</SENT>
```

Tabla 1 Ejemplo de pre procesamiento del corpus, en su etapa de análisis morfológico

- Línea en el archivo de reglas:

```
{NCMS000}_es_una_{NCFS000} HIPER 0/3 N/N 0/0
```

- Patrón a buscar dentro del texto según la regla:

```
{NCMS000}_es_una_{NCFS000}
```

5.1.2 Transiciones del AFD:

Funcionamiento del autómata: Se tiene un mapa, en el cual se cargan en memoria todos los patrones a buscar. Se generan también todos los prefijos posibles de cada patrón. Cada palabra procesada en el texto es buscada dentro de los patrones. Si se encuentra, se identifica que es un prefijo, o un patrón, y si no, se genera el símbolo_otro, como en este caso.

1. δ(edo_pref, símbolo_otro) → edo_alto

Palabra de entrada: El

Forma: el
Lema: el
Etiqueta: DA0MS0

Se busca "el" dentro del mapa de patrones, como no se encuentra, se genera un simbolo_otro

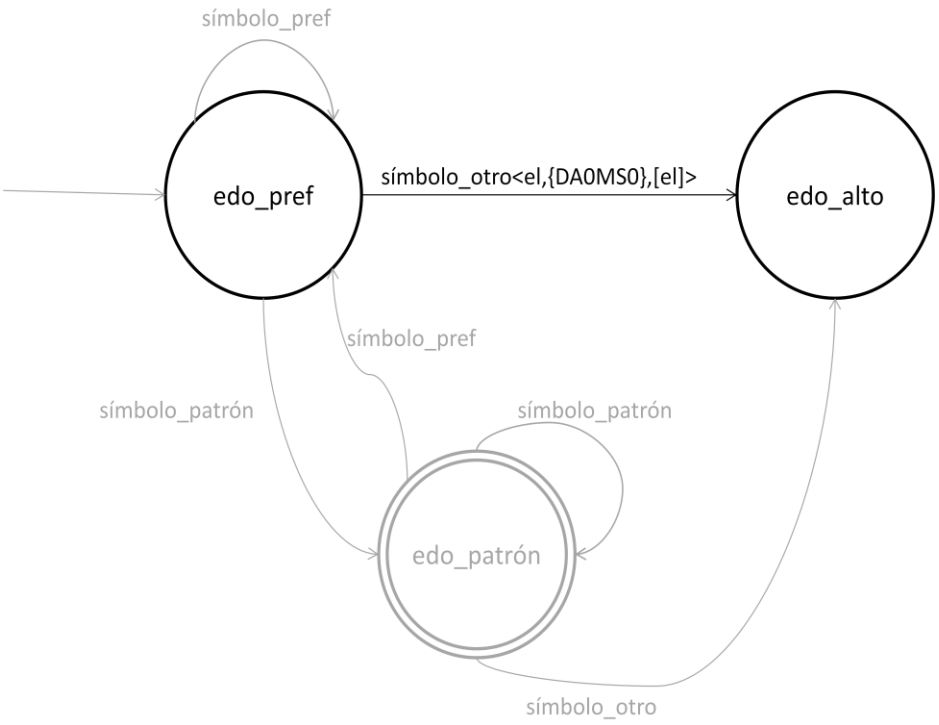

Figura 1 Transición δ(edo_pref, símbolo_otro) → edo_alto

patrón acumulado: <>

Al llegar a un estado_alto, se realiza la verificación del patrón acumulado, se valida y se reinicia el autómata. En este caso el patrón está vacío, por lo que la validación es negativa.

2. δ(edo_pref, símbolo_pref) → edo_pref

   Palabra de entrada: martillo

   Forma: martillo
   Lema: martillo
   Etiqueta: NCMS000

   Se busca dentro del mapa de patrones, se encuentra NCMS000 como un prefijo de patrón, por lo que se genera un símbolo de prefijo, específicamente de prefijo por etiqueta, ya que la etiqueta de la palabra es la que forma parte del patrón.

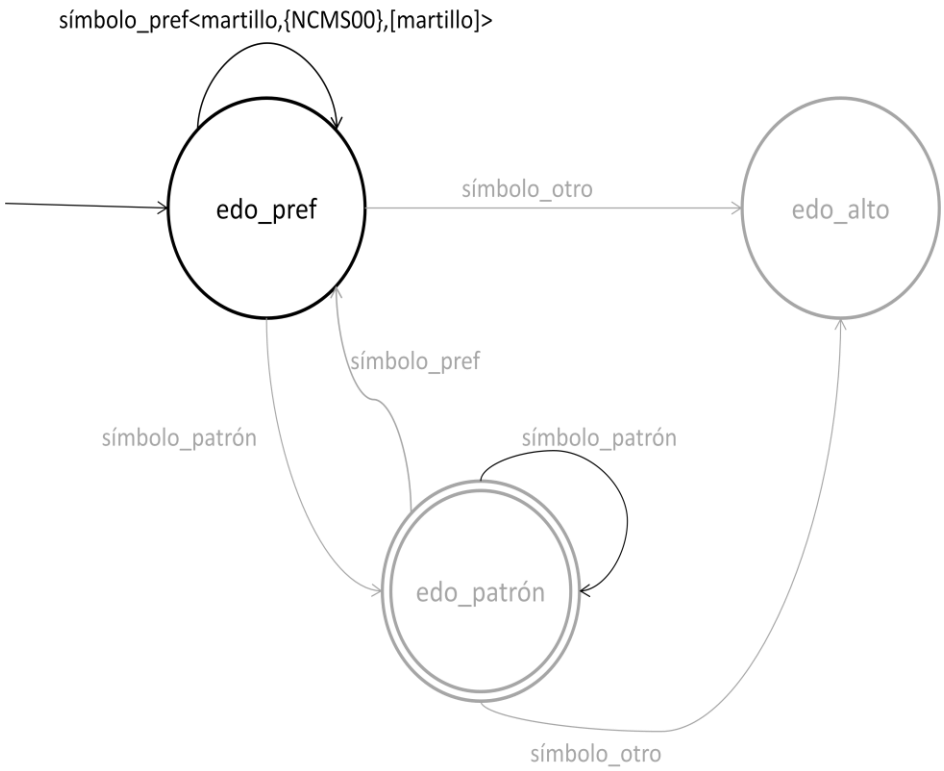

Figura 2 Transición δ(edo_pref, símbolo_pref) → edo_pref

Al reconocer un símbolo que junto con el patrón acumulado al momento (en este caso, el patrón acumulado es vacío), forma parte de un prefijo, se agrega el símbolo recibido al patrón acumulado.

patrón acumulado: `<NCMS000>`

forma patrón acumulado: `martillo`

3. δ(edo_pref, símbolo_pref) → edo_pref

   Palabra de entrada: es

   Forma: es
   Lema: ser
   Etiqueta: VSIP3S0

   Se busca dentro del mapa de patrones, concatenándolo al patrón ya acumulado, se encuentra que "<NCMS000>_es" es un prefijo, por lo que se actualiza el patrón acumulado y se genera como símbolo símbolo_pref.

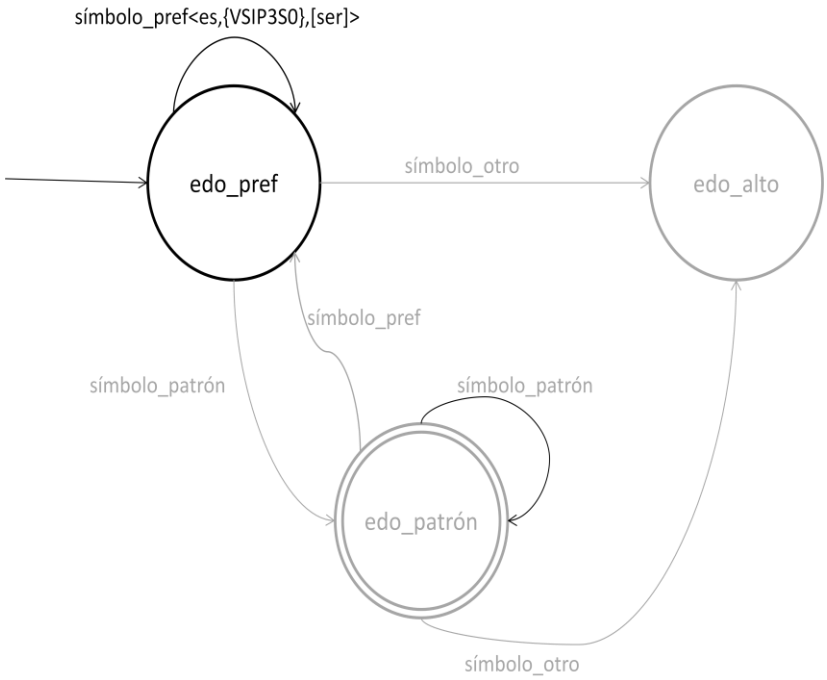

Figura 3 Transición δ(edo_pref, símbolo_pref) → edo_pref

Al reconocer un símbolo que junto con el patrón acumulado al momento forma parte de un prefijo, se agrega el símbolo recibido al patrón acumulado.

patrón acumulado: `<NCMS000_es>`

forma patrón acumulado: `martillo es`

4. δ(edo_pref, símbolo_pref) → edo_pref

   Palabra de entrada: una

   Forma: una
   Lema: una
   Etiqueta: DI0FS0

   Se busca dentro del mapa de patrones, concatenándolo al patrón acumulado al momento, en este caso se encuentra que <NCMS000>_es_una forma parte de un prefijo, por lo que se actualiza el prefijo acumulado y se genera el simbolo_pref.

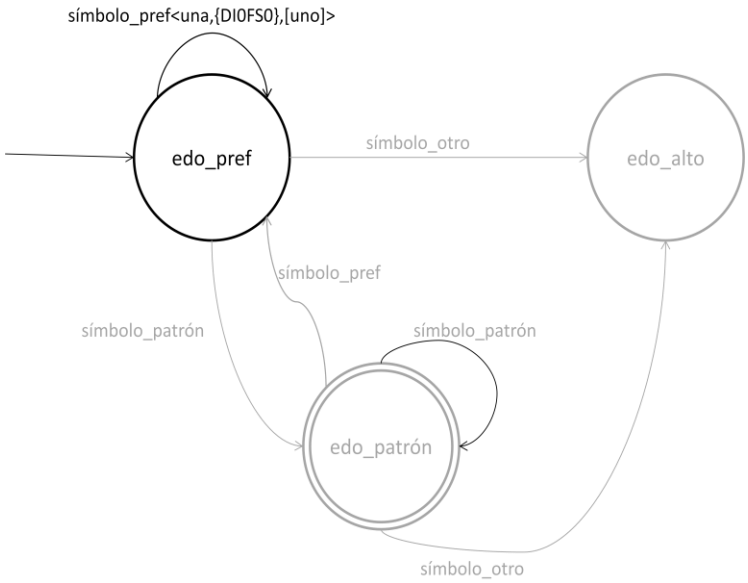

Figura 4 Transición δ(edo_pref, símbolo_pref) → edo_pref

Mientras el símbolo recibido como entrada al autómata, concatenado con el patrón acumulado al momento continúe siendo parte de un prefijo, se agrega el símbolo recibido al patrón acumulado.

patrón acumulado **:** `<NCMS000_es_una>`

forma patrón acumulado: `martillo es una`

5. δ(edo_pref, símbolo_patrón) → edo_patrón

   Palabra de entrada: herramienta

   Forma: herramienta
   Lema: herramienta
   Etiqueta: NCFS000

   Se busca el dentro del mapa de patrones, concatenándolo al patrón ya acumulado, se encuentra que "<NCMS000>_es_una<NCFS000>" es un patrón completo, por lo que se genera el símbolo edo_patrón y se actualiza el prefijo acumulado. El prefijo se actualiza, ya que el autómata siempre buscará el patrón más largo posible.

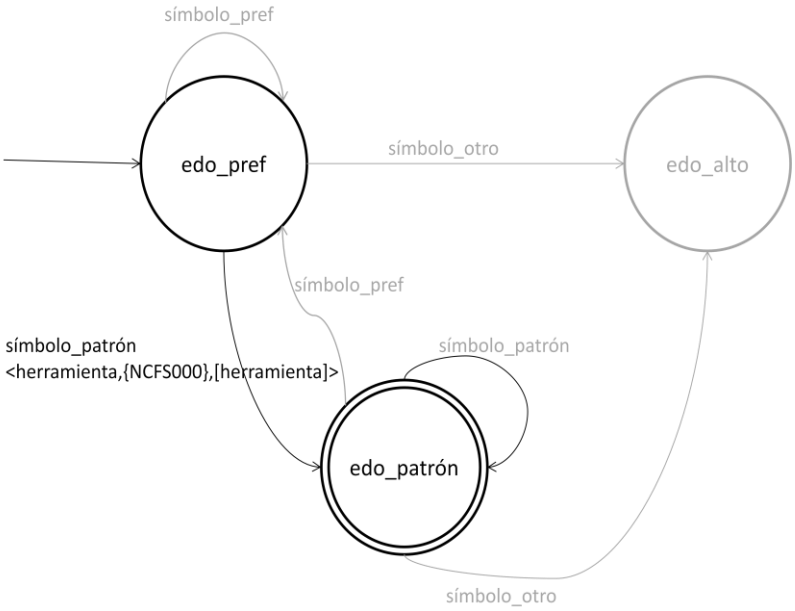

Figura 5 Transición δ(edo_pref, símbolo_patrón) → edo_patrón

En este momento, el símbolo recibido concatenado al patrón acumulado hasta el momento, es un patrón completo, en este punto la transición es al estado valido final edo_patrón, sin embargo el proceso continúa hasta llegar a un estado de alto.

patrón acumulado **:** `<NCMS000_es_una_NCFS000>`

forma patrón acumulado: `martillo es una herramienta`

6. δ(edo_patrón, símbolo_otro) → edo_alto

   Palabra de entrada: utilizada

   Forma: utilizada
   Lema: utilizar
   Etiqueta: VMP00F

   Se buscan dentro del mapa, la concatenación del patrón acumulado con:

   - la forma "<NCMS000>_es_una<NCFS000>_utilizada"
   - el lema "{NCMS000}_es_una{NCFS000}_<utilizar>"
   - la etiqueta "<NCMS000>_es_una<NCFS000>_{ VMP00F }"

Al no encontrarse ninguna posibilidad, se genera un símbolo_otro.

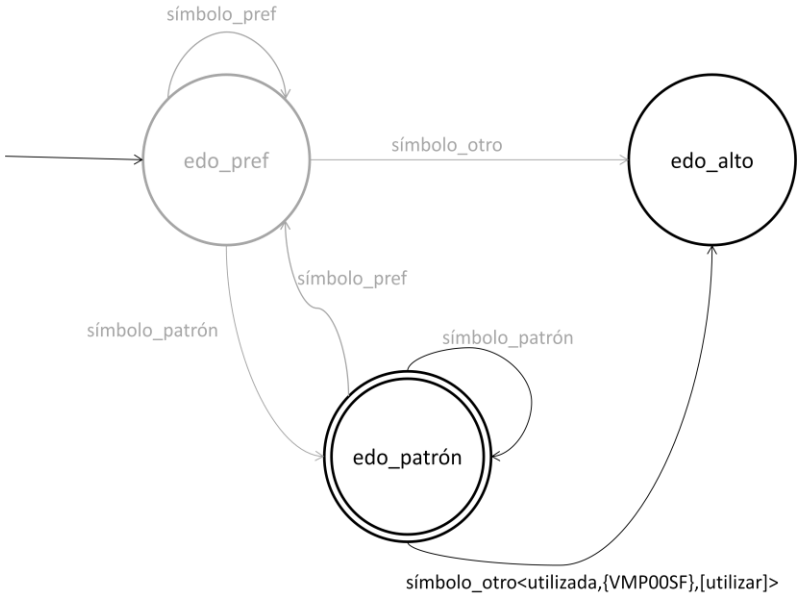

Figura 6 Transición δ(edo_patrón, símbolo_otro) → edo_alto

Al llegar al estado edo_alto, se realiza la verificación del patrón acumulado, en caso de que el patrón acumulado sea un patrón válido, se procede a aplicar la segunda parte de la regla para encontrar la relación a la que ésta hace referencia.

patrón acumulado: `<NCMS000_es_una_NCFS000>`

forma patrón acumulado: `martillo es una herramienta`

En este caso, el patrón acumulado es un patrón completo válido, por lo que se procede a encontrar los elementos de la relación, basándose en la regla.

Notar que la búsqueda de cada patrón y prefijo de patrón, el autómata la delega al mapa, el cual, en un mismo paso, verifica todas las posibilidades, con lo cual el autómata solo se encarga de saber si se recibió un prefijo de patrón o un prefijo.

5.1.3 Aplicación de la regla

En la primera etapa del algoritmo, el AFD encontró un patrón que corresponde a una de las reglas definidas, en esta segunda etapa se debe aplicar la segunda parte de la regla, para que en base al patrón encontrado se defina la relación que corresponde.

- Regla: `HIPER 3/0 {N}/{N} 0/0`

Los valores de los elementos de la regla para este caso son los siguientes:

- ***R***: HIPER
- ***ai***: 0
- ***bj***: 3

- ***at:*** {N}
- ***bt***: {N}
- ***ae***: 0 (índice ai no estricto)
- ***be***: 0 (índice bj no estricto)

Para una explicación del significado de cada elemento, ver sección **¡Error! No se encuentra el origen de la referencia.**.

La regla que se desea encontrar es de la forma Rel(A,B), para localizar los elementos A y B, se verifican los índices ai y bj definidos en la regla, cada palabra en el texto, dentro y fuera del patrón tiene un índice, siendo el primer símbolo del patrón el elemento con índice 0.

1. Posicionar los apuntadores a los elementos A y B de acuerdo a los índices ai y bj.

Forma	El	martillo	es	una	herramienta	utilizada
Patrón		{NCMS000}	es	una	{NCFS000}	
Índice	**-1**	**0**	**1**	**2**	**3**	**4**

2. Verificar para cada elemento, si se cumple la condición establecida en *at* y *bt*.
   2.1. Si se cumplen pasar al paso 3
   2.2. Si no se cumplen
      2.2.1. Si el índice del símbolo que no cumplió con la categoría es estricto, la regla no se aplica. FIN.
      2.2.2. Si el índice del símbolo que no cumplió con la categoría no es estricto buscar la siguiente palabra que cumpla con la categoría.
         2.2.2.1. Si se encuentra pasar al paso 3.
         2.2.2.2. Si no se encuentra, la regla no se aplica. FIN
3. Se definen los elementos A y B de acuerdo a los índices encontrados.

En este caso:

A = martillo

B = herramienta

La relación resultante es:

HIPER(herramienta, martillo)

5.1.4 Representación OM

Finalmente, una vez encontradas las relaciones, se representa el conocimiento adquirido en notación OM, para el caso del ejemplo anterior, la ontología resultante es:

```
<concept>cosa
      <concept>herramienta
            <language>spanish<word>herramienta</word></language>
            <subset>cosa</subset>
            <concept>martillo
                  <language>spanish<word>martillo</word></language>
                  <subset>herramienta</subset>
            </concept>
      </concept>
</concept>
```

En una representación de grafo, la ontología es la siguiente:

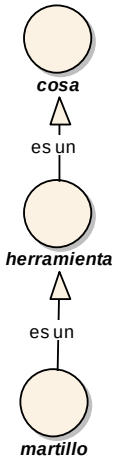

Figura 7 Ejemplo relación hiperonimia

5.2 Ejemplos

Se muestran los ejemplos de acuerdo a la siguiente estructura:

1. Corpus: se presenta el corpus utilizado, el cual como se menciono anteriormente debe ser un texto descriptivo de algún objeto concreto del mundo real, se indica la fuente de donde se obtuvo el corpus y se presenta un análisis de las características del texto que permiten que el análisis presente mejores resultados. En algunos casos, no se muestra el corpus completo sino sólo las partes del mismo que fueron relevantes para el análisis y de las cuales se obtuvieron los resultados másnotables.
2. Análisis del corpus en la herramienta SERCDD: Se presentan ejemplos del funcionamiento básico de la herramienta y se muestra cómo se presentan los resultados.

3. Análisis e interpretación de los datos obtenidos en el paso 2.

5.2.1 Ejemplo 1 Martillo

- Corpus

Concepto descrito: Martillo

Fuente: Wikipedia (http://es.wikipedia.org/wiki/Martillo)

Martillo
El martillo es una herramienta utilizada para golpear una pieza, causando su desplazamiento o deformación. El uso más común es para clavar (incrustar un clavo de acero en madera u otro material), calzar partes (por la acción de la fuerza aplicada en el golpe que la pieza recibe) o romper una pieza. Los martillos son a menudo diseñados para un propósito especial, por lo que sus diseños son muy variados. Tiene una cuña en la parte trasera para la remoción de clavos.
Historia
Los primeros martillos datan de la Edad de Piedra del año 8000 a. C.; estos martillos constaban de una piedra atada a un mango con tiras de cuero. Más tarde, en el año 4000 a. C., con el descubrimiento del cobre los egipcios comenzaron a fabricar la cabeza de los martillos en este material. Después, en el año 3500 a. C., durante la era de bronce se fabricaron con este material. Tiempo después aparecieron los martillos con orificios para el mango. El martillo actual comenzó a utilizarse en tiempo de los romanos.
Forma
La forma básica del martillo consiste de un mango (comúnmente de madera) con una cabeza pesada (comúnmente de metal) en su extremo. Los martillos son utilizados en diferentes profesiones y es una de las herramientas básicas junto con el cuchillo.

Formas conocidas del martillo son:

- Martillo de cola
- Martillo de oso carpintero
- Martillo de chapista
- Martillo de construcción, incluyendo la maza
- Martillo de galponero
- Martillo de guerra
- Martillo mecánico
- Martillo de uña

Para grandes esfuerzos existen heces, que se utilizan en minería y en la construcción. El martillo neumático es un taladro percutor portátil que basa su funcionamiento en mecanismos de aire comprimido. Realmente funciona como un martillo, pues no agujerea sino que percute la superficie con objeto de romperla en trozos.

También existen martillos hidráulicos con el mismo principio de funcionamiento que los martillos neumáticos solamente que aquí el fluido es aceite hidráulico en vez de aire comprimido, Estos martillos los llevan acoplado las excavadora industriales.

Asimismo es importante la gama de martillos no férricos que existen, con bocas de nylon,

plástico, goma o madera y que son utilizados para dar golpes blandos donde no se pueda deteriorar la pieza que se está ajustando. Martillos más utilizados Martillo de orejas: es el martillo por excelencia. Su peso es de medio kilo y su cabeza se caracteriza por poseer dos caras. Una redonda, para clavar los clavos, y otra con ranura, para sacarlos. Para los clavos pequeños conviene utilizar uno fino de cabeza cuadrada, ligero y estrecho, que no golpee los dedos al sujetar las puntas pequeñas. Martillo de bola: de uso en mecánica. La bola, aparte de equilibrar el martillo, sirve para concentrar los golpes, en el forjado de una pieza cóncava o al deformar los bordes de un remache o roblón para realizar una unión por remachado. Martillo de cuña: de uso en mecánica. La cuña sirve para el corte en caliente de piezas, de forma similar al uso de la tajadera para piezas mayores, o al cortafríos para espesores menores.

Tabla 2 Corpus Ejemplo 1 Martillo

- Análisis del corpus en la herramienta SERCDD

Para el análisis del corpus en la herramienta, se deben tener previamente configurados todos los parámetros en el archivo de configuraciones del sistema.

En este ejemplo los parámetros de configuración indican que se muestre el análisis morfológico y el sintáctico (parsing) del texto, este análisis lo provee Freeling (Carreras, Chao, Padró, & Padró, 2004) y se muestra en la herramienta como ayuda y referencia del proceso realizado.

Al momento de ejecutar la aplicación se crean todos los procesos necesarios para realizar el análisis necesario según la configuración que se haya seleccionado.

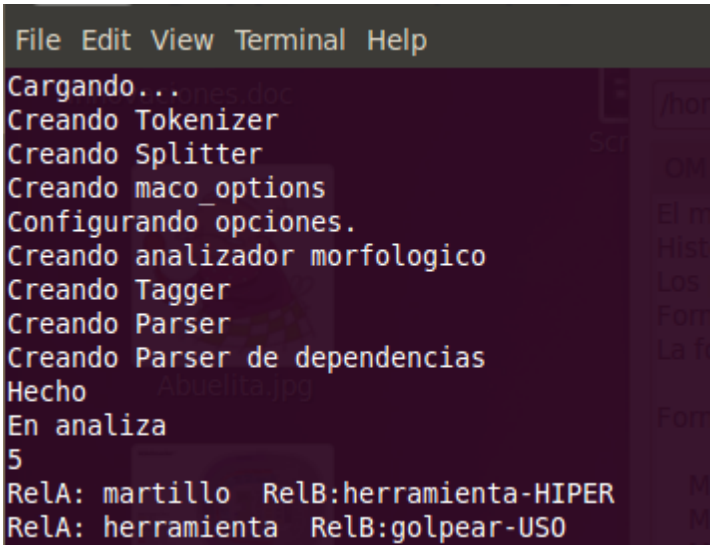

Figura 8 Salida en consola del SERCDD

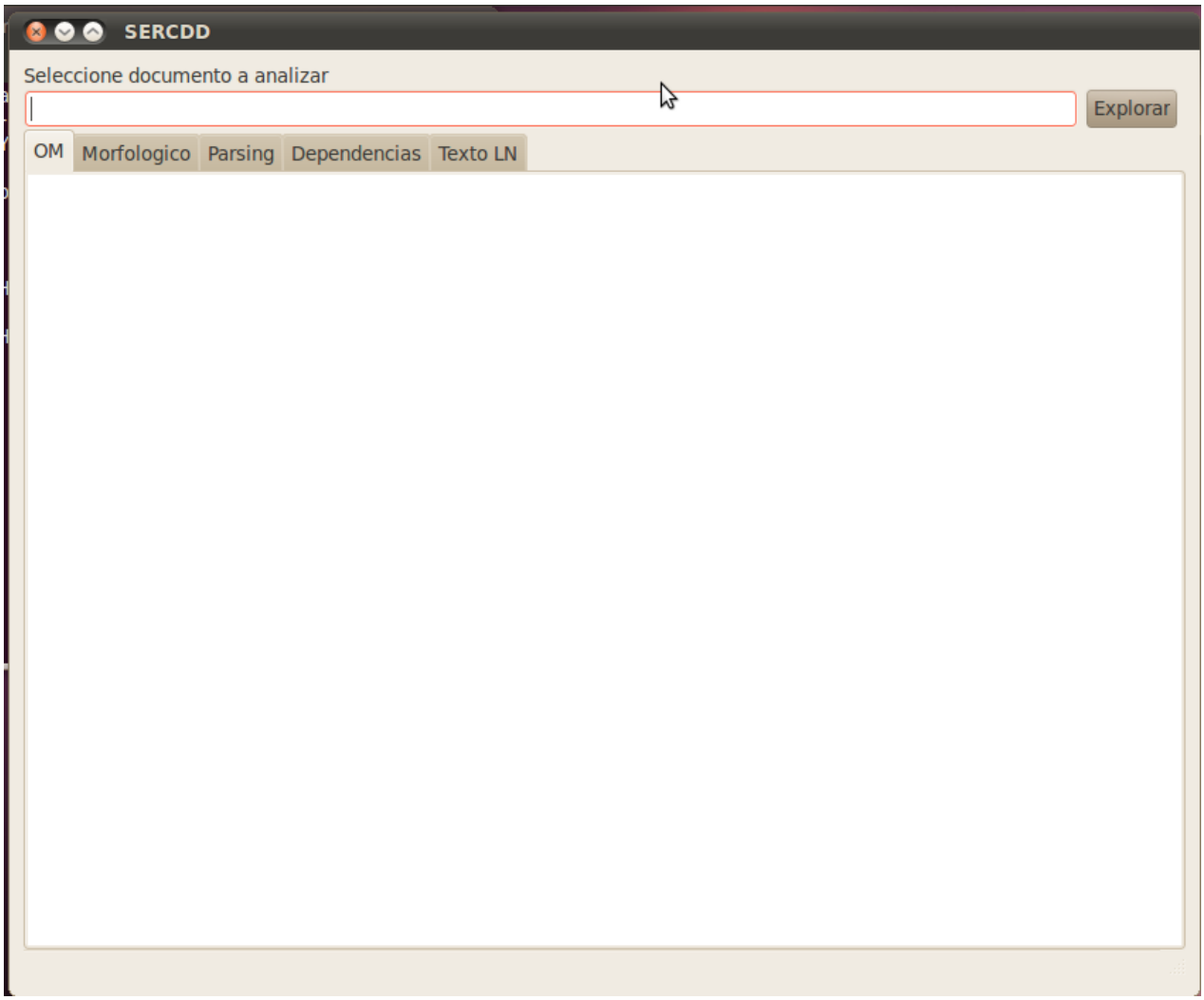

Figura 9 Interfaz de usuario de inicio del SERCDD

El paso inicial para el análisis es indicar el archivo que contiene el texto a analizar.

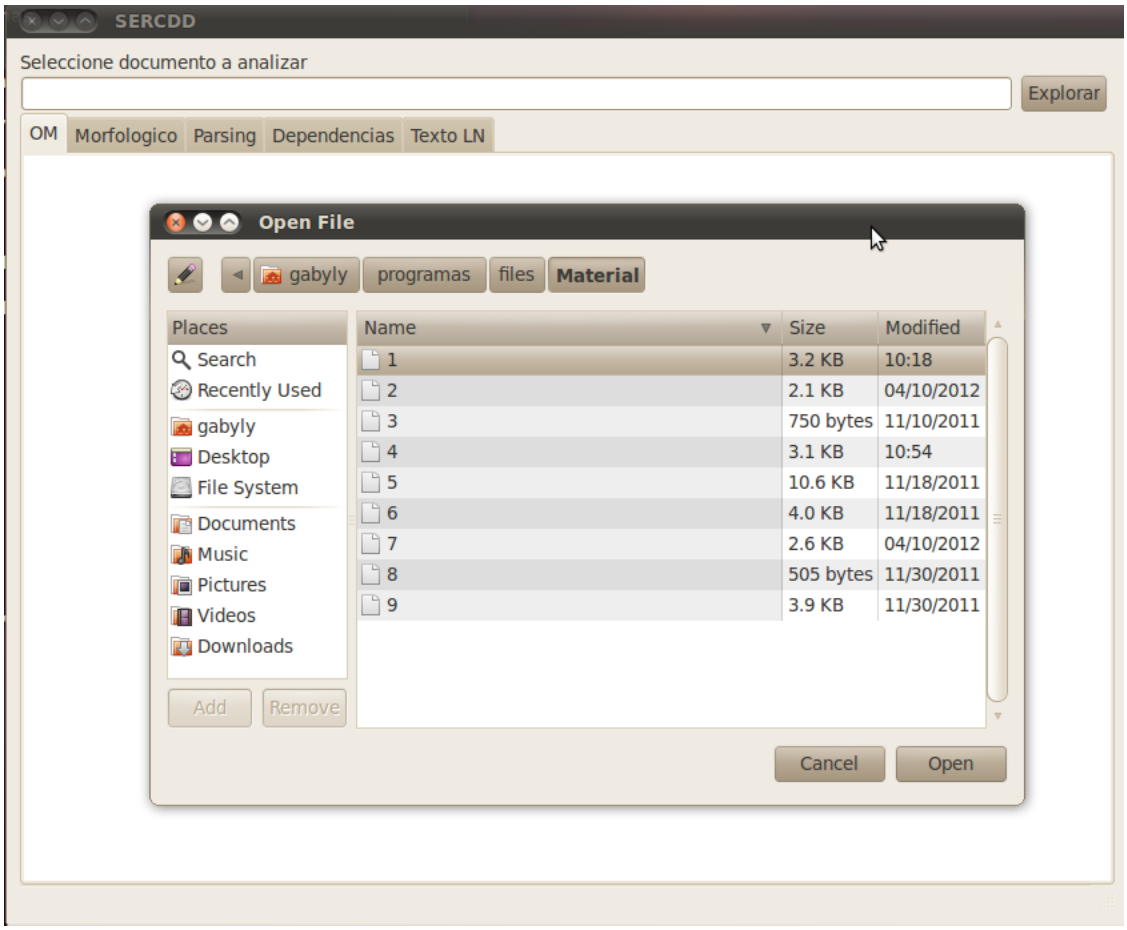

Figura 10 Interfaz de Usuario del SERCDD Seleccionar archivo

Una vez seleccionado el corpus, se inicia el proceso de análisis y se muestran los resultados en la pantalla en los formatos que se haya pedido según la configuración.

Dado que las pestañas de los análisis que realiza Freeling no son relevantes sólo se describe el resultado de la pestaña OM.

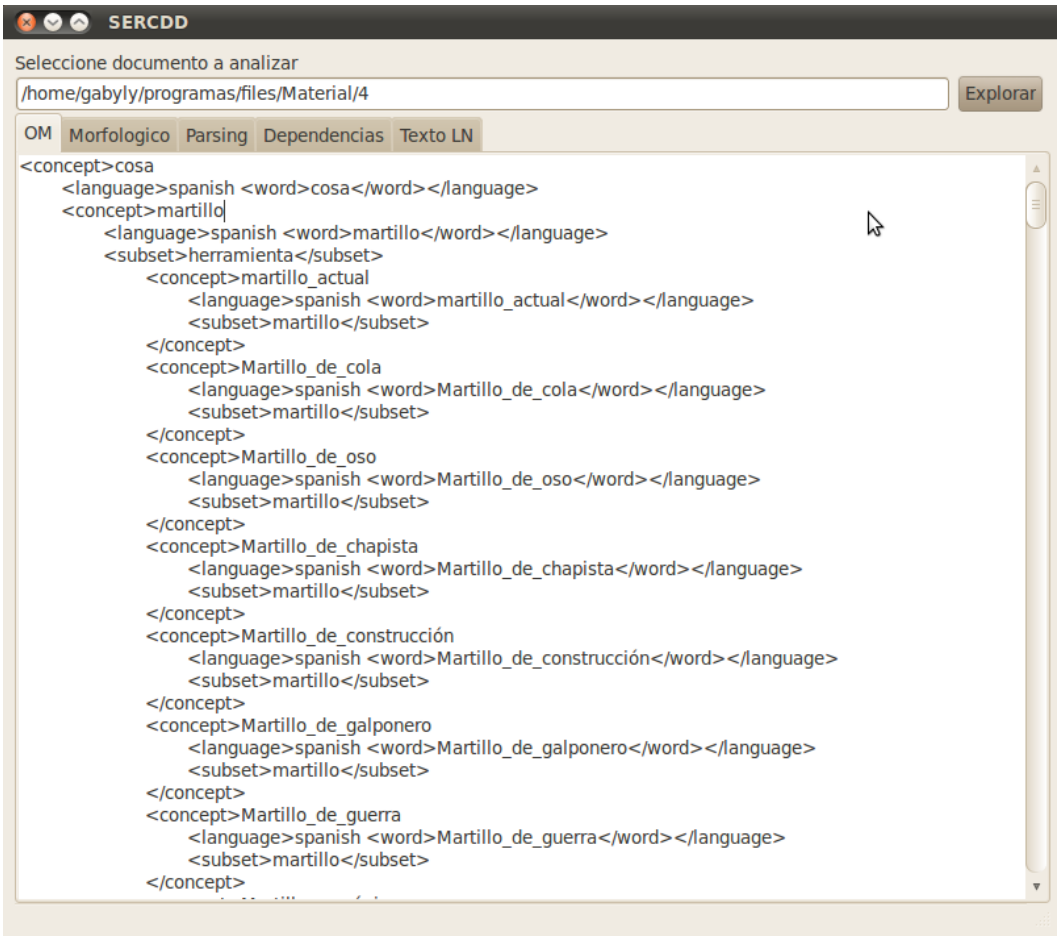

Figura 11 Interfaz de usuario del SERCDD Resultados ejemplo 1 en notación OM

- Análisis de los resultados obtenidos

En la figura 18 se muestran los resultados obtenidos en lenguaje OM, el cual permite especificar de manera no ambigua las relaciones de:

- Hiponimia
- Hiperonimia

Según las reglas que se definieron se identificaron correctamente las relaciones que se muestran en la Tabla 9.

Relación	Elemento A	Elemento B	Correcto
Hiperonimia	herramienta	martillo	Sí
Hiponimia	martillo actual	martillo	No
	martillo de cola	martillo	Sí
	martillo de oso	martillo	Sí
	martillo de chapista	martillo	Sí
	martillo de	martillo	Sí

	construcción		
	martillo de galponero	martillo	Sí
	martillo de guerra	martillo	Sí
	martillo mecánico	martillo	Sí
	martillo de cuña	martillo	Sí
	martillo neumático	martillo	Sí
	martillo de bola	martillo	Sí

Tabla 3 Análisis de Resultados Ejemplo 1

La figura 19 muestra a manera de árbol los resultados obtenidos del análisis

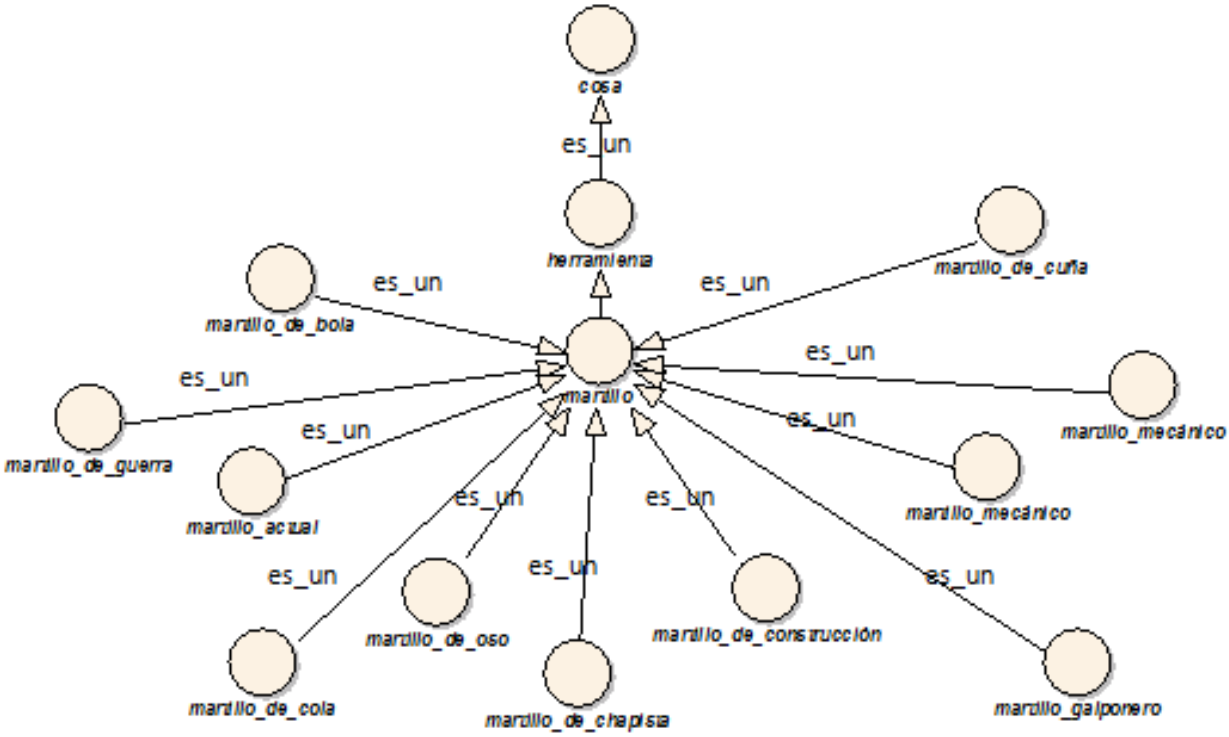

Figura 12 Ontología resultante del ejemplo 1 Martillo

5.2.2 Ejemplo 2 Escalpelo

- Corpus

Concepto descrito: Escalpelo

Fuente: Wikipedia (http://es.wikipedia.org/wiki/Escalpelo)

El escalpelo o bisturí, también llamado lanceta o cuchillo de cirujano, es un instrumento en forma de cuchillo pequeño, de hoja fina, puntiaguda, de uno o dos cortes, que se usa en procedimientos de cirugía, disecciones anatómicas, autopsias y vivisecciones.
…
Tipos de bisturí

Bisturí médico clásico: La forma y tamaño de los bisturís quirúrgicos tradicionales dependen del uso y del lugar anatómico. Pueden tener una hoja fija o desechable. Por ser instrumentos esencialmente de corte o incisión se fabrican con hojas extremadamente afiladas, solamente tocando un escalpelo médico levemente con las manos cortará la piel. La hoja normalmente es plana y recta, permitiendo realizar fácilmente cortes rectos o en línea y en caso de los de hoja fija ésta generalmente se curva gradualmente para una mayor precisión. Las hojas o cuchillas intercambiables tiene una ranura central para encajar en el mango y se distinguen numeradas por su forma según el tipo de corte que se desea hacer. Los mangos son ligeramente corrugados o con muchos surcos de sujeción y son igualmente numerados del 1 al 15, siendo los mas usados los No 3 (también denominado Bard-Parker o estándar), No 4 y No 7. Los mangos de hoja intercambiable son metálicos pero existen de desechables de material plástico.

El bisturí de diamante creado por el científico y médico venezolano Humberto Fernández Morán, cuya hoja fabricada con diamante se emplea en microcirugía como la oftalmológica. También se utiliza para realizar cortes ultrafinos en materiales desde tejidos biológicos hasta muestras lunares traídas a la Tierra por astronautas. Algunos orfebres lo utilizan para seccionar tramos precisos de materiales blandos, como la plata.

Además de los bisturís metálicos convencionales existen otros instrumentos para cortar o hacer diéresis o disección quirúrgica, y que por la tecnología incorporada permiten hacer hemostasia mediante cauterización en forma simultánea al corte, y son:

Bisturí eléctrico: o electrobisturí que puede ser de modalidades: unipolar o bipolardos según las diferentes tipos de energía que aplica.
Bisturí de rayos gamma(Gamma-Knife), rayos X (X-Knife) o de protones : que propiamente son formas de radioterapia concentrada en dosis altas y únicas.
Bisturí armónico : usa ultrasonido.
Bisturí láser: usa diferentes tipos de láser: YAG, de CO2, KTP.

Tabla 4 Corpus Ejemplo 2 Escalpelo

- Análisis del corpus en la herramienta

La Imagen muestra el resultado obtenido del análisis, una vez más se muestra el árbol de con las relaciones ontológicas extraídas en lenguaje OM.

Al igual que en el Ejemplo 1, no es relevante para el presente trabajo explicar los resultados de las pestañas adicionales.

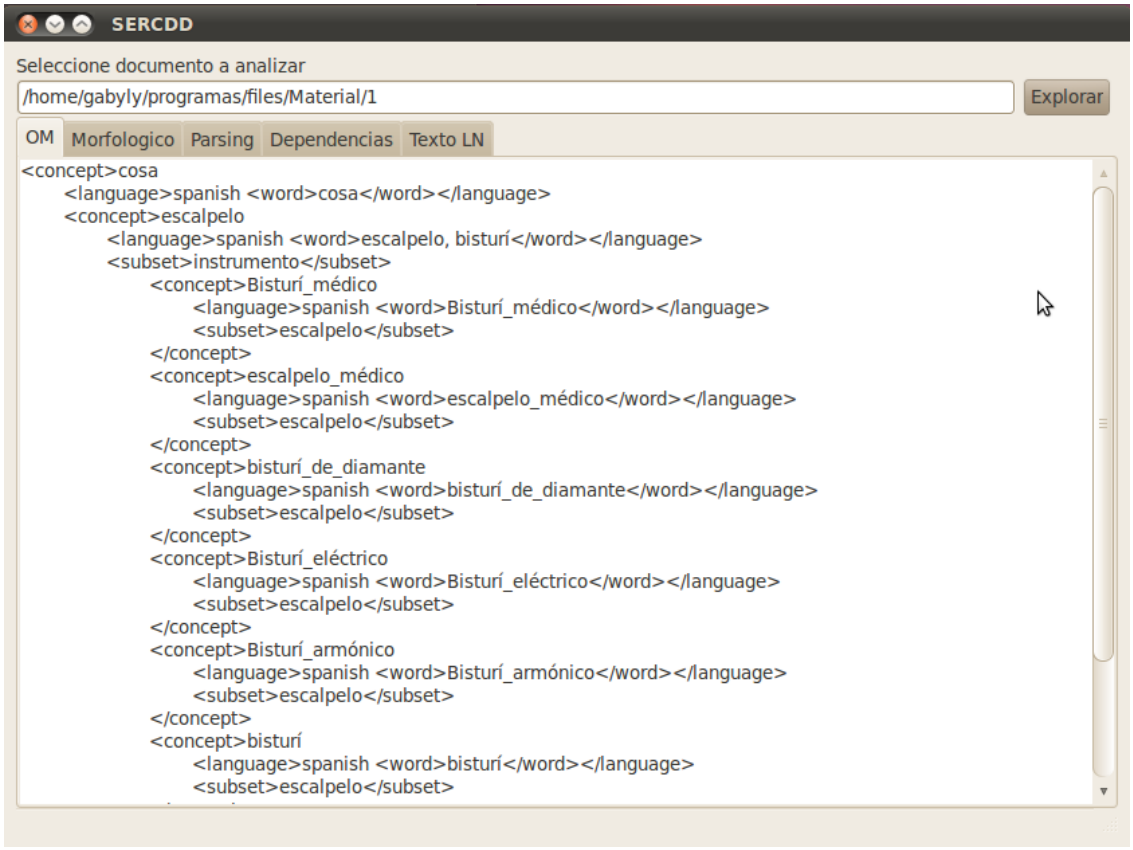

Figura 13 Interfaz de usuario del SERCDD Resultados ejemplo 2 en notación OM

- Análisis de los resultados obtenidos

En este segundo ejemplo, el corpus presenta información sobre una relación adicional, por lo tanto, en este caso las relaciones ontológicas identificadas son:

- Hiponimia
- Hiperonimia
- Sinonimia

Relación	Elemento A	Elemento B	Correcto
Hiperonimia	instrumento	escalpelo	Sí
Hiponimia	bisturí médico	escalpelo	Sí

	escalpelo médico	escalpelo	Sí
	bisturí de diamante	escalpelo	Sí
	bisturí eléctrico	escalpelo	Sí
	bisturí armónico	escalpelo	Sí
Sinonimia	escalpelo	bisturí	Sí
	lanceta	cuchillo	Sí
	corte	incisión	Sí

Tabla 5 Análisis de resultados Ejemplo 2

El árbol resultante del análisis es:

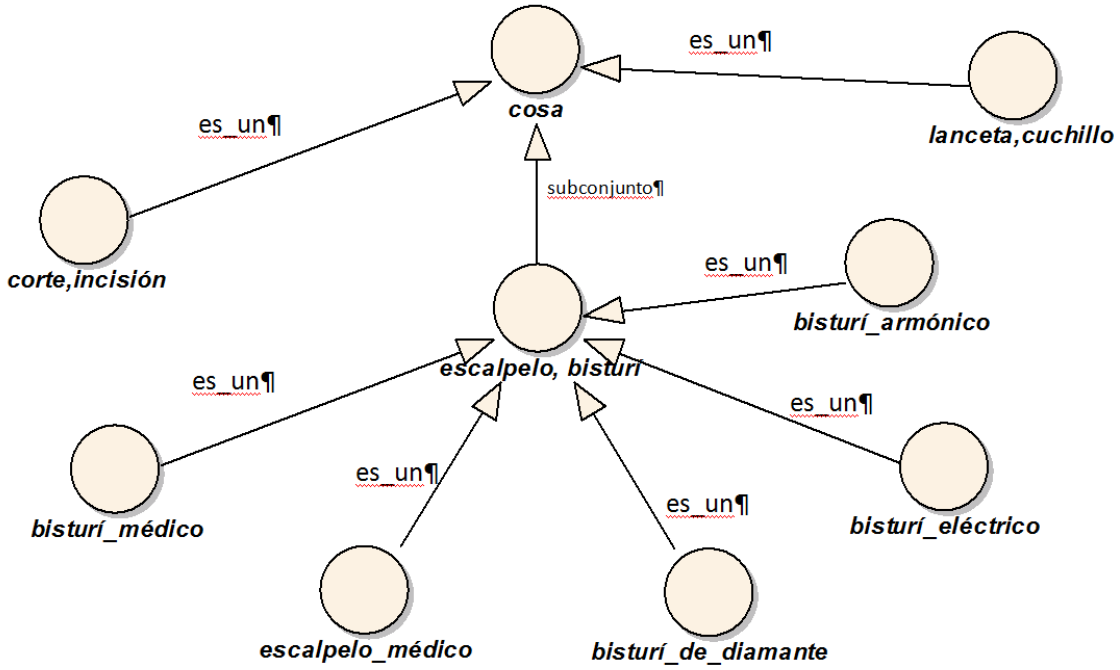

Figura 14 Ontología resultante del ejemplo 2 Escalpelo

5.2.3 Ejemplo 3 Formón

- Corpus

Concepto descrito: Martillo

Fuente: Wikipedia (http://es.wikipedia.org/wiki/Formón)

Formón
El formón o escoplo es una herramienta manual de corte libre utilizada en carpintería. Se compone de hoja de hierro acerado, de entre 4 y 40 mm de anchura, con boca formada por un bisel, y mango de madera. Su longitud de mango a punta es de 20 cm aproximadamente. El ángulo del filo oscila entre los 25-40º, dependiendo del tipo de madera a trabajar: madera blanda, menor ángulo; madera dura, mayor ángulo.

Los formones son diseñados para realizar cortes, muescas, rebajes y trabajos artesanos artísticos de sobre relieve en madera. Se trabaja con fuerza de manos o mediante la utilización de una maza de madera para golpear la cabeza del formón. .

Tabla 6 Corpus Ejemplo 3 Formón

- Análisis del corpus en la herramienta SERCDD

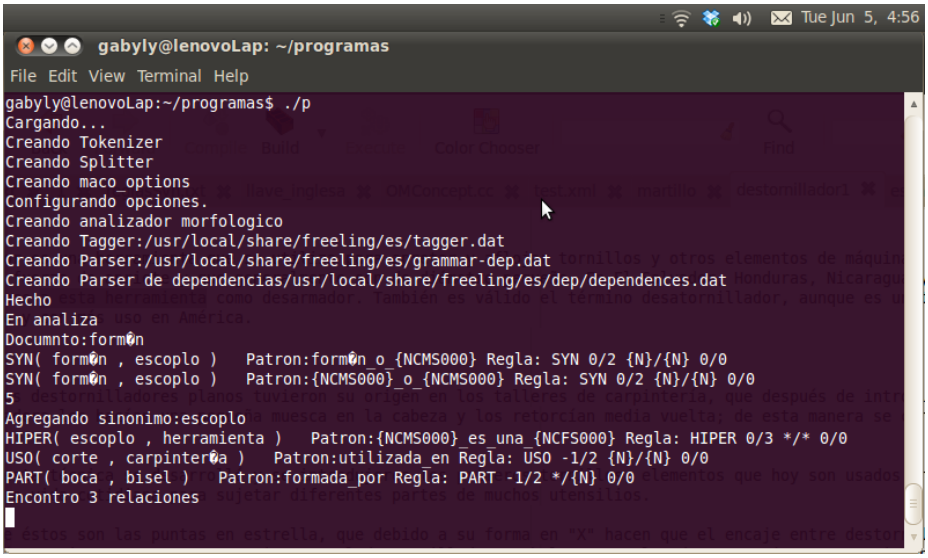

Figura 15 Salida en consola Ejemplo 3 Formón

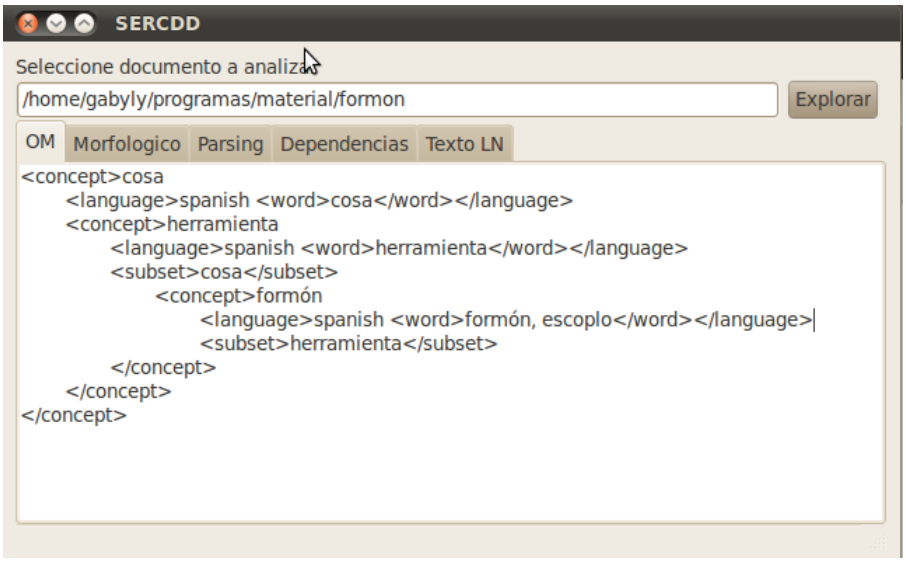

Figura 16 Interfaz de usuario del SERCDD Resultados ejemplo 3 en notación OM

- Análisis de los resultados obtenidos

Relación	Elemento A	Elemento B	Correcto
Hiperonimia	herramienta	formón	Sí
Sinonimia	formón	escoplo	Sí

Tabla 7 Análisis de los resultados obtenidos Ejemplo 3 Formón

La figura siguiente muestra a manera de árbol los resultados obtenidos del análisis

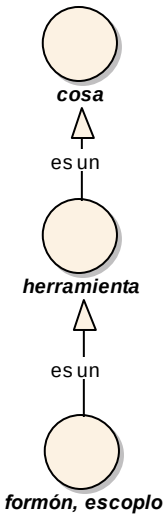

Figura 17 Ontología resultante del Ejemplo 3 Formón

5.2.4 Ejemplo 4 Banjo

- Corpus

Concepto descrito: Banjo

Fuente: Wikipedia (http://es.wikipedia.org/wiki/Banjo)

Banjo El banjo o banyo es un instrumento musical de cuatro, cinco, seis (banjo guitarra) o diez cuerdas constituido por un aro o anillo de madera circular de unos 35 cm de diámetro, cubierto por un "parche" de plástico o piel a modo de tapa de guitarra. El parche y el anillo de madera se ensamblan con tornillos metálicos (y el resonador de madera que se añade posteriormente también). La mezcla de materiales que conforman el banjo consigue uno de los instrumentos musicales con un sonido más característico e inconfundible que existen. Este instrumento procede de África occidental y fue introducido en el siglo XIX en Estados Unidos, donde los músicos negros explotaron sobre todo sus posibilidades rítmicas.

Tabla 8 Corpus Ejemplo 4 Banjo

- Análisis del corpus en la herramienta SERCDD

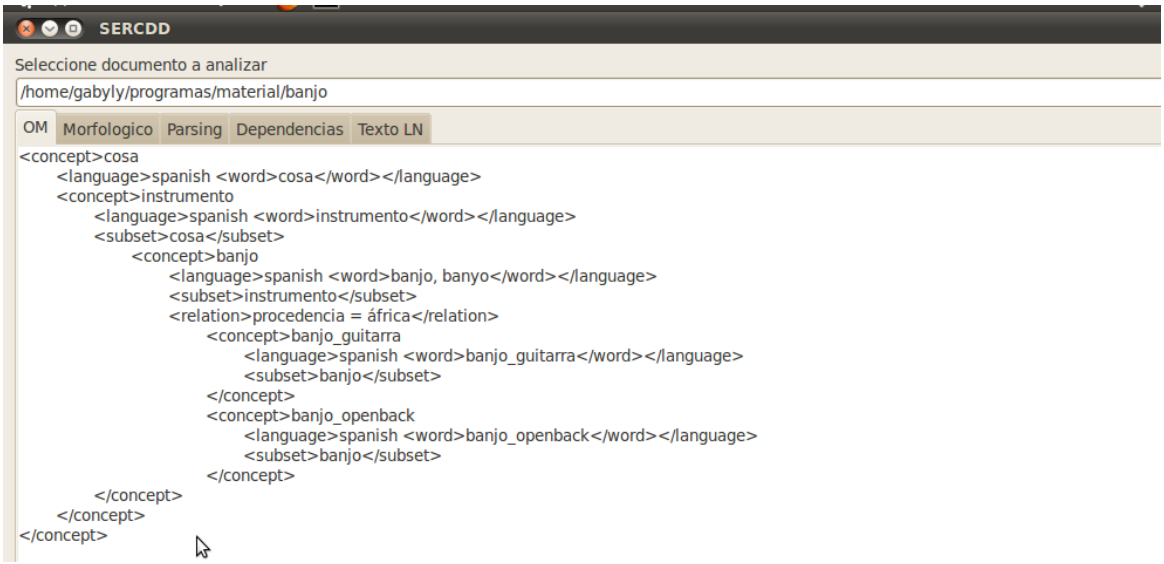

Figura 18 Interfaz de Usuario del SERCDD Resultados ejemplo 4 en notación OM

Relación	Elemento A	Elemento B	Correcto
Hiperonimia	instrumento	banjo	Sí
Sinonimia	banjo	banyo	Sí
Hiponimia	banjo guitarra	banjo	Sí
	banjo openback	banjo	Sí

Tabla 9 Análisis de resultados Ejemplo 4 Banjo

5.2.5 Ejemplo 5 Plato

- Corpus

Concepto descrito: Plato

Fuente: Wikipedia (http://es.wikipedia.org/wiki/Plato)

Plato
Utensilio doméstico común a todas las culturas, el plato es un recipiente útil para muy diferentes usos pero esencialmente empleado como pieza de la vajilla para comer. Los diccionarios lo definen: vasija circular y casi plana, ligeramente cóncava en su centro y borde extendido, diferenciando platos soperos u hondos y platos llanos. Vasijas hermanas son: el cuenco, la escudilla y la fuente. La cultura del plato lo ha convertido en un lujoso objeto de adorno, presente en los mejores museos del mundo.
Tipos de plato y la etiqueta en la mesa
Platos llanos.
Platos hondos, para tomar cremas, sopas y otros platos de cuchara.
Platos de postre, de menor diámetro que los anteriores utilizados para servir el postre.
Platos de café, los más pequeños de los cuatro tipos, para servir con la taza del café.
Platos de consomé, que combinan con sus correspondientes tazones.

Tabla 10 Corpus ejemplo 5 Plato

- Análisis del corpus en la herramienta SERCDD

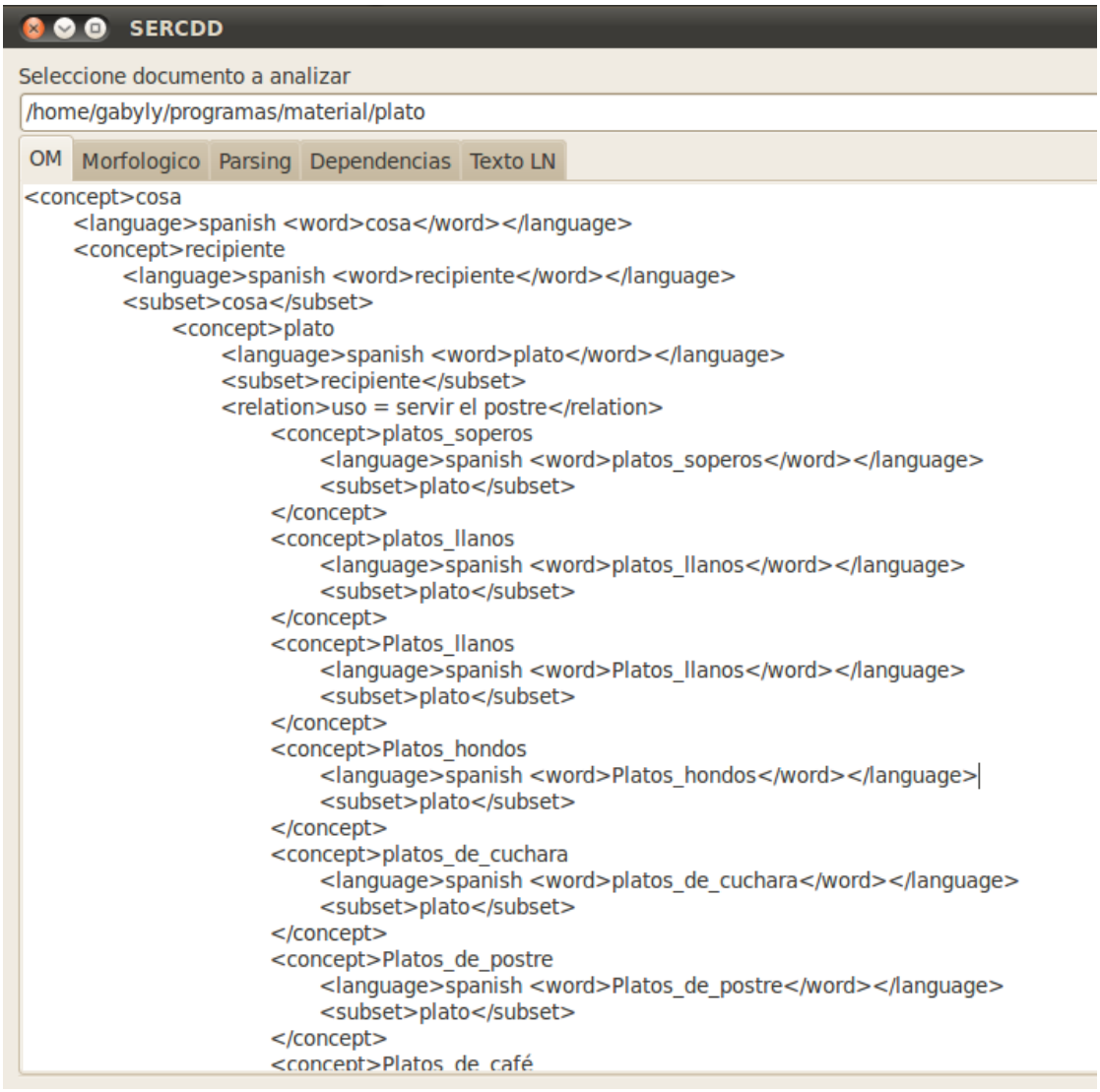

Figura 19Interfaz de Usuario del SERCDD Resultados ejemplo 5 en notación OM

Relación	Elemento A	Elemento B	Correcto
Hiperonimia	recipiente	plato	Sí
Hiponimia	platos soperos	plato	Sí
	platos llanos	plato	Sí
	Platos hondos	plato	Sí
	Platos de postre	plato	Sí
	platos de cuchara	plato	Sí
	Platos de café	plato	Sí
	Platos de consomé	plato	Sí
	platos de decoración	plato	Sí
	platos de aperitivo	plato	Sí
	platos iguales	plato	No

Tabla 11 Análisis de resultados Ejemplo 5

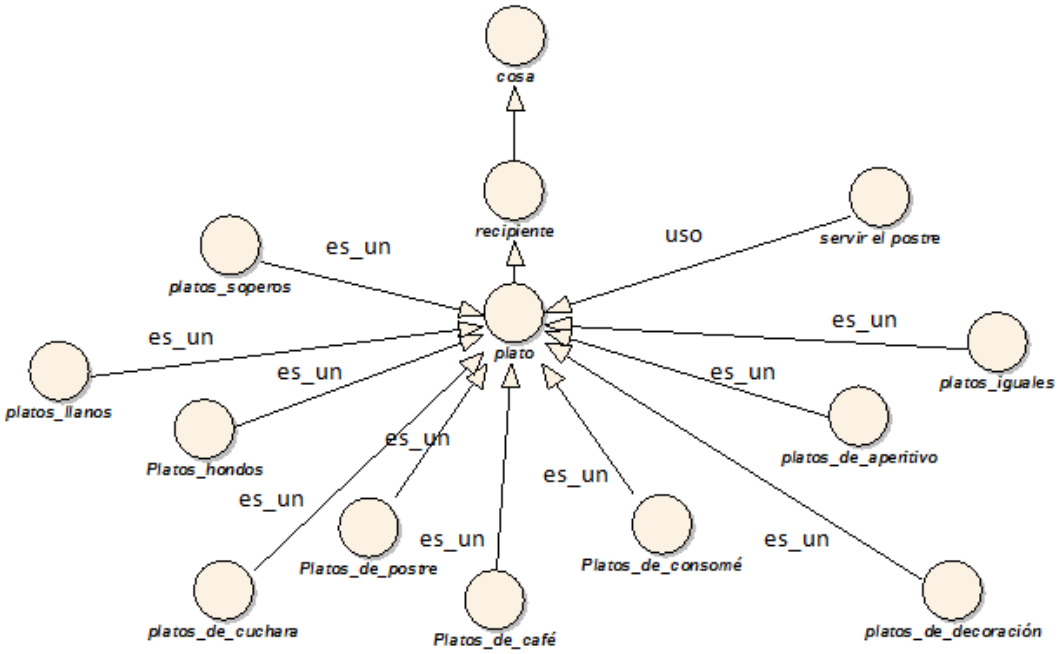

Figura 20 Ontología resultante Ejemplo 5 Plato

5.3 Análisis de resultados

Para el análisis de los resultados obtenidos, se presenta una medición de porcentajes, para cada relación. Se compara el porcentaje de relaciones encontradas contra aquellas relaciones contenidas en el texto.

El análisis se realiza sobre 100 documentos analizados. Los documentos fueron obtenidos de diversas fuentes, siendo la principal de ellas, y de la cual se obtuvieron los mejores resultados wikipedia.

Los documentos analizados describen objetos concretos en la siguiente proporción:

- 50 herramientas
- 10 instrumentos
- 10 utensilios de cocina
- 10 animales
- 10 plantas
- • 5 piezas de ropa
- • 5 muebles

La tabla 18 muestra el porcentaje de relaciones encontradas y el porcentaje de relaciones que el sistema falló en encontrar. Se hace el análisis para cada relación buscada.

El sistema está diseñado y configurado para encontrar relaciones de: hiponimia, hiperonimia, sinonimia, uso y material. Aquellas relaciones adicionales que los documentos pudieran contener, no son consideradas. Se consideran solo aquellas relaciones definidas en el archivo de reglas.

No todos los documentos describen todas las relaciones, por lo que el porcentaje no está calculado en base al número total de documentos analizados, sino en base al número de documentos que contienen información acerca de cada relación.

Relación	Encontradas	No encontradas	Razón principal de fallo
Hiperonimia	98	2	Complejidad de la oración.
Hiponimia	96	4	Complejidad de la oración
Sinonimia	89	11	Distancia entre los elementos.
Uso	87	13	Distancia entre los elementos.
Partes	73	27	Complejidad de la oración. Distancia entre los elementos. Anáforas
Material	72	28	Complejidad de la oración. Anáforas.

Tabla 12 Análisis de resultados

5.4 Muestra de integración manual con OM

En esta sección se muestra un ejemplo simple de la integración que el Sistema de Extracción y Representación del Conocimiento a partir de documentos descriptivos tiene con el sistema Ontology Merging (Cuevas Rasgado, 2006).

La integración de ambos sistemas se realiza de manera manual, almacenando en archivos de texto plano los resultados obtenidos en el SERCDD y posteriormente, cargarlos al sistema OM.

En el trabajo OM, todas las ontologías utilizadas, fueron hechas a mano, a partir de documentos leídos por la autora y las relaciones extraídas manualmente.

En este ejemplo, las Ontologias A y B, fueron generadas completamente de manera automática. (Guzmán Arenas & Cuevas Rasgado, 2010)

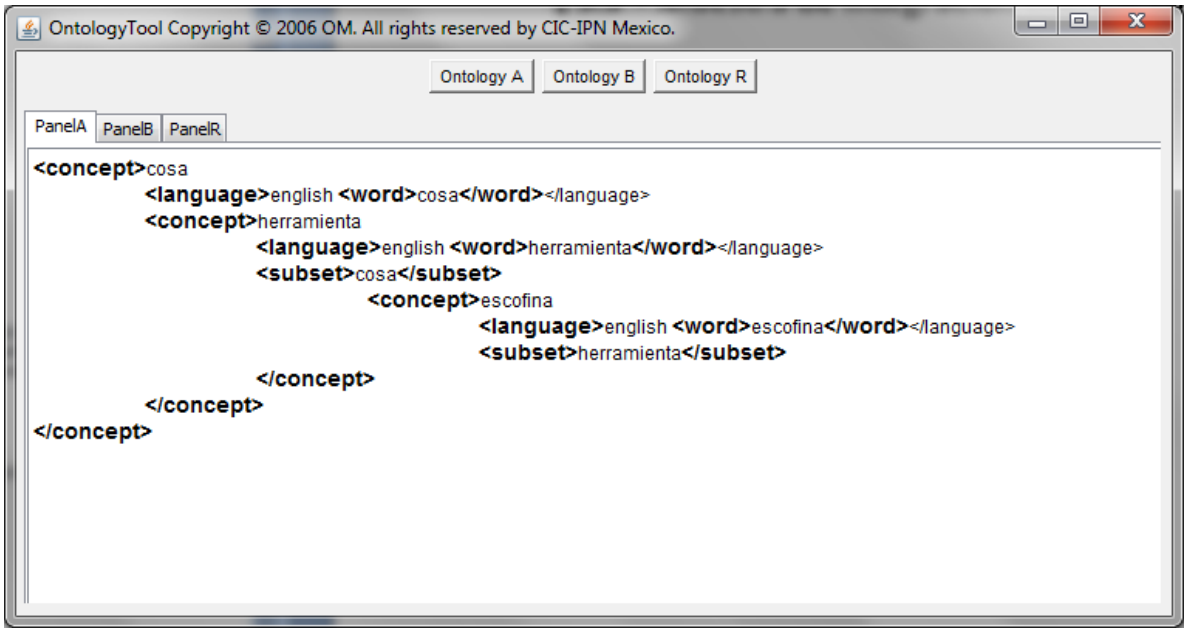

Figura 21 Muestra integración SERCDD-OM Ontologia A

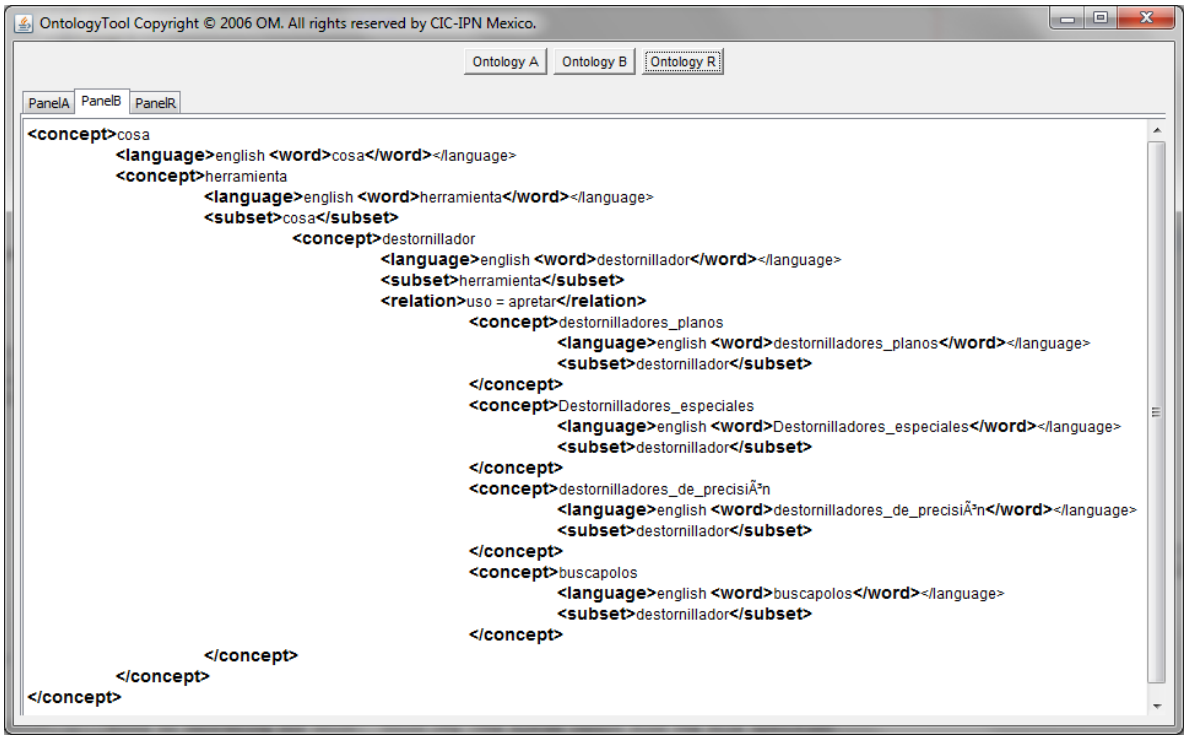

Figura 22 Muestra integración SERCDD-OM Ontologia B

La Figura 30 muestra la Ontología R, la cual es el resultado de la fusión de la Ontología A (Figura 28) y la Ontología B (Figura 29). La fusión la realiza de manera automática el fusionador de ontologías OM (Cuevas Rasgado, 2006).

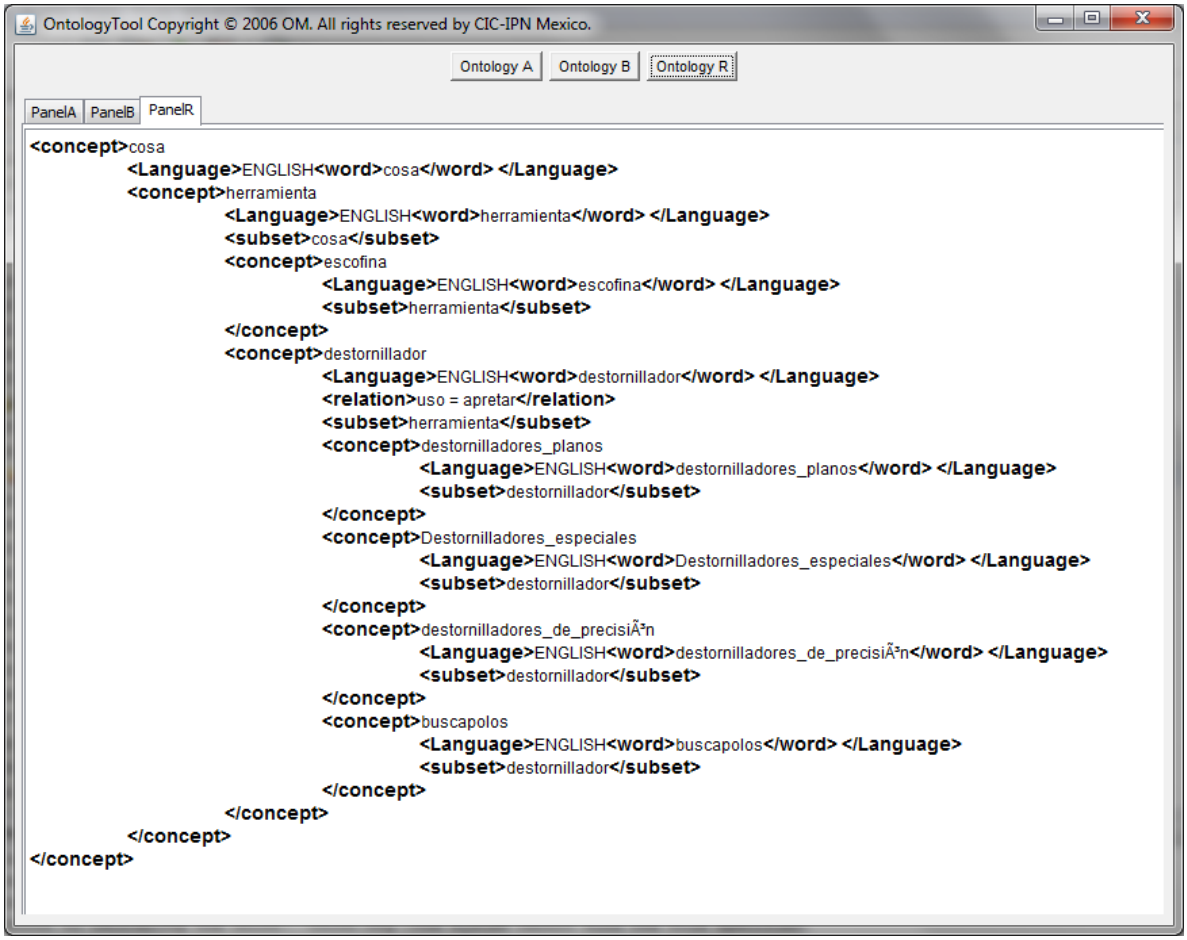

Figura 23 Muestra integración SERCDD-OM Ontologia R

6 LIMITACIONES

La principal limitación del SERCDD se presenta cuando los elementos de una relación dentro del texto están a una gran distancia. Y especialmente, cuando se encuentran en diferentes oraciones.

Ejemplo:

La forma y tamaño de los bisturís quirúrgicos tradicionales dependen del uso y del lugar anatómico. Pueden tener una hoja fija o desechable
El formón o escoplo es una herramienta manual de corte libre utilizada en carpintería. Se compone de hoja de hierro acerado, de entre 4 y 40 mm de anchura, con boca formada por un bisel, y mango de madera.

Tabla 1 Corpus problemático con elementos de la relación a gran distancia.

Relación contenida en el texto:

```
TIENE(bisturí_quirurjico, hoja)

TIENE(formón, hoja)

TIENE(formón, mango)
```

El concepto al que hace referencia la segunda oración está en una oración anterior, e incluso se puede dar el caso en que dicho concepto se encuentre en oraciones aún anteriores.

La relación TIENE(A,B) que se describe en el corpus de la tabla 19, no podría ser encontrada por el SERCDD tal y como esta. La frase "pueden tener" y el verbo "se" hacen referencia a un elemento mencionado con cierta anterioridad, anáforas como estas deberían ser resueltas en un paso previo, como se menciona en los trabajos futuros, agregar como un pre-procesamiento del corpus un módulo de resolución de anáforas mejoraría considerablemente los resultados obtenidos por el SERCDD.

7 APLICACIONES

Las aplicaciones que se le pueden dar a un sistema de aprendizaje de ontologías son muchas, el presente trabajo está pensado en ser aplicado al objetivo de contestar preguntas, lo cual es parte del proyecto OM desarrollado en el CIC-IPN.

A pesar de haber sido desarrollado pensando en específico en una aplicación, existen distintas aplicaciones para las que podría ser adaptado, algunas de ellas son:

- Procesamiento del Lenguaje Natural y la traducción automática
- Web Semántica
- Ingeniería del Conocimiento y Gestión
- Comercio electrónico
- Recuperación de Información e Integración de la Información
- Motores de búsqueda inteligentes
- Bibliotecas Digitales
- Business Process Modeling

8 CONCLUSIONES

El presente trabajo es un avance en el área de aprendizaje de ontologías, el cual está categorizado en varias ramas según el origen de los datos de entrada. Este trabajo se centra en el extraer datos de textos en lenguaje natural.

La condición de que los textos a analizar sean descriptivos es necesaria para poder tener una cierta confianza en que las estructuras del texto suelen ser sencillas, el lenguaje claro y la información presentada puede ser prevista.

Las reglas generadas mostraron buenos resultados dentro de los documentos analizados, las relaciones ontológicas son las que presentan menor índice de error.

- Hiperonimia
- Hiponimia
- Sinonimia

Analizando textos con una estructura relativamente predecible es posible limitar el número de casos a considerar cuando se realiza un análisis en busca de ciertas relaciones específicas.

Un pre procesamiento del corpus a analizar es importante para poder obtener resultados confiables, ya que se puede contar con mayor información de la que un texto sin pre procesamiento ofrece, por ejemplo las clases gramaticales, ontológicas y lemas de las palabras en un texto es información que resulta muy útil para realizar análisis más complejos.

El método propuesto es independiente de idioma y puede aplicarse el mismo principio a cualquier idioma con tan solo modificar los archivos de configuración y crear el sistema de reglas acorde al idioma deseado.

Una ventaja importante del método es que no necesita de grandes cantidades de información previa ni gran poder de cómputo para presentar buenos resultados. Trabajos existentes requieren de grandes cantidades de corpus previamente analizados para tener datos de probabilidades.

No es necesaria la interacción del usuario, el método es no supervisado, sin embargo provee la posibilidad de tener un cierto grado de interacción o entrada por parte del usuario para obtener mejores resultados.

9 Trabajos futuros

El presente trabajo es parte de un proyecto de investigación dentro del Centro de Investigación en Computación del Instituto Politécnico Nacional, cuyo objetivo final es poder obtener información de la web para contar con un banco de conocimiento virtualmente ilimitado y tener la posibilidad de ingresar consultas en forma de pregunta y obtener una respuesta concreta.

El trabajo futuro directamente relacionado con el proyecto OM sería la integración del SERCDD a los subsistemas que forman parte del proyecto.

9.1 Integración de un buscador jerárquico automático (araña)

Se plantea la idea de conectar el SERCDD con un buscador que automáticamente recorra la web en busca de documentos y sin necesidad de un usuario entregue la lista de documentos al sistema para que sean analizados automáticamente.

El buscador deberá identificar que los documentos cumplan con las características necesarias para un correcto procesamiento.

9.2 Integración de un sistema de resolución de anáforas

Como se mencionó en la sección 6, la mayor limitación del SERCDD se encuentra en aquellas relaciones que no incluyen todos sus elementos a una cierta distancia, en particular dentro de la misma oración. Esto es muy común en el idioma español, por lo que la integración de un paso adicional al pre-procesamiento del corpus, en el cual se integrara un sistema de resolución de anáforas, permitiría que los resultados arrojados por el SERCDD fueran más y mejores. Dentro del proyecto OM, ya existe un sistema de resolución de anáforas, desarrollado por alumnos de ESCOM (Toledo Gómez & Valtierra Romero, 2012). Queda como trabajo futuro, la integración de este sistema al SERCDD.

9.3 Integración con el sistema de unión de ontologías

De igual manera que en el punto anterior, la salida del SERCDD debería ser automáticamente enviada al sistema de OM para su fusión con la base de conocimientos global del proyecto.

En la sección 5.3 se demostró el modo de integración de ambos sistemas, aunque los resultados son los mismo y definitivamente el utilizar SERCDD previo a OM es de gran ayuda, sin embargo lo ideal es que este proceso este integrado de manera transparente al usuario.

Para tal fin se deberá contar con un método dentro de SERCDD que identifique aquellos documentos que requieran de fusión y que generen mejores resultados.

9.3.1 Relaciones no binarias

El presente trabajo presenta un método para extraer relaciones a partir de un documento en lenguaje natural, todas las relaciones que se localizan son del tipo binario, sin embargo es posible extender el método para considerar relaciones no únicamente binarias sino n-arias y además de distintos tipos, no sólo ontológicas o específicas de ámbito.

9.3.2 Análisis de documentos con temporalidad

El análisis de los documentos y todas las pruebas realizadas se limitó a ciertos documentos cuya característica principal es que la información que presentan es estática.

Dada la naturaleza del proyecto OM es necesario contar con un sistema análogo al SERCDD pero que cuente con la capacidad de identificar tiempo en los documentos, con este trabajo sería posible extender el análisis a documentos descriptivos de objetos no concretos o de mayor complejidad, como países o personas.

Un ejemplo claro de documentos con un alto grado de información de tiempo es el de las biografías de personajes, ya que la información que presentan es esencialmente temporal.

Dentro de este trabajo, adicional a la necesidad de un extractor de información temporal es necesario extender el concepto de definiciones dentro del lenguaje OM.

Referencias

Agrawal, R., Imieliński, T., & Swami, A. (1993). Mining association rules between sets of items in large databases. *Proceedings Of the ACM SIGMOD Conference on Management of Data*, (pp. 207-216). Nueva York, EUA.

Álvarez, M. (1988). *Tipos de escrito I: Narración y Descripción.* Madrid: Arco/Libros.

Aussenac-Gilles, N., Despres, S., & Szulman, S. (2008). The TERMINAE Method and Platform for Ontology Engineering from Texts. *Proceeding of the 2008 conference on Ontology Learning and Population: Bridging the Gap between Text and Knowledge.* IOS Press.

Biébow, B., & Szulman, S. (1999). TERMINAE: a linguistic-based tool for the building of a domain ontology. *Proceedings of the 11th European Workshop on Knowledge Acquisition, Modelling and management* (pp. 49-66). Dagstuhl, Alemania: Springer-Verlag.

Buitelaar, P., Olejnik, D., & Sintek, M. (2003). OntoLT: A protégé plug-in for ontology extraction from text. *Proceedings of the International Semantic Web Conference*, (pp. 31-44).

Carreras, X., Chao, I., Padró, L., & Padró, M. (2004). FreeLing: An Open-Source Suite of Language Analyzers. *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC'04).*

Cimiano, P. (2005). Ontology Learning and Population: Algorithms, Evaluation and Applications. Universidad de Karlsruhe.

Cuevas Rasgado, A. D. (2006). Unión De Ontologías Usando Propiedades Semánticas. México D.F.: Centro de Investigación en Computación.

Davis, R., Shrobe, H., & Szolovits, P. (1993). What is a Knowledge Representation? *AI Magazine , 14* (1), 17-33.

Faure, D., & Poibeau, T. (2000). First experiments of using semantic knowledge learned by ASIUM for information extraction task using INTEX. *Proceedings of the Workshop on Ontology Learning, 14th European Conference on Artificial Intelligence.* Berlín, Alemania.

Genesereth, M. R., & Nilsson, N. (1987). *Logical Foundations of Artificial Intelligence.* San Mateo, CA.: Morgan Kaufmann Publishers.

Gruber, T. (1993). A translation Approach to portable ontology specifications. . *Knowledge Acquisition , 5*, 199-220.

Hahn, U., & Schulz, S. (2000). Towards Very Large Terminological Knowledge Bases: A Case Study from Medicine. *Proceedings of the 13th Biennial Conference of the Canadian Society on Computational Studies of Intelligence: Advances in Artificial Intelligence*, (pp. 176-186).

Hearst, M. (1992). Automatic acquisition of Hyponyms from large text corpora. *Proceedings of the Fourteenth International Conference on Computational Linguistic.* Nantes, Francia.

Hsu, W.-L., Wu, S.-H., & Chen, Y.-S. (2001). Event Identification Based On The Information Map - INFOMAP. *Natural Language Processing and Knowledge Engineering Symposium of the IEEE Systems*, (pp. 1661-1672). Tucson, Arizona, EUA.

Jurafsky, D., & Martin, J. H. (2009). *Speech and Language Processing. An Introduction to Natural Language Processing,.*

Kietz, J.-U., Mädche, A., Maedche, E., & Volz, R. (2000). A Method for Semi-Automatic Ontology Acquisition from a Corporate Intranet. *Proceedings of the EKAW'2000 Workshop "Ontologies and Text".* Juan-Les-Pins, Francia.

Lampert, A. (2004). *A Quick Introduction to Question Answering.*

Maedche, A., & Staab, S. (2000). Discovering Conceptual Relations from Text. *Proceedings of the 14th European Conference on Artificial Intelligence*, (pp. 21-25). Berlín.

Maedche, A., & Staab, S. (2001). Ontology Learning for the Semantic Web. *IEEE Intelligent Systems , 16* (2), 72-79.

Maedche, A., & Volz, R. (2001). The Text-To-Onto Ontology Extraction and Maintenance. *Proceedings of the ICDM Workshop on integrating data mining and knowledge management.* San Jose, California, EUA.

Meneses, J. C. (2011). *Sistema Analizador de Frases Temáticas.* Trabajo Terminal Escuela Superior de Cómputo.

Minsky, M. (1974, Junio). A Framework for Representing Knowledge. *MIT-AI Laboratory Memo 306* .

Morin, E. (1999). Automatic acquisition of semantic relations between terms from technical corpora. *Proceedings Of the Fifth International Congress on Terminology and Knowledge Engineering.* Vienna.

Musen, M. (1993). An overview of Knowledge Acquisition. Berlín: Springer-Verlag.

Singhal, A. (2001). Modern Information Retrieval: A Brief Overview. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering* , 35-43.

Studer, R., Benjamins, V., & Fensel, D. (1998). Knowledge engineering: Principles and methods. *Data & Knowledge Engineering , 25* (1 y 2), 161-197.

Velardi, P., Navigli, R., & Missikoff, M. (2002, Noviembre). Integrated approach for Web ontology learning and engineering. *IEEE Computer* .

Vossen, P., Bloksma, L., Rodriguez, H., Climent, S., Calzolari, N., Roventini, A., et al. (1998). The EuroWordNet Base Concepts and Top Ontology. París, Francia.

W3C. (2009, Octubre 27). *OWL 2 Web Ontology Language Document Overview*. Retrieved Mayo 2012, from W3C Recommendation: http://www.w3.org/TR/2009/REC-owl2-overview-20091027/

W3C. (2004, 02 10). *Resource Description Framework (RDF)*. Retrieved from http://www.w3.org/RDF/

Wu, S.-h., & Hsu, W.-L. (2002). SOAT: A Semi-Automatic Domain Ontology Acquisition Tool from Chinese Corpus. *Proceedings of the 19th International Conference on Computational Linguistics.* Taipei, Taiwan.

Xu, F., Kurz, D., Piskorski, J., & Schmeier, S. (2002). A Domain Adaptive Approach to Automatic Acquisition of Domain Relevant Terms and their Relations with Bootstrapping. *Proceedings of LREC 2002, the third international conference on language resources and evaluation.* Las Palmas, España.