Benemérita Universidad Autónoma de Puebla

Facultad de Ciencias de la Computación

Extracción automática de relaciones no taxonómicas en corpus de dominio

Tesis presentada para obtener el grado de:

Licenciatura en ingeniería en Ciencias de la Computación

presenta

Cabrera Moreno Irvin Yair

Asesor: Dra. Mireya Tovar Vidal
Asesor: Dr. José de Jesús Lavalle Martínez

Diciembre 2019

Dedicatoria

Este trabajo es dedicado a todas aquellas personas que han puesto su confianza en mí incluso llegando a creer más en mí de lo que a veces yo he creído.

A mi padre, que ha dedicado gran parte de su vida a trabajar para su familia, siendo un padre amoroso y bondadoso que a pesar de ser un hombre de pocas palabras sus grandes acciones han forjado mi carácter para poder lograr los objetivos que me ponga. A mi madre, gracias a su amor, paciencia, interés, disciplina y todas aquellas noches en las que se desvelaba por apoyarme con mis deberes escolares, he podido llegar hasta este momento.

Espero que todo el esfuerzo que han puesto para que sea una persona con profesión y de bien se esté reflejando en las metas que he logrado conseguir hasta ahora. A ustedes les debo todo y deseo poder contar con el tiempo para demostrarles que su esfuerzo no ha sido en vano y puedan estar orgullosos de la persona que he me convertido, decir que los amo es poco para ustedes.

A mis hermanos, son las personas con quienes he compartido la mayor parte de mi vida y sé que ustedes también confían en que pueda lograr algo bueno, los amo y esto también es gracias a ustedes. A ti Ana, que nunca has fallado como hermana mayor y has estado ahí para mí en complicidad de hermanos, sé que tienes fé en mí y espero no defraudarla. Erick, vivimos una buena infancia llena de juegos e imaginación, no pude haber tenido mejor compañero para crecer, gracias por creer en mí hermano. Los amo.

A Jazmín una de mis mejores amigas, siempre has estado cuando más he necesitado a alguien y sé que tu felicidad por mí es genuina, la confianza que has depositado en mí para lograr mis metas espero poder estarla respondiendo de buena forma.

A una de mis personas favoritas, Alejandra, mi gran amiga, agradezco enormemente las palabras de amor y ánimo que siempre me das, no hay duda que gracias a ti también he logrado esto, compartimos la carrera y salones de clases, risas, proyectos y más, no sé como regresar algo de lo que me das con palabras pero quiero que sepas qué formas parte de los logros que pueda alcanzar, gracias por la confianza que tienes en mí.

Todos ustedes se han vuelto la fuerza que me hace seguir creciendo como persona y profesionista, mi único deseo es que la vida nos permita seguir creciendo juntos, de verdad muchas gracias por todo.

- Yair

Agradecimientos

Esta investigación es apoyada por el Fondo Sectorial de Investigación para la Educación, con el proyecto CONACyT CB/257357 bajo el número de becario 28617 y por el proyecto VIEP-BUAP 100409344-VIEP2019.

Resumen

La identificación de relaciones no taxonómicas es una tarea que se realiza con la creación automática de ontologías. Además, la construcción manual de ontologías para expertos e ingenieros de conocimiento es una tarea costosa y lenta, por lo que es necesario crear algoritmos automáticos y/o semiautomáticos que agilicen el procedimiento.
Este problema nos demuestra la importancia de crear sistemas que extraigan de forma automática relaciones no taxonómicas de fuentes de texto para mejorar y agilizar la creación o evaluación de ontologías, esto sería posible con la ayuda de un enfoque de extracción automática. En esta investigación se propone un algoritmo para la extracción de relaciones no taxonómicas en una ontología de Inteligencia Artificial (IA), las cuales son evaluadas a través de una técnica de minería de datos: *reglas de asociación*, que cuenta con medidas estadísticas que determinan la probabilidad de ocurrencia entre los conceptos y el verbo conector relacionados. Los resultados experimentales indican que el 72 % de las relaciones obtenidas en el algoritmo existen en la ontología de IA.

Índice general

Capítulo 1

Introducción

El fin de este capítulo es presentar la investigación, exponiendo sus antecedentes y una breve explicación de cómo nace la idea para desarrollar nuestra solución, planteando así los objetivos que se desean alcanzar con el desarrollo de las actividades que se realizaron.

El origen de esta investigación fue el trabajo desarrollado por Tovar, Pinto, Montes, González y Vilariño [1] el cual propone una evaluación automática de relaciones en dominios de espacio restringido, haciendo uso de patrones léxicos sintácticos con el fin de evaluar las relaciones de inclusión de clase y ontológicas que estén dentro de la ontología. Su enfoque está centrado en un corpus de referencia que sirve para comprobar la validez de la relación. El enfoque es capaz de brindar una medida de exactitud para cada ontología que se ha evaluado, este valor está asociado con la calidad que existe en las relaciones dentro de la ontología.

Esta puntuación se obtiene con cierto grado de confiabilidad, mediante la comparación de los resultados dados por el enfoque contra la evaluación de expertos y una base. En su trabajo se explica las relaciones taxonómicas y ontológicas, los patrones léxico sintácticos que fueron usados para lograr su objetivo, así como también la mención de hipónimos e hiperónimos.

La creación de ontologías y su representación en un lenguaje formal comúnmente se realiza por ingenieros del conocimiento junto con expertos

del dominio. Este proceso implica la extracción, conceptualización, evaluación y formalización del conocimiento de dominio. La construcción manual de ontologías se ha identificado en gran medida como una tarea costosa, tediosa y con tendencia a errores [2]. Dificultades técnicas, como la falta de estándares para reutilizar las ontologías existentes y la ausencia de métodos de extracción automática de conocimientos, son problemas que dificultan la creación de ontologías [3].

Actualmente, la creación de ontologías a partir de textos utilizando métodos de aprendizaje automático y minería de datos se ha propuesto como un método que facilita el proceso de ingeniería ontológica. En este contexto, en [4] el aprendizaje ontológico se ha identificado como un campo que ayuda a los ingenieros del conocimiento, así como a los usuarios finales en la creación de ontologías. Puede verse como un campo multidisciplinario, con disciplinas como la ingeniería ontológica, el aprendizaje automático y el procesamiento del lenguaje natural.

El uso de estas tecnologías se distribuye en tres tareas principales, extracción de entradas léxicas, de taxonomías y relaciones no taxonómicas [4]; todos juntos permiten construir una ontología desde cero o mejorar una existente utilizando diferentes fuentes de información.

El aprendizaje ontológico a partir de textos constituye un medio prometedor para que los ingenieros ontológicos aceleren la creación de ontologías, de modo que se han propuesto enfoques para cubrir las diferentes tareas. La fase de extracción de relaciones no taxonómicas ha sido reconocida como uno de los problemas con más dificultad [4] y menos cubierta [5].

Esta fase se puede dividir en dos problemas diferentes: descubrir la existencia de una relación entre un par de conceptos y luego etiquetar esta relación de acuerdo con su significado semántico. La asignación de etiquetas a las relaciones también es difícil ya que son posibles varias relaciones entre instancias de los mismos conceptos generales [4]. Por esta razón, en esta investigación se propone un método de extracción automático de relaciones no taxonómicas en corpus de dominio especifíco abordando el tema de reglas de asociación.

El objetivo principal del proyecto fue desarrollar un enfoque que permita la extracción de relaciones no taxonómicas, haciendo uso de técnicas adecuadas en el campo de procesamiento de lenguaje natural. Cumpliendo con los objetivos:

- Analizar técnicas para la identificación de relaciones no taxonómicas en artículos científicos.
- Implementar un algoritmo capaz de encontrar las relaciones entre palabras en un dominio ontológico.
- Evaluar el algoritmo propuesto.

Con el propósito de cumplir los objetivos y crear un sistema que cumpla las expectativas, se usó un modelo clásico para el desarrollo de software, iniciando con la obtención de requerimientos, para después hacer un análisis que nos guió al diseño de nuestra solución la cuál fue implementada y se realizaron pruebas.

Después de esta introducción al trabajo, se tiene el capítulo 2 titulado Estado de arte en donde se presenta todo aquel conocimiento que motivó la inspiración para este trabajo. Este conocimiento esta repartido en diferentes reseñas de artículos que fueron leídos.

Seguido del Estado del arte se tiene el capítulo 3 Marco teórico, en este tercer capítulo se abordan todos los temas relevantes para este trabajo de tesis, con el fin de facilitar al lector la comprensión de este trabajo.

En el capítulo 4 llamado Diseño, se presentan los algoritmos que fueron usados para lograr nuestros objetivos, estos son los algoritmos de extracción de términos y extracción de relaciones no taxonómicas.

Capítulo 2

Estado del arte

En este capítulo se presenta un estudio acerca de los trabajos de otros autores, sobre sus investigaciones en tareas de extracción de conceptos y extracción de relaciones no taxonómicas, estos trabajos sirvieron como ideas para la investigación. El capítulo está divido en dos partes, la primera parte consta de aquellos enfoques que sirvieron como inspiración para la extracción de conceptos. La segunda consta de aquellos enfoques que están relacionados con la extracción de relaciones no taxonómicas su evaluación y descubrimiento, así como también el aprendizaje ontológico.

2.1. Enfoques para la extracción de conceptos

Los trabajos presentados en esta sección del capítulo son enfoques que realizan extracción de conceptos clave en documentos de texto, estos trabajos son de gran ayuda para nuestra investigación, ya que los conceptos clave contienen información importante del texto, por lo cual una buena extracción de estos conceptos es importante ya que en ellos junto con algún verbo se da la relación no taxónomica que deseamos encontrar.

De los primeros antecedentes de extracción de frases clave es presentado por Eibe en conjunto con Paynter, Witten, Gutwin y Nevill-Manning

[6] donde proponen KEA (*Keyphrase Extraction Algorithm*) un sistema de extracción de frases clave en textos basado en el algoritmo de aprendizaje supervisado Naïve Bayes.

Tratan la extracción de frases clave como una tarea de clasificación, en esta tarea se determina la probabilidad de que una frase pertenezca o no a la categoría de frase clave y usan aprendizaje supervisado para generar ese espacio de clasificación. El modelo Naïve Bayes se construye usando frases clave de un conjunto de entrenamiento donde se consideran como atributos medidas de ponderación de términos como TF-IDF (*Term Frequency Inverse Document Frequency*) y la distancia de la frase clave desde el inicio del documento hasta donde está la frase clave.

KEA sigue el siguiente procedimiento para realizar la extracción de frases clave en textos, primero crea un conjunto de frases clave candidatas partiendo el documento en oraciones y creando secuencias de a lo más tres palabras, elimina frases clave candidatas que tengan palabras vacías (*stopwords*), (Khair [7] define a las palabras vacías como palabras muy comunes dentro de un texto y que contienen poca información útil, ejemplos en inglés son: *the, of, for, etc.*) al inicio y al final o aquellas que sean nombres propios.

Luego KEA calcula la medida TF-IDF y la distancia de las frases clave candidatas y posteriormente usa el modelo de clasificación Naïve Bayes para determinar la probabilidad de que una frase clave candidata sea una frase clave del documento. Al final, ordena las probabilidades y extrae las mejores frases claves.

Barker y Cornacchia [8] crean un sencillo sistema de extracción de frases clave llamado *B&C*, se realiza una elección de frases conocidas como frases nominales (*noun phrases*). El sistema funciona en tres etapas: la primera es buscar en el documento frases nominales base, a cada una de ellas se le asigna un puntaje que depende de la frecuencia y longitud tomando en cuenta la frecuencia del sustantivo de la frase nominal (*noun phrase head*).

Seguido de esto se filtran para elegir las que cuenten con un mejor puntaje. Durante la búsqueda de frases nominales se intenta encontrar una

estructura que tenga una secuencia se sujetos y adjetivos terminando con un sujeto o rodeado por palabras que no sean sujetos o adjetivos, para lograr esto hacen uso de dos diccionarios que etiquetan cada palabra del documento procesado.

En la segunda parte, el sistema elige frases clave con base a la frecuencia de su *noun phrase head* para elegir las que tengan mayor ocurrencia dentro del documento. Una vez que el sistema produce un conjunto de k frases clave candidatas por documento, continúa con la tercera parte que consta de la eliminación de frases clave candidatas que contengan sub-frases para evitar la generalización. El conjunto restante de los pasos anteriores se selecciona como frases clave del documento.

El sistema de extracción de frases clave propuesto por Ortiz [9] parte de la idea de que las frases clave es una secuencia de dos o más palabras que capturan la idea central de un documento para crear un enfoque de extracción que toma ventaja de dos diferentes técnicas de análisis: secuencias de frecuencia máxima para extraer una o más palabras de un documento y el algoritmo de *PageRanking* que se usa para conseguir las secuencias que representan las ideas centrales, además usan un módulo de extracción de acrónimos para ayudar al sistema en la elección.

El sistema se divide en dos módulos, el primero extrae secuencias de palabras del texto, posteriormente se las pasa a *PageRanking* para determinar las secuencias más importantes. Se usa la estructura del documento para determinar relaciones entre las secuencias de palabras encontradas. Las relaciones se identifican usando un método llamado "Criterio de vecindad" que encuentra las relaciones y las ordena.

Después de ordenarlas, son seleccionadas aquellas de una determinada longitud. Posteriormente, el módulo de acrónimos regresa un máximo de tres acrónimos con sus frases multi-términos asociadas como candidatas a frases clave. Por último, se determina la longitud y cantidad de las secuencias para conseguir un conjunto final de frases clave.

Las investigación de Rose [10] realiza un sistema llamado RAKE (*Ra-*

pid Automatic Keywords Extraction), un método automático, independiente del dominio e independiente del lenguaje para extraer palabras clave de documentos individuales. RAKE se basa en la observación de que las palabras clave frecuentemente contienen varias palabras, pero raramente contienen signos de puntuación o palabras vacías (*stopwords*).

RAKE usa un conjunto de *stopwords*, un conjunto de delimitadores de frases y un conjunto de palabras delimitadoras como parámetros de entrada. Los parámetros anteriores funcionan para dividir el documento en palabras clave candidatas, que son secuencias de palabras de contenido tal como aparecen en el texto.

Las co-ocurrencias de palabras dentro de estas palabras clave son significativas y permiten identificar la co-ocurrencia de la palabra sin la aplicación de otro tipo de medición arbitraria. Se mide el grado de asociación de las palabras con el contenido del texto de tal manera que se adapta automáticamente al estilo y contenido del texto, permitiendo también una medición adaptiva y una medición de grado de la co-ocurrencia de las palabras que serán usadas para asignarle una puntuación a las palabras clave candidatas.

De forma general, el texto se parte en palabras clave candidatas usando las listas de entrada, posteriormente se le saca una medida de asociación con el contenido del texto de cada palabra que forma las palabras clave candidatas, luego cada frases clave candidata se califica usando las medidas de asociación de cada una de las palabras que la conforman y finalmente se seleccionan las palabras con mejor calificación como palabras clave del texto.

El autor Gelbukh [11] presenta un método basado en la comparación con un corpus de referencia para extraer términos simples de un dominio específico. La idea se basa en comparar las frecuencias de un término en dos corpus, si una palabra aparece mucho en un corpus de dominio es un término probable. Para localizar los candidatos se realizan pasos de pre-procesado de texto para quitar palabras vacías, puntuación, números y símbolos, a cada candidato se le cálcula su ocurrencia para conocer su frecuencia en ambos corpus, si un candidato aparece más veces en el corpus general que en el corpus de dominio es automáticamente rechazado como

término.

Del conjunto de términos restantes del paso anterior, son seleccionados aquellos con un puntaje más alto para agruparlos mediante el algoritmo de *k-Means*, posteriormente son eliminados de los grupos palabras vacías que se hayan incorporado, palabras que tengan un sentido científico muy general o verbos, solo dejando términos con gran similitud.

Nuestra investigación hace uso de técnicas para la extracción de términos relevantes en corpus de dominio de documentos de texto en inglés, nuestro objetivo es crear una lista de términos que podrían ser evaluados por expertos. Nuestro enfoque para la extracción de frases nominales está basado en lo que se conoce como patrones regulares, el cuál previo un análisis de "Parte del Discurso" en donde se agregan etiquetas a las palabras mediante a su función dentro de la oración (adjetivos, adverbios, sustantivos, etc.).

Un patrón regular es una sucesión de etiquetas, en nuestro caso un patrón que forme la estructura de una frase nominal, la cuál puede incluir adjetivos, sustantivos e incluso en el idioma inglés verbos conjugados.

2.2. Enfoques para extracción de relaciones

En esta sección se presenta una síntesis de trabajos anteriores de algunos autores que trabajaron con ontologías. En sus enfoques realizan la extracción de relaciones no taxonómicas como una de sus tareas y gracias a esto, podemos generar un punto de partida a nuestra investigación propuesta y crear nuestro propio enfoque como contribución.

Serra y Girardi en [12] proponen un proceso semiautomático para extracción de relaciones no taxonómicas de ontologías de fuentes de texto, haciendo uso de técnicas de procesamiento de lenguaje natural para identificar las relaciones no taxonómicas y técnicas de minería de datos para sugerir un nivel alto dentro de la jerarquía de la ontología en inglés.
Este proceso está dividido en tres fases:

- Extracción de las relaciones candidatas:
  Donde se usa la técnica de procesamiento de lenguaje natural, donde el texto es divido en oraciones, entonces, se realiza una búsqueda en las oraciones para encontrar aquellas que tengan al menos dos conceptos de la ontología. El objetivo es encontrar el verbo que indique la relación no taxonómica.
- Análisis de un nivel jerárquico apropiado:

  Para la sugerencia de un nivel apropiado se hace uso del algoritmo para el descubrimiento de reglas generalizadas asociadas propuesto por Srikant y Agrawal [13]; el algoritmo usa un conjunto de transacciones que contiene un conjunto de elementos y cada elemento es un conjunto de conceptos, el algoritmo computa las reglas de asociación con los cuales se obtienen los valores de medición.
- Selección manual de relaciones:
  Esta fase la realiza un experto, ya que los autores creen que no hay mejor decisión que la que toma un experto para evitar ambigüedad. Por lo que la meta en esta fase es dar al usuario las mejores sugerencias para que al final tome la decisión.

Mäedche y Staab [14] establecen un nuevo enfoque que amplía los enfoques actuales en la adquisición semiautomática de taxonomías, para el descubrimiento de relaciones conceptuales no taxonómicas de texto, para facilitar la ingeniería de estas relaciones no taxonómicas. Usan un algoritmo de reglas de asociación generalizadas que no sólo detecta relaciones entre conceptos, también, el nivel de abstracción apropiado el cual determina la relación.

Se hace uso de métodos de procesamiento de texto plano para identificar pares de palabras relacionadas, esto comprende un etiquetador que busca palabras u oraciones complejas para crear abreviaciones, también hace uso de análisis léxico para el reconocimiento de entidades o la recuperación de información de dominios específicos y el algoritmo de reglas de asociación es el propuesto por Srikant y Agrawal [13]; la salida de la asociación de reglas son pares de conceptos que son entregados al ingeniero para que las

incluya en la ontología de relaciones no taxonómicas.

Weichselbraun, Scharl, Granitzer, Neidhart y Juffinger presentan en su documento [15] un enfoque automático de sugerencias de relaciones en ontologías, ya sean taxonómicas o no taxonómicas. La investigación presentada en este documento se centra en agregar la relación descubierta a una ontología semiautomática basada en una pequeña ontología semilla y un corpus específico de dominio que contiene un gran número de documentos web no estructurados.

Su enfoque detecta relaciones taxonómicas al facilitar técnicas y bases de datos personalizadas de procesamiento de lenguaje natural. La categoría no taxonómica se basa en relaciones previamente aprendidas, suponiendo que relaciones similares entre conceptos se expresen a través de verbos similares. Al comparar la representación del espacio vectorial de los verbos que coinciden con los conceptos objetivo con los vectores verbales conocidos utilizando la métrica de similitud de coseno, se obtiene el tipo de relación de la relación desconocida.

Las relaciones no taxonómicas son descubiertas gracias a tres pasos que se realizan en su sistema: un análisis de los principales sustantivos donde son agregadas frases que normalmente forman sustantivos compuestos y de esta forma agregarlas a su base de conocimiento y formar palabras que tengan un significado más amplio al que muchas otras palabras puedan pertenecer, el segundo paso es crear sinónimos a estas frases que ya se tienen y agregarlas a lo que denominan WordNet en su sistema de ontología extendida. El último paso es un análisis de subsunción, en el cuál se asume que el documento está compuesto por oraciones de un conjunto de otros documentos y bajo las condiciones de Sanderson y Croft[16].

Porzel y Malaka [17] presentan un enfoque de evaluación cuantitativa de ontologías basado en tareas y un enfoque poblacional. Esta evaluación busca conocer que tan efectiva es una ontología dada una tarea definida. La efectividad en este sentido significa: si se va a emplear una ontología para una tarea determinada, se puede usar para desempeñarse mejor o peor de una manera medible. Por lo tanto, se debe seleccionar un conjunto de

evaluación de modo que el resultado medible relacionado con la tarea del conjunto dependa tanto como sea posible de la ontología utilizada.
Tal evaluación (o prueba) de una ontología puede juzgar en al menos tres niveles básicos:

- El alcance del vocabulario (conceptos).
- Generalización de una taxonomía del tipo '*es un*'.
- Relaciones no taxonómicas, es decir, el ajuste de las relaciones semánticas.

Dada una tarea apropiada y maximizando los algoritmos que trabajan en la ontología resolviendo las tareas y con los estándares de la evaluación, se pueden calcular las tasas de error correspondientes a la ontología. Donde la tasa de error corresponde a comparar una segmentación de salida con una oración segmentada a mano, viendo cuántas palabras difieren. La tasa de error es la distancia de edición mínima en palabras entre la salida y la oración escrita a mano. Los resultados deben mostrar lo siguiente:

- errores de inserción; indican conceptos superfluos, relaciones *es un* y semánticas
- errores de eliminación; indican conceptos faltantes, relaciones *es un* y semánticas
- errores de sustitución; indican conceptos ambiguos o fuera del objetivo, relaciones *es un* y semánticas

Al aplicar el esquema de evaluación se puede probar y medir las mejoras que son hechas por el enfoque de aprendizaje que se enfoca en los temas de la ontología.

Sánchez y Moreno [18] en su trabajo presentan un método automático que extrae conceptos de relaciones no taxonómicas y etiquetado de relaciones usando toda la web como corpus. Para el descubrimiento y extracción de las relaciones no taxonómicas en la red hacen uso de los siguientes métodos:

- Técnicas de análisis ligero, son usadas para tener una mejor escalabilidad en un entorno como la web. Lo que hacen es reducir el análisis, solo analizando texto simple, directo y no ambiguo o como es llamado texto *nugget*.
- Análisis estadístico, son aplicadas en las tareas de adquisición del conocimiento. Ya que los motores de búsqueda de la web proveen medidas confiables, si la consulta está bien hecha, en ese caso, es muy importante ya que se obtienen estadísticas robustas sobre distribución de la información.
- Patrones lingüísticos, una técnica muy efectiva para no consultar a expertos. Para descubrir las relaciones no taxonómicas, además de un conjunto de patrones definidos, se debe aprender patrones adecuados como frases verbales del dominio previamente, ya que este tipo de relaciones se dan normalmente por un verbo que relaciona dos conceptos.
- Bootstrapping, es usado para restringir las consultas hechas por el motor de búsqueda web para obtener corpus de documentos de dominio específicos.

Para el aprendizaje no taxonómico se descubren patrones que expresen estas relaciones no taxonómicas. Como el número de verbos en inglés es grande, se debe encontrar los verbos que sean apropiados para el dominio. Entonces, se hace una búsqueda en la web con la palabra clave para obtener el conjunto de recursos de ese dominio específico. Para cada uno se hace el análisis ligero haciendo uso de términos vecinos con el verbo inicial para encontrar frases de verbos y así crear candidatos.

Estos candidatos son clasificados de acuerdo a su posición en la oración y se evalúan para verificar si pertenecen al dominio. Cada frase candidata extraída es medida mediante una fórmula que corresponde a la clasificación que tiene, estas fórmulas son usadas para medir el grado de relación que tienen dos palabras, en este caso, el verbo inicial contra la frase candidata encontrada. Los valores devueltos forman un rango de lista que existe en la lista de candidatos de patrones lingüísticos dependientes del dominio y se seleccionan aquellos que están más cerca del dominio.

Cuando los patrones lingüísticos están listos, lo siguiente es descubrir conceptos que no son taxonómicos con el verbo inicial. Para eso se hace una nueva búsqueda en la web con " patrón lingüístico verbo inicial " o viceversa, el resultado será un conjunto de recursos web con la búsqueda especifica y el objetivo ahora es evaluar su contenido. Después se buscan conceptos no taxonómicos relacionados con el verbo inicial etiquetando su relación con el patrón lingüístico. Se miden estos candidatos y se seleccionan los más cercanos al dominio específico.

Kavalec, Mäedche y Svátek presentan en su trabajo [19] una combinación entre análisis de texto plano, minería de datos y modelado de conocimiento. Además, para la extracción de relaciones no taxonómicas usan una técnica que se basa en el método de *Text-to-Onto*, el cual produce, basado en los documentos del corpus, un conjunto de relaciones de pares entre conceptos, estas relaciones son etiquetadas por un experto y pasan a ser parte de la ontología.

El método usado para el descubrimiento de las relaciones en el texto del corpus está basado en las reglas de asociación, donde dos o más léxicos pertenecen a una transacción si estos se encuentran juntos en un documento o en un texto definido, las transacciones más frecuentes son salidas como asociaciones entre sus objetos.

Se plantea entonces que la predicción de una relación no taxonómica se caracteriza por verbos que se encuentran frecuentemente en la vecindad de pares de entradas léxicas que corresponden a asociaciones de conceptos. Los candidatos para etiquetar una relación no taxonómica entre dos conceptos son aquellos verbos que el número de transacciones que se mantienen entre un verbo v, concepto c_1 y concepto c_2, si c_1 y c_2 aparecen dentro de n palabras a partir de una aparición de v, con alguna razonable n.

Mäedche y Staab describen en su trabajo [20] un enfoque de minería de datos para relaciones conceptuales no taxonómicas de corpus creado de técnicas de procesamiento de texto plano, este enfoque está basado en el algoritmo de reglas de asociación para encontrar las relaciones y también para definir un nivel de abstracción en estas relaciones. El algoritmo de

reglas de asociación generalizadas está basado en el algoritmo propuesto por Srikant y Agrawal [13], el cuál es muy conocido en minería de datos ya que puede encontrar las relaciones que ocurren en un par de elementos.

La regla de asociación básica consiste en un conjunto de transacciones, donde cada transacción tiene un conjunto de elementos y cada uno de los elementos forma parte de un conjunto de conceptos. Srikant and Agrawal extendieron su algoritmo para determinar las asociaciones a un buen nivel de la taxonomía, pero para descubrir las relaciones conceptuales, los autores construyen un esquema de aprendizaje donde modifican el conjunto de transacciones y las métricas para generar las reglas de asociación.

Por otra parte, Lutz en su artículo [21] presenta un enfoque que reconoce la importancia de las relaciones no taxonómicas. Este enfoque usa operaciones definidas en el domino de la ontología como un formato para descripciones semánticas de entradas y salidas de servicios web. El trabajo parte de un servicio web llamado Susan el cuál calcula la distancia entre dos plantas industriales. Susan usa un tipo complejo de salida que consta de coordenadas de longitud y latitud y este tipo de salida es un requerimiento para realizar su búsqueda y calcular la distancia entre dos puntos.

Es así como en su trabajo se describe un problema con una herramienta llamada *Semantic Services Matchmaker (SSM)*, la cual provee emparejamientos entre descripciones semánticas de servicios y solicitudes de usuarios. SSM está basada en los algoritmos de emparejamiento de LARKS y OWL-S los cuales constan de cinco filtros y uno de ellos es un filtro de entrada y salida. El filtro de entrada y salida verifica que los parámetros de entrada y salida de un servicio estén bien acoplados y sus parámetros de entrada y salida son definidos como clases de la ontología.

Se presenta una alternativa usando un formato de operaciones como un punto en común para las capacidades y requerimientos del servicio web; estas operaciones son parte de la ontología. Los parámetros definidos en las operaciones sirven para encontrar conceptos más específicos, y estos pueden describir mejor las entradas y salidas de un servicio en específico. Estos conceptos siempre deben tener una relación taxonómica con los conceptos

del dominio que describen la operación de los parámetros.

El SSM genera un formato a partir del documento WSDL del servicio, en el que el usuario debe seleccionar un concepto de la ontología que coincida con cada parámetro de entrada y salida. Susan, en cambio, usa la ubicación como requerimiento en su búsqueda para un servicio de distancia que acepta la salida del servicio de ubicación. Sin embargo, como sólo existen relaciones no taxonómicas entre ellas, no encuentra ninguna.

En su investigación Nabila, Basir y Mamat en [22], presentan un método para extraer relaciones no taxonómicas usando la similitud entre relaciones que existen en más de una oración. Las mejores afirmaciones son usadas como punto de referencia entre las relaciones de conceptos que se encuentren no solo en una oración sino en otras más. El propósito del método es mejorar el proceso de recuperación de relaciones no taxonómicas en dominios específicos de un texto. Su modo de actuar consiste en extraer conceptos con herramientas de preprocesamiento donde el texto es divido en oraciones y se realiza un análisis estadístico para extraer los términos más importantes del texto.

Con eso, se generan pares de conceptos usando producto cartesiano entre las dos listas, después los conceptos se clasifican en dos listas una de temas y otra de objetos, estas listas son determinadas en base a los temas y objetos que haya en los textos, para evitar la existencia de relaciones taxonómicas o jerárquicas se usan algunas restricciones tales como: que el tema y el objeto no sean la misma palabra, sinónimos, parte de lo mismo o “es un“ como relación entre ellos.

Por último se realiza la extracción y asignación de buenas relaciones entre los pares de conceptos, donde se tienen contemplados tres posibles casos. El primer caso sucede cuando el tema, el objeto y el predicado se encuentran en la misma oración, en este caso, se toma al predicado como relación entre el par de conceptos.

El segundo caso se presenta cuando el predicado del tema y el predicado del objeto se encuentran en oraciones diferentes pero el predicado es el

mismo, este caso se usa para identificar las relaciones entre conceptos que aparecen en diferentes oraciones, asumiendo que como el predicado es el mismo, se puede tomar como relación.
Y en el tercer caso se tiene cuando el predicado del tema y el predicado del objeto están en diferentes oraciones y los predicados son sinónimos, aquí se puede tomar cualquier predicado como relación, ya que al ser sinónimos ambos representan el mismo significado.

Villaverde en su documento [23] propone una técnica para el descubrimiento de relaciones no taxonómicas y la extracción de elementos léxicos que sirven como conectores entre los conceptos relacionados. Su enfoque está basado en el análisis de estructuras sintácticas y dependencias entre conceptos. Su técnica intenta encontrar información semántica de los conceptos denotados por verbos que son usados generalmente para conectarlos.

Usando como base esa información, la técnica busca respaldar el trabajo de los ingenieros ontológicos al sugerir relaciones entre conceptos y ayudarlos a seleccionar las etiquetas más adecuadas de acuerdo con el análisis y procesamiento del corpus de dominio. Para el descubrimiento de relaciones no taxonómicas su técnica toma como entrada un corpus de textos específicos de dominio y una jerarquía de conceptos que describe las relaciones taxonómicas entre conceptos y busca otras posibles relaciones en el texto.

Estos textos primero son transformados en representaciones adecuadas para ser utilizados como entrada en algoritmos de preprocesamiento de texto. En esta etapa de preprocesamiento toma los documentos y elimina las llamadas '*stopwords*'. Después de la eliminación, los documentos son transformados en una representación de bolsa de palabras ('*bag-of-words*') de acuerdo al modelo de espacio vectorial.

El siguiente paso consiste en indexar los textos para propositos de recuperación de información, este paso permite una eficiente recuperación de oraciones, basada en consultas clave para una búsqueda de patrones recurrentes. Los pares de conceptos son generados usando un modelo de jerarquía, un número de conceptos de dominio y de forma opcional sus re-

laciones taxonómicas. Los conceptos incluyen sus sinónimos usando '*WordNet.*' Se usa el etiquetador POS para asignar etiquetas a las palabras de contenido e identificar frases de verbos y nominales en cada oración.

Este análisis se aplica a todas las oraciones encontradas que contienen ambos conceptos candidatos que cumplan con el patrón:

"<*término*><*verbo*><*término*>"

Una vez que las relaciones candidatas estén disponibles, son validadas bajo el criterio de las reglas de asociación. Estas reglas proveen información estadística de los conceptos y verbos que ocurren con frecuencia dentro de las oraciones de los textos de dominio.

Capítulo 3

Marco teórico

Dado que este trabajo se refiere al área de procesamiento del lenguaje natural, en particular a temas relacionados con aprendizaje de ontologías y métodos de extracción de términos, resulta necesario dar algunas definiciones para facilitar la comprensión del documento al lector. Además también se abordan temas que fueron necesarios para la extracción de las relaciones no taxonómicas.

3.1. Extracción de información

La extracción de información es el proceso de extraer información y convertirla en datos estructurados. Esto puede incluir llenar una fuente de conocimiento estructurada con información de una fuente de conocimiento no estructurada [24].

La información contenida en la base de conocimiento estructurada se puede utilizar como un recurso para otras tareas, como responder consultas en lenguaje natural o mejorar en motores de búsqueda estándar con formas de conocimiento más profundas o más implícitas que las expresadas en el texto. Por fuentes de conocimiento no estructuradas, nos referimos al texto libre, como el que se encuentra en el periódico, artículos, publicaciones de blogs, redes sociales y otras páginas web, en lugar de tablas, bases de datos y ontologías, que constituyen texto estructurado.

Al considerar la información contenida en el texto, hay varios tipos de información que pueden ser de interés. A menudo considerados como los componentes clave del texto son nombres propios, también llamados *entidades nombradas*, como personas, ubicaciones y organizaciones. Junto con los nombres propios, las expresiones temporales, como fechas y horas, a menudo también se consideran entidades nombradas.

Las entidades nombradas están conectadas entre sí al denotar que alguien que es el CEO de una compañía está conectado a la relación de que alguien es un empleado de una compañía, por medio de una relación de sub-propiedad, ya que el CEO es un tipo de empleado. Un tipo de información más complejo es el de *evento*, que puede verse como un grupo de relaciones basadas en el tiempo. Los eventos generalmente tienen participantes, una fecha de inicio y finalización, y la ubicación, aunque parte de esta información puede ser solo implícita.
En la figura 3.1 se muestran ejemplos de entidades nombradas:

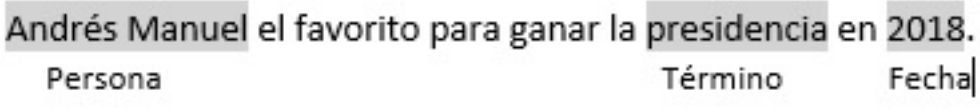

Figura 3.1: Ejemplo de entidades nombradas

3.2. Procesamiento del Lenguaje Natural

El lenguaje natural es el principal medio de comunicación usado por el ser humano para expresarse. En la práctica, diferentes categorías sintácticas son usadas para describir entidades lógicas; como son: pronombres, verbos, adverbios, adjetivos, frases de preposición, etc.

El Procesamiento de Lenguaje Natural (PLN) es una rama de las ciencias de la computación, relacionada con la inteligencia artificial y la lingüísti-

ca. El PLN elabora sistemas computacionales para la comunicación eficiente entre personas y máquinas a tráves de lenguajes naturales. Diseña mecanismos para que los programas ejecuten o simulen la comunicación, implica aspectos cognitivos, de memoria y de comprensión del lenguaje.

Entre las aplicaciones de PLN destacan la minería de datos y la recuperación de información. La minería de datos [25], entendida como la extracción de patrones y reglas significativas de una gran cantidad de información, es útil en cualquier campo en el cual exista una importante cantidad de datos y algo valioso que aprender. Mientras que Meadow [26] piensa que la recuperación de la información es una disciplina que involucra la localización de una determinada información dentro de un almacén de información o base de datos.

Sosa [27] describe el PLN como el reconocimiento y utilización de la información expresada en lenguaje humano a tráves del uso de sistemas informáticos. En su estudio intervienen diferentes disciplinas tales como lingüística, ingeniería informática, filosofía, matemáticas y psicología. Debido a las diferentes áreas del conocimiento que participan, la aproximación al lenguaje en esta perspectiva es también estudiada desde la llamada ciencia cognitiva.

La ciencia cognitiva es un campo interdisciplinario, de base empírica, preocupado por el estudio de la naturaleza de la mente humana. Según Gardner, también están comprometidos aspectos epistemológicos, motivo por el cual dicho autor definió la ciencia cognitiva "como un empeño contemporáneo de base empírica para responder a interrogantes epistemológicos de información antigua, en particular los vinculados a la naturaleza del conocimiento, sus elementos componentes, sus fuentes, evolución y difusión" [28].
El PLN consiste en la aplicación secuencial de componentes de análisis, como pueden ser sintáctico, semántico o contextual. El PLN analiza la estructura del lenguaje a cuatro niveles, los cualés son descritos por Mateos [29] a continuación :

- *Análisis morfológico*: El análisis de las palabras para extraer raíces, rasgos flexivos, unidades léxicas compuestas y otros fenómenos.

- *Análisis sintáctico*: El análisis de la estructura sintáctica de la frase mediante una gramática de la lengua en cuestión.
- *Análisis semántico*: La extracción del significado (o posibles significados) de la frase.
- *Análisis pragmático*: El análisis de los significados más allá de los límites de la frase, por ejemplo, determinar los antecedentes referenciales de los pronombres.

En la figura 3.2 se muestra el flujo común de un sistema de procesamiento del lenguaje natural o un sistema básico de extracción de información.

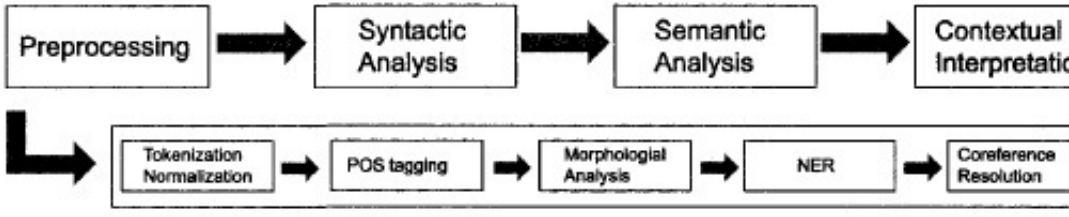

Figura 3.2: Tubería clásica de procesamiento. [30]

En los siguientes párrafos se presenta una introducción de los diferentes componentes.

3.2.1. Pre procesamiento

Entenderemos el paso de procesamiento como un conjunto de los siguientes subpasos:

- Tokenización y normalización
- Part-of-Speech(POS)
- Lematización / Stemming / Análisis morfológico
- Reconocimiento de entidades nombradas (NER)
- Resolución de la referencia

El propósito de la tokenización es detectar oraciones así como también limites de palabras. Algunos detalles en este paso, por ejemplo, los signos de puntuación como puntos, los cuales pueden denotar un fin de oración o una abreviación o pueden ser usados para especificar números telefónicos, fechas, horas, entre otros. Otro problema son los espacios en blanco que no siempre indican el límite de una palabra, tal es el caso de nombres de entidades como "Nueva York".

En la normalización se trata de encontrar fechas, horas, etc., y transformarlas en un formato estándar. A veces, la normalización también incluye la expansión de abreviaciones, para cada correspondiente abreviación se necesita un lexicón.

El etiquetado de partes de la oración *Part-Of-Speech* (POS) es la tarea de asignar a cada token su correspondiente parte del lenguaje, es decir, su categoría sintáctica de palabra tal como sustantivo, verbo, adjetivo, etc. Diferentes etiquetados tal y como diferentes paradigmas han sido aplicados a esta tarea. La lematización es típicamente aplicado como un paso de normalización, mapeando varientes morfológicas a su correspondiente forma base.

El reconocimiento de entidades nombradas (NER) consiste en reconocer las llamadas entidades nombradas, es decir, nombres que se refieren a objetos únicos en el mundo, tal como nombres de países, marcas reconocidas o personajes públicos. El reconocimiento de entidades ha sido restringido a un número pequeño de clases, considerando las clases de persona, organización, ubicación, fecha, etc.

El sistema de reconocimiento primero intenta reconocer y clasificar las entidades nombradas que aparezcan en el texto haciendo una mirada en lo que se llama *listas de gazetter*. Estas listas contienen nombres así como sus correspondientes tipos, clases o etiquetas. La Resolución de la referencia es a veces visto como un paso de pre procesamiento. En cualquier caso, es visto como un simple ordenamiento de referencias.

3.2.2. Análisis sintáctico

Tiene como función etiquetar cada uno de los elementos sintácticos que aparecen en la oración y analizar cómo las palabras combinan para formar construcciones gramáticalmente correctas. El resultado de este proceso consiste en generar una estructura correspondiente a las categorías sintácticas formadas por cada una de las unidades léxicas que aparecen en la oración. El análisis sintáctico lo podemos dividir en dos funciones el *Chunking* y el análisis gramátical, los cuáles son descritos a continuación.

El ***Chunking*** o también llamado analizador parcial o ligero, aplica técnicas de procesamiento ligero para agrupar palabras que juntas constituyen una más grande, comúnmente con un cabecera la cuál es modificada por otras palabras en la unidad. La cabecera es la principal unidad de significado con un complemento sintáctico. El verbo es la principal unidad de significado en una frase verbal y además su cabecera.

La palabra principal con significado en una frase de sustantivo en inglés es comúnmente el sustantivo que esté más a la derecha. En "*the exciting modern art museum*", '*museum*' es la palabra con mayor significado, mientras que las otras palabras son esenciales modificadores con una función de restricción de significado.

Las unidades sintácticas son generalmente llamadas *chunks* (fragmentos). Los *chunks* no están sobrepuestos, no son recursivos ni exhaustivos. No son recursivos significa que los chunks no están embebidos dentro de otros *chunks* y no son exhaustivos ya que puede haber palabras en una oración de la cuál no pertenezca a un *chunk*. Los chunkers, así, descubren grandes cantidades de palabras con las cuales construyen una unidad sintáctica.

Generalmente, los chunkers aplican tecnologías de estado finito en llamadas en cascada, donde la salida de un nivel forma la entrada de la otra, así es capaz de reusar grupos de palabras detectadas en las primeras frases. Los chunkers proceden primero detectando las unidades más sencillas y después las más complejas. En general, no descubren relaciones gramáticales como un sujeto, objeto, complemento o modificador. Además, adoptan una

estrategia conservadora y tienden a evitar errores, muchos de los chunkers disponibles, no tratan de resolver ambigüedades semánticas o sintácticas.

El **análisis gramatical** en contraste con el chunking, su objetivo es revelar toda la estructura sintáctica de una oración de entrada dada. La estructura sintáctica es representada usando dos principales paradigmas: dependencia gramátical o estructura de la frase gramatical. Mientras que las dependencias sintácticas son representadas diferentes en ambos paradigmas, ambos tienen el objetivo de descubrir grandes unidades sintácticas de palabras. Es decir, frases y hacer su relaciones de dependencias explicítas.

Su espacio de búsqueda es comúnmente tan grande que no hay analizador gramátical que no pueda evitar explorar diferentes alternativas de algún escenario. Además, entre más gramáticas se obtengan, más cantidad de ambigüedad con la que se tiene que tratar, llevando a un abrumador número de análisis por oración. Este es exactamente el orden de complejidad que es evitado cuando se usa un análisis de chunk.

3.2.3. Dependencias sintácticas

Una posibilidad para extraer características contextuales que describen un término es analizar la colección de texto y extraer dependencias sintácticas entre un verbo y su sujeto, objeto y complemento de PP de los árboles de análisis correspondientes mediante el uso del comando *tgrep*. En esencia, *tgrep* proporciona soporte para la búsqueda de ciertas rutas en árboles. Los verbos pueden ser lemmatizados.

Como se mencionó anteriormente, la lematización asigna una palabra a su forma básica y puede usarse para normalizar el texto. Una posibilidad para extraer características contextuales que describen un término es analizar la colección de texto y extraer dependencias sintácticas entre un verbo y su sujeto, objeto y complemento de PP de los árboles de análisis correspondientes mediante el uso de tgrep.

En esencia, tgrep proporciona soporte para la búsqueda de ciertas rutas en árboles. Los verbos pueden ser lemmatizados. Como se mencionó ante-

riormente, la lematización asigna una palabra a su forma básica y puede usarse para normalizar el texto.

3.2.4. Funciones de contexto

Para muchas aplicaciones de PLN, es crucial representar el contexto de ciertas palabras. Esto es importante para el sentido que la palabra no sea ambigua, es decir, la tarea de encontrar el correcto significado de una palabra dada en su contexto. Por ejemplo, la palabra en inglés '*bank*' tiene varios significados, dos de ellos son: uno en el sentido de un instituto financiero y otra en el sentido de la orilla de un río. El significado correcto de una palabra ambigua puede solo ser determinado con respecto a su contexto.

En una gran cantidad de trabajo se consideraron modelos de ventana de palabras, en las que n palabras a la izquierda y derecha de la palabra objetivo se consideran características para describir el contexto de un término. Aunque este es un enfoque válido, no está claro en qué medida todas estas palabras dentro de una ventana nos dicen algo sobre la naturaleza de la palabra objetivo.

Como se ha mencionado antes, hay varias construcciones en lenguaje natural que transmiten más información sobre los argumentos o las palabras que modifican. Este es el caso de los verbos, los adjetivos y las frases preposicionales, hay varias construcciones en lenguaje natural que transmiten más información sobre los argumentos o las palabras que modifican. Este es el caso de verbos, adjetivos y frases preposicionales.

3.3. Relaciones semánticas

La semántica es la parte lingüística que estudia el significado de las palabras, oraciones y expresiones del lenguaje. El estudio del significado se enfrenta siempre a cierta imprecisión, ya que depende tanto del contexto lingüístico como del extralingüístico. El contexto lingüístico de una palabra lo constituyen las demás palabras que la rodean: así, la posible polisemia de la palabra llave queda aclarada en la oración: *Alcánzame esa llave inglesa.*

El contexto extralingüístico es la situación en la que se pronuncia una palabra; por ejemplo, el grito de *¡Fuego!* en una cafetería repleta de gente no significa lo mismo que si se grita en unas maniobras militares.
La semántica analiza fenómenos como:

- campos semánticos
- denotación y connotación
- sinonimia y antonimia
- polisemia y homonimia

3.3.1. Campo semántico

Todas las palabras que mantienen entre sí una relación de significado forman parte de un mismo campo semántico. Por ejemplo:

- Clavel, rosa, amapola, tulipán, ... pertenecen al campo semántico de las flores.
- Estantería, mesa, silla, sofá, cama, armario, sillón, ... forman parte del campo semántico de los muebles.

El campo semántico queda definido por el sema o los semas[1] que comparten todas las palabras que pertenecen a él.
La extensión de un campo semántico depende, lógicamente, del sema que lo define; así, el campo semántico de los medios de transporte terrestre incluiría tren, autobús, coche, y muchas más, y el de los medios de transporte en general sería bastante más amplio (todos los anteriores y además los transportes marítimos, fluviales y aéreos).
Si el número de componentes de un campo semántico es fijo se llama campo cerrado (el de los meses del año, por ejemplo); en el caso contrario, es un campo abierto.

3.3.2. Denotación y connotación

El significado de las palabras está formado por un conjunto de semas o rasgos significativos mínimos. Sin embargo, no todos esos semas son igualmente compartidos por los hablantes de una lengua, sino que hay algunos

[1] Cada uno de los rasgos mínimos en que puede descomponerse el significado de una palabra se llama sema

de ellos que siempre están presentes, mientras que otros varían.

Es decir, el significado de una palabra no es siempre exactamente igual en todos los casos. Por ejemplo, si preguntásemos a varias personas por el significado de una palabra habitual como *playa*, observaríamos que el significado se compone de dos partes:

- componente común a todos: la playa es la '*ribera arenosa del mar o de un río grande*'.
- componente variable: al que suele ir allí de vacaciones, playa le sugiere descanso; al camarero que trabaja en un restaurante del paseo marítimo, la palabra playa le trae a la mente esfuerzo; para un habitante de la costa, este término no significa lo mismo que para el que procede de tierra adentro; tampoco pueden considerarlo igual un constructor y un ecologista.

En resumen, el significado de una palabra está formado por:

- Denotación: es la parte del significado objetiva y común a todos los hablantes; constituye un significado primario que no cambia según el contexto.
- Connotación: es la parte subjetiva del significado, la que depende de las circunstancias del hablante; es cualquier significado secundario que se asocia a un término.

Los diccionarios recogen en sus definiciones los semas que corresponden a la denotación, porque son los que resultan válidos para todos; en cambio, los semas connotativos solo aparecen si es una connotación compartida por numerosos hablantes. Sin embargo, el Diccionario no puede recoger otras muchas posibles connotaciones más personales, como las asociaciones que la palabra *negro* tiene para un diseñador que acaba de presentar una colección de ropa de ese color.
Las connotaciones pueden clasificarse en dos grupos:

- Connotaciones compartidas o colectivas: son las comunes a un grupo importante de hablantes. Existen connotaciones compartidas por los que tienen una misma cultura (la palabra *plaza* se asocia en los países

mediterráneos con la vida pública), por los que tienen un determinado trabajo (las connotaciones de la palabra *paloma* no pueden ser las mismas para un ornitólogo que para un empleado que limpia las calles que las palomas ensucian) o por los que habitan en la misma zona (el término *nieve* produce seguramente alegría y expectación, pero no a los habitantes de un pueblo que suele quedar aislado durante el invierno).

- Connotaciones individuales: son los significados secundarios que una persona concreta asocia a cierta palabra a partir de su experiencia. Por ejemplo, la palabra *perro* puede tener connotaciones desagradables para una persona que haya sido atacada por dicho animal. A veces, estas connotaciones son fundamentales en poesía: para Federico García Lorca, el color verde lleva asociada la idea de muerte, mientras que para la mayoría de personas el verde es un color relacionado con la esperanza.

3.3.3. Sinonimia

La sinonimia es el fenómeno que se produce cuando signos distintos y con diferente significante aluden a un mismo significado, o, más precisamente, es la relación semántica que se da entre signos que poseen alguna parcela de significación común: oscuro, sombrío, nocturno.

La sinonimia total es muy rara y se da tan sólo en el caso de términos del vocabulario de las ciencias: oftalmólogo/ oculista, pretérito perfecto simple/ pretérito indefinido, odómetro/ velocímetro, etc. Las palabras que habitualmente llamamos sinónimas no son intercambiables en todos los contextos.

Las tres puestas como ejemplo arriba, sin ir más lejos, no tienen exactamente el mismo significado, sino significaciones aproximadamente parecidas. Se puede discutir si existen sinónimos perfectos. La respuesta parece negativa, porque la identidad total de los significados supondría la coincidencia absoluta de todos los semas; según Lyons [31] deberían compartir hasta sentidos potenciales. Por ello, podemos hablar de clases de sinonimia:

- Sinonimia conceptual: Coinciden los semas denotativos, es la más

habitual: *morir, fallecer, fenecer...*

- Sinonimia referencial: Los términos remiten al mismo referente pero no '*significan*' lo mismo: *estrella de la mañana, lucero de la mañana, lucero del alba,...*
- Sinonimia contextual: Conmutabilidad de dos términos en un contexto sin alterar el significado de la secuencia: *Los garbanzos son pesados / indigestos.*
- Sinonimia de connotación: Cuando dominan las connotaciones afectivas puede borrarse totalmente el contenido conceptual y ciertos términos son equivalentes: *¡Eres un bestia / salvaje / monstruo!*

3.3.4. Antonimia

La antonimia se produce entre dos palabras de significados opuestos. Desde el punto de vista formal, los antónimos pueden ser de dos tipos:

- Antónimos gramaticales: Se forman con la ayuda de prefijos de sentido negativo: *humano/inhumano, proporción/desproporción.*
- Antónimos lexicales: Se producen entre unidades lexicas: *no/sí, nunca/siempre, dormirse/despertarse.*

Desde el punto de vista semántico, existen distintas clases de antonimia:

- Antonimia en sentido estricto: Oposición de significados que admiten gradación: *alto / bajo, grande / pequeño.* También existen términos intermedios: *mediano.*
- Complementariedad: Opuestos donde no es posible la gradación ni los términos medios: *presente / ausente, tónico / átono, vivo / muerto.*
- Reciprocidad: Términos que se implican mutuamente: *comprar / vender, padre / hijo, dar / recibir.*

3.3.5. Polisemia y homonimia

La **polisemia** consiste en la existencia de varios significados para un mismo significante, éstos dependen del contexto en que se use. La mayoría de las unidades léxicas son polisémicas. Un ejemplo es el significante banco,

que tiene distintos significados: *'asiento para varias personas', 'conjunto de peces', 'conjunto de datos', 'institución financiera'.*

La **homonimia** consiste en una coincidencia entre significantes puramente casual, por razones históricas (etimológicamente proceden de términos distintos). Con frecuencia, las palabras homónimas pertenecen a categorías gramáticales distintas. Los términos homónimos pueden ser:

- **Homógrafos:** Se pronuncian y se escriben igual:
  - haya (árbol) / haya (subjuntivo de haber)
  - haz (conjunto) / haz (imperativo de hacer)
- **Homófonos:** Se pronuncian igual, pero se escriben diferentes:
  - haya / aya (niñera);
  - vaya (subjuntivo de ir) / valla (cercado)

3.4. Reglas de asociacion

Las reglas de asociación son una técnica de minería de datos que describen la relación entre los elementos de un conjunto de datos y permiten extraer información detectando elementos de una transacción que implican la presencia de estos elementos expresando la afinidad entre ellos [32]. Estas reglas nacen de la investigación de Agrawal, Imielinski y Swami [33] donde consideran la colección de datos que generan las compras en un supermercado, sirviendo como apoyo para saber que conjunto de productos se compran y generar por medio de las reglas de asociación promociones.

En [33] definen las reglas de asociación como: Dado $T := \{t_i \mid i = 1 \dots n\}$ como la base de datos de transacciones, donde cada transacción t_i consiste en un conjuto de elementos $t_i = \{a_{i,j} \mid j = 1 \dots m_i, a_{i,j} \in C\}$ y cada elemento $a_{i,j}$ es un conjunto de elementos en C. El algoritmo calcula las *reglas de asociación* $X_k \Rightarrow Y_k (X_k, Y_k \subset C, X_k \cap Y_k = \{\})$ tal que las medidas de *soporte*(3.1) y *confianza*(3.2) sean iguales o mayores a las deseadas. Se le denomina a X_k como el *antecedente* de la regla y se le denomina como

consecuente a Y_k.

El soporte de una regla $X_k \Rightarrow Y_k$ es el porcentaje de transacciones que contiene $X_k \cup Y_k$ como un subconjunto, y la confianza de una regla $X_k \Rightarrow Y_k$ está definida como el porcentaje de transacciones donde Y_k aparece si X_k se encuentra en una transacción.

$$Soporte(X_k \Rightarrow Y_k) = \frac{|\{t_i \mid X_k \cup Y_k \subseteq t_i\}|}{n} \tag{3.1}$$

$$Confianza(X_k \Rightarrow Y_k) = \frac{|\{t_i \mid X_k \cup Y_k \subseteq t_i\}|}{|\{t_i \mid X_k \subseteq t_i\}|} \tag{3.2}$$

Las métricas se pueden interpretar como, una regla con bajo soporte indicaría que habrá aparecido por casualidad, sin embargo, una regla con baja confianza indicaría que no existe relación entre el antecedente y el consecuente. Además, existe una diferencia entre $X_k \Rightarrow Y_k$ y $Y_k \Rightarrow X_k$, debido a que las reglas comparten el mismo soporte pero su confianza tiende a ser distinta.

3.5. Ontología

El término ontología viene del griego '*ontos*' y '*logos*' etimológicamente remite al estudio del ser o ciencia del ente. Platón fue uno de los primeros filósofos que mencionó el "*mundo de las ideas o formas*" en contraste de los objetos reales u observados, que según su punto de vista solo eran '*sombras*' de las ideas. Filosóficamente, ontología es la ciencia de *¿qué es?*, es una explicación sistemática de la existencia, de los tipos de estructuras, categorías de objetos, propiedades, eventos, procesos y relaciones en cada área de la realidad.

En el lenguaje moderno de ciencias de la computación, no se habla de la 'ontología' como una ciencia que estudia al ser, sino, las ontologías como una especificación formal de una conceptualización explícita en el sentido de Gruber [34]. Una conceptualización es una vista simplificada y abstracta del mundo que deseamos representar para algún próposito en

específico, definiendo un vocabulario controlado. Explícita significa que el tipo de conceptos utilizados sean explícitamente definidos, esto es que si también pueden describir otros conceptos del mismo tipo, se definan detalladamente. Formal se refiere al hecho de que la ontología debe ser legible por la máquina, esto quiere decir, que se almacene en un formato digital.

Así que, mientras 'ontología' fue originalmente una ciencia, 'las ontologías' han recibido el estado de los recursos que representan el modelo conceptual que está en un dominio específico, describiéndolo de manera declarativa y, por lo tanto, separándolo de los aspectos de procesos.Las ontologías están formadas de los siguientes componentes que servirán para representar el conocimiento de algún dominio en específico [35]:

- **Conceptos**: son las ideas básicas que intentan formalizar, estos conceptos pueden ser clases de objetos, métodos, planes, estrategias, procesos de razonamiento, etc.
- **Relaciones**: representan la interacción y el enlace entre los conceptos del dominio. Suelen formar la taxonomía del dominio.
- **Funciones**: son un tipo concreto de relación, donde se identifica un elemento mediante el cálculo de una función que considera varios elementos de una ontología.
- **Instancias**: son utilizadas para representar objetos determinados de un concepto.
- **Axiomas**: son teoremas que se declaran sobre relaciones que deben cumplir lo elementos de la ontología.

Mientras que las aplicaciones para las ontologías en ciencias de la computación crecen de forma constante, la necesidad para aclarar la definición de una ontología aumenta al mismo tiempo. En el pasado, han sido muchas propuestas para un lenguaje de ontologías con un sintaxis bien definida y semántica formal, especialmente en el contexto de web semántica, tal como *Ontology Inference Layer* (OIL) [36], RDFS [37] o OWL [38].
En [30] se define a una ontología; un ontología es una estructura:

$$O := (C, \leq c, R, \sigma_R, \leq_R, A, \sigma_A, \tau)$$

que consiste en:

- Cuatro conjuntos disjuntos C, R, A y τ cuyos elementos se denominan identificadores de concepto, identificadores de relación, identificadores de atributo y tipos de datos, respectivamente,
- una retícula semi-superior $\leq_C$ en C con elementos superiores $root_C$, llamados conceptos jerárquicos o taxonómicos,
- una función $\sigma_R : R \to C^+$ llamada relación de firma,
- un order parcial $\leq_R$ en R, llamada relación jerárquica, donde $r_1 \leq_R r_2$ implica $|\sigma_R(r_1)| = |\sigma_R(r_2)|$ y $\pi_i(\sigma_R(r_1)) \leq_C \pi_i(\sigma_R(r_2))$, para cada $1 \leq i \leq |\sigma_R(r_1)|$, y
- una función $\sigma_A : A \to C \times \tau$, llamada atributo de firma,
- un conjunto τ de tipos de datos tales como cadenas, enteros, etc.

De este modo, $\pi_i(t)$ es el componente $i-th$ de la tupla t. En algunos casos, cuando queda claro en el contexto si se está refieriendo a una relación o atributo, simplemente se usará σ.
Además una retícula semi superior $\leq$ cumple con las siguientes condiciones:

$$\forall x \leq x(reflexiva) \tag{3.3}$$

$$\forall x \forall y(x \leq y \wedge y \leq x \to x = y)(antisimetrica) \tag{3.4}$$

$$\forall x \forall y \forall z(x \leq y \wedge y \leq z \to x \leq z)(transitiva) \tag{3.5}$$

$$\forall x x \leq top(elementosuperior) \tag{3.6}$$

$$\forall x \forall y \exists z(z \geq x \wedge z \geq y \wedge \forall w(w \geq x \wedge w \geq y \to w \geq z))(supremo) \tag{3.7}$$

Así que cada dos elementos tienen un supremo único más específico. En el contexto de las ontologías se refiere a este elemento como el subsumidor menos común.

Dominio y rango

Para una relación $r \in R$ con $|\sigma(r)| = 2$, se define su dominio y rango como: $dom(r) := \pi_1(\sigma(r))$ y $rango(r) := \pi_2(\sigma(r))$.
Si $c_1 <_C c_2$, para $c_1, c_2 \in C$, entonces c_1 es un subconcepto de c_2, y c_2 es un superconcepto de c_1. Si $r_1 <_R r_2$, para $r_1, r_2 \in R$, entonces r_1 es una subrelación de r_2, y r_2 es una superrelación de r_1.

Sistema de axiomas-$\boldsymbol{\iota}$

Sea $\boldsymbol{\iota}$ un lenguaje lógico. Un sistema de axioma $\boldsymbol{\iota}$ para una ontología $O := (C, \leq_C, R, \sigma_R, \leq_R, A, \sigma_A, \tau)$ es una tripleta:

$$S := (AS, \alpha, \boldsymbol{\iota})$$

Donde:

- AS es un conjunto cuyos elementos se denominan esquemas de axiomas y
- $\alpha : AS \to AS_{\boldsymbol{\iota}}$ es una asignación de AS de esquemas de axiomas definidos en $\boldsymbol{\iota}$

Una ontología con un sistema de axiomas $\boldsymbol{\iota}$ es un par

$$(O, S)$$

donde O es una ontología y S es un sistema de axiomas $\boldsymbol{\iota}$ para O.

3.6. Aprendizaje ontológico

La construcción semiautomática o automática de ontologías es también llamada aprendizaje ontológico. Este término está generalmente atribuido a Mädche y Staab [4] y puede ser descrito como la adquisición de un modelo de dominio de información. Esto es comúnmente relacionado con la semántica Web, en la que se construyen modelos ontológicos o formalismos lógicos restringidos para escoger fragmentos de lógica de primer grado, en especial, descripción lógica. De esta forma los modelos de dominio están restringidos en su complejidad y expresividad.

El aprendizaje ontológico necesita de una entrada de información de la cuál aprenderá los conceptos relevantes del dominio dado, sus definiciones y las relaciones que se establezcan entre ellos. Por lo tanto, un requisito fundamental es que los datos de entrada sean representativos del dominio para el que se pretende aprender una ontología. Los datos de entrada pueden ser esquemas como XML - DTD, diagramas UML o esquemas de bases

de datos.

Llamamos a este tipo de aprendizaje ontológico: '*lifting*' [39], ya que consiste principalmente en mapear definiciones del esquema a las definiciones ontológicas correspondientes. El aprendizaje ontológico también se puede realizar sobre la base de fuentes semiestructuradas como documentos XML o HTML o estructuras tabulares. En caso de que el aprendizaje ontológico se realice sobre la base de recursos textuales no estructurados, hablaremos de *aprendizaje ontológico de texto.*

El aprendizaje ontológico puede considerarse en cierta medida como un proceso de ingeniería inversa. El autor de cierto texto o documento tiene en mente un modelo del mundo o dominio que se comparte con otros autores que escriben textos sobre el mismo dominio. Este modelo de dominio, entre muchos otros factores, como el mensaje deseado, da forma al contenido del texto resultante.

La tarea de reconstruir el modelo completo del autor o incluso del modelo compartido por un autor diferente, se puede ver como ingeniería inversa. La tarea es intrínsecamente compleja y difícil debido principalmente a dos razones. En primer lugar, es típico que solo una pequeña parte del conocimiento del dominio de los autores esté involucrado en el proceso de creación, de modo que el proceso de ingeniería inversa, en el mejor de los casos, solo puede reconstruir parcialmente el modelo del autor.

Segundo, y mucho más importante, el conocimiento, a menos que se esté considerando un libro de texto o diccionario. De hecho, el conocimiento suele estar contenido solo como una implicación en los textos en la forma en que los autores utilizan ciertas palabras o estructuras lingüísticas.El desarrollo ontológico se ocupa principalmente de axiomatizar la definición de conceptos, así como las relaciones entre ellos.

Para algunas aplicaciones de ontologías en la minería de textos o el procesamiento de lenguaje natural, así como para la legibilidad humana, también es importante relacionar los conceptos y la adquisición de conocimientos lingüísticos sobre los términos que se utilizan para referirse a

conceptos específicos y sinónimos potenciales de estos términos.

Una ontología consiste además en una jerarquía de conceptos así como en otras relaciones no jerárquicas. Para restringir la interpretación de conceptos y relaciones, se pueden instanciar esquemas axiomáticos tales como la desunión de conceptos, simetría, reflexividad, transitividad, etc. Finalmente, también se está interesado en usar una ontología para derivar hechos que no están explícitamente modelados en la base de conocimiento pero que pueden derivarse de ella.

Todo lo mencionado, describe las primitivas ontológicas que se pueden organizar en un pastel de capas, de acuerdo con las tareas cada vez más complejas dentro de la ontología que aprenden a adquirirlas. La capa muestra las diferentes subtareas de aprendizaje de una ontología, i.e.

- adquisición de terminología relevante
- identificación de términos sinónimos / variantes lingüísticas
- formación de conceptos
- organización jerárquica de conceptos
- relaciones de aprendizaje, propiedades o atributos, juntos con el dominio y rango apropiados
- organización jerárquica de las relaciones
- ejemplificación de esquemas de axiomas
- definición de axiomas arbitrarios

3.6.1. Tareas del aprendizaje ontológico

A continuación se describen las diferentes subtareas del aprendizaje ontológico a lo largo de su diferentes capas (3.3) .

Términos

La extracción de términos es un prerrequisito para todos los aspectos del aprendizaje ontológico de texto. Los términos son realizaciones lingüísticas de conceptos específicos del dominio y, por lo tanto, son fundamentales para

Figura 3.3: Capas del aprendizaje ontológico [30]

tareas más complejas. La tarea aquí es encontrar un conjunto de términos o signos relevantes para conceptos y relaciones que sean característicos del dominio representado en la colección de texto y que proporcionen la base para definir un léxico para una ontología.

Desde un punto de vista lingüístico, los términos son palabras simples o compuestos de varias palabras con un significado muy específico, posiblemente técnico, en un contexto o dominio dado. Nuestra definición de término es ligeramente más general en el sentido de que nos referiremos a cualquier palabra única o compuesto de varias palabras relevante para el dominio en cuestión como un término.

Por lo tanto, la entrada a esta tarea es una colección de documentos que representan el dominio de interés, mientras que la salida es un conjunto de cadenas que representan términos que se utilizarán como signos de conceptos y relaciones, respectivamente.

Sinónimos

La tarea del descubrimiento de sinónimos consiste en encontrar palabras que denoten el mismo concepto y que, por lo tanto, aparezcan en el mismo conjunto para un concepto dado. Hasta cierto punto, estos elementos pueden considerarse como sinónimos. Es bien sabido que los sinónimos reales casi no existen, ya que existen diferencias sutiles incluso entre las palabras que comúnmente se consideran como tales.

Por lo tanto, nuestra definición de sinónimo es menos estricta. Considerará dos palabras como sinónimos si comparten un significado común que puede usarse como base para formar un concepto relevante para el dominio en cuestión [40].

Se considerará que existe una superposición significativa entre esta definición de sinonimia y la relación léxica de co-hiponimia. La co-hiponimia se define típicamente como la relación entre los hipónimos de un hipernimo común. Es importante mencionar que la sinonimia, la co-hiponimia, la hipernimia y la hiponimia son relaciones léxicas que no pueden considerarse equivalentes a las nociones de igualdad, superconcepto y relaciones de subconcepción entre conceptos, que se definen de manera extensiva. Las relaciones léxicas se definen en el nivel de las palabras.

Conceptos

En el punto de vista de [41] la *formación de conceptos* idealmente debería proporcionar una definición intensional de conceptos, su extensión y los signos léxicos que se utilizan para referirlos. Así, para los propósitos del aprendizaje ontológico, se define un concepto como una tripleta $< i(c), \|c\|, Ref_c(c) >$, donde $i(c)$ es la intensidad del concepto, $\|c\|$ su extensión y Ref_c describe su relación lexica en el corpus.

El lexicon también puede contener estructuras más complejas enriquecidas con información estadística como se describe en [42], o incluso análisis de árboles, marcos de subcategorización, etc. Aunque no existe una definición explícita de intensidad dentro del modelo, se asumirá la intensidad como una descripción en lenguaje natural del significado intuitivo de un concepto de atributos en línea con las glosas de WordNet o una colección de atributos en línea con la teoría del Análisis de Conceptos Formales.

Jerarquías de conceptos

A continuación se presentan las tareas relacionadas con la inducción, extensión y refinación de la parte fundamental de la ontología, es decir, su jerarquía de conceptos.

- **Inducción de la jerarquía de conceptos**
  Definimos la inducción de la jerarquía de conceptos como la tarea de, dado un conjunto de conceptos C, típicamente junto con su realización léxica Ref_C, aprendiendo pares (c_i, c_j) donde c_i, c_j en C tal que $\leq c = \bigcup_{i,j} \{(c_i, c_j)\}$ forma una retícula semi superior. La tarea aquí entonces es inducir una jerarquía de conceptos desde cero. Empezando con un conjunto de conceptos C, la tarea sería derivar una relación $\leq_C$ reflejo.
- **Refinamiento**
  Se define el refinamiento de jerarquía de conceptos como la tarea de, dado un conjunto de conceptos C así como una retícula semi superior $\leq_c$ en C, pares de aprendizaje (c_i, c) tal que $c \in C$. El refinamiento de jerarquía $C' := C \bigcup_i c_i and \leq_{C'} = \leq_C \cup \bigcup_i \{(c_i, c)\}$ todavía debe formar una retícula semi superior$(C', \leq'_C)$. La tarea aquí es extender la jerarquía de conceptos existente con subconceptos adicionales de conceptos ya existentes, refinando así la jerarquía.
- **Extensión léxica**
  Se define la extensión léxica o refinamiento léxico de una jerarquía de conceptos, como la tarea de, dado un concepto c junto con su función de referencia léxica $Ref_C(c)$, encontrando nuevas realizaciones léxicas s_i del concepto c, extendiendo así $Ref_C(c)$, i.e. $Ref'_C(c) :=$

$Ref_C(c) \bigcup_i \{(s_i)\}$.

Relaciones

En las relaciones binarias se define el aprendizaje de las relaciones como la tarea de aprender los identificadores de las relaciones o las etiquetas r, así como su dominio adecuado $dom(r)$ y su rango $range(r)$. Aquí se distinguen las siguientes tareas:

- encontrar conceptos en C en alguna relación ontológica no taxónomica,
- especificando R, i.e. encontrar etiquetas apropiadas e identificadores de relación sobre la base del corpus dado,
- dada una cierta relación $r \in R$, que determina el nivel correcto de abstracción con respecto a la jerarquía de conceptos para el dominio y el rango de la relación,
- aprender un orden jerárquico $\leq_R$ entre las relaciones en R

Esquemas de axiomas

En lo que respecta a la definición axiomática de conceptos y relaciones, el objetivo del aprendizaje ontológico no es aprender los esquemas de axiomas en sí. Se asume la existencia de algún sistema $\iota - axioma$, que define esquemas de axiomas que se utilizan a menudo en ingeniería ontológica y, por lo tanto, merecen un estado especial.

Para los conceptos, se tiene por ejemplo, separación o equivalencia de axiomas, mientras que para las relaciones se tienen axiomas que describen las propiedades de la relación, es decir, la transitividad, la simetría, etc. La tarea aquí es aprender qué conceptos, relaciones o pares de conceptos aplican los axiomas del sistema, i.e. se puede querer conocer qué pares de conceptos son desarticulados, qué relaciones son simétricas, la cardinalidad mínima o máxima de una relación.

Axioma general

La situación es diferente para la tarea de aprender axiomas generales, en los cuales los axiomas mismos deben aprenderse y no meramente instanciarse. Aquí el tipo de axiomas depende en gran medida del formalismo lógico usado en el fondo. Los axiomas generales pueden considerarse como implicaciones lógicas que limitan la interpretación de conceptos y relaciones.

Se diferencias los esquemas axiomáticos que no ocurren con tanta frecuencia y, por lo tanto, no merecen ningún estatus especial. La tarea de aprender axiomas puede entenderse así como consistir en derivar relaciones y conexiones más complejas entre conceptos y relaciones. Estos axiomas se pueden representar, por ejemplos, utilizando un fragmento Horn de la lógica de primer grado.

3.7. Herramientas

Durante la elaboración de esta investigación se ha usado el lenguaje de programación *python*, este lenguaje cuenta con una biblioteca, la cual contiene funciones para trabajar el procesamiento del lenguaje natural. A continuación serán descritas estas dos herramientas.

3.7.1. Python

Es un lenguaje muy robusto que permite trabajar más rápidamente e integrar tus sistemas más eficazmente (python, 2018). El lenguaje cuenta con bibliotecas para trabajar con temas específicos, en este caso particular, el procesamiento del lenguaje natural.

En [43] se explica que python es un excelente lenguaje para aprender a programar. Hay muchas razones para esto, pero la explicación simple es que es fácil de leer y rápido de escribir; No toma mucho tiempo crear un código de trabajo que haga algo significativo. Python tiene una sintaxis muy amigable para los humanos, lo que facilita la escritura de código elegante.

El lenguaje básico es bastante simple y, por lo tanto, fácil de recordar, y luego tiene una extensa biblioteca de funciones predefinidas que puede usar para facilitar las tareas más comunes de la computadora. Escribir aplicaciones efectivas en Python puede ser tan simple como jugar con bloques de construcción conceptuales.

Funciona realmente bien para escribir una pequeña aplicación de dos líneas para realizar alguna tarea de administración de sistema de rutina o para proporcionar funciones interactivas en una página web, pero tiene la potencia y la flexibilidad suficientes para crear con comodidad aplicaciones mucho más grandes y complejas con interfaces gráficas que no se distinguen de los programas que está acostumbrado a ejecutar desde el menú principal de su computadora.

Si sigue las sugerencias expuestas en este libro sobre cómo escribir un código autoexplicativo, en varios meses, incluso años, podrá volver a sus programas y ver de inmediato que se supone deben hacer y cuáles fueron sus intenciones originales; Esto hace que el mantenimiento de programas sea mucho más simple también.

3.7.2. NLTK

Python cuenta con una librería llamada NLTK (Natural Language Tool-Kit) con excelente funcionalidad para procesar datos lingüísticos. Con NLTK [44] se pueden realizar tareas como el acceso al corpus, procesamiento de cadenas de texto, aprendizaje de máquina, chunking, etiquetado, interpretaciones semánticas y entre muchas otras tareas en el área del procesamiento del lenguaje natural.

En su página oficial [45] se informa que NLTK es una plataforma líder para la creación de programas Python para trabajar con datos en lenguaje humano. Proporciona interfaces fáciles de usar para más de 50 recursos corporales y léxicos , como WordNet, junto con un conjunto de bibliotecas de procesamiento de texto para clasificación, tokenización, derivación, etiquetado, análisis y razonamiento semántico, envoltorios para bibliotecas

PLN de gran solidez industrial. y un foro de discusión activo.

Gracias a una guía práctica que presenta los aspectos básicos de la programación junto con los temas de lingüística computacional, además de la documentación completa de API, NLTK es adecuado para lingüistas, ingenieros, estudiantes, educadores, investigadores y usuarios de la industria. NLTK está disponible para Windows, Mac OS X y Linux. Lo mejor de todo es que NLTK es un proyecto gratuito, de código abierto y dirigido por la comunidad.

NLTK ha sido llamada *una herramienta maravillosa para enseñar y trabajar en lingüística computacional usando Python* y *una biblioteca increíble para jugar con lenguaje natural*. El procesamiento del lenguaje natural con Python proporciona una introducción práctica a la programación para el procesamiento del lenguaje. Escrito por los creadores de NLTK, guía al lector a través de los fundamentos de la escritura de programas Python, el trabajo con corpus, la categorización del texto, el análisis de la estructura lingüística y más.

NLTK utiliza el segmentador de sentencias Punkt [46]. Esto utiliza un enfoque no dependiente del lenguaje para la detección de límites de oraciones, basado en la identificación de abreviaturas, iniciales y números ordinales. Su detección de abreviaturas, a diferencia de la mayoría de los divisores de oraciones, no se basa en listas precompiladas, sino que se instala en base a métodos para la detección de colación, como la probabilidad de registro.

3.7.3. RDF

RDF o *Resource Description Framework*, no es realmente un lenguaje que sirva para el modelado de ontologías, sino más bien para la definición de metadatos, ya que proporciona un modo sencillo de expresar afirmaciones acerca de recursos web, tratando de aportar interoperabilidad ante la multiplicidad de formatos incompatibles existentes. No es un lenguaje que permite modelas ontologías, porque una ontología se compone de una clase, relaciones, atributos, etc., y estos elementos no son del todo implementables en un documento RDF.

Lo que pretende RDF es ser un soporte para la expresión de relaciones entre recursos de cualquier tipo con carácter universal y distribuido de modo que facilite la identificación de la información sin dar lugar a ambigüedades principalmente encaminado a ser procesado por aplicaciones en lugar de clientes humanos. Está basado en el uso de afirmaciones (*staments*). Por ejemplo, una afirmación podría ser: '*la página https:"www.lawebsemantica.com*' tiene como creador a Santiago Martínez. En el estándar RDF, las afirmaciones se componen de una serie de elementos principales, como son [47]:

- **Sujeto**: Sobre qué vamos a hacer una afirmación (la página en el ejemplo anterior). También se suelen denominar recursos y la idea es ver los recursos como objetos de los que queremos hablar. Todo sujeto o recurso queda identificado por una URL (*Uniform Resource Locator*), que en el aso de los documentos RDF hay que entenderlas como identificadores que apuntan a partes específicas del documento.
- **Predicado**: La propiedad del recurso que estamos describiendo (su creador en el ejemplo). Se utilizan para describir las relaciones entre los sujetos y también quedan identificados por una URL.
- **Objeto**: Lo que vamos a asignar como valor a la propiedad anterior (el nombre de su autor).

Existen varias formas de representar las afirmaciones RDF, como:

- **Declaración Explícita**: Se basa en el uso de una fórmula lógica del tipo: (x, P, y), donde el predicado binario P relaciona al objeto x con el objeto y. Cabe mencionar que RDF solo acepta predicados binarios.
- **Grafos RDF**: Se basa en la identificación de la terna: $(sujeto - predicado - objeto)$, como un grafo en el que se conectan los dos nodos correspondientes a los elementos sujeto y objeto a tráves de un arco que representa el predicado. En RDF es un aspecto importante que el sujeto de una afirmación puede ser sujeto de otra declaración diferente.

3.7.4. OWL

En [47] se dice que OWL (*Web Ontology Language*) o Lenguaje de Ontologías para la Web, está diseñado para usarse cuando la información contenida en los documentos necesita ser procesada por programas o aplicaciones, en oposición a situaciones donde el contenido solamente necesita ser presentado a los seres humanos. OWL puede usarse para representar explícitamente el significado de términos en vocabularios y las relaciones entre aquellos términos.

Esta representación de los términos y sus relaciones se denomina una ontología. En realidad, OWL es una extensión del lenguaje RDF y emplea tripletas de RDF, aunque es un lenguaje con más poder expresivo. OWL posee más funcionalidades para expresar el significado y semántica que XML, RDF y RDFs, pero OWL va más allá que estos lenguajes pues ofrece la posibilidad de representar contenido de la Web interpretable por máquina.

XML es un meta-lenguaje que nos permite definir lenguajes de marcado adecuados a usos determinados permitiendo definir etiquetas personalizadas para descripción y organización de datos [48].

OWL es una revisión de lenguaje de ontologías web DAML (*DARPA's Agent Markup Language*) + OIL (*Ontology Inference Layer*) que incorpora lecciones aprendidas desde el diseño y aplicaciones DAML+OIL. OWL es en realidad un lenguaje de etiquetado semántico para publicar y compartir ontologías en la *World Wide Web* (WWW).

OWL se ha desarrollado como una extensión del vocabulario RDF. El motivo de desarrollo de este lenguaje ha sido la puesta en marcha de la Web Semántica, o sea, una visión para el futuro de la Web en el cuál el significado de la información será dado de forma explícita haciendo que las máquinas automaticen de forma más fácil los procesos e integren la información disponible en la Web.

Capítulo 4

Diseño

Gracias al conocimiento adquirido a tráves del estudio del estado del arte y los conceptos teóricos vistos en el capítulo anterior, se propone un enfoque de solución para la extracción de relaciones no taxonómicas utilizando la técnica de minería de datos: reglas de asociación. Después de obtener las relaciones del corpus, se someten a una evaluación para determinar su relevancia en el dominio.

Este capítulo consta de dos partes: el primero presenta el algoritmo propuesto para la extracción de términos de un corpus de dominio, que serán considerados posteriormente como conceptos. La segunda parte presenta el enfoque usado para la extracción de las relaciones no taxonómicas basado en la técnica de reglas de asociación. Estas dos partes forman el sistema completo propuesto en esta investigación.

4.1. Algoritmo de extracción de términos

En esta fase para la extracción de términos del corpus de dominio se utiliza funciones de procesamiento del lenguaje natural para extraer mejor el contenido de la información del corpus de dominio y poder obtener los mejores resultados en cuanto a calidad de conceptos. Esta fase está descrita en el algorimo 4.1.

El funcionamiento general del algoritmo 4.1 consta en: identificar los acrónimos existentes en el corpus mediante la función *Acronimos*, la cuál por medio de una expresión regular, son seleccionados aquellos tokens de las oraciones que se deseen, especificando un patrón con la forma en que se desean encontrar las palabras. En este caso, se seleccionan aquellas palabras que cumplan con tener dos o más letras mayúsculas consecutivas sin que la palabra tenga minúsculas en ella. Después el corpus es dividido en oraciones en la función *Oracion*, esta división está dada por cada punto que aparezca en el corpus.

Algoritmo 4.1: Algoritmo de extracción de términos propuesto

Datos: Corpus de dominio
Resultado: Corpus de dominio pre-procesado

```
1 a ← Acronimos(corpus)
2 sentences ← Oracion(corpus)
3 sentences, dic ← prepro(sentences,a)
4 resul ← pos_tag(sentences) inds ← buscarNPS(resul)
5 cand ← relacionesCandidatas
6 triple ← reglasAsociacion
```

Esto con la finalidad de que en la función *prepro* se realice una expansión de términos que es detallada en el algoritmo 4.2. La función *prepro* recorre cada oración en busca de los acrónimos obtenidos en el paso anterior y los almacena en la variable *dic* que tiene función como diccionario de acrónimos. Cada oración es divida en tokens y se realiza una comparación entre los tokens de las oraciones y los acrónimos obtenidos, para comprobar qué acrónimos tiene la oración. Si el resultado de la comparación tiene elementos, la función continúa con verificar cada elemento encontrado en la comparación.

Si el acrónimo no está en el diccionario de acrónimos, se busca su significado tomando el tamaño del acrónimo para buscar hacia la izquierda las palabras que formen el significado, de coincidir las primeras letras de cada palabra con las letras que conforman el acrónimo, el acrónimo es agregado al diccionario y se elimina de la oración dejando solo su significado para evitar redundancia. Si el acrónimo ya se encuentra en el diccionario, solo

se realiza la sustitución del acrónimo con su significado en la oración. Al finalizar el proceso, la función regresa el corpus en forma de lista de oraciones con sus acrónimos ya expandidos y un diccionario que contiene todos los acrónimos del corpus con su significado.

Algoritmo 4.2: Función *prepro*

```
   Datos: Oraciones del corpus, acrónimos
   Resultado: Oraciones con acrónimos expandidos, diccionario de
              acrónimos
 1 Para i = longitud-oraciones hacer:
 2   oracion ← oraciones[i]
 3   comp ← Para elemento en acronimos Si elemento está en
     oracion
 4   Si comp no está vacío hacer:
 5     Para palabra en comp hacer:
 6       indice← oracion.index(oracion)
 7       Si la palabra está en dic hacer:
 8         Para letra en palabra:
 9           Se toma el indice del acronimo, se elimina y se expande
10         FinPara
11     FinPara
12       Sino:
13         rango ← len(palabra)
14         Para j<rango:
15           Si oracion[ind-(rango-j)][0] == palabra[0] hacer:
16             acronimo = acronimo + oracion[ind-(rango-j]
17           Sino
18             eliminar acronimo de acronimos
19           FinSi
20         FinPara
21       FinSi
22   FinSi
```

El siguiente paso es volver las oraciones en su forma minúscula para lograr una comparación entre las palabras más exacta. Después se realiza el etiquetado *Parte del discurso*; se hace un etiquetado al contenido y función de las palabras logrando así, identificar las clases de palabras en las

oraciones. Mediante una gramática, las frases nominales de las oraciones se identifican. Esta gramática es muy parecida a una expresión regular, la diferencia es el uso de las etiquetas de POS para crear el patrón de frase que se desea encontrar. Las frases nominales son aquellas que constan de varias palabras que juntas forman un solo sustantivo. Estas frases nominales solo son etiquetadas en el análisis y son devueltas como tipo árbol por cada oración, así que se hace una extracción de las frases nominales en la función *buscarNPS* vista en el algoritmo 4.3.

La función *BuscarNPS* consta en recorrer cada árbol generado en el paso anterior, con la intención de encontrar las frases nominales que se agruparon. Para poder ser encontradas se debe considerar que, al etiquetar una palabra con el analizador POS de la herramienta NLTK, cada palabra se vuelve una tupla[1], que consta de la palabra y su etiqueta de clase. Sabiendo esto, se pueden tener tres escenarios posibles para la identificación de nuestras frases nominales deseadas.

El primer escenario es considerar que una frase nominal puede constar de más de dos palabras, es decir, el nodo agrupa tres o más tuplas. Esto se ve reflejado en el paso seis del algoritmo 4.3, la función *len* de python devuelve el valor entero del tamaño de la variable que se defina en ella. Al ser verdadero se toma el primer valor de cada tupla que haya en el nodo (la palabra) y se agrega a una lista por cada oración. En el segundo escenario se considera, si el tamaño del nodo es igual a dos, puede ser por dos motivos.

El primer motivo es porque el nodo contiene dos tuplas y si es así, se pregunta si la primer tupla contiene alguna de las etiquetas NN (sustantivo), JJ (adjetivo) o DT (determinante) como se observa en el paso 12. Estas etiquetas son las que se determinaron en el patrón para crear las frases nominales y de contener alguna de ellas el nodo, indica que es una frase nominal y debe guardarse.

El segundo motivo y tercer escenario, es si el nodo es tan solo una tupla, aquí se considera que el árbol resultante da el nodo de frase nominal como

[1]En python una tupla es un tipo de lista que solo consta de dos elementos

un árbol y si solo tiene un elemento, es decir, una palabra, consta de una sola tupla. Eso indica que su tamaño sea igual a dos, aquí basta con saber si el primer elemento es una tupla y si lo es agregar la palabra.

Algoritmo 4.3: Función *BuscarNPS*

```
   Datos: Árbol de POS (resul )
   Resultado: ind <indiceOracion, nps>
 1 i ← 0
 2 Para oracion en resul hacer:
 3   nps ← [ ]
 4    palabra ← ' '
 5   Para nodo en oracion hacer:
 6     Si len(nodo) > 2 hacer:
 7       Para j <len(nodo) hacer:
 8         palabra ← palabra +nodo[j][0]
 9       FinPara
10       nps ← agrega(palabra)
11     FinSi
12     Sino Si nodo[0][1] contiene la etiqueta NN, DT o JJ hacer:
    Para j <len(nodo) hacer:
13         palabra ← palabra +nodo[j][0]
14       FinPara
15       nps ← agrega(palabra)
16     FinSi, FinSino
17     Sino Si nodo[0] es una tupla
18       Para j <len(nodo) hacer:
19         palabra ← palabra +nodo[j][0]
20       FinPara
21       nps ← agrega(palabra)
22     FinSi, FinSino
23     FinPara
24   aux ← [i,nps]
25   ind ← agrega(aux)
26   i ← i + 1
27 FinPara
```

Cuando se tiene la lista de frases nominales que ocurren en una oración, se guardan junto con el índice de la oración en el que se encuentran, representado por i y se devuelve como salida esa nueva lista de índices con frases nominales.

Una vez disponibles las frases nominales junto con el índice de la oración en la que se encuentran, se continúa con la extracción de las relaciones no taxonómicas.

4.2. Algoritmo de extracción de relaciones no taxonómicas

En el algoritmo propuesto para la extracción se tienen dos funciones: *relacionesCandidatas* y *reglasAsociacion*. El procedimiento que sigue la función *relacionesCandidatas* se detalla en el algoritmo 4.4.

En esta función se observa que la entrada son las oraciones del corpus y la lista de frases nominales e índices de su oración de ocurrencia, la salida es las tripletas candidatas para la ontología. La intención del algoritmo de la función *relacionesCandidatas* comienza encontrando una frase verbal que conecte a dos frases nominales que ocurren en la oración.

Para que el verbo sea encontrado se toma el texto que hay entre las dos frases nominales que hay en la oración. Del paso 3-5, se toman los índices de donde comienzan las dos frases nominales para conocer el texto que hay entre ellas.

Este texto comienza en el número del índice de la primer frase nominal más su tamaño más uno y termina en el índice de la segunda frase nominal. Este texto es analizado por el etiquetador *POS* de NLTK para conocer que tipo de palabras son el conector entre estas frases nominales. Y nuevamente se tiene diferentes escenarios.

El primero, si el texto se constituye solo de una palabra, se debe verificar que la palabra sea un verbo, si no lo es, se ignora y se continua en el

ciclo pero de serlo, se agrega a la tripleta junto a las frases nominales con la forma: <*frase1, verbo, frase2*>.

El segundo caso es que el texto entre el par de frases nominales sea menor que cinco pero mayor a uno, se toma este rango ya que de contener más palabras es posible que no sea una frase verbal la que conecta a los sustantivos. En este caso surgen los dos escenarios cuando se trata de encontrar las frases nominales, esto se debe a que se trata con árboles.

De la misma forma, se crea una gramática para identificar las frases verbales que ocurran en el texto obtenido. Si su longitud es igual a uno, indica que es una sola frase nominal y ésta debe ser almacenada como tripleta. De ser mayor la longitud del nodo se busca el verbo en el texto obtenido para que sea agregado como tripleta si es que existe.

Después de obtener las tripletas candidatas son evaluadas bajo las métricas de *soporte y confianza* de las *reglas de asociación* [33] (ver sección 3.4). En este contexto, una transacción en T representa la ocurrencia de un par de conceptos con algún verbo de enlace en el cuerpo del texto.

La fuerza de la asociación de ambos conceptos con el verbo estará dada por la regla de confianza. Se deben encontrar las frases nominales que tienen soporte de transacciones por encima del soporte mínimo. El soporte para un conjunto de elementos es el número de transacciones que contienen el conjunto de elementos.

Algoritmo 4.4: Función *relacionesCandidatas*

```
   Datos: oraciones, ind
   Resultado: tripleta
 1 Para i <len(oraciones) hacer:
 2   Para j < len(ind[i][1]-1) hacer:
 3     ind1 ← oraciones[i].index(ind[i][1][j])
 4     ind2 ← oraciones[i].index(ind[i][1][j+1])
 5     vp ← oraciones[i][len(ind[i][1][j])+ind1+1:ind2]
 6     vp2 ← nltk.postag(vp)
 7     Si len(vp2) == 1 y vp2[0][1] == V hacer:
 8       aux← [ind[i][1][j],vp,ind[i][1][j+1]]
 9       tripleta← agrega(aux)
10     Sino Si 5 > len(vp2)>1 hacer:
11        vp2 ← parse(vp2)
12       nps ← agrega(palabra)
13       Si len(vp2) == 1 hacer:
14         aux← [ind[i][1][j],vp,ind[i][1][j+1]]
15         tripleta← agrega(aux)
16       Sino hacer:
17         Para k <len(vp2) hacer:
18           Si len(vp2[k]) >1 y su tipo es un árbol hacer
19             Para v en vp2[k] hacer:
20               verbo ← verbo + v[0] +
21               verbo ← verbo[:len(verbo)-1]
22               aux ← [ind[i][1][j],verbo,ind[i][1][j+1]]
23                cand ← agrega(aux)
24             FinPara
25           FinSi
26         FinPara
27       FinSi
28     FinSi
29   FinPara
30 FinPara
```

Capítulo 5

Resultados

En este capítulo se muestran los resultados de nuestro enfoque propuesto para la extracción de conceptos y de las relaciones no taxonómicas. También se explica la métrica utilizada para evaluar los resultados de nuestro conjunto de datos.

5.1. Medidas de evaluación

A continuación se muestran la medida utlizada para determinar el rendimiento de los resultados obtenidos en la extracción de relaciones no taxonómicas.

5.1.1. Exactitud

La medida se refiere a la evaluación del sesgo de las predicciones, es decir, responde a la pregunta: *¿Cuál es el promedio de las predicciones correctas?* [49]. La fórmula de exactitud se presenta en la ecuación 5.1.

$$Exactitud = \frac{CantidadDeCasosCorrectos}{TotalDeCasos} \tag{5.1}$$

5.2. Conjunto de datos

El conjunto de datos se encuentra descrito en el trabajo de Zouaq [50], el cuál consta de dos ontologías: Inteligencia Artificial (IA) y Estándar e-

learning (SCORM). Cada ontología cuenta con un número determinado de documentos, tokens o palabras y vocabulario. Esta información se muesta en la tabla 5.1

Tabla 5.1: Datos del corpus de dominio

Ontología	Documentos	Tokens	Oraciones	Vocabulario
IA	8	10,805	460	1,510
SCORM	36	32,592	1642	2457

Para la ontología IA, durante el procesamiento de sus datos, se obtuvieron 276 conceptos relevantes y 61 relaciones no taxonómicas que serán extraídas y evaluadas con nuestro método propuesto. Mientras que la ontología de SCORM cuenta con un total de 1,461 conceptos relevantes y 759 relaciones no taxonómicas para evaluar. En la Tabla 5.2 se muestran los totales para cada ontología.

Tabla 5.2: Total de conceptos y relaciones no taxonómicas en la ontología de dominio de IA

Ontología	Conceptos	Relaciones no taxonómicas
IA	276	61
SCORM	1,461	759

5.3. Extracción y validación de conceptos

En esta sección se muestran los resultados del sistema propuesto para la extracción de conceptos del corpus de dominio. Nuestro sistema se basa en patrones regulares, como se vio en el capítulo 4 se realiza un proceso

de preprocesamiento en el cuál se expanden acrónimos y el texto es llevado a minúsculas para después someter el texto al etiquetado de parte del discurso (*POS*).

Con la ayuda de estas etiquetas se identifican los conceptos por medio de una gramática la cuál es formada por una sucesión de etiquetas del análisis *POS*, es decir, donde se encuentre algún número de adjetivos (JJ) seguido por algún número de sustantivos (NN) se podría encontrar un concepto con significado mayor. Sabiendo esto, en nuestro sistema se extendió esa sencilla gramática, la cuál se describe en la tabla 5.3 que se presenta a continuación:

Tabla 5.3: Gramática para extraer conceptos

Gramática	Ejemplo
<RB>?<JJ.*>*<VBN>?<VBG>?	
<VBD>?<NN.*>+<VBN>?<VBG>?	
<VBD>?(<IN>?<DT>?<VBG>?	
<JJ.*>*<NN.*>+<VBG>?)?	*rational utility based agent*

¿Qué es lo que nos dice esta gramática? con ayuda de la tabla 5.4, en donde se describe que representa cada etiqueta y símbolo. Conociendo el significado de cada etiqueta y símbolo, se puede notar que la gramática usada en nuestro sistema describe un concepto (frase nominal) de la forma:

Puede empezar o no con algún adverbio, seguido podría encontrarse con uno o varios adjetivos, continuando con encontrar o no un verbo conjugado (en inglés los verbos conjugados en gerundio, pasado o pasado participio, pueden ser considerados como sustantivos).

Después uno o varios sustantivos para nuevamente encontrarse o no con algún verbo conjugado y entonces encontrarse o no con un grupo de etiquetas que inicia con una preposición y nuevamente un formato parecido al ya descrito.

Este patrón que conforma la gramática fue creado analizando la estructura de una frase nominal en inglés, es por eso que los verbos son considerados y también algunos adverbios. Dentro de nuestro sistema se especifíca solo algunas preposiciones que pueden dar pie a un mismo significado de frase nominal. En el ejemplo de la tabla 5.3 puede verse una frase nominal en la que participa un verbo en pasado.

Hubo dos enfoques usados para la obtención de nuestros conceptos, el

Tabla 5.4: Significado de etiquetas y operadores para conceptos

Etiqueta	Significado
<RB>	Adverbio
<IN>	Preposición
<JJ>	Adjetivo
<NN>	Sustantivo
<VBN>	Verbo en pasado participio
<VBG>	Verbo en gerundio
<VBD>	verbo en pasado

Operador	Significado
*	de cero a más elementos
+	de uno a más elementos
?	cero o un elemento
( )	Agrupación de etiquetas
<Etiqueta.*>	agrupa todas las etiquetas de un tipo

primero (Sistema uno) usó esta gramática, mediante el uso de patrones regulares y el segundo (Sistema dos), en el cual además de usar los patrones regulares se usa *lematización* para usar solo la raíz de las palabras y haya mejor coincidencia, .

Se logró obtener en la ontología de IA con el primer sistema un total del 88 % de la medida de exactitud vista en la sección 5.1. Con el segundo sistema hubo una pequeña mejora del 2 %, notando que al eliminar el plural y las conjugaciones de las palabras se puede obtener un mejor resultado.

En el caso de la ontología SCORM el porcentaje de la medida de exactitud fue menor con ambos sistemas, el más bajo fue para el primer sistema, el cuál no cuenta con la técnica de *lematización*, en esta ontología se nota nuevamente el ligero incremento del 2 % para el segundo sistema que sí cuenta con la técnica de *lematización* como puede verse en la tabla 5.5.

	IA	SCORM
Sistemas	Exactitud	Exactitud
Primer Sistema	0.88	0.70
Segundo Sistema	0.90	0.72

Tabla 5.5: Resultados de la extracción de los conceptos de la ontología

Las puntuaciones demuestran que la gramática vista en la tabla 5.3, funciona mejor en la ontología de Inteligencia Artificial al contrario que la ontología de SCORM. Para ambos corpus de dominio se obtuvo un mejor resultado con el segundo sistema que aplica la *lematización* para volver a la raíz de las palabras, esto determina que pueda existir ciertas diferencias entre conjugaciones de palabras o inconvenientes entre palabras singulares y plurales.

Cabe mencionar que algunos de los conceptos que no se encontraron es debido a que se notó que en la ontología existen conceptos que realmente consisten en la idea principal de una oración, por lo que el porcentaje de exactitud podría llegar a ser mayor si no se tomaran este tipo de conceptos en cuenta.

5.4. Extracción y validación de relaciones no taxonómicas

Para la evaluación de relaciones no taxonómicas entre un par de conceptos, se consideraron las relaciones obtenidas del algoritmo 4.4 del capítulo 4.

Estas relaciones fueron evaluadas con las medidas de soporte 3.1 y confianza 3.2, gracias a estas medidas estadísticas se logra segmentar los resultados para obtener las relaciones que tengan un mayor grado de relación. Los resultados obtenidos se evaluaron con los de la ontología provista en [50] (IA) y SCORM, para después calcular la *exactitud*. Nuestro sistema extrajo un total de casos correctos de 44 para la ontología de IA y 599 para SCORM.

En la tabla 5.6 se muestran las puntuaciones de exactitud correspondientes a cada ontología, también se presentan los casos correctos en comparación, es decir, los resultados que coinciden entre nuestro sistema y el contenido de las ontologías.

	IA	SCORM
Coincidencias	41	599
Exactitud	0.72	0.79

Tabla 5.6: Puntuaciones de las ontologías

La observación de estos resultados provocó que se hiciera una investigación para comprender el comportamiento de nuestros resultados. En esta observación nos dimos cuenta que las estructuras de oración en el idioma inglés tienen diferentes formas, en las que el verbo que vincula a un sustantivo con un objeto puede aparacer al final de la oración, además de, como se mencionó en la sección anterior, existen conceptos en las ontologías que son formados de ideas principales de oraciones y estos conceptos están involucrados en las relaciones no taxónomicas de la ontología.

Conclusiones

En esta investigación se presenta una propuesta para la extracción y validación de conceptos y relaciones no taxonómicas de dos corpus de dominio inteligencia artificial y SCORM mediante preprocesamiento de datos y la técnica de reglas de asociación.

Esta técnica describe la probabilidad de que exista una relación entre objetos, en nuestro caso entre un par de conceptos y un verbo que los conecta en una oración en el corpus de dominio. Iniciando con técnicas de preprocesamiento para el corpus de dominio en el cual, en el caso de existir se expanden acrónimos, se lleva el texto a minúsculas y se eliminan símbolos.

Después del preprocesamiento sigue la extracción de los conceptos candidatos mediante una gramática expresada por patrones regulares en las que se especifica la estructura de la frase nominal deseada según la función que cumple cada palabra, esta gramática es complementada con una técnica llamada *lematización* en la cuál las palabras son llevadas a su forma raíz, esto con la finalidad de mejorar la precisión en la obtención de conceptos.

El primer sistema de extracción de conceptos obtuvo un menor rendimiento comparado con el segundo, este sistema solo usa la gramática o expresión regular, diseñada para encontrar estructuras de frases nominales, obteniendo un puntaje con la medida de *exactitud* de 0.88 para la ontología IA y 0.70 para la ontología SCORM. Esto nos indica que la gramática es buena pero puede mejorarse.

En el caso del segundo sistema el cuál contempla la gramática de patrones regulares, es completando con el uso de la técnica de *lematización*. Los resultados mejoraron para ambas ontologías, aunque es un aumento muy ligero de apenas 0.02, dando un así un total de *Exactitud* del 0.90 para IA y el 0.72 para SCORM.

Se pudó notar que la técnica de *lematización* tiene alguna ventaja pero hay algo que hace que no se llegue a mejores resultados pese a mejoras, una de las razones que se encontraron fue que las ontologías cuentan con conceptos o frases nominales, que realmente tienen función de *idea principal de la oración*, esto quiere decir que aún teniendo una gramática mejorada, no se podrían obtener estos conceptos, debido a que no están en el corpus, sino que son conceptos creados a partir de una oración.

Se puede concluir también, que entre más grande la ontología más de estos conceptos aparecen y esto se ve reflejado en los porcentajes que obtuvieron cada una de las ontologías, la ontología IA que tiene un corpus más pequeño que SCORM, obtuvo un 0.90, mientras que SCORM que cuenta con un corpus más extenso obtuvo 0.72.

En la sección de validación de relaciones no taxonómicas, se considera un sistema basado en la identificación de un verbo que conecte a un par de conceptos, este verbo debe cumplir con encontrarse entre este par de conceptos. La salida de sistema se comparó con las relaciones con las que cuenta cada ontología y los resultados fueron buenos. Se logró obtener el 72 % de relaciones no taxonómicas existentes en la ontología de IA [50] y 79 % con la ontología de SCORM, usando la medida de exactitud.

Se observó que al contar con un corpus pequeño, el soporte que hay entre los conceptos y el verbo es muy bajo, es decir, menor al 2 %, ya que esta medida representa la probabilidad de encontrar al par de conceptos y el verbo que los conecta dentro del dominio. Sin embargo, esto provoca que la confianza sea mayor al 50 %, ya que describe la probabilidad de que esta relación sea verdadera, es decir, que al encontrar los dos conceptos, el verbo se encuentre en la misma oración.

Además, cabe mencionar que algunas de las relaciones que no detectó el algoritmo, y existen en la ontología, son aquellas que la localización del verbo está al final de la oración y no de manera intermedia. En nuestra investigación se considera la estructura de oración en inglés, donde el verbo se encuentra entre los dos conceptos.

Como se dijo anteriormente, existen conceptos dentro de las ontologías que son formados por la idea principal de una oración, estos conceptos están involucrados en las relaciones no taxonómicas también, por lo que no fue posible su descubrimiento y el puntaje de comparación se ve afectado por la falta de estas relaciones y conceptos.

Como trabajo a futuro se propone implementar una propuesta de solución que identifique relaciones no taxonómicas en diferentes tipos de estructuras de la oración en inglés. Así mismo, aplicar el enfoque a otras ontologías y comparar los resultados. También se buscarán técnicas que permitan encontrar la idea principal de las oraciones y poder tomar estas como conceptos.

Referencias

[1] Tovar M., Pinto D., Montes A., Gonzalez G., and Vilarino D. Evaluacion de relaciones ontologicas en corpora de dominio restringido. *Computacion y sistemas*, 5:139–149, 2015. doi: 10.13053/Cys-19-11-954.

[2] Mehrnoush Shamsfard and Ahmad Abdollahzadeh Barforoush. The state of the art in ontology learning: a framework for comparison. *The Knowledge Engineering Review*, 18(4):293–316, 2003. doi: 10.1017/S0269888903000687.

[3] Mehrnoush Shamsfard and Ahmad Abdollahzadeh Barforoush. Learning ontologies from natural language texts. *International Journal of Human-Computer Studies*, 60:17–63, 2004. doi: 10.1016/j.ijhcs.2003.08.001.

[4] Mäedche A. and Staab S. Ontology learning for the semantic web. *IEEE Intelligent Systems*, 2:72–79, 2001. doi: 10.1109/5254.920602.

[5] Sanchez D. and Moreno A. Learning non-taxonomic relationships from web documents for domain ontology construction. *Data and Knowledge Engineering*, 64(3):600–623, 2008. doi: 10.1016/j.datak.2007.10.001.

[6] Eibe Frank, Gordon W. Paynter, Ian H. Witten, Carl Gutwin, and Craig G. Nevill-Manning. Domain-specific keyphrase extraction. *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, IJCAI 99*, page 668–673, 1999. URL `http://dl.acm.org/citation.cfm?id=646307.687591`.

[7] Ibrahim Abu El-Khair. Effects of stop words elimination for arabic information retrieval: a comparative study. *International Journal of Computing Information Sciences*, (3):119–133, 2006.

[8] Ken Barker and Nadia Cornacchia. Using noun phrase heads to extract document keyphrases. *En Proceedings of the 13th Biennial Conference of the Canadian Society on Computational Studies of Intelligence: Advances in Artificial Intelligence, AI 00*, pages 40–52, 2000. URL `http://dl.acm.org/citation.cfm?id=647461.726264`.

[9] Roberto Ortiz, David Pinto, Mireya Tovar, and Héctor Jiménez-Salazar. Buap: An unsupervised approach to automatic keyphrase extraction from scientific articles. *En Proceedings of the 5th International Workshop on Semantic Evaluation, Association for Computational Linguistics, Uppsala, Sweden*, page 174–177, 2010. URL `http://www.aclweb.org/anthology/S10-1037.`

[10] Stuart Rose, Dave Engel, Nick Cramer, and Wendy Cowley. Automatic keyword extraction from individual documents. *Text Mining: Applications and Theory*, pages 1–20, 2010.

[11] Alexander Gelbukh, Grigori Sidorov, Eduardo Lavin-Villa, and Liliana Chanona Hernandez. Automatic term extraction using log-likelihood based comparison with general reference corpus. *Springer Berlin Heidelberg, Berlin, Heidelberg*, page 248–255, 2010. doi: https://doi.org/10.1007/978-3-642-13881-2_26.

[12] Serra I. and Girardi R. A process for extracting non-taxonomic relationships of ontologies from text. *Intelligent Information Management*, pages 119–124, 2011.

[13] R. Srikant and R. Agrawal. Mining generalized association rules. *Future Generation Computer Systerms*, 13(2-3):161–180, 1997. doi: 10.1006/ijhc.1995.1081.

[14] Mäedche A. and Staab S. Mining non-taxonomic conceptual relations from tex. *IN: R.DIENG and O CORBY. EKAW-00 EUROPEAN KNOWLEDGE ACQUISITION WORKSHOP. OCTOBER 2-6, 2000, JUAN-LES-PINS*, 2000. doi: 10.1.1.41.4860.

[15] Weichselbraun A., Wohlgenannt G., Scharl A., Granitzer M., Neidhart T., and A. Juffinger. Discovery and evaluation of non-taxonomic relations in domain ontologies. *International Journal of Metadata, Semantics and Ontologies*, 3:212–222, 2009.

[16] M. Sanderson and W. B. Croft. Deriving concept hierarchies from text. *22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 206–213, 1999.

[17] Porzel R. and Malaka R. A task-based approach for ontology evaluation. *European Media Laboratory GmbH, SchlossWolfsbrunnenweg*, 2004.

[18] Sanchez D. and Moreno A. Discovering non-taxonomic relations from the web. *7th International Conference on Intelligent Data Engineering and Automated Learning*, pages 629–636, 2006.

[19] Kavalec M., Maedche E., and Svatek V. Discovery of lexical entries for non-taxonomic relations in ontology learning. *Proceedings of SOFSEM 2004: Theory and Practice of Computer Science*, pages 249–256, 2004. doi: 10.1.1.10.2718.

[20] Mäedche A. and Staab S. Mining non-taxonomic conceptual relations from tex. *IN: R.DIENG and O CORBY. EKAW-00 EUROPEAN KNOWLEDGE ACQUISITION WORKSHOP. OCTOBER 2-6, 2000, JUAN-LES-PINS*, 2000.

[21] Lutz M. Non-taxonomic relations in semantic service discovery and composition. *1st Óntology in ActionẂorkshop, in conjunction with 16th Conference on Software Engineering and Knowledge Engineering (SEKE 2004)*, pages 482–485, 2004.

[22] Nabila N., Basir N., and Mamat A. Synonymous non-taxonomic relations extraction. *ARPN Journal of Engineering and Applied Sciences*, 10:402–406, 2015.

[23] Villaverde J., Persson A., Godoy D., and Amandi A. Supporting the discovery and labeling of non-taxonomic relationships in ontology learning. *Elsevier*, 36:10288–10294, 2009. doi: 10.1016/j.eswa.2009.01.048.

[24] R. Gaizauskas and Y. Wilks. Information extraction: Beyond document retrival. *Journal of documentation*, 54(1):70–105, 1998. doi: 10.1108/eum0000000007162.2.

[25] J. Han and M. Kamber. *Data Mining: Concepts and Techniques.* 2000.

[26] C. T. Meadow. *Text Information retrieval Systems. San Diego.* 1993.

[27] Sosa Eduardo. Procesamiento de lenguaje natural: revisión del estado actual, bases teóricas y aplicaciones (parte i). *El profesional de la investigación. Revista internacional científica y profesional*, 1997. URL `http://www.elprofesionaldelainformacion.com/contenidos/1997/enero/procesamiento_del_lenguaje_natural_revisin_del_estado_actuall_bases_tericas_y_aplicaciones_parte_i_.html`.

[28] H. Gardner. La nueva ciencia de la mente: Historia de la revolución cognitiva. *Editorial Paidós*, 1985.

[29] M. Mateos and R. Ruíz. Procesamiento del lenguaje natural. *Depto. Ciencias de la computación e Inteligencia Artificial, Universidad de Sevilla*, 2012-2013. URL `https://www.cs.us.es/cursos/ia2/temas/tema-06.pdf`.

[30] P. Cimiano. *Ontology Learning and Population from Text. Algorithms, Evaluation and Application.* Springer, 2006.

[31] J. Lyons. *Introducción al lenguaje y a la lingüística.* Teide, 1984.

[32] J. Villena, R. Crespo, and J. García. Inteligencia en redes de comunicaciones. 2011-2012.

[33] Rakesh Agrawal, Tomasz Imielinski, and Arun Swami. Mining association rules between sets of items in large databases. *IN: PROCEEDINGS OF THE 1993 ACM SIGMOD INTERNATIONAL CONFERENCE ON MANAGEMENT OF DATA, WASHINGTON DC (USA)*, pages 207–216, 1993. doi: 10.1145/170035.170072.

[34] Gruber T. Toward principles for the design of ontologies used for knowledge sharing.in formal analysis in conceptual analysis and knowledge representation. *Kluwer*, 1993.

[35] Barry Smith. Ontology and an information systems. *Stanford Encyclopedia of Philoshopy*, 2004.

[36] Horrocks I., Fensel D., Broekstra J., Decker S., Erdmann M., Goble C., van Harmelen F., Klein M., Staab S., Studer R., and Motta E. Oil: The ontology inference layer. Reporte técnico, Vrije Universiteit Amsterdam, Faculty os Sciencies, 2000.

[37] Brickley D. and Guha R. Rdf vocabulary description language 1.0: Rdf schema. Reporte técnico, W3C Working Draft, 2002.

[38] Bechhofer S., van Harmelen F., Hendler J., Horrocks I., McGuiness D., Patel-Schneider P., and Stein P. Owl web ontology language reference. Reporte técnico, W3C Working Draft, 2004.

[39] R. Volz, D. van Oberle, S. Staab, and R. Studer. Ontolift prototype. Reporte técnico, Institute AIFB University of Karlshue, 2003.

[40] C. Fellbaum. Wordnet, a electronic lexical database, 1998.

[41] P. Buitelaar, T. Declerck, A. Franck, S. Racciopa, M. Kiesel, R. Sintek, M.And Engel, M. Romanelli, D. Sonntag, B. Loos, V. Micelli, R. Porzel, and P. Cimiano. *Linginfo: Design and applications of a model for the integration of linguistic information in ontologies.* In Proceedings of the OntoLex06 Workshop at LREC, 2006.

[42] Gruber T. Semantic lexicons. In *Ontologies and Lexical Knowledge Bases*, pages 10–24. In Simov, K. and Kiryakov, A. editors, 2000.

[43] T. Hall and J-P Stacey. *Python 3 for absolute beginners.* Apress. Springer Verlag, 2009.

[44] S. Bird, E. Klein, and E. Loper. *Natural Language Processing with Python– Analyzing Text with the Natural Language Toolkit.* NLTK Project, 2015.

[45] NLTK Project. Nltk 3.4 documentation: Natural language toolkit. url https://www.nltk.org/, 2018.

[46] Tibor Kiss and Jan Strunk. Unsupervised multilingual sentence boundary detection. *Computational Linguistics*, 32(4):485–525, 2006. doi: 10.1168/coli2006.32.4.485.15.

[47] S. Márquez. *La web semántica*. Santiago Máquez Solis, 2007. ISBN 978-1-84753-192-6.

[48] S.L. Expertos en Servicios de Consultoría Exes. Manual de xml. http://www.mundolinux.info/que-es-xml.htm.

[49] K. Vazques. Monitoreo de opiniones en redes sociales sobre la calidad del servicio. *Tesis Licenciatura, Benemérita Universidad Autónoma de Puebla.*, 2017.

[50] A. Zouaq, D. Gasevic, and M. Hatala. Linguistic patterns for information extraction in ontocmaps. *In: Proceedings of the 3rd International Conference on Ontology Patterns. 929. CEUR-WS.org*, pages 61–72, 2012.