Extracción automática de relaciones taxonómicas de tipo hipónimo / hiperónimo en corpus de dominio

Hugo Raziel Lasserre Chávez

Asesora: Dra. Mireya Tovar Vidal
Facultad de Ciencias de la Computación

Benemérita Universidad Autónoma de Puebla

Tesis presentada para obtener el grado de

Maestro en Ciencias de la Computación

Puebla, Puebla. - Enero 2018

Agradecimientos

Agradezco a la Facultad de Ciencias de Computación de la Benemérita Universidad Autónoma de Puebla por permitirme realizar mis estudios de maestria en el programa de Maestria en Ciencias de la Computación con especialidad en base de datos y recuperación de información.

Agradezco al Consejo Nacional de Ciencia y Tecnologia, CONACYT, por las dos becas que me fueron otorgadas una durante Julio 2015 - Julio 2017 y la otra durante el periodo de agosto 2017, la segunda mediante el proyecto de investigación con referencia CB 2015/257357.

Gracias a todos los que, en algún momento determinado, me alentaron a no desistir en este proceso que para mí es más que una tesis de maestría, llegue a esta maestría esperando diferentes cosas, pero no tantas como las que pase en su transcurso. Lleve a cabo muchas actividades, conocí lugares y gente que en otro momento no imaginé llegar a vivir y por todo eso estoy profundamente agradecido.

Mis inquietudes profesionales y académicas fueron acogidas por la Dra. Mireya Tovar Vidal, quien es mi asesora en este trabajo de tesis y a quien le agradezco por haberme encaminado para poder realizarla durante estos dos años de maestría y quien fue un apoyo incondicional para que esto se terminara por completo y por lo cual estoy infinitamente agradecido.

De igual manera agradezco a la Dra. María Josefa Somodevilla García por su paciencia con Andie y conmigo quien con sus palabras de aliento y ayuda incondicional mediante todo este proceso nos ayudó a terminar nuestro trabajo a ambos.

Agradezco al Dr. José Alejandro Reyes Ortiz, quien a pesar de no ser de la universidad y de las dificultades de que en algunos casos no estuviera presente físicamente en mis evaluaciones, siempre estuvo pendiente de mi trabajo ya fuera personalmente o por videollamada, sus comentarios fueron de muchísima ayuda para la finalización de este trabajo de tesis.

Gracias a Andie, mi compañera de universidad, maestría y de vida, quien fue la que me convenció de entrar a esta maestría (¡juntos entramos, juntos nos vamos!) a pesar de todas las peleas y situaciones en las que nos vimos envueltos a lo largo de esta etapa de vida, siempre salimos adelante los dos apoyándonos principalmente uno en el otro.

De igual manera agradezco a todo mi jurado evaluador de tesis por ayudarme en las etapas finales de este trabajo y quienes fueron la última palabra de evaluación para este trabajo.

Agradezco infinitamente a mi familia, especialmente a mi mama y a mi papa por ese apoyo incondicional que me brindaron siempre, de una u otra forma algunas cosas se vuelven difíciles y ellos nunca dudaron de mi ni me dejaron de apoyar de cualquier tipo de forma que fuera necesario. Los amo a todos!.

Resumen

Hoy en día muchas aplicaciones de procesamiento de lenguaje natural hacen uso de tesauros, listas de palabras o de términos clasificados de manera taxonómica, empleados para representar conceptos, como el sistema WordNet [1] , que sirve como un diccionario de conocimiento léxico para el procesamiento de la semántica de palabras y documentos.

Construir dichas taxonomías puede ser una tarea difícil y extremadamente lenta. Por lo que ha surgido un creciente interés en encontrar métodos que puedan aprender relaciones taxonómicas y construir jerarquías semánticas de manera automática [2].

Las jerarquías de conceptos son importantes porque permiten la estructuración de información mediante categorías. Las relaciones de tipo "is-a" (es un) son un problema importante en la construcción de taxonomías, por lo que la adquisición automática de ese tipo de relaciones puede ser utilizado para construir una taxonomía e incluso una ontología.

En este trabajo de investigación se proponen modelos para la extracción de relaciones tipo hipónimo e hiperónimo a través de métodos que permitan la identificación de estos tipos de relaciones en corpus de dominio específico, como son los métodos de agrupamiento (Análisis Formal de Conceptos) y/o los métodos por patrones léxico-sintácticos, utilizando técnicas de procesamiento de lenguaje natural.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

Este capítulo introduce la problemática del trabajo, de donde sale la idea del mismo, los objetivos, así como también la metodología que se sigue para resolverlo y que más adelante en el capítulo 4 se especifica.

1.1. Antecedentes

SemEval (Semantic Evaluation) es un conjunto de evaluaciones o pruebas de sistemas computacionales de análisis semántico. SemEval inició en 1998 llamado Senseval y en 2001 se convirtió en SemEval. Los organizadores de dichas pruebas son diferentes universidades que se enfocan en el ámbito del Procesamiento de Lenguaje Natural y de la semántica de la lingüística computacional.

Esta tesis tiene como origen la tarea #13 del SemEval-2016 llamada Taxonomy Extraction Evaluation (TExEval-2). Esta tarea provee un corpus que es Wikipedia. Dicha tarea se divide en cuatro subtareas:

1. Construcción de taxonomía.
2. Identificación de hiperonimia.
3. Construcción de taxonomía multilenguaje.

4. Identificación de hiperonimia multilenguaje

SemEval-2016 provee datos de prueba para la evaluación de la propuesta que realizarán los participantes. Se proveen conceptos que tratan de taxonomías ya hechas, por ejemplo, conceptos de una taxonomía manual de vehículos, conceptos de vehículos obtenidos de WordNet, conceptos de plantas de una taxonomía manual y conceptos de plantas obtenidos de Eurovoc y WordNet. La propuesta que realicen los participantes tiene que identificar hiperonimia y construir su taxonomía de manera automática.

Esta tesis se enfoca en la segunda subtarea propuesta por SemEval-2016 en su *Task* #13, es decir, la identificación de relaciones semánticas de tipo hipónimo / hiperónimo.

1.2. Objetivos generales y específicos

A continuación, se describe el objetivo general y los objetivos específicos a lograr con la realización de esta tesis:

1.2.1. Objetivo general

Desarrollar modelos para la extracción automática de relaciones taxonómicas de tipo hipónimo/hiperónimo en un corpus de dominio específico mediante la aplicación de métodos basados en patrones y agrupamiento.

1.2.2. Objetivos específicos

1. Construir un sistema de recuperación de información para la extracción de información obtenida desde Wikipedia.

2. Identificar características que faciliten el agrupamiento de términos.

3. Construir modelos que permitan la detección de relaciones de tipo hipónimo e hiperónimo.

4. Probar y evaluar los modelos propuestos sobre los datos de entrada y prueba proporcionados.

1.3. Organización del documento

El resto del documento se organiza de la siguiente manera:

- En el Capítulo 2 se aborda la investigación de la terminología que se utilizará a lo largo de este documento, es decir, el marco teórico.
- En el Capítulo 3 se investiga el estado del arte en cuanto a todos los tipos de extracción automática de relaciones semánticas, métodos basados en diccionarios, agrupamiento, en patrones, por estadística y por medio del análisis formal de conceptos.
- El Capítulo 4 aborda de manera específica las propuestas de solución al problema presentado previamente.
- En el Capítulo 5 se presentan los resultados experimentales de las propuestas de solución.
- En el Capítulo 6 se presentan las conclusiones de este trabajo de tesis.

Finalmente se abordan las conclusiones de este trabajo así como también la bibliografía utilizada a lo largo de este documento.

Capítulo 2

Marco teórico

En este capítulo se abordará el marco teórico de la terminología que se abordará en la tesis, definiendo su significado para mayor comprensión de la misma.

2.1. Lenguaje

De acuerdo a [22] un lenguaje se puede definir de diferentes formas, desde el punto de vista funcional lingüístico se define como una función que expresa pensamientos y comunicaciones entre la gente, mediante escritura o voz. Desde un punto de vista formal se define como un conjunto de frases que se forma con combinaciones de elementos tomados de un conjunto llamado alfabeto, respetando un conjunto de reglas de formación es decir reglas sintácticas o gramaticales y de sentido (semánticas).

2.1.1. Lenguaje natural

De la misma forma [22] define al lenguaje natural al medio que utilizamos de manera cotidiana para establecer nuestra comunicación con las demás personas, el cual se ha venido perfeccionando a partir de la experiencia y que puede ser utilizado en situaciones completas y razonar de manera muy sútil.

2.1.2. Lenguaje de programación

El autor de [22] define un lenguaje de programación como un lenguaje formal definido con un conjunto de elementos llamados componentes léxicos los cuales están organizados a través de constructores (Reglas gramaticales) que permiten escribir un programa y que este sea entendido por una máquina y que este pueda ser trasladado a diferentes maquinas para su funcionamiento similar en otro sistemas. Un lenguaje de programación es un conjunto de instrucciones secuenciales ordenadas que permiten realizar una tarea o trabajo específico.

Wordnet Wordnet es una inmensa base de datos léxica en inglés. Sustantivos, verbos, adjetivos y adverbios son agrupados en conjuntos de sinónimos llamados synsets, cada uno expresando un concepto distinto. Los synsets se ínter-relacionan mediante relaciones conceptuales-semánticas y léxicas. Wordnet puede ser visto como una combinación de diccionarios y tesauros. Su principal uso es el análisis de texto y aplicaciones de inteligencia artificial [1].

Python Python es un lenguaje de programación interpretado, orientado a objetos y de alto nivel con semántica dinámica. Es un lenguaje de programación sencillo y enfatizado en la legibilidad por lo cual reduce el costo de mantenimiento de programas, también soporta módulos y paquetes que motivan a los usuarios a programar de manera modular y a reutilizar código. [23]

AWK AWK es un lenguaje de programación extremadamente versátil para trabajar con archivos, es un excelente manejador y escritor de reportes. Este lenguaje de programación hace uso amplio de listas asociativas (listas indexadas por claves) y de expresiones regulares. Fue una de las primeras herramientas en aparecer en los sistemas tipo UNIX. [24]

2.1.3. Procesamiento de lenguaje natural (PLN)

El procesamiento de lenguaje natural (PLN) es el enfoque computarizado para analizar texto que se basa tanto en un conjunto de teorías como en un conjunto de tecnologías. Es una gama de técnicas computacionales para analizar y representar textos que se producen de forma natural en uno o más niveles de análisis lingüístico, con el propósito de lograr un estilo similar al de una persona al procesar el lenguaje en una variedad de tareas o aplicaciones. [https://surface.syr.edu/istpub/63/]

El PLN consiste en utilizar un lenguaje natural para comunicarnos con una máquina, debiendo esta entender las oraciones o frases que le sean proporcionadas. Esto facilita el desarrollo de programas que realicen tareas relacionadas con el lenguaje.

Se analiza la estructura del lenguaje a cuatro niveles

Análisis morfológico El análisis de las palabras para extraer raıces, rasgos flexivos, unidades léxicas compuestas y otros fenómenos. Se le llama análisis morfológico a un método que determina la categoría gramatical de las palabras que conforman una oración.

Este análisis se puede realizar en dos formas, mediante palabras y oraciones.

Palabras: El objetivo es separar el lexema y el morfema de la palabra, donde el lexema nos indica la raíz de la palabra y el morfema nos indica el número o género.

Oraciones: El objetivo es separar las partes de la oración e indicarnos que es cada una de las partes que conforman la oración.

Ejemplo de análisis morfológico de una oración [28].

Oración: «La cocinera Josefa cocina carne de res»

1. La = Artículo (femenino en singular).
2. Cocinera = Sustantivo (común singular femenino)
3. Josefa = Sustantivo (singular femenino)
4. Cocina = Verbo (primera persona del singular en presente)

5. Carne = Sustantivo (común masculino en singular)

6. De = Preposición

7. Res = Sustantivo (sustantivo y objeto directo).

Análisis sintáctico Es el análisis de la estructura sintáctica de la frase mediante una gramática de la lengua en cuestión, se representa por medio de un esquema la organización de enunciados y oraciones. El análisis sintáctico esta al servicio de la comprensión del enunciado ya que esta determina una parte importante del significado del mismo. Muestra la organización del enunciado y, sobre todo de la oración en sus distintos niveles [29].

El análisis se realiza mediante los siguientes pasos:

1. Delimitar, separar y clasificar el sujeto y el predicado.

2. Observar, delimitar y clasificar otros elementos, ajenos al sujeto y al predicado como el sintagma nominal que es una palabra o un grupo de palabras cuyo núcleo es un sustantivo y sintagma vocativo cuyo contenido hace referencia al interlocutor al que se pretende llamar la atención.

3. Localizar el núcleo del sujeto y el núcleo del predicado

4. Delimitar, separar y clasificar los complementos que lleva cada núcleo.

Ejemplo [30]:

Oración: ¡Laura, la puerta continúa abierta!

Laura es el vocativo, pues es a quien se llama la atención; en cambio no se debe confundir con el sujeto puesto que el sujeto de la oración en este caso es la puerta.

Análisis semántico La extracción del significado (o posibles significados) de la frase. En definitiva, comprobará que el significado de lo que se está leyendo es valido. La semántica se divide en denotación y connotación.

- Denotación: Esta es la expresion original, o formalmente aceptada de la palabra, esta es la palabra que formalmente se encuentra en diccionarios, enciclopedias y que es universalmente aceptada.
- Connotación: Forma alterna o secundaria en la que se utiliza una palabra, como la palabra burro, que en forma denotativa implica al animal de tipo equino y en la forma connotativa al hombre o persona tonta.

Ejemplo [31]:

- El burro subió y bajó el cerro. (Detonación o denotativa)
- El burro de José no entendió la semántica (Connotativa)

Análisis pragmático. El análisis de los significados más allá de los límites de la frase, por ejemplo, para determinar los antecedentes referenciales de los pronombres

Las distintas fases y problemáticas del análisis del lenguaje se afrontan principalmente con las siguientes técnicas:

- Técnicas lingüísticas formales: Se basan en el desarrollo de reglas estructurales que se aplican en las fases de análisis del lenguaje
- Técnicas probabilísticas: Se basan en el estudio en base a un conjunto de textos de referencia (corpus) de características de tipo probabilístico asociadas a las distintas fases de análisis del lenguaje

Modelos para el procesamiento del lenguaje natural:

- Lógicos (gramáticas)
- Probabilísticos (basados en corpus)

Áreas de aplicación del procesamiento del lenguaje natural [32]:

- Comprensión del lenguaje

- Recuperación de la información
- Extracción de la información
- Búsqueda de respuestas
- Generación de discurso
- Traducción automática
- Reconstrucción de discurso
- Reconocimiento del habla síntesis de voz.

Arquitectura de un sistema de PLN La arquitectura de un sistema de procesamiento de lenguaje natural se divide en niveles, estos son: fonológico, morfológico, sintáctico, semántico y pragmático.

- Nivel Fonológico: Estudio de los fonemas de una lengua, es decir, como las palabras se relacionan con los sonidos que representan. Es el encargado de dar sonido estructural a los textos líricos, logrando un tono particular, pausas uniformes, acentos etc. Se clasificación en, verso, métrica, rima, estrofa y ritmo.
- Nivel Morfológico: Como las palabras se construyen mediante unidades de significado mas pequeñas llamadas morfemas, por ejemplo. Rápida + Mente = Rápidamente.
- Nivel Sintáctico: Como las palabras pueden unirse para formar oraciones, verificando la estructura que cada palabra tiene en la oración y que sintagmas son parte de otros sintagmas.
- Nivel Semántico: Significado de las palabras y de cómo los significados se unen para dar significado a una oración.
- Nivel Pragmático: Cómo las oraciones se usan en diferentes situaciones y de cómo el uso afecta el significado de las oraciones.

2.1.4. Taxonomía

El esquema conceptual o la intensión de una taxonomía dinámica es una taxonomía simple diseñada por un experto en el dominio: una jerarquía de conceptos que va desde los conceptos más generales a los más específicos y que no requiere ninguna otra relación además de las suposiciones. Un concepto A está subsumido por un concepto B ($A \leq B$) si el conjunto de instancias clasificadas en A está intensivamente restringido para ser igual o un subconjunto del conjunto de instancias clasificadas en B: $A \subseteq B$. Modelos de subsunción de relaciones taxonómicas de tipo *IS-A*. En este caso, $A \leq B$ significa que $A \equiv B$ o que A es un descendiente de B en la taxonomía, por lo que las suposiciones definen un orden parcial entre los conceptos.

El autor de [25] explica que de forma muy esquemática, una taxonomía es una lista de elementos estructurada y ordenada de manera jerárquica que presenta una forma arbórea. Una taxonomía es la organización jerárquica del conjunto de categorías (palabras clave) bajo las que se clasifican las unidades de contenido. Por ejemplo la taxonomía «estudios» se identifican las categorías «estudios superiores», «estudios primarios», «estudios universitarios» y «otros estudios». La función de la taxonomía es la posibilidad de agrupar términos jerárquicamente relacionados. El conjunto de hipónimos e hiperónimos puede ser visto como una taxonomía.

Un ejemplo gráfico se muestra en la figura 2.1:

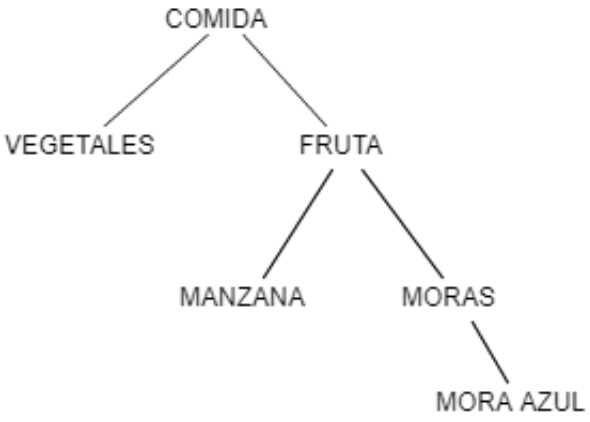


Figura 2.1: Taxonomía de comida

Hipónimo Un hipónimo es una palabra cuyo significado incluye el de otra, por ejemplo. Gorrión es un hipónimo de pájaro.

Hiperónimo Un hiperónimo es una palabra cuyo significado está incluido en el de otras, por ejemplo. Pájaro es hiperónimo de gorrión.

2.1.5. Tesauro

De igual manera [25] dice que los tesauros se pueden considerar una taxonomía con extras, ya que permiten representar la realidad mediante términos no solo organizados de forma jerárquica, sino que permiten otro tipo de relaciones entre ellos, como la relación de equivalencia y asociación. Un tesauro es un vocabulario controlado y dinámico, compuesto por términos que tienen entre ellos relaciones semánticas genéricas y que se aplica a un dominio particular del conocimiento.

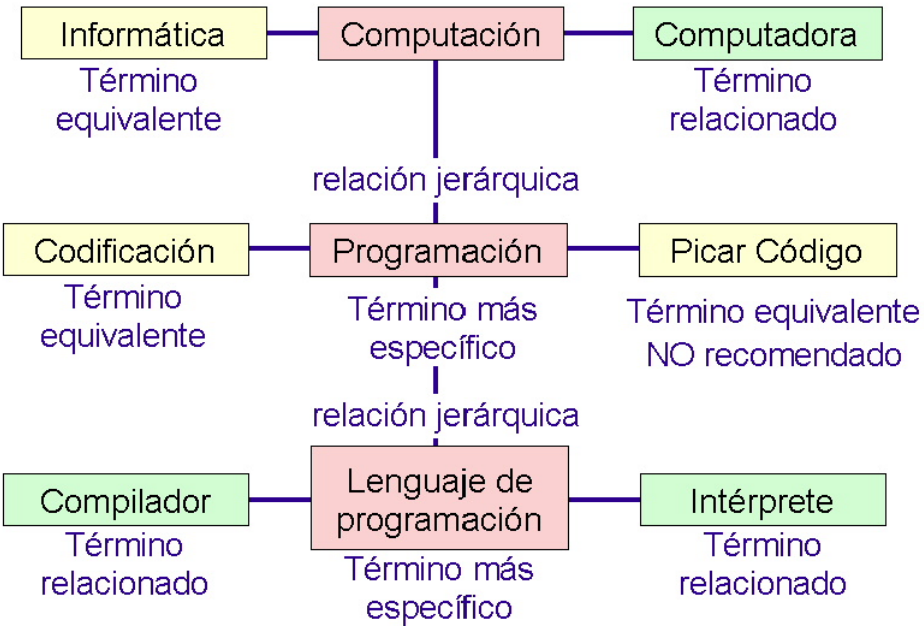


Figura 2.2: Ejemplo de tesauro [33]

2.2. Ontología

El término ontología de acuerdo a [34] viene del campo de la filosofía , y se define como la rama de la filosofía que se ocupa de la naturaleza y organización de la realidad, es decir de lo que "existe".

De acuerdo al autor de la tesis [26], la palabra ontología se deriva del griego *ontos* (estudio del ser) y *logos* (palabra). Filosóficamente una ontología es la ciencia de que es, es explicación sistemática de la existencia de los tipos de estructuras, categorías de objetos, propiedades, eventos, procesos y relaciones. Una ontología es una especificación explícita y formal de una conceptualización compartida.

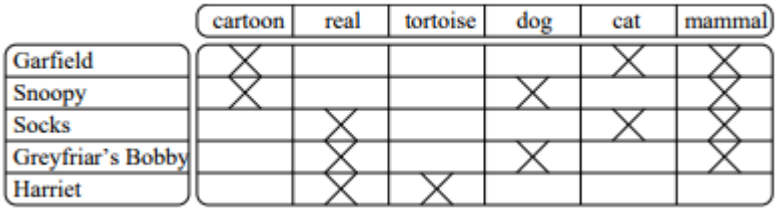
El autor de [27] define en términos prácticos el desarrollo de una ontología como:

- Definir clases en la ontología
- Colocar las clases en una jerarquía de taxonomías (Subclase-superclase)
- Definir los atributos y describir los valores permitidos para esos
- Rellenar los valores de los atributos con ejemplos.

Figura 2.3: Ontología de organizacion de una unidad educativa [Fuente: 35]

En la figura 2.3 podemos observar un ejemplo de las clases una ontología que modela la estructura de una unidad educativa. Donde podemos observar que «profesor» y «administrativo» pertenecen a clase «empleado», la cual a su vez pertenece a la clase «persona». De la misma manera vemos que «dirección», «salón», «laboratorio», «cubículo» y «biblioteca» pertenecen a la clase espacio físico de la unidad educativa. Dentro de estas clases se encuentran individuos, en el caso de la clase «profesor» cuenta con individuos los cuales heredan propiedades de las clases superiores. Por ejemplo,

Tabla 2.1: Matriz de un concepto formal de «animales famosos en inglés»



	cartoon	real	tortoise	dog	cat	mammal
Garfield	X				X	X
Snoopy	X			X		X
Socks		X			X	X
Greyfriar's Bobby		X		X		X
Harriet		X	X			

la clase persona tiene una propiedad que se llama «nombre» y la clase empleado tiene una propiedad que se llama «id de empleado», por lo cual al crear un individuo en la clase profesor, este tendrá de propiedades un nombre y un id de empleado.

2.4 Análisis Formal de Conceptos

El análisis formal de conceptos (FCA por sus siglas en ingles) formaliza la extensión e intensión de un concepto y sus relaciones mutuas. Basada en la teoría de la retícula, permite derivar una jerarquía de conceptos desde un conjunto de datos [36]. Es un método para análisis de datos, representación de conocimiento y manejo de información que es ampliamente desconocido entre las personas que se dedican a la recuperación de información en Estados Unidos a pesar de que esta tecnología tiene un potencial significativo para diferentes aplicaciones. FCA fue inventado por Rudolf Willie en los años 80s (Willie, 1982).

En una definición formal FCA es una tripleta K = (G,M,I) donde G es un conjunto de elementos llamado objetos, M es un conjunto de elementos llamados atributos e I es una relación binaria entre G y M.

Por ejemplo [37]:

Consideremos el conjunto de objetos {*Garfield, Snoopy, Socks, Greyfriar's Bobby, Harriet*} sobre los cuales se observaron las propiedades {*cartoon, real, tortoise, dog, cat, mammal*}, con lo cual obtenemos la relación dada por la siguiente figura.

La tabla 2.1 muestra de manera grafica la relación entre cada uno de los objetos con los atributos en la cual podemos ver que *Garfield* y *Snoopy* son un *cartoon*, *Socks*,

Greyfriar's Bobby y *Harriet* son real, *Harriet* es una *tortoise*, Snoopy y *Greyfriar's Bobby* son un *dog*, *Garfield* y *Socks* son un *cat* y *Garfield*, *Snoopy*, *Socks* y *Greyfriar's Bobby* son un *mammal*.

Los conceptos de un contexto pueden ser parcialmente ordenados de forma natural en un diagrama llamado retículo: un concepto C1 es "menor" que otro C2 cuando los objetos de C1 lo son también de C2. Una retícula de conceptos consiste en un conjunto de conceptos de un contexto formal y su relación de subconcepto-superconcepto entre dichos conceptos, en el ejemplo anterior se puede visualizar la reacción entre los conceptos del contexto mediante la siguiente figura.

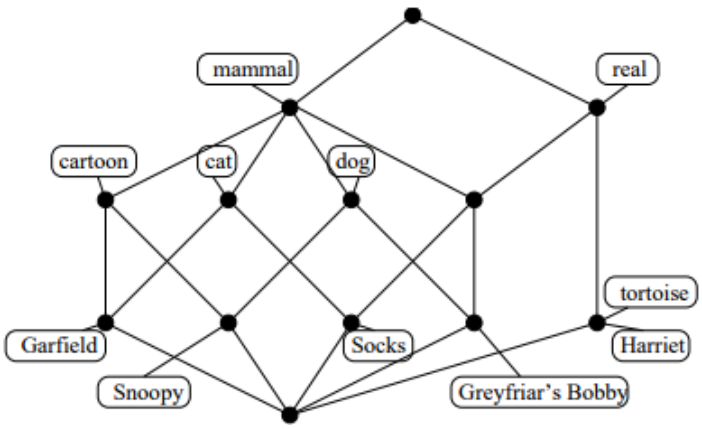


Figura 2.4: Una retícula de conceptos para el contexto formal del ejemplo anterior.

La relación subconcepto-superconcepto es transitiva, lo que significa que un concepto es un subconcepto de cualquier otro concepto que pueda ser alcanzado siguiendo el camino hacia arriba de la retícula. Si un concepto formal tiene un atributo dicho atributo es heredado por todos sus subconceptos, es decir la relación que se denota en la retícula es una relación de jerarquía, previamente descrita en el párrafo anterior.

2.3. Trabajos realizados por otros participantes de la tarea

A continuación, se presenta una breve descripción de los trabajos que los equipos que participaron en la Tarea #13 del SemEval-2016 presentaron.

En esta tarea participaron seis instituciones, la información disponible al momento de la escritura de esta tesis es reducida, pero los sistemas de manera general son los siguientes:

1. JUNLP: El sistema está basado en dos módulos de detección de hiperónimos, el primero trata con las relaciones semánticas que pueden ser encontradas para un término, en vez de analizar los 50GB de Wikipedia para una extracción mediante patrones, optaron por la extracción de relaciones de hiponimia de Babel Net (una red semántica que conecta conceptos y entidades nombradas con una gran red de relaciones semánticas). El segundo módulo intenta identificar subtérminos presentes en la lista de términos que puede ser un posible hiperónimo para ese término.

2. TAXI (Taxonomy Induction): Este método se basa en dos fuentes de evidencia, coincidencias de subcadenas y los patrones de Hearst. Analizan todo Wikipedia en busca de los patrones de Hearst y extraen esas relaciones, también hacen uso de diferentes corpus como GigaWord, ukWac y CommonCrawl.

3. NUIG-UNLP: El sistema implementa un método semi-supervisado que encuentra candidatos de hiperonimia para sustantivos representandolos como vectores de distribución. Este método asume que los hiperónimos pueden ser inducidos agregando un vector de compensación al hipónimo correspondiente generado por GloVe. El vector es obtenido como el promedio de la compensación entre 200 pares de hipónimo / hiperónimos en el mismo espacio del vector.

4. USAAR: Frecuentemente, hipónimo multi-palabra son construcciones que con-

tienen otra palabra que funciona de la misma manera que una parte de la misma palabra. Por ejemplo: “Apple pie” es esencialmente un “pie”. Este sistema exploró el número de términos que son de la misma manera (multi-palabra) en inglés.

5. QASSIT: Método semi-supervisado para la adquisición de taxonomías léxicas basadas en algoritmos genéticos. Está basado en la teoría de pre topología (Generalización de un concepto de espacio topológico) que ofrece un poderoso modelo formal de relaciones semánticas y transforma una lista de términos en un espacio de términos estructurados en combinación con diferentes criterios de discriminación. En particular, raras, pero precisas piezas de conocimiento son usadas para parametrizar los diferentes criterios definiendo el espacio de término pre topológico. Un algoritmo estructural es usado para transformar el espacio pre topológico en una taxonomía léxica.

Capítulo 3

Estado del Arte

En este capítulo se abordará el estudio e investigación de lo que se ha realizado previamente en este tema de tesis, así como también el estudio de algunas técnicas que pueden servir para su desarrollo.

3.1. Introducción

En los últimos años, ha surgido la necesidad de procesar y/o clasificar información de manera automática debido al crecimiento acelerado de la información disponible en Internet, empresas, organizaciones y repositorios en general. Este tipo de procesamiento requiere que la información sea representada de tal manera que sea entendible por las computadoras para que dicho procesamiento se pueda realizar de manera automática.

La importancia de las relaciones semánticas aplicadas a este trabajo radica en la necesidad de clasificar información de manera automática, es decir, necesitamos saber por ejemplo si una cierta información en un texto habla del mismo tema que otra parte de información en un texto totalmente diferente.

En el presente trabajo se aborda un estudio acerca de los trabajos realizados por diferentes investigadores en el área de Procesamiento de Lenguaje Natural enfocado

al análisis y extracción de relaciones taxonómicas en corpus de dominio y se hace una propuesta de solución basada en las investigaciones expresadas en el estado del arte. Tomando en cuenta métodos utilizados por la comunidad global en procesamiento de lenguaje natural como por ejemplo los métodos basados en diccionarios que parten de la idea de que una relación semántica de hiperonimia se puede encontrar en la primera frase del significado de una palabra en específico. Métodos basados en agrupamiento que afirman que textos o palabras similares comparten contextos similares por lo cual la relación entre ellos está en el contexto que tenga el texto o palabra. Y métodos basados en patrones que son reglas ya definidas que se tienen que cumplir para encontrar ciertas relaciones.

A continuación, se presentan algunas definiciones teóricas y algunos trabajos relacionados con diferentes enfoques de extracción de relaciones semánticas.

3.2. Relaciones taxonómicas

El procesamiento de lenguaje natural (PLN) consiste en la utilización de un lenguaje natural para comunicarnos con la computadora, un lenguaje natural es aquel que ha evolucionado con el tiempo para fines de comunicación humana, como el español. Una computadora debe entender las oraciones que le sean proporcionadas, el uso de estos lenguajes naturales facilita el desarrollo de programas que realicen tareas que sean basadas en el lenguaje o bien desarrollar modelos que ayuden a comprender los mecanismos humanos relacionados con el lenguaje [4].

El PLN cuenta con una arquitectura de diferentes niveles como son el nivel de integración del discurso, nivel pragmático, semántico, sintáctico, morfológico, y fonológico. Nos enfocaremos en el nivel semántico el cual estudia la codificación del significado dentro de las expresiones lingüísticas [5].

Dentro del nivel semántico encontramos relaciones conocidas como "es-un", en inglés "is-a", la cual es una relación entre clases donde una clase A es subclase de otra

clase B, dentro de esta relación encontramos otra, la relación de hipónimo e hiperónimo las cuales son relaciones entre tipos de clases definiendo una relación taxonómica donde mediante una relación de herencia un hipónimo tiene una relación de "tipo-de" (es-un) con su hiperónimo. Por ejemplo, el hiperónimo de "lunes y martes" es "días de la semana" es decir, lunes es un tipo de día de la semana y martes también es un tipo de día de la semana [6].

La computación de relaciones semánticas tiene una aplicación directa y relevante en algunas tareas de procesamiento de lenguaje natural como desambiguación de sentido de palabras, detección de sinónimos o corrección automática de palabras [7].

Las ontologías pueden ser vistas como un grafo directo en las cuales los conceptos están interrelacionados mediante relaciones taxonómicas de tipo (is-a) "es un". Haciendo un mapeo de los conceptos ontológicos por el significado de sus etiquetas textuales se puede encontrar un método para calcular su similitud el cual consiste en calcular el "camino más corto" conectando sus nodos ontológicos correspondientes mediante relaciones de tipo "es un", diciendo así que el camino más largo significa que los términos están semánticamente menos relacionados [7].

3.3. Extracción automática de relaciones semánticas

En esta sección se abordan diferentes conceptos, tesis, algoritmos y trabajos enfocados a la extracción automática de relaciones semánticas dentro de textos y/o corpus de dominio, haciendo énfasis en las relaciones semánticas llamadas Taxonomías enfocadas a la hiponimia (inclusión semántica de un término en otro), las cuales son usadas ampliamente para organizar conocimiento de manera ontológica usando relaciones de generalización y especialización a través de una herencia simple o múltiple que puede ser aplicada para la clasificación de información. Se espera que un sistema o método para la obtención automática de relaciones semánticas cumpla con las siguientes características:

- Rendimiento: Debe de generar relaciones de alta precisión.
- Supervisión Mínima: Debe de requerir mínima interacción humana o ninguna.
- Generalidad: Debe ser aplicable a diferentes tipos de relaciones.

De acuerdo a Ortega Mendoza [8], los trabajos que tratan la relación de hiponimia usan diversas técnicas para realizar la tarea. Entre los enfoques más comunes se encuentran: los métodos basados en diccionarios, métodos basados en agrupamiento y métodos basados en patrones.

3.3.1. Métodos basados en diccionarios

Estos métodos asumen que los diccionarios ya están en un formato legible por una máquina y que contienen conocimiento explícito de manera estructurada. Parten de la idea de que el hiperónimo de una palabra puede aparecer en la primera frase nominal de la definición de la misma. Por ejemplo: Primavera: "La estación entre invierno y verano en la cual aparecen flores". Ahí podemos extraer "estación" como hiperónimo de "primavera" por estar en la primera frase nominal de la definición.

Este tipo de métodos son muy precisos, pero presentan ciertas desventajas. Los diccionarios no contemplan términos específicos de un dominio como los corpus, casi siempre son términos muy generales y de diferentes dominios.

En el trabajo de [9] se usan datos disponibles en DBPedia.org para construir un conjunto de definiciones de términos en inglés. Para cada concepto que se obtiene del artículo que está siendo analizado en ese momento, un par (c, d) es construido donde C es el título exacto de un artículo en Wikipedia y d la definición del artículo. La extracción automática de relaciones semánticas basadas en diccionarios tiene su principio en que palabras similares tienen definiciones similares. El método propuesto usa una medida de similitud que toma como entrada un conjunto de conceptos y da como resultado relaciones entre ellos. Por ejemplo, el conjunto de términos (cocodrilo, animal, construcción, casa) daría como resultado (cocodrilo, animal), (construcción,

casa), etc. Lo que quiere decir que el cocodrilo es un animal y una casa es una construcción, un concepto abarca al otro.

3.3.2. Métodos basados en agrupamiento

Esta técnica es la que suele dar mejores resultados, dado que para construir las relaciones se requieren ciertos datos de entrada que se pueden recoger en una práctica de campo y así tener características o clasificaciones más específicas, lo que nos brinda la posibilidad detectar de mejor manera cuando hay una relación.

Los métodos basados en agrupamiento toman como base la hipótesis de Harris citada por Cimiano (2006), la cual indica que las palabras similares comparten contextos similares. Gracias a este enfoque las palabras se caracterizan por su contexto y se agrupan de acuerdo con la similitud entre contextos.

Los autores de [10] han experimentado con un corpus basado en métodos para construir relaciones semánticas semiautomáticas. Su sistema usa un corpus de texto y un conjunto de palabras "semilla" para cada categoría y así tener la posibilidad de identificar otras palabras que también pertenezcan a esa categoría. El algoritmo usa estadísticas simples para generar una lista ordenada por ranking de posibles palabras para cada categoría.

De acuerdo a [11] diferentes métodos han sido propuestos en la literatura para atacar el problema de obtener la derivación jerárquica de un texto de manera semiautomática o automática y estas pueden ser agrupadas en dos clases, los algoritmos basados en similitud y los conjuntos teóricos. El primer tipo de método se caracteriza por el uso de una medida de similitud o distancia con el fin de calcular la similitud por parejas entre los vectores correspondientes a los términos, con el fin de decidir si son semánticamente similares y por lo tanto ser agrupados o no. Más a fondo estos métodos pueden ser categorizados en métodos de aglomeración (bottom-up) y de división (top-down) que son estrategias de procesamiento de información.

El trabajo de [12] aborda la co-ocurrencia de términos en un corpus para la extrac-

ción de relaciones semánticas. Esta técnica asume que el hiperonimo de un término será encontrado en los términos que ocurran más veces cerca del término en el corpus. Su estrategia es la siguiente:

- Se toman términos como semilla de la taxonomía a ser construida. Un término semilla es un término que sirve como punto de construcción de una taxonomía, puede ser cualquier término, que tenga el mayor peso para el dominio de la taxonomía o que se repita lo suficiente en los documentos a analizar.

- Se analizan las relaciones léxicas de los términos inspeccionando cuales son los términos con una mayor co-ocurrencia con los términos semilla. Se analiza el primer orden de co-ocurrencia, que busca la coocurrencia de términos dada una ventana de contexto.

- Después se busca el segundo grado de coocurrencia que se refiere a buscar la relación de entre un término A con C cuando A coocurre con el término B y B también co-ocurre con C.

- En la última etapa los términos son ordenados en una taxonomía. De acuerdo a [13] la adquisición automática de relaciones de hiponimia es un problema básico en la obtención de conocimiento desde texto y es comúnmente usado en la construcción y verificación de ontologías y bases de conocimiento. Dados los conceptos C_1 y C_2, si la extensión de C_2 incluye la extensión de C_1, entonces se puede pensar que C_1 es el hipónimo de C_2. Una manera simple de comprobar que esto sea correcto es juzgar si la sentencia "C_1 es un tipo de C_2" o "C_1 es parte de C_2" es correcto.

El método que [13] propone es la extracción de relaciones semánticas basadas en agrupamiento de jerarquías. El primer problema que jerarquía de agrupamiento necesita resolver es cómo usar un vector para representar una palabra concepto. El modelo *WordSpace* es un modelo de espacio multidimensional que es construido con un

vector, en el, cada vector representa una palabra concepto. Una palabra puede ser representada contando las palabras de co-ocurrencia en el corpus, estas co-ocurrencias pueden construir un vector. Una palabra que informe sobre el contexto, tiene información abundante de la semántica de la palabra, cada palabra siempre tiene semántica diferente en diferente contexto.

3.3.3. Métodos basados en patrones

A lo largo de los años se ha visto un considerable trabajo relacionado a la extracción de información basada en patrones. Hearts (1992) fue la pionera en el uso léxico-sintáctico de patrones para la extracción automática de relaciones semánticas. Ella encontraba relaciones de hipónimo basadas en un pequeño conjunto de patrones previamente definidos como "*X, Y and/or other Z*" y también patrones como "*Z such as X and/or such as Y*" [14].

En [15] se identifica un método para reconocer patrones léxico-sintácticos. Esto implica la búsqueda de términos específicos que están conectado mediante alguna relación semántica y derivando posibles patrones de los resultados en un corpus. Si estos patrones devuelven de manera correcta relaciones entonces estos pueden ser aplicados independientemente del dominio en el que se quiera aplicar para identificar y extraer definiciones. Los patrones léxico-sintácticos pueden modelar diferentes relaciones, pero la relación de hiponimia ha dado los mejores resultados desde 1992.

Estos métodos se apoyan en la idea de que existen frases, convenciones o estilos de palabras que las personas repiten al momento de relacionar un homónimo con su hiperónimo dentro de un texto. Estos patrones si ya se encuentran registrados pueden permitirnos extraer instancias de la relación de hiponimia al aplicarse a un corpus.

Las primeras pruebas bajo patrones que se realizaron fueron construidas manualmente, es decir, después de observar la forma en la que los conceptos se describen y relacionan en un texto, un experto de dominio identificaba y formaba un conjunto de patrones sintácticos para crear una pareja hipónimo-hiperónimo.

Estos métodos se apoyan en la idea de que existen frases, convenciones o estilos de palabras que las personas repiten al momento de relacionar un hipónimo con si hiperónimo dentro de un texto. Estos patrones si ya se encuentran registrados pueden permitirnos extraer instancias de la relación de hiponimia al aplicarse a un corpus.

Las primeras pruebas bajo patrones que se realizaron fueron construidas manualmente, es decir, después de observar la forma en la que los conceptos se describen y relacionan en un texto, un experto de dominio identificaba y formaba un conjunto de patrones sintácticos para crear una pareja hipónimo-hiperónimo.

En el trabajo realizado por Patrick Panel y Marco Pennacchiotti [16] se comenta que debido al reciente crecimiento de atención en problemas de enriquecimiento de conocimiento como responder preguntas de manera automática se ha motivado a los investigadores en procesamiento de lenguaje natural a desarrollar algoritmos para automáticamente buscar recursos semánticos. Con casi un sin fin de información textual a nuestra disposición, tenemos una grandiosa oportunidad para crecer de manera automática recursos ontológicos y bancos de datos. Su método es el siguiente:

Inducción de patrones

En la fase de inducción de patrones, su algoritmo infiere un conjunto de patrones P que conecta a todas las instancias posibles dado un corpus. Cualquier patrón de aprendizaje funciona para esta etapa, se elige el mejor algoritmo y para cada instancia de entrada primero se obtienen todas las sentencias que contengan dos términos "X" y "Y", estas sentencias son generalizadas en un nuevo conjunto de sentencias reemplazando todas las expresiones terminológicas por una etiqueta terminológica. La generalización de términos es útil para pequeños conjuntos de documentos.

Clasificación y selección de patrones

Un patrón confiable es aquel que es preciso y que puede extraer un número mayor de instancias posibles.

Extracción de instancias

En esta fase, se extraen las instancias "I" que coincidan con el patrón "P", a continuación, se filtran las instancias incorrectas de acuerdo a un algoritmo propiedad de Patric & Marco [16].

En un conjunto de archivos o datos pequeños el número de instancias extraídas puede ser demasiado pequeño como para garantizar suficiente evidencia o entrenamiento para que en la siguiente iteración el algoritmo descubra de manera correcta instancias.

Cuentan con dos métodos para obtener instancias nuevas, vía web y vía sintáctica.

- Expansión Web: Nuevas instancias son extraídas de la web, usando el motor de búsqueda de Google, el sistema crea un conjunto de peticiones usando un patrón P instanciado con un concepto Y, por ejemplo, "*Italy, Country*" y el patrón "Y such as X", entonces la búsqueda en Google será, "*country such* *", las instancias entonces son creadas de acuerdo al resultado de la búsqueda.

- Expansión Sintáctica: Nuevas instancias son creadas extrayendo expresiones correspondiendo a los términos más importantes del texto.

Otro método por patrones propuesto por [8] aborda el problema de la extracción automática de parejas hipónimo-hiperónimo a partir de textos no estructurados tomados de la web. Su idea es formar un catálogo de hipónimos relacionado a un vocabulario predefinido y su método se basa en el uso de patrones. El método propuesto en su trabajo de investigación trata con patrones expresados en un nivel exclusivamente léxico su construcción es simple y no se necesita un fuerte conocimiento del idioma, no depende de analizadores sintácticos. Su trabajo considera como una pareja hipónimo-hiperónimo confiable y valida si es extraída en varias iteraciones o por varios patrones, y la confiabilidad de un patrón será mayor de acuerdo al número de parejas correctas que este recupere.

Descripción del método:

- Etapa 1: Descubrimiento de patrones mediante semillas.
- Etapa 2: Aplicar los patrones encontrados en la etapa uno y extraer tuplas de hipónimo-hiperónimo de una colección de documentos.
- Etapa 3: Ordenamiento de las tuplas obtenidas en la etapa dos, con el objetivo de ubicar las tuplas con mayor probabilidad de ser correctas en las primeras posiciones del catálogo.

Otro trabajo que se enfoca en una metodología por patrones propuesto en [2] dice que las taxonomías semánticas y tesauros como *WordNet* son el recurso clave de conocimiento para aplicaciones de procesamiento de lenguaje natural, dando información estructurada acerca de las relaciones semánticas entre palabras. En este trabajo se hace un enfoque de la construcción automáticas de clasificaciones para las relaciones de hipónimo/hiperónimo. Una palabra "X" es un hipónimo de una palabra "Y" si "X" es un subtipo de "Y", en ese sentido, "perro" es un hipónimo de "canino", "mesa" es un hipónimo de "muebles". El trabajo propuesto en [2] usa el método de patrones, pero no el método con patrones descritos anteriormente. La idea es extraer ejemplos de todos los pares de hiperonimia almacenados en *WordNet*. Para cada uno de esos pares encontrar sentencias en un corpus en el que aparezcan las dos. Procesar las sentencias encontradas y automáticamente extraer patrones los cuales indicarían un buen método para encontrar hiperonimia.

De acuerdo a [17] el acceso a nuevas tecnologías en la web ha permitido la creación de conjuntos muy grandes de información textual, esta información se puede encontrar en diferentes formatos como noticias, emails, blogs, tweets etc. lo que significativamente representa una fuente de experiencias colectivas. Para guardar, consultar o inferir conocimiento de esta información es necesario que dicha información sea procesable por una máquina. Una ontología es apta para este tipo de tareas, pero como la construcción de ontologías de manera automática es una tarea muy difícil hoy en día el área que se investiga más es aprendizaje ontológico. La idea principal del aprendizaje

ontológico parte de la búsqueda de relaciones semánticas o taxonómicas. El aprendizaje ontológico trata de descubrir conceptos y buscar la forma de que esos conceptos puedan ser agrupados, relacionados o sub divididos de acuerdo a su semántica. Se usan herramientas de Procesamiento de Lenguaje Natural para procesar los documentos o textos de los que se quiera obtener información, enfocándose en la extracción de conceptos y relaciones taxonómicas. La mayoría de las técnicas que se utilizan para extraer relaciones no toman en cuenta que algunas palabras son ambiguas y comparten un contexto semántico similar, en este caso PANTEL (2003) creó un algoritmo llamado "Agrupamiento por Comités" (CBC: Clustering By Committee) que puede asignar palabras a diferentes grupos usando conjuntos de elementos representativos llamados comités para descubrir centroides no ambiguos para describir a los miembros de una posible clase. Este método sólo crea grupos de términos, pero no crea la estructura jerárquica. De acuerdo a Gruber (1993) en un sentido semántico la identificación de relaciones de hiperonimia e hiponimia entre términos es obligatoria para la construcción de conceptos jerárquicos. Un hipónimo puede ser definido como una palabra de más significado específico que un término general aplicado al mismo. En contraste, un hiperónimo es una palabra que constituye una categoría bajo términos más específicos. Dado que la web se ha convertido en una fuente de conocimiento colectivo es una opción demasiado atractiva para encontrar hiperónimos.

El trabajo propuesto en [17] se divide en dos tareas subsecuentes. La primera es la extracción de conceptos y la segunda es la extracción de relaciones taxonómicas, estos subprocesos son aplicados al texto para extraer los conceptos y sus relaciones taxonómicas utilizando *Wordnet*. Para la extracción de conceptos [17] usa la estructura de términos en tripletas, sujeto, verbo y objeto. Haciendo uso de un etiquetador de dependencias sintácticas, "Stanford Parser" y eliminando ciertas palabras como pronombres personales y "stopwords", etiquetan las palabras de la sentencia como, sujeto, verbo y objeto y se quedan con las relaciones de dependencia de verbos y sustantivos se convierten en su lista de conceptos.

Para la extracción de relaciones taxonómicas, dado que el sistema está hecho para consultas web, primero se construye una "query" (consulta) y se obtienen diferentes páginas donde los términos que se están analizando aparezcan.

Una vez que se obtienen las páginas donde se encuentran los términos se procesa la información dejando sólo sentencias ignorando "stopwords" y se etiqueta cada palabra de cada una de las sentencias, cada una de las palabras que sea etiquetada como sustantivo es un candidato a ser un hiperónimo.

Para la obtención de hiperónimos se hace una nueva búsqueda en web donde se utilizan patrones de Hearst y Snow et al.

- A, and other B
- A, or other B
- A is a B
- B, such as A
- B, including A
- B, especially A
- B, particulary A
- B, for example A
- B, among which A.

Y la nueva consulta quedaría por ejemplo "<sustantivo>, and other <candidato a hiperónimo>" y así para cada uno de los patrones léxicos, el resultado obtenido es un número de resultados, el candidato a hiperónimo que se toma como resultado es el que devuelva el mayor número de resultados en la búsqueda sobre el total de valores buscados.

Para la extracción automática de patrones [21] usa tres tipos de recursos, corpus, ontologías y analizadores. Esta metodología es sólo para la extracción de patrones que se pueden ocupar para encontrar relaciones semánticas y no para encontrar relaciones semánticas directamente.

En [21] utilizan el corpus general "British National Corpus" (BNC) y el corpus específico "Harrison's Book, Principles of Internal Medicine" puesto que ellos se enfocan en el ámbito médico. Se utilizan las ontologías de "Princenton Wordnet" y UMLS (Unified Medical Language System).

Haciendo uso de un script en Perl se extraen del corpus las sentencias que contengan palabras que están directa o indirectamente en una relación de hiponimia con los diccionarios. Usar solamente relaciones de hiponimia directas puede llevar a un resultado no satisfactorio, es necesario ver las relaciones que hay en la ontología para encontrar relaciones no directas mediante la jerarquía ya realizada en la ontología.

El algoritmo en el script en Perl en [21] realiza lo siguiente:

- Para cada sentencia del corpus se buscan los sustantivos y verbos.
- Para cada sustantivo y verbo se busca si hay un hiperónimo en la misma sentencia. El hiperónimo es extraído desde WordNet.
- Se extraen las sentencias que tengan el hiperónimo en la misma sentencia que se está analizando.
- Se agrupan las sentencias extraídas de acuerdo a la similitud léxica del contexto entre hipónimo e hiperónimo.
- Se toma como patrón todo lo que se encuentra en medio del hipónimo e hiperónimo.

La salida del algoritmo en [21] es una lista de patrones léxicos.

3.4. Método de extracción de términos por medio de estadística

Estos métodos estadísticos son aplicados para adquirir la relevancia de un término para un dominio específico. Un método estadístico popular es la frecuencia de un término. El método de extracción de términos abordado en [18] usa la frecuencia de un término en los documentos de corpus de dominio y el número inverso de corpus en donde el término aparece. Entre más alta sea la frecuencia del término en comparación con la frecuencia de documentos en los que aparece, mayor relevante es el término.

Una vez que se obtienen los términos más relevantes utilizando el método anterior, se procede a la formalizan de esos conceptos agrupándolos con sus atributos. Para derivar atributos de un corpus específico se filtra y extraen las dependencias verbo/objeto y verbo/sujeto.

Para cada sustantivo que aparece en la frase que se está analizando el verbo se utiliza como atributo para construir el contexto de la frase. Identificando atributos similares de múltiples objetos conlleva a que las relaciones entre ellos puedan ser definidas [19].

3.5. Método de extracción de términos por medio de análisis formal de conceptos

De acuerdo a [20] las ontologías son una especificación explícita de la conceptualización de un dominio, desarrolladas con el propósito de compartir y re-usar conocimiento.

Diferentes metodologías se han propuesto para la construcción de ontologías como aquellas que se basan en el Análisis Formal de Conceptos (FCA por sus siglas en inglés). FCA es una metodología matemática para la extracción de jerarquías conceptuales de un conjunto de individuos. Estos individuos y sus propiedades son obtenidos

de corpus de texto usando herramientas de PLN.

El Análisis Formal de Conceptos (AFC), introducido en 1982 por Rudolf Wille [37], es una tecnica de aprendizaje capaz de extraer estructuras conceptuales de un conjunto de datos. Esta basada en la idea filosófica de que un "concepto" consta de dos partes: su extensión, formada por todos los objetos que pertenecen a dicho concepto; y su intención, que comprende todos los atributos compartidos por dichos objetos.

En [38] se explica que el análisis formal de conceptos es un metodo de analisis de datos con una popularidad en crecimiento en varios dominios. FCA analiza relaciones entre datos y un conjunto particular de objetos junto con un conjunto particular de atributos. FCA produce una salida de dos tipos. El primero es una latice de conceptos. Una latice de conceptos es una colección de conceptos formales en los datos los cuales son ordenados jerárquicamente por una relación de subconcepto y superconcepto. Los conceptos formales son agrupamientos particulares que representan conceptos de tipo "entendimiento humano" como "organismos viviendo en el agua", "coche con sistema de conducción", "numero divisible entre 3 y 4". La segunda salida de FCA es una colección de las llamadas implicaciones de atributos. Una implicación de atributos describe una dependencia particular la cual es validada en los datos como "cualquier número es divisible entre 3 y 4 si es divisible entre 6", "cualquier persona con una edad mayor a 60 esta retirada" etc.

Se obtienen dos beneficios principales en la aplicación de FCA para la obtención de relaciones:

1. La caracterización formal de una jerarquía de conceptos que se basa en FCA, provee una base para la especificación formal de la ontología derivada.

2. Diferentes operaciones eficientes han sido diseñadas en FCA para mantener la jerarquía de conceptos en la evaluación de los datos.

Se creó una extensión de FCA llamada "Análisis de Conceptos Relacionales" (RCA por sus siglas en inglés) para tratar con descripciones de individuos más complejas.

RCA es usado para derivar jerárquicas conceptuales donde conceptos formados reflejan aspectos comunes en los enlaces de objetos. RCA muestra vínculos entre los individuos al rango de las relaciones entre conceptos cuyo significado es similar a los roles en las ontologías. RCA produce una salida de conceptos organizados en una relación de orden parcial, lo cual se traduce en componentes para una ontología.

De acuerdo a [19] las taxonomías o jerarquías de conceptos son cruciales para cualquier sistema basado en conocimiento. Las jerarquías de concepto son importantes porque permiten estructurar la información en categorías para su búsqueda y re utilización.

El método utilizado en [19] para la extracción de taxonomías automáticas es el siguiente:

En una primera instancia el corpus es etiquetado mediante etiquetas "part-of-speech" utilizando TreeTagger, lo que devuelve cada una de las palabras del corpus con una etiqueta de sustantivo, verbo, adjetivo, adverbio etc. Del resultado generado del etiquetado se obtienen las tuplas frase verbo/sujeto, verbo/objeto y verbo/preposición. El verbo es lematizado, es decir, se obtiene su forma base, por ejemplo "jugando" su lema es "jugar". Se asignan pesos a las tuplas mediante una medida estadística y sólo los pares que alcancen un cierto límite de esa medida son transformados a un contexto formal en el cual se aplica un "Análisis Formal de Concepto" (FCA: Formal Concept Analysis). FCA es un método usado primordialmente en el análisis de datos para derivar relaciones implícitas entre objetos descritos mediante un conjunto de atributos. Los datos están estructurados en unidades que son abstracciones formales de conceptos del pensamiento humano, permitiendo la interpretación comprensible de los mismos. A pesar de que FCA puede ser visto como una técnica de agrupamiento o clustering permite también descripciones intencionales para conceptos abstractos o unidades de datos que produce. Una tripleta (G,M,I) es llamada contexto formal si G y M son conjuntos y si el resultado de que (I es un subconjunto del producto de G x M) es una relación binaria entre G y M. Un par (A, B) es un

concepto formal de (G, M, I) si y sólo sí, los atributos que tiene A son idénticos con los que tiene B, igual de manera inversa. Por ejemplo, en el contexto del dominio de la palabra turismo, uno sabe que, hotel, apartamento, carro, bicicleta o viaje pueden ser objetos que pueden ser "reservados" para su uso. Podemos "manejar" un coche, pero solo podemos "montar" una bicicleta. Podemos "unirnos" a una excursión o un viaje. Teniendo este tipo de atributos, la lattice producida por FCA sería la siguiente:

- Hotel puede ser: Reservable
- Un apartamento puede ser: Reservable y Rentable.
- Un carro puede ser: Reservable, Rentable y Manejable.
- Una bicicleta puede ser: Reservable, Rentable, Manejable y Montable.
- Una excursión y un viaje son cosas a las que uno se puede Unir y Reservar.

Esta lattice generada puede ser interpretada como una jerarquía de conceptos removiendo el elemento vacío, introduciendo conceptos ontológicos para cada concepto formal e introduciendo un subconcepto para cada elemento en la extensión del concepto formal en cuestión.

3.6. Tabla comparativa de los métodos y herramientas utilizadas por los autores

En la Tabla 3.1 se presenta una tabla comparativa de la metodología utilizada por los autores que se mencionan en el estado del arte y de la propuesta de solución del proyecto. Que metodología se usa, el corpus utilizado para la extracción de relaciones semánticas y los recursos léxicos aplicados.

Tabla 3.1: Tabla comparativa de trabajos descritos

Autores	Metodología Utilizada	Corpus	Recursos Léxicos
Panchenko A. et. Al. [9]	Basada en Diccionarios	DBPedia, Wikipedia	Wordnet
Riloff E. et. Al. [10]	Basada en Agrupamiento	No especificado.	No especificados.
Cimiano P. et. Al. (2004) [11]	Basada en Agrupamiento	No especificado.	No especificados.
Nazar R. et. Al. [12]	Basada en Agrupamiento	No especificado.	No especificados.
Mirkin S. et. Al. [14]	Basada en Patrones	No especificado.	Patrones de Hearst
Klaussner C. et. Al. [15]	Basada en Patrones	No especificado.	Patrones de Hearst y Automáticos
María R et. Al. [8]	Basada en Patrones	Texto no estructurado en web.	WordNet
Rios A. et. Al. [17]	Basada en Patrones	No especificado.	Patrones de Hearst y Snow. Stanford Parser.
Cimiano P. et. Al. (2015) [19]	Basada en Patrones	No especificado.	Patrones de Hearst y Automáticos. Uso de Treetagger.
Pantel P. et. Al. (2006) [16]	Basada en Patrones	Google	Patrones de Hearst
Pantel P. (2003)	Basada en Agrupamiento	No Especificado.	No especificados.
Meijer K. et. Al. [18]	Basada en Agrupamiento	No Especificado.	No especificados.
Snow R. et. Al. [2]	Basada en Patrones	No Especificado.	WordNet
Propuesta	Basada en Agrupamiento y Patrones	Wikipedia	Wordnet. Treetagger

Capítulo 4

Propuestas de solución

Como se mencionó previamente en la introducción se plantea crear un sistema que permita la extracción automática de relaciones taxonómicas en corpus de dominio mediante dos acercamientos. El primero esta basado en patrones los cuales son generados por un sistema de extracción semiautomática que posteriormente se aplicará a las listas propuestas por el SemEval para así extraer de manera automática el hipónimo / hiperónimo de los textos. El segundo acercamiento es análisis formal de conceptos, que tomará ciertas características o atributos a cumplir con cierto conjunto de conceptos para así inferir que hay una relación de hiperonimia.

4.1. Metodología

Para el desarrollo del sistema la metodología en particular a seguir es la siguiente:

- Fase 1 - Pre-procesamiento y sistema de recuperación de información: Desarrollo de un sistema de recuperación de información que permita el procesamiento del corpus Wikipedia en inglés, en esta fase también se considera el pre-procesamiento del corpus, quitar signos de puntuación y convertir todas las palabras a minúsculas.

- Fase 2 - Modelos propuestos para la extracción de relaciones semánticas de tipo

hipónimo/hiperónimo: Desarrollo e implementación de modelos que permitan la extracción de relaciones semánticas de tipo hipónimo / hiperónimo en corpus de dominio.

- Fase 3 - Evaluación de resultados: Prueba y evaluación de los resultados de las relaciones semánticas extraídas para la Tarea #13 mediante la metrica de evaluación exactitud.

A continuación, se detallan las tres fases anteriores.

4.1.1. Fase 1 – Pre-procesamiento y sistema de recuperación de información

En general, dado que el sistema estará enfocado a la solución de la segunda subtarea de la Tarea #13 del SemEval-2016 "*Taxonomy Extracion Evaluation*", identificación de hiperonimia, se hará uso del corpus Wikipedia. Dicha tarea provee de dos listas una del domino de palabras relacionadas con plantas y otra del dominio de palabras relacionado con vehículos de elementos a los cuales es necesario encontrarles sus relaciones de tipo hipónimo / hiperónimo [1]

Para el primer objetivo específico es necesario procesar Wikipedia a manera de que pueda ser entendible por una máquina por lo cual se hará uso de "Wikipedia Extractor" [2] la cual es una herramienta que genera una salida en texto plano de toda la base de datos almacenada en Wikipedia.org. En este caso cada elemento almacenado en Wikipedia como, por ejemplo: El término "Benemérita Universidad Autónoma de Puebla" en Wikipedia tiene como URL: https://en.wikipedia.org/wiki/Benem %C3 %A9rita_Universid

%C3 %B3noma_de_Puebla su contenido se toma como un documento el cual es almacenado en formato XML con la siguiente sintaxis ejemplo.

[1] SemEval 2016. Task 13: Taxonomy Extraction Evaluation. http://alt.qcri.org/semeval2016/task13/

[2] Giuseppe Attardi. Wikipedia Extractor. https://github.com/attardi/wikiextractor

<doc id="10273182" url="https://en.wikipedia.org/wiki?curid=10273182"
title="Benemérita Universidad Autónoma de Puebla">
Benemérita Universidad Autónoma de Puebla
The Benemérita Universidad Autónoma de Puebla (BUAP) (Meritorious Autonomous University of Puebla) is the oldest and largest university in Puebla, Mexico. Founded on 15 April 1587 as Colegio del Espíritu Santo, the school was sponsored by Society of Jesus during most of the Spanish colonial era before turning into a public college in 1825 and eventually into a public university in 1937. The religious origins can be seen in many of BUAP's buildings in Puebla city centre, which were once colonial-era churches. </doc>

Dado que la base de datos de Wikipedia, al momento de la escritura de esta tesis, tiene un tamaño de 53 GB es necesario limitar dicho corpus por medio del sistema de recuperación de información. Se optó por limitarlo a sólo los documentos en los que aparecen los términos que la Tarea #13 que el SemEval-2016 proporcionó y así obtener dos corpus de dominio específico basados en Wikipedia uno que contenga la terminología de vehículos y otro de plantas. [3]

Los pasos a seguir por el sistema de pre-procesamiento y de recuperación de información son los siguientes:

- Se convertirán todas las palabras del corpus a minúsculas.
- Se removerán signos de puntuación y números en todos los documentos.
- Se tomarán en cuenta para el corpus reducido sólo los documentos en los que aparezca la terminología de plantas o de vehículos.
- El sistema devolverá el corpus reducido en un formato como el siguiente:
    - Sustantivo|Documento Relacionado 1#Documento Relacionado 2

[3]SemEval 2016. Task 13: Taxonomy Extraction Evaluation Data. alt.qcri.org/semeval2016/task13/index.php?id=data-and-tools

En la Figura 4.1, como podemos apreciar, en el sistema de pre-procesamiento entra la base de datos de Wikipedia descargada desde Wikipedia.org, esta base de datos pasa por el sistema de procesamiento que removerá signos de puntuación y convertirá todas las palabras a minúsculas y les asignará un identificador. Esta salida entra al sistema de reducción de corpus junto con la lista de términos que SemEval-2016 proporcionó, dejando como resultado una salida que sólo contiene documentos donde aparezcan los términos que SemEval-2016 proporcionó, en otras palabras, el corpus estará limitado a un dominio específico.

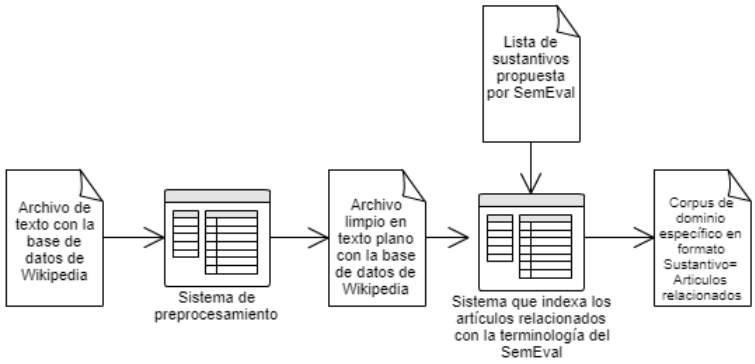


Figura 4.1: Fase 1 - Pre-procesamiento y recuperación de la información

4.1.2. Fase 2 – Modelos propuestos para la extracción de relaciones semánticas de tipo hipónimo/hiperónimo

Las propuestas de solución al problema de la extracción automática de relaciones semánticas de tipo hipónimo / hiperónimo son las siguientes:

Propuesta de extracción basada en patrones Se propone un modelo para la extracción de este tipo de relaciones mediante patrones, dichos patrones también serán extraídos automáticamente y la metodología sería la siguiente:

Del corpus de dominio, identificar todos los sustantivos contenidos en cada uno de los documentos, al tener una lista de todos los sustantivos del corpus solicitar a una base de conocimiento léxico como *WordNet* el hiperónimo de cada sustantivo,

con la tupla obtenida de tipo: <sustantivo extraído>:<hiperónimo identificado por WordNet > se propone hacer una limitación del corpus de dominio a los documentos en los que aparezca el sustantivo extraído y el hiperónimo identificado a una distancia máxima de "K" palabras. Con expresiones regulares se propone la extracción de todo lo que se encuentre entre el sustantivo extraído y el hiperónimo identificado recordando siempre que lo que este en medio de ambos tiene que ser de tamaño máximo K y esto se convertiría en un candidato a patrón. Para reducir el margen de error de patrones que puedan no ser siempre correctos, se tomaran solo aquellos candidatos a patrones que tengan mayor frecuencia en otros documentos. Un ejemplo se desarrolla en el siguiente párrafo.

En la lista de sustantivos extraídos de todo el corpus obtenemos la palabra *lion* (león). Solicitamos a *WordNet* su hiperónimo y nos dice que es *animal* por lo que ahora se busca en el corpus donde aparezcan las palabras *lion* y *animal*. Obtenemos la sentencia "*Lion is an animal*", extraemos "*is-an*" y se convierte en un candidato a patrón. Si el candidato a patrón "*is-an*" se repite en varias relaciones es tomado como un patrón para la extracción de relaciones semánticas y más adelante en este trabajo será enviado a los expertos para su validación.

Posterior a la validación de los patrones candidatos por los expertos, un sistema toma dichos patrones y se utilizan para extraer relaciones de hiperonimia finales. Es decir, si un experto valido el patrón "*is a*", este patrón se une a cada uno los sustantivos de entrada de SemEval de modo que si uno de los sustantivos es "*car*" al unirlos se genera una expresión regular "*car is a* (.*)" y como *is-a* es un patrón que de acuerdo a nuestros expertos si genera una relación de hiponimia la palabra que este siguiente a ese patrón es el hiperónimo del sustantivo de entrada, generando así un archivo final en formato <sustantivo de entrada>:<hiperónimo descubierto>.

En la Figura 4.2 se observa el proceso realizado en la fase 2 de manera gráfica. Dado que en la Fase 1 generamos un corpus de dominio específico este será la entrada al sistema de extracción de relaciones, en conjunto con los términos que el SemEval-

2016 proporcionó y a los cuales es requerido encontrarles su hiperónimo mediante WordNet, la salida este dicho sistema es una lista de términos con su respectivo hiperónimo devuelto por WordNet. La segunda parte de este sistema es la extracción de patrones candidatos para ser enviado a validación por expertos. Y por último la tercera parte es el sistema que genera expresiones regulares con la entrada del SemEval y los patrones validados para descubrir relaciones de hiperonimia finales.

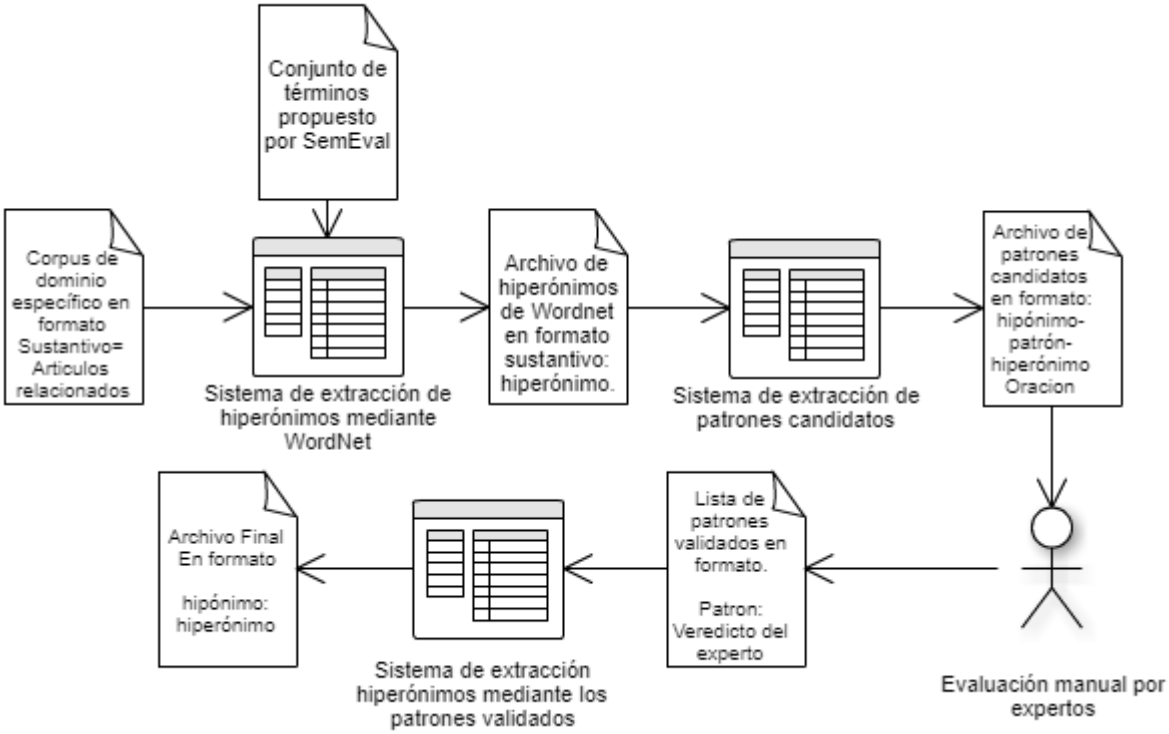


Figura 4.2: Fase 2 - Modelo propuesto para la extracción de relaciones semánticas de tipo hipónimo/hiperónimo mediante patrones.

Propuesta de solución basada en agrupamiento mediante FCA El análisis formal de conceptos proporciona una metodología para derivar una jerarquía de conceptos (como una taxonomía) a partir de una colección de objetos y las propiedades que verifican. Cada concepto de la jerarquía obtenida representa, simultáneamente, un conjunto de objetos que comparten los mismos valores para cierto conjunto de propiedades o atributos [3].

Para el desarrollo de este proyecto se propone una metodología que permite la construcción de la jerarquía de conceptos en la cual, la colección de objetos que se define en el párrafo anterior serán los datos que proporcionó el SemEval-2106, es decir, los términos del dominio de plantas y vehículos. Como se mencionó anteriormente, cada concepto debe cumplir con un conjunto de propiedades o atributos y con esta información podremos concluir que conceptos (objetos) comparten los mismos conjuntos de atributos o propiedades y, manifestando algún tipo de relación semántica. Los atributos (propiedades) que se toman en este trabajo serán sustantivos.

En la Figura 4.3 se observa el proceso realizado en la fase 2 mediante el agrupamiento de términos y atributos utilizando el método Análisis Formal de Conceptos de manera gráfica. De manera similar a la figura 4.2 dado que en la Fase 1 generamos un corpus de dominio específico este será la entrada al sistema de identificación de relaciones, en conjunto con los términos que el SemEval-2016 proporcionó y a los cuales es requerido encontrarles su hiperónimo, la salida de dicho sistema es una lista de términos con su respectivo hiperónimo.

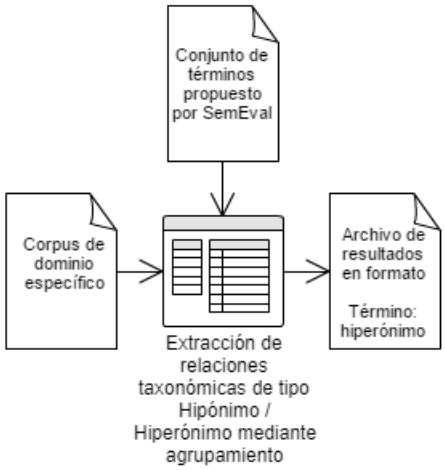


Figura 4.3: Fase 2 - Modelos propuestos para la extracción de relaciones semánticas de tipo hipónimo/hiperónimo mediante el análisis formal de conceptos.

4.1.3. Fase 3 – Evaluación de resultados

Una vez obtenidos los resultados de la aplicación de los modelos propuestos para la extracción automática de hipónimo / hiperónimos, es necesario realizar la evaluación de los mismos. La medida de evaluación que se utiliza para este trabajo es exactitud es decir el porcentaje de cuantos elementos clasificó de manera correcta el modelo realizado por el usuario entre el número total de relaciones resultas por SemEval-2016.

En la Figura 4.4 podemos observar el proceso de la tercera fase, en la cual los resultados que se obtengan de buscar las relaciones taxonómicas de la fase anterior serán evaluados. Este sistema evaluará las coincidencias con los resultados esperados y emitirá un resultado numérico en porcentaje de exactitud.

Figura 4.4: Fase 3 - Evaluación de resultados

4.2. Algoritmos de programación de las propuestas

En esta sección se abordan los diferentes algoritmos utilizados para la extracción e identificación de relaciones taxonómicas de hiperonimia.

4.2.1. Reducción de corpus general a corpus de dominio

Esta sección perteneciente a la fase 1 de este trabajo, se reduce el corpus general, es decir, Wikipedia se transforma en un corpus reducido, el cual solo aborda un dominio específico. Plantas y vehículos respectivamente.

El código [ver Apéndice 1, 6.1] recibe como entrada el archivo de la base de datos de Wikipedia, así como también las respectivas listas propuestas por el SemEval para el dominio de plantas y vehículos, su salida es un archivo por cada dominio específico.

4.2.2. Extracción de sustantivos de tipo hipónimo / hiperónimo mediante WordNet

Para la creación de la primera propuesta mediante patrones de la fase 2, se propone generar una lista de sustantivos en formato "hipónimo <tabulador> hiperónimo" la cual servirá para extraer patrones candidatos mediante la ejecución del programa que se detalla en la sección 4.2.3.

En el código [ver Apéndice 1, 6.2] primero se importan herramientas necesarias para el funcionamiento de este módulo del sistema, las cuales son *WordNet* e *Itertools*. *Wordnet* nos permitirá encontrar el hipónimo / hiperónimo de cualquier sustantivo que nosotros le ingresemos y dado que pueden existir uno o más de ellos, la herramienta *itertools* como su nombre lo describe nos permitirá iterar entre ellos. Una vez que iteramos entre ellos, generamos una lista en formato "hipónimo <tabulador> hiperónimo" como se muestra en el siguiente ejemplo.

Si el sistema recibe el sustantivo "*water scooter*", *Wordnet* devolverá los hiperónimos *motorboat* y *powerboat* por lo cual nuestra lista resultante será:

- *water scooter* <tabulador> *motorboat*
- *water scooter* <tabulador> *powerboat*

La lista mencionada se almacena en un archivo de texto para posteriormente ser utilizada en la sección 4.2.3

4.2.3. Extracción automática de patrones

Para la segunda etapa de la fase 2 en la propuesta mediante patrones se creó el siguiente programa en *Python* [ver Apéndice 1, 6.3].

El código recibe dos archivos de entrada, el corpus completo a utilizar para esta tesis, en este caso el corpus de dominio específico, así como también la lista de sustantivos generada en la sección 4.2.2 y su funcionamiento es el siguiente:

- Leer el corpus línea por línea, cada una de las líneas es un artículo de *Wikipedia*.
- Leer linea por linea la lista de palabras obtenida en la sección 4.2.2
- Verificar si en la línea que se está procesando del corpus existen las dos palabras de la lista anterior, en caso de que existan se verifíca la distancia a la que se encuentran una de otra.
- Si la distancia entre el hipónimo / hiperónimo es menor a la ventana "K" (descrita en el paso 2 de la extracción semiautomática de patrones) entonces se toma lo que se encuentre entre ambas palabras como patrón candidato y se agrega a una lista de términos en el lenguaje *Python*, si es la primera vez que este texto extraído aparece entonces se le asigna el valor de uno, sí es la segunda o más veces, se suma uno al valor anterior cada vez que se vuelva a encontrar.
- Al final se obtiene una lista de frecuencia de patrones candidatos que es guardada en un archivo para su posterior validación por expertos.

4.2.4. Extracción de oraciones completas para validación por expertos

Para la validacion de los patrones por parte de los expertos, es necesario extraer las oraciones completas donde se encuentren las relaciones de hiperonimia y así el experto pueda entender el contexto del patrón y términos [ver Apéndice 1, 6.4].

El código correspondiente recibe como entrada el corpus reducido del dominio que se este procesando, así como también la lista de patrones que se obtuvo con los códigos anteriores, su función es generar un archivo de texto que sea entendible para

el experto que validará la información proporcionada. El código imprime un resultado con el formato:

<hipónimo> <patrón> <hiperónimo> <oración completa donde se encuentran hipónimo, patron e hiperónimo>

Tomando un ejemplo del dominio de plantas un resultado seria el siguiente:

tea **which is a** *beverage* In the late 1990s there was a television commercial of vita lemon *tea* **which is a** *beverage* shown on the local television channels in hong kong

Podemos obser que en la segunda línea "*tea*" es una "*beverage*", en español, podemos decir que el té, es una bebida. En la tercera linea podemos obtener la oración completa donde se encontró dicha relación para que el experto entienda su contexto.

4.2.5. Creación de matriz de propiedades para FCA

Para este acercamiento como se mencionó previamente en la descripción de la fase 2 se utilizan ciertas propiedades para relacionar conceptos, en este momento utilizamos sustantivos.

Para hacer uso del análisis formal de conceptos con el lenguaje Python utilizamos la librería llamada *Concepts* [4] la cual se describe como una implementación simple de FCA para Python.

Esta librería requiere como entrada una matriz en formato CSV la cual tenga como columnas las propiedades y como filas los objetos a evaluar, para generar la matriz con los datos que tenemos.

El código correspondiente [ver Apéndice 1, 6.5] toma como entrada la lista de sustantivos del SemEval del dominio a procesar en ese momento asi como también él corpus reducido del mismo dominio, de manera vertical en la matriz introduce la lista del semeval y va recorriendo el corpus buscando dichos sustantivos en él. Posteriormente la oracion de estos es etiquetada mediante TreeTagger y nos quedamos sólo

[4] *Concepts: Formal Concept Analysis with Python. https://pypi.python.org/pypi/concepts*

con los sustantivos para esta primera ejecución de la propuesta y estos son ingresados como columnas, y se marca con una X la relación que tiene fila y columna.

4.2.6. Algoritmo para la generación de la lattice de FCA

Este algoritmo recibe la matriz creada en la sección 4.2.5 para poder generar una *lattice* de manera gráfica mediante la libreria *Concepts* de Python (ver código) [Apéndice 1, 6.6].

En el siguiente capítulo se muestran los resultados obtenidos con la implementación de estos algoritmos.

Capítulo 5

Resultados

En este capítulo se presentan los resultados obtenidos con los dos enfoques propuestos para la extracción semiautomática de relaciones taxonómicas de tipo hipónimo / hiperónimo.

5.1. Sistema de recuperación de información

Se creó un sistema de recuperación de información el cual redujo el corpus general Wikipedia en dos subcorpus de dominio, plantas y vehículos respectivamente. Este sistema indexa de manera automática las palabras específicas de cada dominio añadiendo todas sus palabras relacionadas en un archivo nuevo de salida para cada uno de los dominios. Este sistema recibe como entrada los sustantivos propuestos por SemEval y como resultado genera dos subcorpus con artículos de Wikipedia donde aparecen dichos sustantivos. Un ejemplo de su funcionamiento es el siguiente:

senna=due to this lack of large game the taino people became very skilled fishermen one technique was to hook a remora also known as a suckerfish to a line secured to a canoe and wait for the fish to attach itself to a larger fish or even a sea turtle once this happened men would jump into the water and bring in their assisted catch another method used by the tainos was to take shredded stems and roots of poisonous **senna** shrubs and throw them into nearby streams or rivers upon eating the bait the fish were stunned just long enough to allow the fishermen to gather them in this poison did not affect the edibility of the fish taino tribesmen mostly young boys also collected mussels and oysters in shallow waters and within the mangroves#this evergreen shrub reaches a height of about one meter it can be grown in temperate climates as it is somewhat frost hardy the soil should be loamy and peaty argentine **senna** may be propagated by cuttings planted in sand in warm and protected conditions eg a glasshouse in the northern hemisphere it flowers in july#**senna** multiglandulosa is a species of flowering plant in the legume family

En el anterior ejemplo podemos observar que para el sustantivo senna, el cual es un sustantivo de entrada del SemEval se encontraron tres articulos los cuales contienen dicho sustantivo, cada articulo se encuentra separado por el signo "#" para su posterior procesamiento por el programa extractor de patrones candidatos.

El conjunto de datos de salida se detalla en la seccion 5.2.

5.2. Conjunto de datos

El sistema de recuperación de información para Wikipedia generó dos subcorpus con las características descritas en la Tabla 5.1. Para el caso del dominio de plantas SemEval proporcionó 512 sustantivos y el sistema de recuperación de información obtuvo 5 millones de lineas de texto, de manera similiar se describe el subcorpus de vehículos.

Tabla 5.1: Características de ambos subcorpus

Subcorpus de Plantas		Subcorpus de Vehículos	
Sustantivos de entrada de SemEval	512	Sustantivos de entrada de SemEval	94
Lineas totales del subcorpus	5,000,000	Lineas totales del subcorpus	1,781,497
Palabras totales del subcorpus	131,129,942	Palabras totales del subcorpus	188,093,785

5.3. Resultados de extracción mediante patrones

Para la fase #2 de la metodología propuesta en el caso de patrones se optó por agregar a la lista propuesta por SemEval su respectivo hipónimo para así proceder a generar las expresiones regulares para extraer una lista de patrones candidatos. En la Tabla 5.2 se muestran cuatro ejemplos de hiponimos / hiperonimos agregados a las listas de los respectivos dominios propuestos por SemEval.

Tabla 5.2: Ejemplos de hipónimo / hiperónimo

Subcorpus de Plantas		Subcorpus de Vehículos	
Hipónimo	Hiperónimo	Hipónimo	Hiperónimo
senna	*shrub*	*water scooter*	*motorboat*
eelgrass	*water plant*	*coach*	*car*
guava	*edible fruit*	*rocket*	*vehicle*
eelgrass	*aquatic plant*	*chariot*	*transport*

En base a los ejemplos de la Tabla 5.2 obtenemos expresiones regulares como "senna (.*) shrub" o "coach (.*) car". Con los cuales extraemos lo que se encuentre en medio de ambos sustantivos en las oraciones y esto se vuelve un patrón candidato. Cabe mencionar que previamente se definió una ventana K=10 la cual indica el máximo de palabras que un patrón candidato puede tener. Este valor se definió mediante la observación de patrones existentes en la literatura.

Una vez obtenidas las expresiones regulares generadas previamente se ejecutó el

algoritmo 4.2.3 y se encontrarón 16,251 patrones candidatos de los cuales 6,056 se repiten dos o más veces. La Tabla 5.3 muestra algunos de los resultados obtenidos.

Tabla 5.3: Extracción de 16 resultados parciales

Candidato a patrón	Cantidad de veces encontrado
"vacio"	9829
a	369
and	288
or	233
the	173
by	152
's	100
of the	78
in a	66
for	64
is a	57
to	49
like a	37
from	25
sense of	25
type of	23

En la Tabla 5.3, el primer resultado es el "vacío", lo que quiere decir que el hipónimo fue encontrado exactamente a un lado del hiperónimo. Un ejemplo de ello se encontró en el corpus con la palabra *pickup* seguida de *truck (pickup truck),* que en español una pickup es una camioneta, lo cual quiere decir que *truck* es uno de los términos generales para *pickup* cumpliéndose la relación de hiperonimia.

Un ejemplo de la salida de patrones se describe en el siguiente párrafo. En este caso el corpus cuenta entre su texto con las siguientes dos oraciones:

1. Portrait of a child boy on a river ***boat or barge*** looking at distance.

2. An aluminum pilot house protects from the elements and is 6 ft wide by 5 ft deep. With 30" diameter pontoons a length of 30' and a width of 8.5' this work ***boat or barge*** is ideal for carrying equipment, material and workers to job sites.

El sistema que busca patrones candidatos en su lista de sustantivos en formato hipónimo / hiperónimo cuenta con *boat* y *barge* que en español es bote y barcaza, lo que quiere decir que un bote es una barcaza o que barcaza es un termino mas general que barco, el patrón *or* de acuerdo a los patrones de Hearst nos permite encontrar relaciones de hiperonimia el sistema al encontrar esas oraciones y encontrar las dos palabras (sustantivos) extrae la parte de en medio, en este caso es la palabra *or* y lo toma como patrón candidato con lo cual podemos observar el que el sistema funciona de manera correcta al encontrar patrones ya estudiados y validados por diferentes autores.

5.3.1. Lista de patrones previa validación

En la Tabla 5.4 se muestra los 14 patrónes candidatos de mayor frecuencia que se obtuvieron en cada uno de los corpus de dominio.

Tabla 5.4: Extracción de 14 resultados parciales para cada dominio

Corpus de Plantas		Corpus de Vehiculos	
Patrón	Frecuencia	Patrón	Frecuencia
to	1906	the	1434
and	1323	and	1493
or	563	a	665
at the	492	or	530
of	444	to	356
the	411	's	139
in the	361	by	119
a	281	and a	73
is a	244	was	68
is	173	on a	68
at	159	on the	64
and other	150	is a	57
like	130	of the	55
is a species of	84	and the	51

Posteriormente se realizó una intersección de los resultados obtenidos entre ambos dominios para obtener patrones generales, obteniendo un total de 1158 patrones. En

la Tabla 5.5 se muestra los patrones de mayor frecuencia intersectados en ambos dominios. Por ejemplo el patrón *and* su frecuencia de repeticion es 1295 en la ambos dominios.

Tabla 5.5: Extracción de 14 resultados parciales

Intersección de patrones		Ejemplos de Vehiculos	Ejemplos de Plantas
Patron	Frecuencia		
and	1295	*hydrogen fuel cell-powered concept car and sport utility vehicle*	*It makes a good container plant and ornamental tree.*
or	530	*A limousine, executive car or sport utility vehicle is usually selected.*	*other room was used to store liquids (oil, wine or honey) in big containers or dolia and other rooms were used to store grain or cereal in pieces of pottery*
the	411	*Two would no longer be able to lift the rocket to launch altitude.*	*Lemon basil is the only basil used much in Indonesian cuisine*
to	356	*The princess had him come into the coach to drive back*	*A report by General Robert E. Lee on August 22, 1864, stated that corn to feed the Southern soldiers was exhausted.*
is a	57	*A rocket is a pyrotechnic firework made out of a paper*	*The African yam bean is a legume that is rich in protein and starch and an important source of calcium and amino acids.*
by	40	*heavy goods vehicles, and public transport by coach and bus*	*It contains the single species Eastwoodia elegans, a flower known by the common name yellow mock aster or yellow aster.*
's	38	*Because the rocket's engine could withstand high heat*	*The plant's flowers and fruits get set in about 10 to 11 months time followed by a maturity period of about 7–8 months and then harvested in about 18 months.*
and the	35	*The Inyo, as well as the express car and the passenger car, originally served the Virginia and Truckee Railroad in Nevada.*	*Predominating plants include the Moriche Palm and the tree "Caraipa llanorum". The dominant vegetation on the non-flooded savannas is grass.*
on the	31	*It is the range of 89-93 % of mean state of charge which means as the blades on the flywheel turn, energy is being stored up between 89-93 % of the given output.*	*Some family members use also an oak leaf on the tree trunk.*

De el total de 1,158 patrones que resultarón de la intersección, 230 se repiten dos o más veces. Los cuales se enviaron a expertos en el área para su validación junto con ejemplos de las sentencias encontradas en donde aparezcan el hipónimo, patrón y el hiperónimo.

5.3.2. Lista de patrones con validación

La lista de 230 patrones fué validada de manera manual por dos expertos. En la Tabla 5.6 se presentan 20 de los primeros patrónes y su validación correspondiente por cada experto.

Tabla 5.6: Lista de 20 patrones con validación manual por expertos

Candidato a patrón	Frecuencia	Experto 1	Experto 2
and	1295	Sí	Sí
or	230	Sí	Sí
the	411	No	Sí
to	356	Sí	Sí
a	281	Sí	Sí
was	68	Sí	Sí
is a	57	Sí	Sí
of the	55	Sí	Sí
by	40	Sí	Sí
's	38	Sí	Sí
and the	35	No	Sí
in the	31	No	Sí
of a	31	No	Sí
on the	31	No	Sí
would	30	No	No
this	29	No	No
in	29	No	Sí
for	27	Sí	Sí
with	25	No	Sí
is	16	Sí	Sí

5.3.3. Resultados de la propuesta basada patrones

Al tener los patrones validados en la sección anterior se procedió a realizar la ejecución del programa con cada uno de los patrones en conjunto con cada sustantivo al cual es necesario encontrarle el hiperónimo, un ejemplo de este proceso es el siguiente.

En el caso del patron *is-a* para el sustantivo *"dump truck" s*e busca el hiperónimo utilizando expresiones regulares de manera que mediante el código 6.4 se busque el texto *"dump truck is a (.*)"* donde *(.*)* sería reemplazado por el hiperónimo de dicho sustantivo. El hiperónimo de acuerdo a los resultados de SemEval es "*truck*" por lo cual el programa debería de encontrar "*dump truck is a truck*" en el corpus reducido de vehículos. El mismo procedimiento se realiza para cada sustantivo de SemEval y cada patrón validado.

En la Tabla 5.7 se muestra una lista de sustantivos de entrada de SemEval, el hiperónimo esperado, si el sistema lo encontro de manera satisfactoria o no, así como también la oracion donde aparece.

Tabla 5.7: Lista de 11 resultados de patrones y sus respectivas extracciones para vehículos

Entrada de SemEval	Salida Esperada	¿Fue encontrado?	Oración
aircraft	craft	Sí	a trial journey undergone by a ship **aircraft or other craft**
airplane	heavier-than-air craft	Sí	a fixed wing aircraft or **airplane is a heavier than air craft** whose lift is generated by air pressure differential between the upper and lower wing surfaces
ambulance	car	Si	law enforcement health services with permanent **ambulance and car**
boat	vessel	Sí	if a **boat or other vessel** can successfully pass through a waterway
car	motor vehicle	Sí	when a **car or other motor vehicle** is used on a public road
ferry	boat	Sí	the rownham **ferry was a boat**
helicopter	heavier-than-air craft	No	N/A
truck	motor vehicle	Sí	the top gear polar special made the **truck the first motor vehicle** to make it to the magnetic north pole
tank	military vehicle	No	N/A
snowmobile	tracked vehicle	No	N/A
yacht	vessel	Si	on the other hand a us navy **vessel such as the yacht** in the example above

De igual manera en la Tabla 5.8, se muestran ejemplos del dominio de plantas.

Tabla 5.8: Lista de 11 resultados de patrones y sus respectivas extracciones para plantas

Entrada de SemEval	Salida Esperada	¿Fue encontrado?	Oración
acacia	tree	Sí	the camel thorn **tree acacia** erioloba forest is one of only two in the world
amaranth	herb	Sí	is a prostrate perennial **herb in the amaranth** family
mandarin	citrus	Si	citrus reshni also known as cleopatra **mandarin is a citrus** tree
melon	gourd	Sí	there are also many orchards and vineyards, **melon and gourd** plantations
mint	herb	Sí	collinsonia canadensis is a perennial medicinal **herb in the mint** family
orange	citrus	Sí	peter owned several florida **citrus orange** groves
passionflower	vine	No	N/A
peppermint	mint	Sí	candies that primarily consist of **peppermint and mint**, such as candy canes
sugarcane	gramineous_plant	No	N/A
teaberry	shrublet	No	N/A
tobacco	herb	Si	the australian **tobacco is a herb** to 15 metres tall growing in new south wales and victoria

5.3.4. Resultados de la evaluación de la propuesta basada en patrones

En Tabla 5.9 se muestran los resultados de la propuesta basada en patrones para cada dominio utilizando la metrica de evaluación exactitud. Se puede observar que en el caso del dominio de plantas se logra una exactitud del 53 % y en el caso del domino de vehículos aproximadamente un 65 %.

Tabla 5.9: Exactitud de la propuesta por patrones

Entrada de SemEval	Dominio de Plantas	Dominio de Vehiculos
Total de Elementos	526	94
Elementos Encontrados	280	61
Exactitud	53.23 %	64.89 %

5.4. Resultados de la propuesta basada en FCA

Como se detalla en la sección 4.1.2 las propiedades a utilizar para FCA son sustantivos, por lo cual se procedió a realizar el etiquetado de cada una de las oraciones en las que aparecen los términos de entrada de SemEval para extraer sólo los sustantivos. El procedimiento fue el siguiente:

1.- Etiquetar las oraciones en las que aparece el termino de entrada de SemEval.

2.- Marcar una relación entre el sustantivo de entrada de SemEval y sus sustantivos adyacentes.

3.- Crear la matriz de relaciones con los datos del paso 1 y 2.

Al realizar este proceso se obtiene una matriz de relaciones entre sustantivos, un extracto de esta matriz se muestra en la siguiente Figura 5.1.

	def-car	def-van	def-truck	def-vehicle	def-wagon	def-airplane
chariot	X			X	X	
bulldozer	X	X	X	X	X	
coach	X	X	X	X	X	X
bicycle				X		
pedicab		X		X		
rocket	X	X	X	X	X	X
canoe	X					
roadster	X	X	X	X	X	X
tramcar	X	X	X	X	X	
ambulance	X	X	X	X	X	
wheeled vehicle	X	X	X	X	X	X
sled	X	X	X	X	X	X
blimp	X	X		X		X
kayak	X	X	X	X		
sport utility	X	X	X	X	X	
police van	X	X	X	X	X	

Figura 5.1: Ejemplo de matriz de relaciones para vehículos

En la Figura 5.1 podemos observar una parte de la matriz generada por una ejecución del algoritmo de FCA. En la parte de filas tenemos la entrada de SemEval, en las columnas los sustantivos que se obtuvieron mediante la extraccion de sustantivos adyacentes con TreeTagger y las "X" marcan la relacion entre los sustantivos de entrada y las extracciones de TreeTagger.

La Figura 5.2 muestra una parte de la lattice generada por una ejecución del algoritmo de FCA a un acercamiento de 300x, en la cual podemos ver una relación obtenida procesando el corpus de vehículos entre *aeroplane* y *airship* con lo cual podemos intuir que existe una relación.

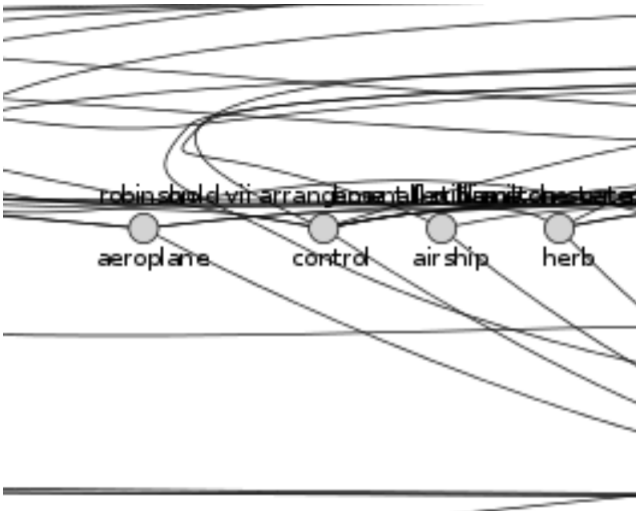


Figura 5.2: Extracto de una lattice de FCA para vehículos

La Tabla 5.1 muestra el resultado final de la ejecución del sistema para la propuesta de FCA. Como podemos observar del total de elementos para el dominio de plantas se encontraron 313 de los 512 sustantivos lo que resulta en una exactitud del 61.13 %, mientras que en el caso de vehiculos se encontraron 69 de los 94 sustantivos lo que resulta en una exactitud de 73.40 %.

Tabla 5.10: Exactitud de la propuesta por FCA

Entrada de SemEval	Dominio de Plantas	Dominio de Vehiculos
Total de Elementos	512	94
Elementos Encontrados	313	69
Exactitud	61.13 %	73.40 %

Como podemos observar es una diferencia significante entre ambos corpus, siendo el dominio de plantas el que menor porcentaje de sustantivos encontrados tuvo. La razón es que se tuvo la necesidad de reducir la cantidad de artículos de Wikipedia a ser procesados por el sistema para el dominio de plantas. El sistema en conjunto con el etiquetador para reconocer sustantivos tardaría un tiempo considerable en etiquetar aproximadamente un total de 162,000 oraciones, cada una de estas oraciones con una cantidad variable de palabras en las cuales aparecen los términos de interés para SemEval. Para el dominio de plantas el sistema utilizó 25,600 oraciones para obtener dicho resultado. De la misma manera, se obtiene una cantidad considerable de relaciones y no es posible generar una lattice completa de la salida final de análisis formal de conceptos.

Conclusiones

La web ha cambiado profundamente él como nos comunicamos, hacemos negocios y nuestro trabajo. Tenemos acceso a millones de recursos en diferentes idiomas independientemente de donde nos encontremos hoy en día.

El problema de la Web es que el contenido o recursos a los que podemos tener acceso crece más rápido de lo que podemos clasificarlo, gracias a la web semántica, los programas permiten procesar su contenido, razonar con el, combinarlo y hacer deducciones lógicas para resolver problemas cotidianos de manera automática, razón por la cual se proponen sistemas que permitan extraer, analizar y procesar información de manera automática.

Los objetivos planteados al inicio de la tesis se han cumplido en su totalidad. Se construyó un sistema de preprocesamiento y recuperación que procesa Wikipedia para los términos de nuestro interés, de igual manera se construyeron los modelos para la extracción de relaciones de hiponimia mediante patrones y análisis formal de conceptos y estos fueron evaluados con los resultados esperados por SemEval.

Como podemos observar los resultados del enfoque basado en patrones, los diferentes corpus contienen patrones similares, es decir podemos encontrar algunos patrones en la ejecución de la primera metodología tanto en el corpus de plantas como en el de vehículos, razón por la cual se optó por hacer una intersección de ambos resultados.

Podemos ver en la lista de patrones obtenidos que el acercamiento que se aborda en este trabajo contiene patrones ya validados por otros autores como los patrones de Hearst, encontrando estos patrones en las primeras posiciones de la tabla por

su alta frecuencia podemos intuir que dicho acercamiento es válido y que puede ser considerado como un complemento a los patrones de Hearst previamente descritos en el estado del arte realizando más pruebas para justificar su validez.

Los resultados para patrones en ambos corpus muestran una similitud en cuanto a su exactitud, analizando dichos resultados se llegó a la conclusión de que ambas propuestas pueden mejorar este porcentaje de sustantivos encontrados satisfactoriamente considerando frases compuestas en lugar de solo considerar palabras, es decir, hacer que el sistema reconozca frases como «automotive vehicle» o «water plant» los sistemas de ambas propuestas reconocen estas frases como dos palabras separadas no como una palabra compuesta.

En el caso de la propuesta basada en FCA la cantidad de sustantivos y de información de los mismo es tan grande que es imposible graficar una lattice y es por eso que se tuvo la necesidad de reducir aun más la información a procesar en el corpus de plantas, lo que llevó a que ambos resultados tuvieran una diferencia significativa.

Como trabajo a futuro se planea realizar la implementación para frases de dos o mas palabras previamente descrita para ambas metodologías, esperando que esto mejore el resultado. En el caso de FCA se espera evaluar los resultados del corpus de plantas sin reducción para así tener un mejor resultado en esta propuesta. De igual manera se planea buscar más características para relacionar términos en análisis formal de conceptos y así poder perfeccionar el sistema.

Referencias

1. Fellbaum, C. (2005). WordNet and wordnets. In: Brown, Keith et al. (eds.), Encyclopedia of Language and Linguistics, Second Edition, Oxford: Elsevier, 665-670.

2. Snow, R., Jurafsky, D. & Ng, A. (2005). Learning syntactic patterns for automatic hypernym discovery. Advances in Neural Information Processing Systems, 17, 1297--1304.

3. Fernando Sancho Caparrini. (2014) Análisis Formal de Conceptos. Dpto. de Ciencias de la Computación e Inteligencia Artificial Universidad de Sevilla.

4. Mg. Augusto Cortez Vásquez, Mg. Hugo Vega Huerta, Lic. Jaime Pariona Quispe. (2009) Procesamiento de Lenguaje Natural. Facultad de Ingeniería de Sistemas e Informática Universidad Nacional Mayor de San Marcos.

5. Julio Villena Román, Raquel M. Crespo García, José Jesús García Rueda. (2012) Procesamiento del Lenguaje Natural. Universidad Carlos III de Madrid.

6. Ríos Ríos, Aura Josefina; Bolívar, Constanza Ivet. (2009) Razonamiento verbal y pensamiento analógico. Universidad del Rosario. 7. Sánchez, D., Batet, M., Isern, D. & Valls, A. (2012). Ontology-based semantic similarity: A new feature-based approach. Expert Syst. Appl., 39, 7718-7728. Pages 1-2.

7. David S., Montserrat B.t, David I., Aida V. (2012): Ontology-based semantic similarity: A new feature-based approach. Departament d'Enginyeria Informà-

tica i Matemàtiques, Universitat Rovira i Virgili, Avda. Països Catalans, 26, 43007 Tarragona, Spain. Pages 1-2

8. Rosa María Ortega Mendoza. (2007) Descubrimiento Automático de Hipónimos a partir de Texto no Estructurado. Instituto Nacional de Astrofísica, Óptica y Electrónica. pages 23--27.

9. Alexander Panchenko, Sergey Adeykin, Alexey Romanov and Pavel Romanov. (2012) Extraction of Semantic Relations between Concepts with KNN Algorithms on Wikipedia. Université catholique de Louvain, Centre for Natural Language Processing, Belgium. Bauman Moscow State Technical University, Information Systems dept. (IU5), Russia pages 78—88.

10. Riloff, E. & Shepherd, J. (1997). A Corpus-Based Approach for Building Semantic Lexicons. CoRR, cmp-lg/9706013. pages 1—3.

11. Cimiano, P., Hotho, A. & Staab, S. (2004). Comparing Conceptual, Divise and Agglomerative Clustering for Learning Taxonomies from Text.. ECAI (p./pp. 435--439). pages 1-2.

12. Nazar, R., Vivaldi, J. & Wanner, L. (2012). Automatic taxonomy extraction for specialized domains using distributional semantics. Terminology, 18, 188--226.

13. Zhan, Q. & Wang, C. (2015). Hyponymy extraction of domain ontology concept based on ccrfs and hierarchy clustering. CoRR, abs/1508.01476. Pages 1-3.

14. Mirkin, S., Dagan, I. & Geffet, M. (2006). Integrating Pattern-Based and Distributional Similarity Methods for Lexical Entailment Acquisition.. In N. Calzolari, C. Cardie & P. Isabelle (eds.), ACL, : The Association for Computer Linguistics. pag 2.

15. Klaussner, C. & Zhekova, D. (2011). Lexico-Syntactic Patterns for Automatic Ontology Building.. In I. P. Temnikova, I. Nikolova & N. Konstantinova (eds.),

RANLP Student Research Workshop (p./pp. 109-114), : RANLP 2011 Organising Committee. pages 1-3.

16. Pantel, P. & Pennacchiotti, M. (2006). Espresso: Leveraging Generic Patterns for Automatically Harvesting Semantic Relations In Proceedings of Conference on Computational Linguistics / Association for Computational Linguistics(COLING / ACL-06). , pp. 113-120 . pages 1--3.

17. Rios-Alvarado, A. B., López-Arévalo, I. & Sosa, V. J. S. (2013). Learning concept hierarchies from textual resources for ontologies construction. Expert Syst. Appl., 40, 5907-5915. Pages. 1,2,5.

18. Meijer, K., Frasincar, F. & Hogenboom, F. (2014). A semantic approach for extracting domain taxonomies from text.. Decision Support Systems, 62, 78-93. pages 1—5.

19. Cimiano, P., Hotho, A. & Staab, S. (2005). Learning Concept Hierarchies from Text Corpora using Formal Concept Analysis. Journal of Artificial Intelligence Research, 24, 305-339. page 2-4

20. Bendaoud, R., Rouane Hacene, M., Toussaint, Y., Delecroix, B. & Napoli, A. (2007). Text-based ontology construction using relational concept analysis. International Workshop on Ontology Dynamics - IWOD 2007, June, Innsbruck, Autriche. Pages 1-2

21. Verginica Barbu Mititelu. (2003) Automatic Extraction of Patterns Displaying Hyponym-Hypernym Co-Occurrence from Corpora. Romanian Academy Research Institute for Artificial Intelligence.

22. Vázquez, A., Huerta, H., & Quispe, J. (2009): Procesamiento de lenguaje natural. Facultad de Ingenieria de Sistemas e Informatica, Universidad Nacional Mayor de San Marcos.

23. Python Software Foundation: What is Python? Executive Summary. https://www.python.org/doc/essays/blurb/

24. Barnett, B. (2001): AWK. The Grymoire. http://www.grymoire.com/Unix/Awk.html

25. Navarra, P (2007).: Los blogs y la organizacion del conocimiento. Editorial UOC. pages 36-38

26. López, S.(2007): Modelo de indexacion de formas en sistemas VIR basado en ontologias. Tesis para obtener el grado de Maestra en Ciencias de la Computacion, Universidad de las Américas Puebla. pages 66-67

27. Cámara, J. (2002): Learning Metadata Standards, Cap. 4: Ontologias. UPF, Barcelona.

28. Ejemplo de Análisis morfológico. Revista Ejemplode.com. Obtenido 11/2017, de http://www.ejemplode.com/12-clases_de_espanol/4349-ejemplo_de_analisis _morfologico.html

29. Analisis Sintactico. Lengua y lingüística. Obtenido el 11/2017 de http://www3.uah .es/innovaciondocentelenguaylinguistica/uah_ana_sin.htm

30. Componentes básicos: Sujeto y Predicado. Componentes extraoracionales. EdAS: Editor de Analisis Sintactico. Obtenido el 11/2017 de http://www.analisissintactico.com/blog/tag/vocativo/

31. Ejemplo de Semantica. Revista Ejemplode.com. Obtenido 11/2017, de http://www.ejemplode.com/12-clases_de_espanol/2437-ejemplo_de_semantica.html

32. Martín, F., Ruiz J. (2012-2013).: Procesamiento del Lenguaje Natural. Dpto. Ciencias de la Computaci´on e Inteligencia Artificial Universidad de Sevilla.

33. Daniel M. Arquitectura de la InformaciÓn. Obtenido 11/2017, de http://www.mordecki.com/Presentaciones/ArquitecturaInformacion/arquitecturainformacion1.htm

34. Laura, G., Elena S., Alicia S. (2001-2002).: Ontologias en Documentacion. Universidad Politecnica de Valencia

35. José B., Andrea T., Hugo L., Orlando R., Mireya T., Darnes V. (2016) : Un modelo ontológico para representar la organización de una unidad educativa. Avances recientes en Ciencias Computacionales - CiComp 2016

36. Steffen S., Rudi S., Steffen S., Rudi S. (2009). Handbook on Ontologies. International Handbooks on Information Systems. Springer-Verlag Berlin Heidelberg.

37. José A. Alonso, J. Borrego, M.J. Hidalgo, F.J. Martín y J.L. Ruiz. (2002): Una introducción al Análisis Formal de Conceptos en PVS *. Dpto. de Ciencias de la Computación e Inteligencia Artificial. Universidad de Sevilla.

38. Radim B. (2008). Introduction to formal concept analysis. Department of Computer Science, Faculty of Science. Palacký University.

Chapter 6

Apéndice 1 - Códigos de Programación

6.1. Código para la reducción de corpus

La función *findWholeWord* permite encontrar la palabra que se le ingrese como parámetro dentro de un texto y regresa verdadero o falso, la función *cleanFileTxt* limpia de caracteres especiales cualquier tipo de texto que se le ingrese y este es convertido a minúsculas, la función *process_line* hace el procesado de linea por linea del corpus Wikipedia. El algoritmo funciona de la siguiente manera:

- Recorre linea por linea Wikipedia
- Cada linea es limpiada de caracteres especiales.
- En la linea que se encuentra procesando se verifica si existe uno de los sustantivos propuestos por el SemEval del corpus que se este ejecutando en ese momento.
- Si en la linea no existe ningún sustantivo de SemEval, esa linea es descartada, caso contrario se imprime en pantalla para ser redireccionada a un archivo.
- El archivo final contendra lineas de texto en las cuales se asegura que se en-

cuentren los sustantivos del SemEval, limitandolo así al dominio de plantas y vehículos.

```
#!/usr/bin/python
# -*- coding: utf8 -*-
from multiprocessing import Pool
import sys, re, codecs
reload(sys)
sys.setdefaultencoding("utf-8")
def findWholeWord(w):
    return re.compile(r'\b({0})\b'.format(w), flags=re.IGNORECASE).search

def cleanFileTxt(texto):
    r = texto.replace('\n', '').replace('\r', '')
    r = r.decode('utf-8').lower()
    r = r.replace('á','a',)
    r = r.replace('é','e',)
    r = r.replace('í','i',)
    r = r.replace('ó','o',)
    r = r.replace('ú','u',)
    r = r.replace('ü','u',)
    r = re.sub(u'[^a-zñ0-9. ]', ' ', r.decode('utf-8'))
    r = " ".join(r.split())
    return r

def process_line(line):
    #print "Procesando Linea"+line
    TEXTO = cleanFileTxt(line)
    GUARDADO = ""
    with open('expresionesRegularesPlantas.txt') as n:
        for palabras in n:
            palabras = palabras.replace("\n","").replace("_"," ").split("\t")
            palabra1 = palabras[0]
            palabra2 = palabras[1]
            splitedText = TEXTO.split(" ")
            longitudTexto = len(splitedText)
            indices = [i for i, x in enumerate(splitedText) if x == palabra1]
            if findWholeWord(palabra1)(TEXTO):
                if TEXTO != GUARDADO:
                    print TEXTO
                    GUARDADO = TEXTO

if __name__ == "__main__":
    pool = Pool(30)
    linea = 1
    with open('text.xml') as source_file:
        results = pool.map(process_line, source_file, 30)
```

6.2. Extracción de sustantivos de tipo hipónimo / hiperónimo mediante WordNet

El siguiente código recibe como entrada los datos proporcionados por el SemEval para obtener su respectivo hiperónimo utilizando WordNet. Su dalida es <sustantivo de SemEval> <tab> <hiperónimo de WordNet>.

```
1  from nltk.corpus import wordnet as wn
2  from itertools import chain
3  palabrasCorpus = {}
4  R = open("WN_vehicles.terms","r")
5  for x in R:
6      x = x.replace("\n","")
7      x=x.split("\t")
8      palabrasCorpus[str(x[1])] = 1
9      palabraHyp = {}
10     for palabra in palabrasCorpus:
11         hiperonimo = []
12        for i,j in enumerate(wn.synsets(palabra)):
13              lista=list(chain(*[l.lemma_names() for l in j.hypernyms()]))
14              for hyp in lista:
15               if palabra in palabraHyp:
16                   palabraHyp[palabra] += ","+hyp
17               else:
18                   palabraHyp[palabra] = hyp
19
20 for x in palabraHyp:
21     print x+" "+palabraHyp[x]
```

6.3. Código del sistema para la extracción automática de patrones

El siguiente codigo recibe como entrada el corpus asi como tambien la lista de expresiones regulares previamente descrita, su salida es una lista de patrones candidatos a evaluar.

```
import sys, re, codecs
patrones = {}
ventana = 10
linea = 1
patronesencontrados = {}
with open('corpus.txt') as f:
    for line in f:
      with open('sustantivoshiponimohiperonimo.txt') as n:
      for palabras in n:
        palabras = palabras.replace("\n","")
        .replace("_"," ").split("\t")
        palabra1 = palabras[0]
        palabra2 = palabras[1]
        texto = cleanfiletxt(line)
        splitedtext = texto.split(" ")
        longitudtexto = len(splitedtext)
        indices = [i for i, x in enumerate(splitedtext)
        if x == palabra1]
        if len(indices) >= 1 and palabra2 in texto:
           for indice in indices:
            for ventananum in range(1,ventana):
            if (indice+ventananum) < longitudtexto:
                if (splitedtext[indice+ventananum])
                 == palabra2:
                   hasta = indice+ventananum
                   patron = ""
                   for pospatron in range(indice+1,hasta):
                   patron += splitedtext[pospatron]+" "
                     if patron in patronesencontrados:
                       patronesencontrados[patron] +=1
                     else:
                       patronesencontrados[patron] =1

for x in patronesencontrados:
    if  int(patronesencontrados[x]) > 1:
        print str(x)+"\t\t"+str(patronesencontrados[x])
```

6.4. Código para la extracción de oraciones completas para validación por expertos

El siguiente código recibe como entrada el corpus de dominio y la lista de expresiones regulares generada previamente, su salida es <sustantivo> <oracion donde aparece>.

```
from itertools import islice
import sys, re, codecs

reload(sys)
sys.setdefaultencoding("utf-8")
LISTATERMINOLOGIA = {}
LISTAPATRONES = []
VALORES = {}
def findWholeWord(w):
    return re.compile(r'\b({0})\b'.format(w), flags=re.IGNORECASE).search

cuantas = 0
ventana = 10
with open('corpusReducidoPlantasPuntos1Sustantivo.txt') as f:
    while True:
        next_n_lines = list(islice(f, 1000))
        if not next_n_lines:
            break
        cuantas = cuantas+1000
        with open('../expresionesRegularesPlantas.txt') as n:
            for palabras in n:
                palabras = palabras.replace("\n","").replace("_"," ")
                            .replace("."," ").split("\t")
                palabra1 = palabras[0]
                palabra2 = palabras[1]
                for palabtasTexto in next_n_lines:
                    TEXTO = palabtasTexto.replace("\n","")
                    if findWholeWord(palabra1)(TEXTO):
                        find = TEXTO.split(".")
                        for y in find:
                            if findWholeWord(palabra1)(y) and
                                findWholeWord(palabra2)(y):
                                splitedText = y.split(" ")
                                longitudTexto = len(splitedText)
                                indices = [i for i, x in enumerate(splitedText)
                                            if x == palabra1]
                                for indice in indices:
                                    for ventanaNum in range(1,ventana):
                                        if (indice+ventanaNum) < longitudTexto:
                                            if (splitedText[indice+ventanaNum])
                                                == palabra2:
                                                hasta = indice+ventanaNum
                                                patron = ""
                                                for posPatron in range(indice+1,hasta):
                                                    patron += splitedText[posPatron]+" "
                                                patron = patron.rstrip()
                                                if patron != "":
                                                    if patron in LISTAPATRONES:
                                                        if patron not in VALORES:
                                                            print "-"*10
                                                            print palabra1+" "+patron+" "+palabra
                                                            print y
```

6.5. Código para la creación de matriz de propiedades para FCA

El siguiente código recibe como entrada el corpus de dominio especifico y etiqueta cada palabra para obtener solo los sustantivos que se encuentren junto al sustantivo de entrada al cual hay que encontrar el hiperónimo, genera un documento en formato CSV la cual la libreria Concepts de Python interpreta como una matriz de relaciones.

```
from itertools import islice
import sys, re, codecs
reload(sys)
sys.setdefaultencoding("utf-8")

from treetagger import TreeTagger
tt = TreeTagger(language='english')

LISTATERMINOLOGIA = []
tamano = len(LISTATERMINOLOGIA)

ARREGLO = {}
contador = 1
with open('salidaVehiculos.txt') as line:
    for palabtasTexto in line:
        #print str(contador), palabtasTexto
        palabtasTexto = palabtasTexto.replace("\n","")
        if contador == 2:
            palabtasTexto2 = palabtasTexto.split(" ")
            hiponimo = palabtasTexto2[0]
        if contador == 3:
            if hiponimo not in ARREGLO:
                ARREGLO[hiponimo] = []
                for i in range (0,len(LISTATERMINOLOGIA)):
                    ARREGLO[hiponimo].append("P")
            etiquetas = tt.tag(palabtasTexto)
            #print etiquetas
            for et in etiquetas:
                palabraEt = et[0]
                tipoEt = et[1]
                if (tipoEt == "NN" or tipoEt == "NNS" or tipoEt == "NP"
                    or tipoEt == "NPS"):
                    if len(palabraEt)>0:
                        if palabraEt in LISTATERMINOLOGIA:
                            posicion = [i for i,x in enumerate(LISTATERMINOLOGIA)
                                        if x == palabraEt]
                            palabraArriba = LISTATERMINOLOGIA[posicion[0]]
                            if palabraArriba != hiponimo:
                                ARREGLO[hiponimo][posicion[0]] = "X"
        contador += 1
        if contador == 5:
            contador = 1

tamano = len(LISTATERMINOLOGIA) for sust in ARREGLO:
    texto = ""
    contador = 1
    for valor in ARREGLO[sust]:
        if valor == "P":
            valor = ""
        texto += valor
        if contador < tamano:
            texto += ","
        contador += 1
    print sust+","+texto
```

6.6. Código para la generación de la lattice de FCA

El siguiente código recibe la matriz generada en la seccion 6.5 y su salida es un grafo en formato PNG haciendo uso de la libreria Concepts y Graphviz.

```
1 from concepts import Context
2 import sys
3 reload(sys)
4 sys.setdefaultencoding("utf-8")
5 r = Context.fromfile('matrizVehiculos.txt', frmat='csv')
6 r.lattice.graphviz(filename='imagen.png',format='png',render=True)
```