Universidad Autónoma de Baja California
Facultad de Ciencias Químicas e Ingeniería
Maestría y Doctorado en Ciencias e Inteniería

"Aprendizaje de Ontologías a partir de corpus no estructurados usando técnicas de Deep Learning"

TESIS

Para Obtener el Grado de

Maestro en Ciencias

Presenta:

Raúl Ignacio Navarro Almanza

Bajo la dirección de:

Dr. Guillermo Licea Sandoval

Co-dirigido por:

Dr. J. Reyes Juárez Ramírez

Tijuana, Baja California, México

Diciembre del 2017

Con amor para mis padres y Giselle, mi motivación para seguir adelante...

Universidad Autónoma de Baja California

FACULTAD DE CIENCIAS QUÍMICAS E INGENIERÍA

COORDINACIÓN DE POSGRADO E INVESTIGACIÓN

FOLIO No. 231

Tijuana, B. C., a 29 de noviembre de 2017

C. Raúl Ignacio Navarro Almanza
Pasante de: Maestro en Ciencias
Presente

El tema de trabajo y/o tesis para su examen profesional, en la Opción TESIS

Es propuesto, por los C. Dres. Guillermo Licea Sandoval y J. Reyes Juárez Ramírez

Quienes serán los responsables de la calidad de trabajo que usted presente, referido al tema "Aprendizaje de Ontologías a partir de corpus no estructurados usando técnicas de Deep Learning"

el cual deberá usted desarrollar, de acuerdo con el siguiente orden:

I.- INTRODUCCIÓN
II.- FUNDAMENTOS TEÓRICOS
III.- ANÁLISIS DE RELACIONES SEMÁNTICAS EN VECTORES DE PALABRAS EMBEBIDAS
IV.- METODOLOGÍA PROPUESTA PARA INFERENCIA DE RELACIONES SEMÁNTICAS
V.- CASO DE ESTUDIO: ONTOLOGÍA DEL PARADIGMA ORIENTADO A OBJETOS
VI.- CONCLUSIONES Y TRABAJO FUTURO

Dr. Guillermo Licea Sandoval
Director de Tesis

Dr. José Luis González Vázquez
Sub-Director Secretario

FACULTAD DE CIENCIAS QUÍMICAS E INGENIERÍA

Dr. J. Reyes Juárez Ramírez
Co-Director de Tesis

Dr. Luis Enrique Palafox Maestre
Director

Agradecimientos

Deseo agradecer a mis directores de tesis Dr. Licea Sandoval y Dr. Reyes Juárez por su guía durante el transcurso de mi estancia, a mis maestros Dra. Olivia Mendoza, Dr. Juan Ramón Castro, Dr. Mauricio Sánchez, Dra. Carelia Gaxiola por su dedicación y esfuerzo para enseñarme.

A la Universidad Autónoma de Baja California y al Concejo Nacional de Ciencia y Tecnología por su apoyo para cumplir mis metas.

A mis compañeros Alan, Andrés, Ángeles, Patricia, Sergio, Samantha y Carlos por tan buenas vivencias haciendo mi estancia muy amena.

De forma muy especial a mi familia por su apoyo incondicional, sobre todo, por su paciencia en todo este tiempo.

Raúl Ignacio Navarro Almanza

Resumen

Las ontologías son artefactos computacionales para representación de conocimiento a través de clases y relaciones. La construcción de estas bases de conocimiento requiere un gran cantidad de esfuerzo y tiempo debido a la necesidad de expertos del dominio e ingenieros de conocimiento para la adquisición y modelado del conocimiento. El aprendizaje de ontologías tiene como propósito lograr la construcción automática de las ontologías a través de diversas fuentes de información, tales como bases de datos, páginas web, archivos multimedia, corpus tanto estructurado como no estructurado, entre otros.

La propuesta principal de este trabajo consiste en la creación de una metodología para la construcción automática de ontologías, cuyo caso práctico se realizó sobre el dominio de la programación orientada a objetos con el fin de ser implementado en un sistema tutor inteligente. La aportación mas importante de esta metodología es la falta de intervención humana para la construcción de características, esto a través del uso de técnicas de aprendizaje profundo. La metodología contempla la implementación de transferencia de aprendizaje mediante técnicas de adaptación de dominio y representación de datos en espacios embebidas, por lo que no se requieren datos estructurados de un dominio específico para el entrenamiento de un modelo de aprendizaje. Los resultados de esta metodología son prominentes considerando la falta de intervención humana e ingeniería de características.

Abstract

Ontologies are computational artifacts to represent knowledge through classes and relations between them. Those knowledge bases require huge human effort to be constructed due to the need for domain experts and knowledge engineers. Ontology Learning aims to automatically build ontologies from data that can be from multimedia, web pages, databases, unstructured text, and so on. In this work, we propose a methodology to automatically build an ontology to represent concepts map of subjects to be used in academic context. The main contribution of this methodology is that does not require handcrafted features by using Deep Learning techniques to identify taxonomic and semantic relations between concepts of some specific domain. Also, due to the implementation of transfer learning is not needed of specific domain dataset, the relation classification model is trained with Wikipedia and WordNet by distant supervision technique and the knowledge is transferred to a specific domain by domain adaptation techniques in word embedding space. The results of this approach are promising considering the lack of human intervention and feature engineering.

Índice general

Índice de figuras

Índice de tablas

Lista de Acrónimos

AHS Adaptive Hypermedia System.

AKE Automatic Keyphrase Extraction.

BGRU Bidirectional Gated Recurrent Unit.

BLSTM Bidirectional Long Short Term Memory.

BPTT Backpropagation through time.

CBOW Continous Bag of Words.

CNN Convolutional Neural Network.

DNC Differentiable Neural Computer.

ITS Intelligent Tutoring System.

LSTM Long Short Term Memory.

NLP Natural Language Procesing.

OL Ontology Learning.

OWL Ontology Web Language.

RDF Resource Description Framework.

ReLU Learning Management System.

ReLU Rectified Linear Unit.

RNN Recurrent Neural Network.

WE Word Embeddings.

Capítulo 1

Introducción

Los estudiantes que reciben una instrucción personalizada por un tutor, quien imparte conocimiento, tienen el potencial de lograr un mejor aprendizaje que aquellos que reciben una instrucción tradicional, es decir varios tutorados por tutor (BLOOM, 1984). El éxito de la instrucción personalizada se debe principalmente a la retroalimentación existente entre tutor y el alumno. Usualmente al haber un número mayor de alumnos, la cantidad de participación de éstos es menor, por lo tanto no se logra el nivel de retroalimentación apropiado.
En México actualmente la relación de cantidad de alumnos por clase (en todos los niveles educativos) es entre 25 y 33 (INEE, 2015). Por ello es conveniente la implementación de sistemas computacionales de apoyo en la enseñanza para incrementar la calidad de aprendizaje de los alumnos.

Los Sistemas de Tutoría Inteligente (ITS, por sus siglas en inglés de *Intelligent Tutoring System*) tienen como propósito emular el comportamiento de un tutor humano. Las interrogantes básicas que deben responder los ITS son las siguientes: qué enseñar, cómo enseñar y a quién enseñar. Éstos sistemas toman en cuenta características del alumno para lograr una mejor adaptación del sistema al estudiante en base a estilos de aprendizaje, estilo cognitivo,

objetivo de aprendizaje, entre otros (Gligora *et al.*, 2015), lo cual permite una instrucción más personalizada de acuerdo a las necesidades del alumno.

La arquitectura de los ITS está conformada por al menos cuatro módulos que se han presentado como estructura de los ITS, los cuales son: el modelo de dominio, el modulo del tutor, el modulo del estudiante y la interfaz de usuario. En el modelo de dominio se representa el conocimiento del dominio de la aplicación del ITS; en el modelo tutor se definen las reglas pedagógicas; en el modelo del estudiante se representa el estado del alumno, es decir todas las características que se consideren para la adaptación del sistema ITS; por último, la interfaz de usuario, en la cual el alumno interactúa con el sistema.

Actualmente el desarrollo de un modelo de conocimiento de dominio en los ITS es usualmente hecho a mano lo que conlleva una gran cantidad de tiempo y esfuerzo. Además se requiere la intervención de un experto que colabore en el proceso de la adquisición del conocimiento. Usualmente el modelo de conocimiento del dominio no es compartible entre otros ITS y están restringidos al dominio para el cual fue creado.

La mayoría de los sistemas de aprendizaje fallan en crear un mapa curricular apropiado para cada alumno, por lo que es necesario la construcción de un modelo de dominio que tome en cuenta también la ruta de enseñanza apropiada para cada individuo (Chen, 2009).

El uso de ontologías para representar del modelo de conocimiento del dominio ha sido ampliamente recomendado debido a su estándar de formalización. Esta formalización hace posible poder compartir y reusar las ontologías entre cualquier ambiente que opere sobre ontologías (Mizoguchi y Canada, 2000). Una ventaja más por las cuales se ha impulsado el uso de las ontologías es que permiten hacer procesos de inferencia para adquirir nuevo conocimiento a partir del existente. Al automatizar el desarrollo de ontologías utilizables por los ITS se podría extender el uso de éstos al requerir menos recursos para su desarrollo, con

lo cual, se facilitaría el acceso a estos sistemas a los alumnos.

En las propuestas actuales se han empleado técnicas de NLP para la extracción de términos relevantes, así como las relaciones entre estos (Larrañaga *et al.*, 2014). Una de las oportunidades de mejora de los métodos actuales es que se requiere la intervención de un científico de datos para el procesamiento y tratamiento de los datos. El empleo de técnicas de NLP se basan generalmente al descubrimiento de patrones entre las sentencias, por ejemplo, determinar que una relación taxonómica esta definida por la frase "es un", es decir X tiene una relación taxonómica con Y si en la oración tiene la siguiente estructura X *es un* Y. Sin embargo, este tipo de técnicas están limitados al tipo de lenguaje empleado e inclusive del idioma y forma de expresión.

Las principales ventajas de la aplicación de Aprendizaje Profundo es el descubrimiento automático de características lo cual reduce la intervención del humano para fabricación de características. Este tipo de técnicas, tienen un gran éxito en tipos de datos complejos como lo son: texto, audio, imágenes, vídeo, entre otros; por el contexto de la tesis, este trabajo se enfocará en aquellas que han superado el estado del arte en procesos de procesamiento de información textual.

Las aportaciones más significativas de DL en el procesamiento de texto, recae en la representación de palabras, ya que tradicionalmente se representaban en un vector de ceros y un uno en la posición que representaba cada palabra. Este tipo de representaciones es llamado comúnmente *one-hot-encoding* y el tamaño de cada vector (donde cada uno representa una palabra) tiene un tamaño igual al tamaño de palabras contenidas en el vocabulario, donde usualmente es de decenas de miles e incluso millones.

Dado los problemas identificados en la representación *one-hot-encoding* se han propuesto representaciones de palabras en un espacio continuo y de baja dimensionalidad, llamados palabras embebidas (WE, por sus siglas en inglés de *Word Embeddings*) , en los cuales se

hace uso de modelos de RNA para la presentación de las palabras en este nuevo espacio. Esta nueva representación de palabras se logra capturar ciertas regularidades semánticas y sintácticas entre ellas, en el trabajo de Fu *et al.* (2015) se evidencia que existen regularidades no lineales entre estos vectores, por ejemplo, la bien conocida expresión $v(\texttt{king}) - v(\texttt{queen}) \equiv v(\texttt{man}) - v(\texttt{woman})$, no se puede generalizar la relación semántica hiperonimia (*es-un*) de la siguiente forma $v(\texttt{king}) - v(\texttt{man})$ y con ella inferir par de entidades con ésta relación semántica por todo el espacio. Dado que las regularidades lineales aparecen en regiones locales, se desprende una de las hipótesis que a través de un clasificador no lineal se podrían detectar, en todo el espacio, las regularidades semánticas que aparecen solo en regiones locales.

Diversos trabajos se han dedicado a emplear la representación de palabras embebidas en varias tareas empleadas en el procesamiento de lenguaje natural y han superado el estado del arte en varios de ellos, tales como Fu *et al.* (2014, 2015); Takase *et al.* (2016); Hyland *et al.* (2016); Fan *et al.* (2016). Por lo que éste será el enfoque que dictará los siguientes procesos de la investigación.

En el contexto del Aprendizaje profundo, en este trabajo se propone una metodología para la automatización de la construcción de ontologías cuyo objetivo principal es el de omitir, en medida de las posibilidades, la intervención humana. Este propósito se busca lograr a través de la implementación de técnicas de Aprendizaje Profundo. De tal forma que se pretende contribuir particularmente al área de Aprendizaje de Ontologías y al área de Construcción Automática de Bases de Conocimiento; esto en el contexto del caso de estudio, el cual implica un método para facilitar el proceso construcción del Modelo de Dominio de los Sistemas Tutores Inteligentes.

En lo sucesivo de este capítulo se presenta de manera formal el planteamiento del problema que en este trabajo se pretende solucionar, seguido de las hipótesis planteadas, posteriormente, los objetivos y metas establecidas para este trabajo. Por último se describe la estructura

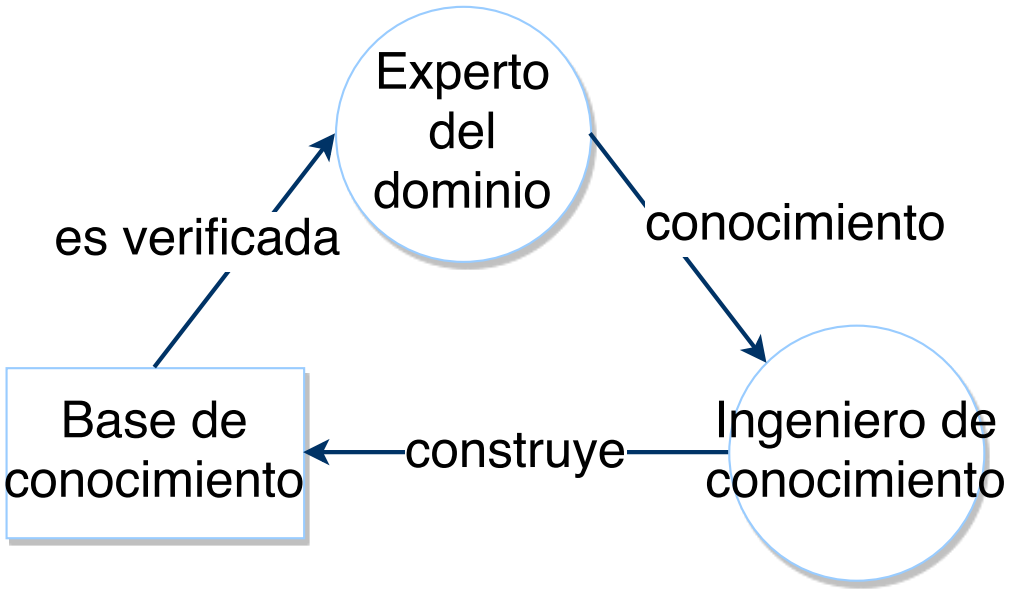

Figura 1.1: Proceso ingeniería de conocimiento

de este documento de las capítulos posteriores.

1.1. Planteamiento del problema

El proceso de obtención de información para construcción de una ontología es un proceso muy costoso. Se requiere de gran cantidad de tiempo y esfuerzo por parte de los expertos en el dominio, así como los ingenieros de conocimiento para construir la estructura que representará el conocimiento dado por el experto. En la figura 1.1 se muestra el proceso general implicado en la ingeniería de conocimiento para la construcción de una base de conocimientos.

La problemática se encuentra en la etapa de adquisición y formalización del conocimiento adquirido y radica en la falta de sistemas para para automatizar la construcción del la base de conocimiento del dominio de los ITS. Los problemas específicos expuestos en Chakraborty *et al.* (2010) son los siguientes:

1. La mayoría de las base de conocimiento en los ITS están restringidas a un solo dominio,

cambiarlas o reorganizarlas para enseñar diferentes asignaturas son extremadamente complicadas o requieren intervención de los desarrolladores.

2. Un solo método de enseñanza empleado por el modelo de tutoría no funcionaría en un ambiente donde se tenga más de un dominio específico.

3. Falta de interoperatividad de los ITS, impidiendo compartir la base de conocimiento en los ITS para lograr el desarrollo en menor en tiempo, siendo este uno de los mayores problemas implícitos en el desarrollo en estos sistemas.

Principalmente la incapacidad de que los modelos de dominio sean compartibles y reusables se debe tanto a la diferencia semántica que existe entre sistemas como en el método de representación del conocimiento del dominio.

Por los motivos anteriormente expuestos se propone el desarrollo de un sistema que automatice el proceso de creación de ontologías a partir de textos no estructurados que contengan información del dominio a modelar, empleando métodos de extracción de información en texto. De esta forma se crearía una estructura que representaría el conocimiento del dominio compartible y reusable en otros sistemas que puedan realizar operaciones sobre ontologías. Por otra parte, al automatizar el proceso de adquisición de información se reduciría el tiempo al no tener que requerir en todo el proceso del ingeniero de conocimiento ni al experto para la adquisición de información.

Existen trabajos relacionados en la extracción de información a partir de texto en sitios web, presentaciones en formato Power Point (Atapattu *et al.*, 2012), materiales didácticos (Gantayat e Iyer, 2011). En este trabajo se propone desarrollar un sistema que genere ontologías a partir libros de texto, extrayendo relaciones semánticas: hiperónimo, hipónimo, holónimo y merónimo.

La problemática se centra en la necesidad de análisis de la información y preparación de

los datos a través de la ingeniería de características, lo cual conlleva tiempo y esfuerzo para su realización. Una de las metas de este trabajo es prescindir de la intervención humana en la preparación de datos aunque se cambie el dominio de aplicación del sistema.

Uno de los principales enfoques de esta metodología es ver el problema de extracción de relaciones como un problema de clasificación, en donde la entrada al modelo de aprendizaje automático se otorga las palabras en las cuales de desea conocer su relación y el contexto en donde se encuentran ambos términos en el corpus. Esto aplicando técnicas de DL con el propósito de generalizar la metodología a cualquier dominio, con especial atención a dominios donde no exista abundante información estructurada y estén conformados por tópicos muy especializados. En el caso de este trabajo se acotará en la generación automática de la representación de conocimiento del dominio del paradigma orientado a objetos, en donde únicamente se considerará información no estructurada como lo son libros de texto. Contemplando que el dominio que se desea modelar puede cambiar y pudiera no ser factible utilizar libros de texto, no se consideró alguna información como el título o tabla de contenidos que pudiera ser automáticamente extraído.

1.2. Hipótesis

General

H1: Empleando modelos no lineales de aprendizaje automático es posible detectar relaciones semánticas entre palabras en un corpus.

H2: Mediante transferencia de aprendizaje es posible inferir relaciones semánticas en un dominio específico a partir de un dominio genérico.

1.3. Objetivos

Generales

1. Automatizar la adquisición de conocimiento de un dominio de conocimiento mediante extracción de conceptos y sus relaciones del tipo: hiperónimo, hipónimo, holónimo y merónimo; utilizando técnicas de aprendizaje de ontologías y minería de texto para identificar conceptos relevantes en un dominio y sus relaciones.

Específicos

1. Proponer una metodología para la creación automática de ontologías a través de corpus no estructurados de un dominio específico.
2. Diseño de sistema de inferencia para identificación de relaciones entre conceptos en un dominio específico.

1.4. Metas

1. Realización de análisis de captura de relaciones semánticas en el espacio de palabras embebidos.

2. Propuesta de metodología para aplicar transferencia de conocimiento de un modelo de clasificación de relaciones semánticas, el cual fue entrenado con un corpus de dominio genérico y empleado en un dominio especializado.

3. Desarrollo del sistema minería de texto para construcción automática de ontologías.

1.5. Contenido del documento

El resto del documento de tesis está dividido en cuatro capítulos, los cuales de describen a continuación:

En el capítulo 2 (Marco teórico), se presentan los temas relacionados con la problemática y a la solución propuesta, en los cuales se encuentran los siguientes: Recuperación de información, Relaciones Semánticas, Sistemas de enseñanza asistido por computadora y Aprendizaje profundo.

En el capítulo 3 titulado *Análisis de relaciones semánticas capturadas en vectores de palabras embebidas* en el cual se describe y realiza el experimento para comprobar la primer hipótesis *Empleando modelos no lineales de aprendizaje automático es posible detectar relaciones semánticas entre palabras en un corpus*, el cual contiene secciones que describen los trabajos relacionados, la metodología que se siguió para la comprobación de la hipótesis, configuración del experimento y resultados.

En el capítulo 4 titulado *Inferencia de relaciones semánticas en vectores de palabras embebidas* se describe la metodología empleada para la comprobación de la hipótesis *Mediante transferencia de aprendizaje es posible inferir relaciones semánticas en un dominio específico a partir de un dominio genérico*, en el cual se presenta los trabajos relacionados, la experimentación, resultados y una discusión de los mismos.

Por último, en el capítulo 6 llamado *Conclusiones y trabajo futuro* se presentan las implicaciones de los resultados obtenidos en la metodología propuesta para automatizar la construcción de ontologías a partir de corpus no estructurado empleando técnicas de aprendizaje profundo disminuyendo la intervención del humano.

Capítulo 2

Fundamentos teóricos

En esta sección se presentan un breve descripción de los conceptos utilizados posteriormente para explicación del estado del arte referente al tema de la problemática expuesta. Además se hace un análisis en la sección de antecedentes de los trabajos relacionados que han propuesto soluciones para resolver el la problemática identificada.

2.1. Recuperación de información

El campo de la recuperación de información es la ciencia de la búsqueda de información en cualquier fuente digital con el fin de obtener información relevante de una colección de fuentes de información (Palka *et al.*, 2008).

2.1.1. Métricas utilizadas en recuperación de información

Se ilustra en la tabla 2.1 los posibles valores existentes en la recuperación de información sin clasificar.

Tabla 2.1: Posibles valores en la recuperación de información

	Relevante	**No relevante**
Recuperado	verdadero positivo (VP)	falso positivo (FP)
No recuperado	falso negativo (FN)	verdadero negativo (VN)

Algunas de las métricas utilizadas para medir la efectividad en la recuperación de información son la precisión y la exhaustividad, estas mediciones refieren a que tan bien el sistema recuperó información relevante.
Se define precisión a la relación entre la cantidad de elementos relevantes recuperados sobre el total de los elementos recuperados (ecuación 2.1).

$$P = \frac{\#(Elementos\ recuperados\ relevantes)}{\#(Total\ de\ elementos\ recuperados)} = \frac{VP}{VP + FP} \tag{2.1}$$

Se define exhaustividad a la relación entre la cantidad de elementos recuperados relevantes sobre la cantidad de elementos relevantes (ecuación 2.2).

$$E = \frac{\#(Elementos\ recuperados\ relevantes)}{\#(Total\ de\ elementos\ relevantes)} = \frac{VP}{VP + FN} \tag{2.2}$$

Una medida que relaciona la precisión con la exhaustividad es la medida F, que es un promedio armónico ponderado de la precisión y la exhaustividad (ecuación 2.3) siendo $\alpha \in [0, 1]$ y por lo tanto $B^2 \in [0, \infty]$.

$$F = \frac{1}{\alpha\frac{1}{P} + (1-\alpha)\frac{1}{E}} = \frac{(\beta^2+1)P \cdot E}{\beta^2 P + E} \text{ donde } \beta^2 = \frac{1-\alpha}{\alpha} \tag{2.3}$$

2.2. Relaciones Semánticas

Una relación semántica comprende aquellas relaciones que se establecen entre las palabras. Es la relación que existe entre dos elementos con significado.

2.2.1. Relaciones hiperonomia-hiponomia

Es la relación estándar subtipo/supertipo que frecuentemente se utiliza en modelado de datos. A es-un B, cuando A es referido como un tipo de entidad específica y B el tipo genérico. Por ejemplo las siguientes oraciones corresponden a relaciones taxonómicas *gerente es-un empleado*, *los carros son un tipo de vehículo*; en estas oraciones gerente y carros pertenecen al conjunto de hipónimos, por lo tanto, empleado y vehículo pertenecen al conjunto de hiperónimos.

Se define el término co-hipónimo cuando dos hipónimos comparten un mismo hiperónimo, por ejemplo el *carro, barco* y *tren* son co-hipónimos al tener como hiperónimo común *vehículo*.

Propiedades

- Relación no conmutativa. Las relaciones de hiponomia son asimétricas, es decir, si X es hipónimo de Y, entonces, Y no es hipónimo de X. $(R_{xy} \wedge R_{yx}) \rightarrow x = y$.
- Relación transitiva. Si X es hipónimo de Y, y Y es hipónimo de Z, por lo tanto, X es hipónimo de Z. $(R_{xy} \wedge R_{yz}) \rightarrow R_{xz}$.

2.2.2. Relaciones holonomia-meronomia

La relación de meronomia ocurre cuando un elemento es parte de otro, por ejemplo *motor es parte-de carro*. La relación de holonomia es la relación opuesta a meronomia. Siendo *motor*

el merónimo y *carro* el holónimo.

Propiedades

- Relación no conmutativa. Las relaciones de holonomia son asimétricas, es decir, si X es holónimo de Y, entonces, Y no es holónimo de X. $(R_{xy} \wedge R_{yx}) \rightarrow x = y$.
- Relación transitiva. Si X es holónimo de Y, y Y es holónimo de Z, por lo tanto, X es holónimo de Z. $(R_{xy} \wedge R_{yz}) \rightarrow R_{xz}$.

2.3. Ontologías

La Ontología es una rama de la metafísica y denota la investigación filosófica de la existencia. Le atañe la pregunta fundamental "¿Qué clases de entidades existen?" y conduce a estudiar categorías generales de todas las entidades que existen. El estudio de ésta se remonta a los tiempos de Aristóteles. Diversas ciencias se han beneficiado del enfoque de la categorización ontológica, una de las cuales ha sido las ciencias computacionales, en los sistemas de información basados en conocimiento. Cuando es aplicada a un dominio limitado de interés en un ámbito de un escenario concreto, la ontología puede ser restringida a contemplar un subconjunto del mundo.

Una ontología es un artefacto computacional que recopila conocimiento acerca de un dominio de una forma procesable por la computadora para hacerlo disponible a los sistemas de información. Gruber (1995) define una ontología como: "una especificación formal explícita de una conceptualización compartida de un dominio de interés", lo que en otras palabras es, una representación detallada de características, precisa y explícita de una percepción abstracta y simplificada de un conocimiento consensuado.

2.3.1. Características de las ontologías

A partir de la definición propuesta por Gruber, Grimm *et al.* (2011) hace explícito las características de una ontología como una especificación del conocimiento del dominio. A continuación se desarrollan las características identificadas :

- **Formalidad**. Una ontología es expresada en un lenguaje de representación de conocimiento basada en los cimientos de la semántica formal. Esto asegura que la especificación del conocimiento del dominio en una ontología es computacionalmente procesable y está siendo interpretada de una forma bien definida. Técnicas de representación simbólica usualmente son construidas sobre los principios de la lógica.

- **Explicitud**. Una ontología representa conocimiento explícitamente para hacerlo accesible para las máquinas. Aspectos que no son explícitamente incluidos en la ontología no son parte de la conceptualización computable, sin embargo los humanos deben de tomarlos por sentido común.

- **Consenso**. Como se definió anteriormente una ontología es una conceptualización acordada entre personas de una comunidad. Entre mayor sea la comunidad, más difícil es llegar a compartir un acuerdo común de una misma conceptualización. En este sentido la construcción de una ontología está asociado con un proceso social para alcanzar un consenso.

- **Conceptualidad**. Una ontología especifica un conocimiento de forma conceptual en términos de símbolos conceptuales que pueden ser intuitivamente captado por humanos, así como su correspondencia con los elementos en sus modelos mentales. Una ontología

describe una conceptualización en términos generales y no solo captura estados particulares de situaciones. En lugar de crear afirmaciones acerca de una situación específica incluyendo individuos particulares, una ontología trata de cubrir tantas situaciones en manera de lo posible que potencialmente puedan ocurrir.

- **Especificación del dominio**. Las especificaciones en una ontología están limitadas al conocimiento acerca de un dominio particular de interés. Entre más pequeña sea el ámbito de un dominio para la ontología, el ingeniero de la ontología se puede enfocar más en capturar los detalles en este dominio en lugar de cubrir un mayor rango de tópicos relacionados.

Las **características esenciales** comunes en la mayoría de las ontologías usadas en sistemas de la información descritas por Grimm, son las siguientes:

- **Interrelación**. Las relaciones entre los conceptos dan lugar a una interconexión, enriqueciendo los nodos conceptuales con una estructura.

- **Instanciación**. Es el mecanismo que hace posible la distinción entre individuos concretos del dominio de interés y las categorías más generales que agrupan objetos con algunas características comunes. Esto se efectúa asignando objetos individuales a clases como sus instancias.

- **Subsunción**. Es la forma más común de interrelacionar nodos conceptuales, expresados como relaciones "es un" que indica la noción de generalización y especialización. Subsunción es el mecanismo detrás de la característica de jerarquía de herencias prevalentes en un modelo conceptual.

- **Exclusión**. Es una forma más sofisticada de referirse a la representación del estado conocimiento negativo en forma de clase de exclusión, evitando que dos nodos concep-

tuales generales se solapen en sus extensiones. Esta forma de conocimiento negativo es encontrado en lenguajes de ontologías expresivas para permitir la negación.

- **Axiomatización**. La sola relación entre nodos conceptuales en muchas ocasiones no son suficiente para expresar un conocimiento amplio o enriquecido. A veces se requiere de declaraciones complejas acerca del dominio de interés, éstas representan el concepto de axioma. La subsunción, exclusión e instanciación son formas básicas de axiomatización. Muchos de los lenguajes de ontologías permiten la formulación de declaraciones mas complejas en forma de axiomas generales.

- **Atribución**. La inclusión de declaraciones acerca de tipos de datos y sus valores es una característica en común en varios lenguajes de ontologías, que representan la atribución de nodos conceptuales por cadenas de texto, números, entre otros tipos de datos. Los tipos de datos y valores son comúnmente una característica indispensable en muchas aplicaciones de ontologías.

2.3.2. Modelo formal de ontología

El modelo formal de ontología (expresión 2.4) definido por Grimm, toma en consideran las características esenciales. Esta formalización es lo suficientemente expresiva para encajar en la mayoría de los formalismos y lenguajes de representación de conocimiento comunes, usados para ontologías en la Web Semántica.

$$\mathcal{O} = (\mathbf{S}, \mathbf{A}) \tag{2.4}$$

Una ontología es una tupla 2.4 de una signatura $\mathbf{S}$ y un conjunto de axiomas $\mathbf{A}$. Los elementos en la signatura, que comprenden las entidades conceptuales usadas para la representación de conocimiento, son separados de las declaraciones actuales que usan estos elementos para

expresar conocimiento acerca del dominio en forma de axiomas. Además las entidades en una signatura de una ontología forma el vocabulario a utilizar por los ingenieros de conocimiento para la formulación de axiomas. Los axiomas por si mismos son expresados en un lenguaje de ontología específico o en un formalismo de representación de conocimiento, también, los predicados de lógica de primer orden son usados para unificar de forma sintáctica la expresión de axiomas.

$$\mathbf{S} = C \bigcup I \bigcup P \tag{2.5}$$

La signatura (expresión 2.5) está compuesta de diversos conjuntos de clases C, instancias I, y propiedades P. Las *instancias* se asignan a nodos individuales en una red semántica, representando objetos individuales del dominio de interés. Las *clases* se asignan a los nodos conceptuales, agrupando instancias que tienen ciertas características en común. Las *propiedades* se asignan a los enlaces en las redes semánticas para expresar las relaciones entre instancias y clases. Los axiomas expresados en predicados de lógica de primer orden, las clases corresponden a predicados un-arios, propiedades a predicados binarios, e instancias a símbolos constantes, todos los que ocurren dentro de los axiomas.

$$C = \mathcal{C} \bigcup \mathcal{D}, I = \mathcal{I} \bigcup \mathcal{V}, P = \mathcal{R} \bigcup \mathcal{T} \tag{2.6}$$

Las entidades de signatura (descritos en la expresión 2.6) también están compuestas por conjuntos de *conceptos* $\mathcal{C}$ y *tipos de datos* $\mathcal{D}$, *individuos* $\mathcal{I}$ y *valores* $\mathcal{V}$, por último, *relaciones* $\mathcal{R}$ y *atributos* $\mathcal{T}$. En éste nivel de distinción, los participantes de instanciación, es decir, clases e instancias, son separadas. Los conceptos abstractos son diferenciados de los tipos de datos como cadena de caracteres o enteros, mientras que la abstracción de objetos individuales del dominio son separados por completo de los valores de datos concretos. De acuerdo a lo anterior, las propiedades son separadas en relaciones que involucran solo objetos abstractos del dominio y los atributos de las clases solo involucran valores de datos. Esta distinción

entre objetos abstractos del dominio y atributos de estos objetos, es contemplado en varias formas de modelado en artefactos de las Ciencias de la Computación, como en el modelado entidad-relación en bases de datos, paradigma orientado a objetos, lenguajes de ontología como OWL, entre otros.

Axiomas específicos

De acuerdo a las características esenciales de una ontología, algunos tipos de axiomas son comunes en la mayoría de los lenguajes de ontologías. En la tabla (2.2) se muestran los tipos de notaciones de axiomas:

- **Instanciación**. Un axioma de instanciación establece una instancia a una clase.
- **Afirmación**. Un axioma de afirmación establece dos instancias por el significado de una propiedad.
- **Subsunción**. Un axioma de subsunción para dos clases establece que cualquier instancia de la clase del subsumido es también una instancia de la clase subsunción, mientras para dos propiedades, establece que cualquiera de las instancias conectadas por una propiedad subsumida están también conectados por una subsunción.
- **Dominio**. Un axioma de dominio de una propiedad y una clase establece que para cualquier conexión de las dos instancias por dicha propiedad, el elemento fuente es una instancia de la clase de dominio.
- **Rango**. Un axioma de rango de una propiedad y una clase establece que para cualquier conexión de dos instancias por medio del elemento objetivo es una instancia de la clase rango.
- **Disyunción**. Un axioma de disyunción de dos clases establece que no puede haber una instancia de una clase que pueda ser instancia de alguna otra clase, por lo tanto, las clases se excluyen unas a otras.

Tabla 2.2: Axiomas comunes en formalismos de ontología

Axioma	Notación	Expresión en lógica de primer orden
Instanciación	$\alpha_{\wedge}(i, C)$	$[C(i)], i \in I, C \in \mathtt{C}$
Afirmación	$\alpha_{\rightarrow}(i_1, p, i_2)$	$[p(i_1, i_2)], i_1, i_2 \in I.p \in \mathtt{P}$
Subsunción	$\alpha_{\Delta}(E_1, E_2)$	$[\forall x : E_1(x) \rightarrow E_2(x)], E_1, E_2 \in \mathtt{C} \bigcup \mathtt{P}$
Dominio	$\alpha_{D\rightarrow}(p, D)$	$[\forall x, y : p(x, y) \rightarrow D(x)], p \in \mathtt{P}, D \in \mathtt{C}$
Rango	$\alpha_{\rightarrow R}(p, R)$	$[\forall x, y : p(x, y) \rightarrow R(y)], p \in \mathtt{P}, R \in \mathtt{C}$
Disyunción	$\alpha_{\oplus}(C_1, C_2)$	$[\forall x : C_1(x) \bigwedge C_2(x) \rightarrow \perp], C_1, C_2 \in \mathtt{C}$

En general, la mayoría de los lenguajes de ontologías permiten la formulación de arbitraria de complejos y sofisticados axiomas para expresar el conocimiento del dominio, los conjuntos básicos de axiomas anteriormente mencionados cubren la mayoría de lo que se encuentra típicamente en los axiomas de las ontologías prevalentes en la Web Semántica.

2.3.3. Ontologías y representación del conocimiento formal

Una ontología es comúnmente vista como una base de conocimiento mantenida por un sistema basado en conocimiento para tener acceso y razonar acerca del conocimiento del dominio.

Representación del conocimiento

La representación y razonamiento del conocimiento busca el diseño de sistemas computacionales que razonen acerca de una representación del mundo entendible por la computadora. Los sistemas basados en conocimiento tienen un modelo computacional de algún dominio de interés (a éste se le denomina como base de conocimiento) cuyos símbolos sirven como sustitutos de los artefactos del dominio en el mundo real, como objetos, eventos, relaciones, etc. Geller (2003). El razonamiento es realizado mediante la manipulación de símbolos en la base de conocimiento.

En contraste con los métodos no simbólicos en la Inteligencia Artificial (como aprendizaje automático estadístico), la representación del conocimiento simbólico trabaja sobre el pro-

cesamiento explícito modelado en secciones de conocimiento que son representados de una forma estructurada.

Una distinción entre la base de conocimiento y una ontología es que la base de conocimiento representa información de una situación particular de los asuntos en el dominio, mientras que la ontología representa una información más general acerca de cada una de las situaciones presentes en el dominio de interés. En resumen una ontología hace referencia a un esquema de conocimiento a través de clases C y sus interrelaciones P expresado mediante axiomas del tipo α_{Δ}, $\alpha_{D\rightarrow}$, $\alpha_{\rightarrow R}$, $\alpha_{\oplus}$, mientras que la base de conocimiento comprende hechos acerca de instancias I concretas expresados mediante axiomas del tipo $\alpha_{\wedge}$ y $\alpha_{\rightarrow}$. Una base de conocimiento es como la especificación de una ontología, es decir, una parte de ésta.

Razonamiento

En los sistemas basados en conocimiento, la idea de razonamiento se refiere al proceso de realizar conclusiones. Los axiomas que están contenidos en la base de conocimiento constituye el conocimiento explícito acerca de un dominio de interés, mientras que la habilidad para procesar el conocimiento explícito computacionalmente por medio del razonamiento permite al sistema derivar conocimiento implícito que lógicamente proviene del conocimiento explícito estipulado. Debido a la formalización de la representación, las propuestas simbólicas para representar el conocimiento permite razonar por medio de lógica formal, lo cual es una herramienta poderosa para el proceso de llegar a una conclusión.

2.3.4. Lenguajes de ontologías

Con el surgimiento en las investigaciones de la Web Semántica han surgido lenguajes especializados para la representación de ontologías. Además de cubrir las características esenciales de una ontología, proveen artefactos que habilitan su uso en la Web, por ejemplo identificación de entidades mediante URIs o formatos de serialización XML. Algunos de los lenguajes

que prevalecen en el ámbito de la Web Semántica son:

- **RDF**. Marco de Descripción de Recursos (RDF, por sus siglas en inglés de *Resource Description Framework*) es un marco de trabajo a surgido como iniciativa de "World Wide Web Consortium" (W3C) como estándar para codificar metadatos y ontologías básicas en la Web. Como lenguaje de ontología tiene expresiones para las características de interrelación, instanciación, y subsunción. Este lenguaje no tiene forma de expresión para representar la negación o exclusión.

- **OWL**. Es un lenguaje de los mas prominentes de los RDF(S), propuesto por la W3C llamado Lenguaje de Ontologías Web (OWL, por sus siglas en inglés de *Ontology Web Language*) , para representar ontologías para su uso en la Web. Existen diversas variantes del lenguaje OWL con distinto nivel de expresividad, siendo el más alto el lenguaje **OWL Full**, que tiene una capa en el tope del esquema RDF, que permite el uso de meta-modelado. El lenguaje que le sigue en cuanto a nivel de expresividad es el llamado **OWL DL**, éste está enfocado al formalismo de la lógica descriptiva (LD). OWL ofrece la construcción de clases complejas a partir de otras más simples por medio de expresiones lógicas, que refiere a una rica axiomatización incluyendo la exclusión entre clases.

2.3.5. Tipos de ontologías

Existen diferentes formas de modelos conceptuales que son interpretados como ontologías, varían en cuanto a dimensiones, formas de apariencia, ámbito, y grado de formalidad.

Tipo de ontología de acuerdo al ámbito

Debido a que el dominio de una ontología puede cubrir cualquier tópico en el que se realice un modelado conceptual, no existe una restricción de que tipo de conocimiento puede ser

representado. Sin embargo, las ontologías pueden clasificarse de diferentes tipos de acuerdo al tipo de conocimiento que éstas representen. La categoría más común entre ellas son las conocidas como *ontologías de alto nivel* que describen categorías generales y pueden ser usados entre dominios; y las conocidas como *ontologías de dominio* las cuales capturan un nivel más profundo de conocimiento en un dominio específico.

Tipo de ontología de acuerdo al grado de formalidad

Las ontologías usadas en el contexto de la Web semántica también se pueden distinguir de acuerdo a su nivel de formalidad. El grado de la formalidad de una ontología indica que tantos axiomas contiene esta.

- **Glosarios**: La última forma formal de una ontología son los glosarios, organizan palabras usadas en un cierto dominio acorde a un criterio léxico. Algunos ejemplos son los diccionarios de un lenguaje específico que también incluya información acerca de sinónimos para usarse en un programa para procesamiento de palabras, o clasificación de términos médicos referentes a enfermedades.

- **Esquemas conceptuales**: Red informal semántica de nodos conceptuales enlazados, los esquemas conceptuales usualmente pueden ser el resultado de actividades de etiquetado colaborativo en un contexto Web.

- **Taxonomías**: Son comúnmente utilizadas para organizar formalmente una jerarquía del conocimiento del dominio por medio de jerarquías basados en el concepto de subsunción.

- **Modelo de datos conceptuales**: Los modelos conceptuales son típicamente usados en ciencias computacionales para representar el diseño de sistemas de información o administración de base de datos. Usualmente cuando se utiliza alguna formalización

lógica es usada para identificar situaciones de datos defectuosos.

- **Basados en reglas y hechos**: En muchas aplicaciones las reglas o los hechos sirven como bases de conocimiento intensivo de datos construidos para el manejo de grandes números de individuos con un poco de razonamiento básico. Algunos ejemplos son las bases de la programación lógica para la derivación de instancias y axiomas, o el de lógica descriptiva para consultar hechos con razonamiento simple sobre jerarquía de clases y propiedades.

- **Teorías generales lógicas**: Esta categoría es la mas formal y expresiva de una ontología, en la cual el dominio de discurso está altamente axiomatizado representado por un formalismo de conocimiento basado en lógica, como lógica de primer orden o incluso de mayor orden. En general permite llegar a conclusiones acerca de diversas situaciones en el dominio en forma de axiomas complejos.

2.3.6. Construcción de ontologías

La creación de ontologías como se hace mención en secciones anteriores, difieren en tamaño, ámbito, expresividad, grado de axiomatización. Siempre que sea posible es recomendable reutilizar ontologías existentes, incluso adaptarlas a los nuevos requerimientos dados, debido a la ardua tarea de la construcción de una nueva ontología. Los cambios y mantenimiento en la ontología demanda gran cantidad de recursos de los expertos en el dominio y de los ingenieros de conocimiento. Para mitigar el esfuerzo requerido para su mantenimiento manual se han propuesto herramientas que ayuden a la tarea de construcción y modificación de ontologías como son editores gráficos, ambientes de desarrollo de ontologías. Todas estas herramientas deberían de ser apoyo en el seguimiento de metodologías para creación de ontologías para prevenir errores que podrían afectar en su aplicación, evitando iteraciones y negociaciones innecesarias en el ciclo de construcción de ontologías, además se lograría un

aumento en la velocidad de desarrollo. Sin embargo la tarea de creación y refinamiento de ontologías sigue siendo un proceso muy tardado y demandante de recursos, así que se han propuesto diversas técnicas para automatizar o semi-automatizar la adquisición de conocimiento de éstas. Éstos métodos para (semi)automatizar los procesos de adquisición de conocimiento son llamados *aprendizaje de ontologías* y operan sobre diversas fuentes de información como son, textos en lenguaje natural, documentos multimedia, etc.

La definición de ingeniería de ontología expuesta en Gómez-Pérez *et al.* (2003) es: "conjunto de actividades que concierne el proceso de desarrollo de ontologías, su ciclo de vida, y las metodologías, herramientas y lenguajes para la creación de ontologías. Los editores y ambientes de desarrollo de ontologías pueden incrementar sustancialmente la velocidad de construcción de éstas además de permitir que el desarrollador de la ontología se enfoque más en la parte sintáctica de la ontología. Las herramientas de apoyo en la creación de ontologías varían de acuerdo al tipo de tarea a realizar, Grimm *et al.* (2011) menciona los siguientes enfoques a tomar en cuenta al seleccionar alguna de estas herramientas en el desarrollo de la ontología: visualización y edición, extensibilidad (etapas que apoya en la ingeniería de ontología), soporte en el razonamiento y desarrollo colaborativo.

Metodologías en la ingeniería de ontologías

Las metodologías empleadas en la construcción de ontologías de acuerdo a han adquirido cada vez mas interés, no por la extensión del uso de las ontologías, sino por el aumento tanto en tamaño como en la complejidad de éstas. Además uno de los retos mas importantes en la ingeniería de ontologías, desde la perspectiva de la Web Semántica, es en la construcción de éstas con el esfuerzo mínimo necesario de los humanos. Las etapas principales en las metodologías empleadas en la ingeniería de ontologías descritas en Gómez-Pérez *et al.* (2003) y Rector *et al.* (2004) son las siguientes:

- **Análisis de requerimientos**. El proceso de ingeniería de ontologías usualmente co-

mienza con esta etapa, se realiza un análisis exhaustivo de los requerimientos sobre el escenario de aplicación de la ontología. El experto en el dominio y/o el ingeniero de conocimiento plasma en un documento de especificación de requerimientos, que sirve como base para el modelado y aseguramiento de calidad en las etapas posteriores.

- **Conceptualización**. En esta etapa el contenido en la ontología es representado en vocabulario semántico y declaraciones acerca del dominio de interés, que involucran las entidades ontológicas seleccionadas y la formulación de axiomas. Siguiendo los lineamientos estipulados en la especificación de requerimientos en la etapa anterior, el ingeniero de conocimiento y el experto del dominio deben llegar a un acuerdo común referente a la estructura básica de la ontología. El resultado de esta etapa es una especificación informal o semi-formal de la conceptualización en común.

- **Implementación**. La formalización explícita de esta especificación en términos de un lenguaje de presentación de ontologías concreto es el paso final en la metodología de la ingeniería de ontologías. Esta fase en particular puede ser apoyada por técnicas de automatización en la adquisición y reutilización de ontologías.

2.3.7. Aprendizaje de ontologías

Debido a la emersión de la Web semántica las investigaciones en el campo de la ingeniería del dominio ontológico han identificado la necesidad de (semi)automatizar el proceso de crear ontologías de dominio. Representar el conocimiento usando ontologías de dominio beneficia la compartibilidad y reusabilidad de las ontologías debido a su formalización. El proceso de automatización en la construcción de ontología es conocida como Aprendizaje Ontológico (OL, por sus siglas en inglés de *Ontology Learning*) . La generación de ontologías puede ser a partir de diversas fuentes de información como documentos multimedia, folksonomías. Éste término es comúnmente utilizado para referirse a la adquisición de conocimiento a partir

de texto en lenguaje natural no estructurado. Este tipo de aprendizaje de ontología puede ser referido como aprendizaje de ontología léxico o lingüísticamente motivado, difiere de por ejemplo, de aprendizaje conceptual, o exploración relacional, en términos de datos de entrada. Las propuestas lógicas usualmente derivan nuevos axiomas a partir de representaciones de conocimiento explícitas y formalizadas, mientras que el aprendizaje ontológico léxico tiene el reto de operar con grandes cantidades de datos informales así como poco fiables.
Las técnicas de aprendizaje varían de acuerdo a diversos criterios comoZouaq y Nkambou (2010):

1. **Grado de automatización**: que puede ser semiautomático o completamente automático.

2. **Conocimiento ontológico a extraer**: conceptos, taxonomía, relaciones conceptuales, atributos, instancias o axiomas.

3. **Fuentes de conocimiento**: textos, bases de datos, documentos xml, etc.

4. **Propósito**: creación de una nueva ontología y/o actualizar las ontologías existentes.

Extracción de conceptos

Es la primer tarea a ser realizada en ingeniería ontológica, la identificación de conceptos. Los conceptos pueden ser descritos como objetos mentales complejos que son caracterizados por un número de características.

Extracción de atributos

Mientras que los conceptos son caracterizados por un número de características, es importante develar los atributos distintivos o propiedades que define un concepto. En Guarino (1992) se hace una distinción entre atributos relaciones y no relaciones. Los atributos relacio-

nales incluyen cualidades y roles relacionales, mientras que las no relacionales incluyen partes.

Extracción de taxonomía

Es la tarea en la cual se organiza el conocimiento en taxonomías, como se propone en Corcho y Gómez-Pérez (2000), las relaciones permiten la herencia entre conceptos y razonamiento automático.

Extracción de relaciones conceptuales

Las relaciones conceptuales se refiere a cualquier tipo de relación entre dos conceptos aparte de las relaciones taxonómicas. Algunas de las relaciones pueden ser sinonimia, parte de, posesión, atributo de, causalidad, así como cualquier otra relación que indique un enlace entre conceptos.

Extracción de instancias

Es una tarea de clasificación cuyo objetivo es el de encontrar instancias de conceptos definidos en una ontología. Es una actividad similar a reconocimiento de entidades y extracción de información.

Extracción de axiomas

Un axioma representa condiciones necesarias y suficientes que restringe la información contenida en la ontología y deduce nueva información.

Técnicas de aprendizaje utilizadas en OL en textos

En el Aprendizaje Ontológico se utilizan técnicas de inteligencia artificial para automatizar la extracción de conocimiento, específicamente cuando se realiza la extracción en fuentes

de texto. Algunas de las técnicas más utilizadas de la inteligencia artificial para apoyar al aprendizaje ontológico, son *Procesamiento de Lenguaje Natural* y técnicas de *Aprendizaje Automático* (Zouaq y Nkambou, 2009).

Técnicas en extracción de conceptos

Una de las tareas más fundamentales en el aprendizaje ontológico léxico es de encontrar términos o frases que son relevantes en un dominio particular (denominado extracción de términos), una de las metodologías empleadas para lograrlo es hacer análisis en las frecuencias de aparición de términos, o considerando información estructural contenida en documentos HTML o XML. Como segundo paso a la extracción de conceptos, basándose en la hipótesis de distribución de Harris (Harris, 1954), dichos términos pueden ser agrupados en grupos de sinónimos o términos relacionados que describen un solo concepto.

Técnicas en extracción de relaciones

Muchas de las técnicas propuestas en la extracción de relaciones se basan en patrones léxico-sintácticos, conocidos habitualmente como *patrones de Hearst* (Heart, 1992). Estos patrones esencialmente expresiones lingüísticas involucran restricciones tanto léxicos como sintácticos en el contexto textual de un concepto de expresión, éstas han sido útiles en detectar relaciones hipónimas o categorías de entidad nombrada.

Varios autores han propuesto diferentes técnicas específicamente para identificar relaciones de subsunción (Caraballo, 1999; Cimiano *et al.*, 2004; Faure y Nedellec, 1999). Se basan en técnicas de agrupamiento jerárquico con el fin de agrupar términos con un comportamiento lingüístico similar en cúmulos organizados de forma jerárquica.

Técnicas en extracción de instancias

Comúnmente la tarea de extracción de instancias han sido enfocadas en la adquisición de hechos, que es información a nivel de instancia. Al igual que en el caso de la extracción de relaciones las técnicas empleadas recaen en el uso de patrones léxico-sintácticos. Además

se han propuesto técnicas para extracción de instancias así como afirmaciones en textos de lenguaje natural basados en FrameNet como un tipo de conocimiento estructurado.

Con el fin de facilitar el proceso de integración de las propuestas de automatización y semi-automatización en los procesos de construcción de ontologías, se han desarrollado varias herramientas y marcos de trabajo. Un marco de trabajo para aprendizaje de ontologías define como una "plataforma que provee diversos métodos de aprendizaje de ontología, que complementa uno o mas tareas en el aprendizaje de ontología; guía al usuario en seleccionar, configurar y aplicar éstos métodos; integran los resultados de cada método en un modelo de conocimiento común; y por último, ayuda al usuario a revisar y exportar éste modelo de conocimiento"(Grimm *et al.*, 2011). Para ayudar a evitar errores y disminuir el pos-procesamiento es necesario lograr una mayor integración entre los sistemas de aprendizaje de ontología (semi)automáticos para evaluación de la ontología.

2.4. Sistemas de enseñanza asistidos por computadora

Existen diversos tipos de sistemas computacionales cuyo objetivo son el de apoyar positivamente el proceso de aprendizaje de los alumnos. Los sistemas que proveen la administración del curso en cuestión son llamados Sistemas de Gestión de Aprendizaje (ReLU, por sus siglas en inglés de *Learning Management System*) , por otra parte se encuentran los sistemas que adaptan ya sea el contenido o el temario del curso de acuerdo a las necesidades del alumno, dentro de éstos sistemas se encuentran los Sistemas Hipermedia Adaptativos (AHS, por sus siglas en inglés de *Adaptive Hypermedia System*) e (ITS). Éstos últimos utilizan técnicas de inteligencia artificial para lograr dicha adaptabilidad.

Vanlehn define a un Tutor Inteligente como sistema de software educativo que utiliza técnicas de Inteligencia Artificial (IA) para representar el conocimiento y otorgar retroali-

mentación al alumno (Vanlehn, 1987). Los ITS tienen como propósito emular el comportamiento de un tutor humano. Las interrogantes básicas que deben responder los ITS son los siguientes: qué enseñar, cómo enseñar. La arquitectura de los ITS está conformada por al menos cuatro módulos que se han presentado como estructura de los ITS (Lage, 2009): el modelo de dominio, el modulo del tutor, el modulo del estudiante y la interfaz de usuario. Los ITS se integran principalmente de los siguientes módulos:

- El modelo de dominio: se define como dominio del conocimiento, tiene como objetivo el principal el de almacenar los conocimientos referentes al dominio de aplicación del ITS.

- El modelo tutor: es el encargado de definir y aplicar estrategias pedagógicas de enseñanza, contiene los objetivos a ser alcanzados y los planes para lograrlo. Monitorea el comportamiento del estudiante y le provee materia de aprendizaje adecuado a las técnicas didácticas.

- El modelo del estudiante tiene por objetivo realizar el diagnostico cognitivo del alumno, y el modelado del mismo para una adecuada retroalimentación del sistema.

- El módulo interfaz de usuario: permite la interacción del estudiante con el sistema.

El uso ITS es una vía para impartir conocimiento emulando el comportamiento de un tutor humano, teniendo la posibilidad de adaptarse a los alumnos para simular una enseñanza uno a uno, y así obtener una mejor retroalimentación para lograr un mejor aprendizaje.

2.4.1. Problemas en el desarrollo de los Sistemas de Tutoría Inteligente

Los ITS proveen los beneficios de la instrucción uno a uno de forma automatizada y eficiencia en cuanto a costo. Sin embargo existe cierto escepticismo, debido a que no son usados ampliamente en los sistemas educativos reales. Una de las razones principales es que su desarrollo es complejo, tiempo de desarrollo extenso, implica un trabajo conjunto de varias personas, incluyendo programadores, tutores, y expertos de un cierto dominio referente al dominio de aplicación de los ITS.

Además de lo anterior, una vez finalizada su construcción, el ITS de un cierto dominio no puede ser reutilizado en algún otro dominio sin invertir mucho tiempo y esfuerzo Escudero y Fuentes-Gonzalez (2012). Ante esta circunstancia, como requerimiento principal en el desarrollo posterior de los ITS debería ser el de desarrollar la base de conocimientos en un lenguaje que sea fácilmente compartible y reutilizable por otros sistemas, fue motivo por lo que (Ibíd.) propusieron utilizar ontologías para representar conocimiento, aunque siendo con el propósito de utilizarla para representación de secuencia de un curso dado, también enfatizan que este tipo de representación de conocimiento puede ser fácilmente utilizable en ITS más complejos en donde se requieren relaciones estructurales como lo son parte-de, es-un, etc.

Las ontologías han sido ampliamente propuesta para dar solución al problema de integración en los sistemas basados en conocimiento debido a la generalidad con la que están compuestas (Bylander y Chandrasekaran, 1987). En Guarino (1998) se define una ontología como un sistema particular de categorías sistematizando cierta visión mundo. El uso de ontologías provee soluciones a los diversos problemas en los sistemas basados en conocimiento, como son la incapacidad de compartir y reutilizar la base de conocimiento debido a las diferencias entre la sintaxis y semántica, además de la falta de protocolos definidos para la

compartición entre sistemas representación de conocimiento.

Así mismo Perez y Benjamins (1999) sustentan el uso de ontologías para reutilización de conocimiento, además éstas al representar conocimiento a un nivel genérico, suponen una buena compatibilidad con los métodos de resolución de problemas como motores de inferencia, ya que éstos operan sobre representación de conocimiento genérico.

El modelo del dominio es una parte esencial de los sistemas de enseñanza asistido por computadora ya que representa el conocimiento de un tema a ser enseñado hacia el alumno. Siendo éste uno de los módulos más complejos de implementar, debido a que se requiere de expertos para otorgar el conocimiento de un dominio dado e ingenieros de conocimiento para representar el conocimiento en una estructura comprensible para la computadora, además de definir las relaciones pedagógicas entre los tópicos a ser enseñados (Conde *et al.*, 2014).

2.5. Aprendizaje Profundo

Anteriormente las técnicas de aprendizaje automático requerían un gran esfuerzo humano para el análisis y diseño de características de los datos recabados. Con la llegada de las técnicas basadas en el Aprendizaje Profundo, ahora los modelos son capaces de lidiar con datos crudos, es decir, sin casi procesamiento, por ejemplo imágenes, audio, vídeo, texto, etc. Las técnicas de DL construyen nuevas representaciones de los datos donde cada nivel de transformación es comúnmente realizada por módulos no lineales. Una de las características más destacadas de las técnicas de DL es que las nuevas representaciones de los datos no son construidas por humanos sino que son automáticamente generada por esta clase de modelos (Urban *et al.*, 2016).

Los modelos basados en Redes Neuronales para representación de palabras han sido el estado del arte en áreas de procesamiento de lenguaje natural. Las actividades en las cuales

han destacado este tipo de modelos en el área de lenguaje natural, tales como extracción de términos relevantes, traducción automática de textos, identificación de entidades, entre otros.

2.5.1. Redes Neuronales

Las técnicas de aprendizaje automático permiten establecer modelos utilizando datos recolectados en el pasado. De tal forma, que estos modelos podrán identificar patrones entre los datos y así construir buenas aproximaciones de un fenómeno dado. Una de las técnicas que pertenecen al aprendizaje automático son basadas en el funcionamiento del cerebro humano, llamadas Redes Neuronales Artificiales.

Las RNA están constituidas por un grupo de unidades de procesamiento llamados neuronas, las cuales están agrupadas en capas. Las neuronas empleadas en las RNA están basadas en el modelo matemático propuesto por McCulloch y Pitts (1943). Cada neurona recibe un conjunto de entradas $\{x_1, x_2, x_3, \ldots, x_D\}$ y devuelve una única salida y. Las neuronas que constituyen una RNA están interconectadas entre sí, cada una de estás conexiones pueden establecerse con mayor o menor intensidad. La intensidad en estos modelos esta representado por pesos sinápticos, de tal forma que cada entrada x_i se encuentra amplificada o atenuada por un peso w_i (figura 2.1).

Para obtener la salida de cada una de las neuronas se debe calcular la suma ponderada (z) de las de las entradas, a la cual se le conoce como activación de la neurona (véase la ecuación 2.7).

$$z = \sum_{i=1}^{D} w_i x_i + b \tag{2.7}$$

Donde b es el umbral que se utiliza para compensar la diferencia entre el valor medio de

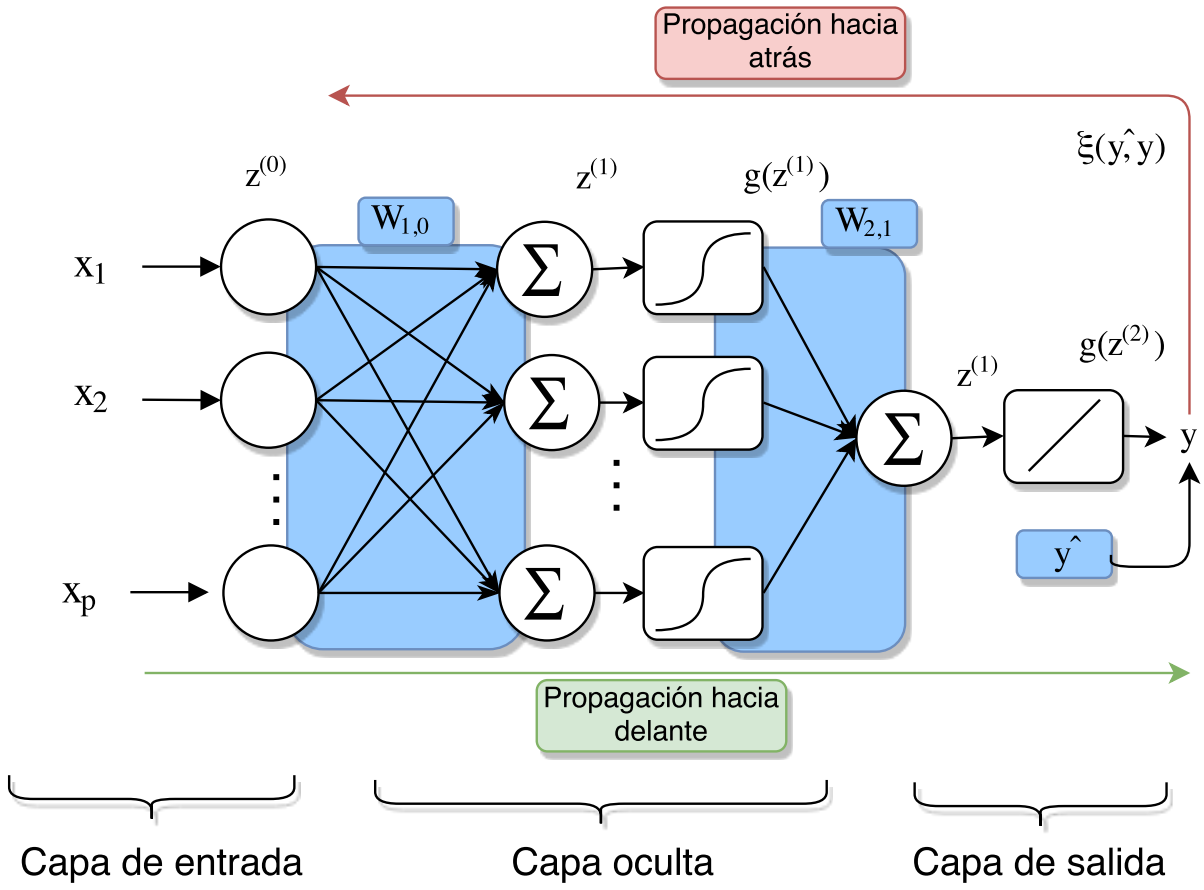

Figura 2.1: Modelo Red Neuronal Artificial

las entradas u el correspondiente valor entre las salidas deseadas. Posterior al cálculo de la activación de la neurona, para obtener el valor de la salida y, se aplica una función sobre la activación z, a esta función se le conoce como función de activación o función de transferencia, por lo tanto se puede representar mediante la ecuación 2.8.

$$y = g(z) = g\left(\sum_{i=1}^{D} w_i x_i + b\right) \tag{2.8}$$

Si se considera al umbral b como un peso mas w_0 y se establece un valor fijo a una entrada $x_0 = 1$ se puede representar la salida de la red neuronal como se muestra en la ecuación 2.9, además, esta ecuación puede ser reescrita con una notación vectorial si se considera a w un vector de pesos y a x como el vector de entrada a la red $y = g(z) = g(w^T \cdot x)$.

$$y = g(z) = g\left(\sum_{i=0}^{D} w_i x_i\right) \tag{2.9}$$

Las funciones de activación más comunes seleccionadas son las siguientes:

Función lineal

$$y = g(z) = z \tag{2.10}$$

Función sigmoidea

$$y = g(z) = \frac{1}{1 + e^{-z}} \tag{2.11}$$

Función tangente hiperbólica

$$y = g(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} \tag{2.12}$$

Función gaussiana

$$y = g(z) = \exp\left(-\frac{(z-\mu)^2}{2\sigma^2}\right) \tag{2.13}$$

Función rectificador (ReLU, por sus siglas en inglés de *Rectified Linear Unit*)

$$y = g(z) = \text{máx}(0, z) \tag{2.14}$$

Para datos categóricos, la función Softmax

$$y = g(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \text{para } j = 1, \ldots, K \tag{2.15}$$

La forma de interconexión de las neuronas y al número de éstas se le conoce como arquitectura o topología de la red. A las RNA compuestas por varias capas o conjuntos de neuronas se les conoce como RNA multicapa, es decir, cuenta con un grupo o más de neuronas entre la entrada y la salida del modelo.

Si se considera una RNA multicapa con una capa oculta que consta de D entradas, M neuronas en la capa oculta, y C unidades en la capa de salida; el nivel de activación z_j de la neurona j en la capa oculta se calcula como la combinación lineal de las entradas x que recibe de la capa anterior (en este caso la de entrada), que al aplicar una función de transferencia se obtiene la salida a_j de ésta neurona. Este proceso se representa en la ecuación 2.16.

$$a_j = \tilde{g}(a_j) = g\left(\sum_{i=0}^{D} w_{ij} x_i\right) \text{ para } j = 1, 2, \ldots, M \tag{2.16}$$

Donde w_{ij} es el peso asociado a cada conexión de la capa anterior, en el caso de que la capa anterior sea la de entrada entonces con x. En la capa de salida k se realiza se forma similar el cálculo de activación de cada una de las neuronas y la aplicación de una función

de transferencia o activación a dichas neuronas. Es importante señalar que en cada una de las capas se puede emplear una función de transferencia diferente. El proceso del cálculo de la salida de la RNA multicapa se puede observar en la ecuación 2.17, y de forma extendida en la ecuación 2.18.

$$y = \tilde{g}(z_k) = \tilde{g}\left(\sum_{j=0}^{M} w_{kj} a_j\right) \tag{2.17}$$

$$y = a_k = \tilde{g}(z_k) = \tilde{g}\left(\sum_{j=0}^{M} w_{kj} g\left(\sum_{i=0}^{D} w_{ij} x_i\right)\right) \text{ para } k = 1, 2, \ldots, C \tag{2.18}$$

Actualización de pesos

El proceso de evaluación del modelo, es decir calcular la salida y de la RNA multicapa dada una entrada a_0, se le conoce como propagación hacia delante. Como se ilustró anteriormente, los pesos w de cada capa modifica la salida de las capas subsecuentes. Por lo tanto, si se desea una determinada salida t se deben ajustar los pesos w de cada una de las capas de tal forma que la salida y se aproxime a t, a este proceso de actualización de pesos se conoce como propagación hacia atrás. Este ajuste se realiza en base al error, es decir, a la diferencia entre la salida deseada t y la salida obtenida y, el algoritmo empleado para dicha actualización en las RNA multicapa es el gradiente descendente o alguna de sus variantes.

Gradiente Descendente

El algoritmo de optimización gradiente descendente consiste en encontrar el mínimo de una función, calculando y restando el gradiente en un punto dado del espacio de solución. El proceso empleado para la actualización de los pesos se muestra en la ecuación 2.19.

$$w_j(\tau + 1) = w_j(\tau) - \eta \nabla E(w_j(\tau)) \tag{2.19}$$

En donde $w_j(\tau + 1)$ es el nuevo valor que adquirirá un determinado peso, $w_j(\tau)$ es el peso actual, η es la tasa de aprendizaje usualmente $0 < \eta < 1$ y $\nabla E(w_j(\tau))$ es la función a minimizar, en este caso la función de error.

Para calcular el gradiente respecto a los pesos ($\nabla E(w)$) se debe aplicar la regla de la cadena, que consiste en la derivación de una composición de dos o mas funciones, en este caso hasta llegar al conjunto de pesos deseados (correspondientes entre las k capas definidas). Como indica el nombre de este proceso, propagación hacia atrás, se inicia con la actualización de los pesos de la capa de salida (w_k), siguiendo en ese orden hasta llegar a la primer capa (w_1). El cálculo del gradiente de la función de error, indicado como en lo sucesivo como$Loss$, respecto a los pesos de la capa de salida w_k, está indicado por la ecuación 2.20 para fines ilustrativos la función de error será $Loss = (t_i - y_i)^2$.

$$\begin{aligned}
\frac{\partial Loss}{\partial w_k} &= \frac{\partial Loss}{\partial a_k}\frac{\partial \tilde{g}(z_k)}{\partial z_k}\frac{\partial}{\partial w_k}\left(\sum_{j=0}^{M} w_{kj}a_j\right) \\
&= \frac{\partial (t_i - y_i)^2}{\partial a_k}\frac{\partial \tilde{g}(z_k)}{\partial z_k}\frac{\partial}{\partial w_k}\left(\sum_{j=0}^{M} w_{kj}a_j\right) \\
&= -2(t_i - y_i)\frac{\partial \tilde{g}(z_k)}{\partial z_k}\frac{\partial}{\partial w_k}\left(\sum_{j=0}^{M} w_{kj}a_j\right) \\
&= -2(t_i - y_i)\dot{\tilde{g}}(z_k)\frac{\partial}{\partial w_k}\left(\sum_{j=0}^{M} w_{kj}a_j\right) \\
&= -2(t_i - y_i)\dot{\tilde{g}}(z_k)a_j
\end{aligned} \tag{2.20}$$

El proceso de actualización de pesos en las capas anteriores dependerá de la propagación del error de las capas posteriores, por lo tanto, se tiene que utilizar nuevamente la regla de la cadena para derivar el error de la salida de la RNA multicapa respecto a los pesos de la capa oculta w_j. El proceso es muy similar al paso anterior, la diferencia es que se tendrá que aplicar la regla de la cadena en la composición de funciones hasta llegar a la función que

esté en términos de w_j. El proceso de aplicación de la regla de la cadena se muestra en las ecuaciones 2.21.

$$\begin{aligned}
\frac{\partial Loss}{\partial w_j} &= \frac{\partial Loss}{\partial a_k}\frac{\partial \tilde{g}(z_k)}{\partial z_k}\frac{\partial}{\partial a_j}\left(\sum_{j=0}^{M} w_{kj}a_j\right)\frac{\partial}{\partial w_j}\left[g\left(\sum_{i=0}^{D} w_{ij}a_i\right)\right] \\
&= \frac{\partial (t_i - y_i)^2}{\partial a_k}\frac{\partial \tilde{g}(z_k)}{\partial z_k}\frac{\partial}{\partial a_j}\left(\sum_{j=0}^{M} w_{kj}a_j\right)\frac{\partial}{\partial w_j}\left[g\left(\sum_{i=0}^{D} w_{ij}a_i\right)\right] \\
&= -2(t_i - y_i)\frac{\partial \tilde{g}(z_k)}{\partial z_k}\frac{\partial}{\partial a_j}\left(\sum_{j=0}^{M} w_{kj}a_j\right)\frac{\partial}{\partial w_j}\left[g\left(\sum_{i=0}^{D} w_{ij}a_i\right)\right] \\
&= -2(t_i - y_i)\dot{\tilde{g}}(z_k)\frac{\partial}{\partial a_j}\left(\sum_{j=0}^{M} w_{kj}a_j\right)\frac{\partial}{\partial w_j}\left[g\left(\sum_{i=0}^{D} w_{ij}a_i\right)\right] \\
&= -2(t_i - y_i)\dot{\tilde{g}}(z_k)w_k\frac{\partial}{\partial w_j}\left[g\left(\sum_{i=0}^{D} w_{ij}a_i\right)\right] \\
&= -2(t_i - y_i)\dot{\tilde{g}}(z_k)w_k\dot{g}(z_j)\frac{\partial}{\partial w_j}\left(\sum_{i=0}^{D} w_{ij}a_i\right) \\
&= -2(t_i - y_i)\dot{\tilde{g}}(z_k)w_k\dot{g}(z_j)a_i
\end{aligned} \tag{2.21}$$

Al agregar mas capas ocultas, se debe extender las derivadas parciales cada ves mas atrás. Como se puede observar hay valores que se pueden reutilizar al calcular el gradiente de la capa posterior, en este caso la operación $-2(t_i - y_i)\dot{\tilde{g}}(z_k)$ es requerida en la ecuación 2.16. Usualmente a esta expresión se deja indicada como $\delta_k = -2(t_i - y_i)\dot{\tilde{g}}(z_k)$, de tal forma que el gradiente en la última capa (k) se puede expresar como $\Delta w_k = \delta_k a_j$. Por lo tanto si se agregan más capas también se podría reutilizar el valor calculado previamente, es decir en la capa i utilizar el cálculo de $\delta_j = \delta_k w_k \dot{g}(z_j)a_j$. De esta forma se logra simplificar las expresiones para actualizar los pesos de la siguiente forma (ecuación 2.22):

$$\Delta w_\ell = \begin{cases} \delta_k a_j, & \text{if } \ell = k, \text{donde } \delta_k = -2(t_i - y_i)\dot{\tilde{g}}(z_k) \\ \delta_\ell a_{(\ell-1)}, & \text{if } \ell < k, \text{donde } \delta_\ell = \delta_{(\ell+1)} w_{(\ell+1)} \dot{g}(z_\ell) a_{(\ell-1)} \end{cases} \tag{2.22}$$

Sin embargo, en esta expresión donde se generaliza el gradiente de los pesos para n capas, solo es correcto cuando se utilice con la función de error *cuadrado de los errores*, por lo tanto, no esta del todo generalizada. Si la naturaleza del problema cambia de ser un problema de regresión a un problema de clasificación, es conveniente utilizar otro tipo de errores, por ejemplo el de entropía cruzada (ecuación 2.23) , el cual es mas conveniente para este tipo de problemas.

$$C(w) = -\frac{1}{N}\sum_{n=1}^{N}\left[t_n \log y_n + (1-t_n)\log(1-y_n)\right] \tag{2.23}$$

Donde N es el número de clases que puede tener como salida la RNA, t_n es el valor $\{0,1\}$ deseado, siendo y_n es el valor obtenido. Dado que es común el cambio de funciones de error, es conveniente generalizar la ecuaciones 2.20 y 2.21, indicando la función de error como $\xi : (t_i, y_i) \rightarrow e$ se obtiene la siguiente expresión para la ecuación 2.20:

$$\begin{aligned}
\frac{\partial Loss}{\partial w_k} &= \frac{\partial Loss}{\partial a_k}\frac{\partial \tilde{g}(z_k)}{\partial z_k}\frac{\partial}{\partial w_k}\left(\sum_{j=0}^{M} w_{kj}a_j\right) \\
&= \frac{\partial \xi(t,a_k)}{\partial a_k}\frac{\partial \tilde{g}(z_k)}{\partial z_k}\frac{\partial}{\partial w_k}\left(\sum_{j=0}^{M} w_{kj}a_j\right) \\
&= \dot{\xi}(t,a_k)\frac{\partial \tilde{g}(z_k)}{\partial z_k}\frac{\partial}{\partial w_k}\left(\sum_{j=0}^{M} w_{kj}a_j\right) \\
&= \dot{\xi}(t,a_k)\dot{\tilde{g}}(z_k)\frac{\partial}{\partial w_k}\left(\sum_{j=0}^{M} w_{kj}a_j\right) \\
&= \dot{\xi}(t,a_k)\dot{\tilde{g}}(z_k)a_j \\
&= \delta_k a_j
\end{aligned} \tag{2.24}$$

Donde δ_k se emplea como se definió anteriormente. La actualización de los pesos correspondientes a la capa oculta usando la nueva expresión del error se muestra en la ecuación 2.24:

$$
\begin{aligned}
\frac{\partial Loss}{\partial w_j} &= \frac{\partial Loss}{\partial a_k}\frac{\partial \tilde{g}(z_k)}{\partial z_k}\frac{\partial}{\partial a_j}\left(\sum_{j=0}^{M} w_{kj}a_j\right)\frac{\partial}{\partial w_j}\left[g\left(\sum_{i=0}^{D} w_{ij}a_i\right)\right] \\
&= \frac{\partial \xi(t,a_k)}{\partial a_k}\frac{\partial \tilde{g}(z_k)}{\partial z_k}\frac{\partial}{\partial a_j}\left(\sum_{j=0}^{M} w_{kj}a_j\right)\frac{\partial}{\partial w_j}\left[g\left(\sum_{i=0}^{D} w_{ij}a_i\right)\right] \\
&= \dot{\xi}(t,a_k)\frac{\partial \tilde{g}(z_k)}{\partial z_k}\frac{\partial}{\partial a_j}\left(\sum_{j=0}^{M} w_{kj}a_j\right)\frac{\partial}{\partial w_j}\left[g\left(\sum_{i=0}^{D} w_{ij}a_i\right)\right] \\
&= \dot{\xi}(t,a_k)\dot{\tilde{g}}(z_k)\frac{\partial}{\partial a_j}\left(\sum_{j=0}^{M} w_{kj}a_j\right)\frac{\partial}{\partial w_j}\left[g\left(\sum_{i=0}^{D} w_{ij}a_i\right)\right] \\
&= \dot{\xi}(t,a_k)\dot{\tilde{g}}(z_k)w_k\frac{\partial}{\partial w_j}\left[g\left(\sum_{i=0}^{D} w_{ij}a_i\right)\right] \\
&= \dot{\xi}(t,a_k)\dot{\tilde{g}}(z_k)w_k\dot{g}(z_j)\frac{\partial}{\partial w_j}\left(\sum_{i=0}^{D} w_{ij}a_i\right) \\
&= \dot{\xi}(t,a_k)\dot{\tilde{g}}(z_k)w_k\dot{g}(z_j)a_i \\
&= \delta_k w_k\dot{g}(z_j)a_i \\
&= \delta_j a_i
\end{aligned}
\tag{2.25}
$$

Donde δ_j se emplea como se definió anteriormente. Por lo tanto la regla de actualización generalizada para cualquier número de capas, en la nueva expresión también se podrá definir cualquier tipo de error (ecuación 2.26).

$$
\Delta w_\ell = \begin{cases} \delta_k a_j, & \text{if } \ell = k, \text{donde } \delta_k = \dot{\xi}(t_i, y_i)\dot{\tilde{g}}(z_k) \\ \delta_\ell a_{(\ell-1)}, & \text{if } \ell < k, \text{donde } \delta_\ell = \delta_{(\ell+1)}w_{(\ell+1)}\dot{g}(z_\ell)a_{(\ell-1)} \end{cases}
\tag{2.26}
$$

Adam

Adam, derivado del inglés de *estimación de momento adaptativo*, es un algoritmo para optimización que usa gradientes de primer orden (Kingma y Ba, 2014). Los parámetros propuestos por el autor son $\eta = 0{,}001$, $B_1 = 0{,}9$, $B_2 = 0{,}999$ y $\epsilon = 10^{-8}$. Las operaciones de

multiplicación sobre vectores son elemento por elemento y las notaciones B_x^τ se refiere al valor de β elevado a la iteración τ. El proceso de actualización de pesos se muestra en la ecuación 2.27. Este algoritmo es del tipo de tasa de aprendizaje adaptativa, en el cual no es necesario una configuración exhaustiva del hiperparámetro η, además, resulta ser empíricamente superior a otros métodos de optimización.

$$\begin{aligned} m_\tau &= \beta_1 m_{\tau-1} + (1-\beta_1)\nabla E(w_\tau) \\ s_\tau &= \beta_2 s_{\tau-1} + (1-\beta_2)\nabla E(w_\tau) \otimes \nabla E(w_\tau) \\ \widetilde{m_\tau} &= \frac{m_{\tau-1}}{1-\beta_1^\tau} \\ \widetilde{s_\tau} &= \frac{s_{\tau-1}}{1-\beta_2^\tau} \\ w_\tau &= w_{\tau-1} - \eta \cdot \widetilde{m_\tau} \oslash (\sqrt{\widetilde{s_\tau}} + \epsilon) \end{aligned} \tag{2.27}$$

2.5.2. Redes Neuronales Convolutivas

Las Redes Neuronales Convolutivas (CNN, por sus siglas en inglés de *Convolutional Neural Network*) son una especialización de las RNA, las cuales están enfocadas en la extracción automática de características de los datos de entrada. Este tipo de red está inspirada en la corteza visual del ser humano y han sido utilizadas para reconocimiento de imágenes desde la década de 1980. Debido al incremento del poder de cómputo de años recientes y la cantidad de datos de entrenamiento, han adquirido mas popularidad. Estos modelos han sido empleados en búsquedas de imágenes, automóviles autónomos, clasificación automática de vídeos.

Aunque las CNN fueron creadas inspiradas en la corteza visual del ser humano, su aplicación no se limita únicamente al procesamiento de imágenes, sino cualquier tipo de datos, por ejemplo reconocimiento de audio o procesamiento de lenguaje natural.

En el artículo Cho *et al.* (2014) se propusieron dos componentes que han sido clave en este tipo de Redes, como las capas convolutivas y las capas de agrupación (*pooling*).

El bloque principal en las CNN es la capa convolutiva, en la cual no está conectada a cada uno de los píxeles de la imagen de entrada sino solo con los campos receptivos. Las capas sucesivas están conectadas solo en ciertas regiones específicas de la primer capa. Este tipo de arquitectura jerárquica permite descubrir características de bajo nivel en la primer capa y posteriormente agregarlas en un nivel más alto de representación en las siguientes capas ocultas.

Una de las características sobresalientes de las CNN es que anteriormente en las RNA multicapa se tenían que transformar a un vector de características, y en éstas nuevas arquitecturas ya no es necesario, ya que soporta la recepción de datos en dos dimensiones.

Cada una de las neuronas en una locación i, j está conectada con las salidas de la capa anterior, localizada en la zona (i) a $(i + f_h - 1)$ y (j) a $(j + f_v - 1)$ donde f_h y f_v son la altura y ancho del campo receptivo. Para lograr que capa actual y anterior contengan la misma altura y ancho, es común agregar ceros para rellenar los espacios entre las entradas. En caso contrario cuando una capa de entrada es sustancialmente mas grande que la siguiente capa, es posible separar los campos receptivos. La distancia entre dos campos receptivos consecutivos es llamado *stride*. Dada una neurona en una locación i, j en una capa posterior conectada con la salida de las neuronas en la capa anterior localizada en los rangos $(i \times s_h)$ a $(i \times s_h + f_h - 1)$ y $(j \times s_w)$ a $(j \times s_w + f_v - 1)$ donde s_h y s_w son las diferencias entre los campos receptivos vertical y horizontal respectivamente.

La salida de la capa convolutiva está mostrada en la ecuación 2.28, salida a la cual se puede aplicar una función de transferencia no lineal (ecuación 2.29).

$$z_{i,j,k}^{\ell} = b^k + \sum_{k=0}^{f_k} \sum_{u=0}^{f_h} \sum_{v=0}^{f_v} w_{u,v}^k a_{i,j,k}^{\ell-1} \tag{2.28}$$

Donde $a^{\ell-1}$ es la salida de la capa anterior, k es el número de kernels, $w_{u,v}^k$ son los pesos que en el contexto de las CNN son llamados filtros o kernel convolutivos. El tamaño de los filtros tiene generalmente una dimensión reducida de los datos de entrada. Al resultado de la aplicación del kernel convolutivo en todo el conjunto de neuronas de una determinada capa se le conoce como mapa de características, en donde las neuronas tiene como salida valores mayores si el filtro concuerda con la salida de la capa anterior $a^{\ell-1}$.

$$a_{i,j,k}^{\ell} = \sigma\left(z_{i,j,k}^{\ell}\right) \tag{2.29}$$

Al agregar diversas capas convolutivas se aplican diversos kernel convolutivos a las salidas de las capas anteriores, por lo tanto, se identifican características de más alto nivel, por lo tanto se logra generalizar mejor los patrones entre los datos.

En las CNN se agregan diversos parámetros a los que contiene comúnmente las RNA multicapa, como el tamaño del filtro horizontal f_h y vertical f_v, k número de filtros $w_{u,v}^k$, distancia entre los campos receptivos horizontal s_h y vertical s_v; sin embargo, el hecho de seleccionar tamaños de kernels convolutivos más pequeños que las dimensiones de los datos de entrada reduce drásticamente el número de pesos a ajustar. Una de las principales ventajas de las CNN es que logran identificar patrones en diferentes locaciones, siendo que en una RNA profunda no sería el caso, sino que detectan el patrón en una locación en particular.

Una capa popular en las CNN son las llamadas de agrupación (*pooling*) cuyo objetivo es el de reducir el procesamiento, memoria y los parámetros de la red. La función en esta capa consiste en el sub-muestreo de características de la salida de la capa anterior. Al igual que en las capas convolutivas, no todas las neuronas están conectadas con las salidas de la capa

anterior, sino con sólo regiones llamados campo receptivos (de forma rectangular), también se consideran parámetros para la distancia entre los campos receptivos horizontal s_h y vertical s_v. Esta capa carece de pesos y solo realiza una agregación de datos mediante alguna función como el promedio o el máximo de los datos en región del campo receptivo. Es importante destacar que se pierde gran información con este tipo de capa por lo que se recomienda usarse en después de capas muy densas.

2.5.3. Redes Neuronales Recurrentes

Las Redes Neuronales Recurrentes (RNN, por sus siglas en inglés de *Recurrent Neural Network*) están ideadas con el fin de procesar información secuencial como series de tiempo, audio, sentencias, documentos, por lo cual son muy útiles en el área de procesamiento de lenguaje natural en tareas como transcribir discursos, traducción automática, análisis de sentimientos, entre otros.

Las neuronas recurrentes se pueden visualizar como una neurona común pero con una conexión hacia atrás también, por lo que cada neurona contiene dos matrices de pesos: una para la entrada y otro para la salida previa.

La salida de cada neurona recurrente para cada instancia se representa a través de la ecuación 2.30.

$$y_{(t)} = \phi\left(x_{(t)}^T \cdot w_x + y_(t-1)^T \cdot w_y + b\right) \tag{2.30}$$

Donde $x(t)$ representa la entrada, $y(t-1)$ salida previa, w_x pesos de la entrada, w_y pesos del anterior. Se puede simplificar la salida de la neurona recurrente si se representa ambas matrices de pesos (w_x y w_y) como una sola, como se muestra en la ecuación 2.31.

$$Y_{(t)} = \phi\left(\begin{bmatrix} X_{(t)} & Y_{(t-1)} \end{bmatrix} \cdot W + b\right) \quad \text{donde} \quad W = \begin{bmatrix} w_x \\ w_y \end{bmatrix} \tag{2.31}$$

La salida de la neurona $Y(t)$ está conformada por una matriz $m \times n$ donde m es el número de instancias y n es el número de neuronas; $X(t)$ una matriz de una dimensión $m \times p$ donde p es el número de entradas; w_x una matriz de una dimensión $p \times n$ que contiene los pesos de las conexiones en el tiempo actual; w_y es una matriz de una dimensión $n \times n$ que contiene los pesos para la salida del tiempo anterior, y b es un vector de n elementos que contiene el umbral de cada neurona.

Como se puede observar en las ecuaciones, el resultado la salida de la neurona en el tiempo t depende de las salidas anteriores $(y_{(t-1)}, y_{(t-2)}, \ldots, y_0)$. Cuando el tiempo $t = 0$, se asume que la respuesta en tiempos anteriores son ceros. La parte de la red que guarda el estado a través del tiempo se le llama celda. Una neurona recurrente o una capa de neuronas recurrentes se considera como una celda básica.

El estado de la celta en un tiempo t se denota como $h_{(t)}$, la cual es una función de algunas entradas en un tiempo dado y su estado en el tiempo previo: $h_{(t)} = f(h_{(t-1)}, x_{(t)})$. Por lo tanto en esta celda básica, la salida de la red $y_{(t)}$ es igual al estado $h_{(t)}$, lo cual en RNN más complejas no siempre es el caso.

Uno de los procesos de entrenamiento de las RNN es llamado propagación hacia atrás a través del tiempo (BPTT, por sus siglas en inglés de *Backpropagation through time*) . Este proceso de entrenamiento consiste en desenrollar la red a través del tiempo y realizar la propagación hacia atrás. Cuando se realiza la propagación hacia delante se obtiene una secuencia

como salida, a la cual se le aplica una función de error (o costo) $Loss(Y_{(t_a)}, Y_{(t_a+1)}, \ldots, Y_{(t_b)})$ donde a y b es el rango de tiempo a evaluar, los gradientes calculados son propagados hacia atrás en la red desenrollada para actualizar los parámetros. Es importante señalar que se aplica los gradientes en base a todas las salidas evaluadas con la función de costo y no solo en la última salida.

Celda de Memoria de largo y corto plazo

Las redes neuronales recurrentes, específicamente las del tipo Redes Neuronales de Gran Memoria de Corto Plazo (LSTM, por sus siglas en inglés de *Long Short Term Memory*) propuestas por Hochreiter y Schmidhuber (1997) y han sido eventualmente mejoradas a través de los años por diversos investigadores como Alex Graves, Hasim Sak, entre otros. Esta nueva arquitectura de RNN puede considerarse como una celda básica, con la gran diferencia que generalmente logra mejor rendimiento que las anteriores, converge notablemente más rápido y detecta largas dependencias entre los datos.

Las RNN con celda LSTM fueron propuestas como una solución al problema desvanecimiento del gradiente. Este modelo incorpora un mecanismo que permite memorizar la característica actual de los datos a utilizar para futuras entradas. Los elementos que componen a estos modelos son la compuerta de entrada (i_t), la compuerta de olvido (f_t), la compuerta de salida (o_t) y el estado de la celda (c_t).

Como puede observarse las ecuaciones que representan el funcionamiento de una celda LTSM por cada instancia 2.32, se puede notar que el estado esta dividido en dos vectores $h_{(t)}$ y $c_{(t)}$. Se puede considerar que el vector $h_{(t)}$ representa el estado a corto plazo y $c_{(t)}$ al estado a largo plazo.

$$
\begin{aligned}
i_{(t)} &= \sigma(W_{x_i}^T \cdot x_{(t)} + W_{hi}^T \cdot h_{(t-1)} + b_i) \\
f_{(t)} &= \sigma(W_{x_f}^T \cdot x_(t) + W_{hf}^T \cdot h_{(t-1)} + b_f) \\
g_{(t)} &= \tanh(W_{xg}^T \cdot x_{(t)} + W_{hg}^T \cdot h_{(t-1)} + b_g) \\
c_{(t)} &= f_{(t)} \otimes c_{(t-1)} + i_{(t)} \otimes g_{(t)} \\
o_{(t)} &= \sigma(W_{xo}^T \cdot x_{(t)} + W_{ho}^T \cdot h_{t-1} + b_o) \\
y_{(y)} &= h_{(t)} = o_{(t)} \otimes \tanh(c_{(t)})
\end{aligned}
\tag{2.32}
$$

En las celdas LSTM hay 4 capas que tienen el siguiente propósito: La capa principal tiene como salida g_t, la cual tiene el rol de analizar las entradas x_t y los estados anteriores $h_{(t-1)}$. En una celda básica La salida en esta capa se dirige a la salida y_t y h_t, en las celdas LSTM no se dirige de forma directa sino que es es almacenada parcialmente en el vector del estado a largo plazo. Las capas restantes son las compuertas controladores, las cuales utilizan una función sigmoidea con salidas de 0 a 1, por lo tanto cuando sus salidas sean 0s se cierra la compuerta y si son 1s se abren las compuertas. Las compuertas que conforman a una celda LSTM son la de olvido $f_{(t)}$ que controla que parte del estado a largo plazo debería ser eliminado. La compuerta de entrada $i_{(t)}$ controla la parte de $g_{(t)}$ que debe ser agregado al estado a largo plazo, por último, la capa de salida $o_{(t)}$ la cual controla que partes del estado a largo plazo debe ser leído y seleccionado como salida en cada paso tato a $h_{(t)}$ como a $y_{(t)}$.

las celdas LTSM pueden aprender a identificar las entradas importantes y almacenarlas en el vector de largo plazo, de tal forma, que la almacene mientras que sea necesario y extraerlas cuando se requiera. Por ello, este tipo de RNN es muy eficiente cuando se requiere identificar patrones en series de tiempo, textos largos, demás información secuencial.

Celda con Unidad de Compuerta Recurrente

Las Redes Neuronales Recurrentes con Unidad de Compuerta Recurrente, fue propuesta por Se puede ver a la celda GRU como una versión simplificada de las conocidas celdas LSTM, además de lograr un rendimiento equiparable a éstas, han ganado popularidad. Las diferencias mas significativas entre las demás Redes Neuronales Recurrentes son que ambos vectores de estados estás integrados en un solo vector $h_{(t)}$, la forma de controlar las compuertas, y el número de compuertas de salida.

En el caso de las compuertas GRU un solo elemento controla tanto la compuerta de olvido como la compuerta de entrada. Si el controlador tiene una salida de 1, la compuerta de entrada es abierta y la compuerta de olvido es cerrada. Por otro lado, si la compuerta del controlador tiene como salida 0, sucede lo opuesto, la compuerta de entrada es cerrada y la compuerta de olvido es abierta. Siempre que la memoria debe almacenar, la locación en donde será guardado es borra primero, acción que es contrastante con las celdas del tipo LSTM.

Otro aspecto a resaltar de las celdas GRU es el hecho que no tienen una compuerta de salida, esto es por que el vector de estado es la salida en paso. Sin embargo, se incluye una nueva compuerta que controla que parte del estado anterior va a ser mostrado en la capa principal.

En la ecuación 2.33 se muestra cómo se calcula el estado de la celda en cada paso.

$$\begin{aligned} z_{(t)} &= \sigma(W_{xz}^T \cdot x_{(t)} + W_{hz}^T \cdot h_{(t-1)}) \\ r_{(t)} &= \sigma(W_{xr}^T \cdot x_{(t)} + W_{hr}^T \cdot h_{(t-1)}) \\ g_{(t)} &= \tanh(W_{xg}^T \cdot x_{(t)} + W_{hg}^T \cdot (r_{(t)} \otimes h_{(t-1)})) \\ h_{(t)} &= (1 - z_{(t)}) \otimes \tanh(W_{xg}^T \cdot h_{(t-1)} + z_{(t)} \otimes g_{(t)})) \end{aligned} \tag{2.33}$$

El éxito de las Redes Neuronales Recurrentes se debe principalmente a las celdas tipo LSTM o GRU, sobre todo en aplicaciones en el procesamiento de lenguaje natural.

2.5.4. Representación de palabras

Para crear modelos para representar algún fenómeno que involucre el procesamiento de lenguaje natural se requiere representar cada una de las palabras de tal forma que sean procesables por la computadora. La forma tradicional de representación de las palabras es llamada *one-hot-encoding* en la cual cada una de las palabras se presenta como un vector en un espacio binario, en donde el valor en una dimensión en el vector es 1, en el cual se representa una palabra única del vocabulario. De tal forma que dado un vocabulario de N palabras (ej. 100,000), cada uno de los vectores que representen cada una de las palabras en el vocabulario tendrá una dimensión de N, lleno de valores 0 en todas sus dimensiones a excepción de una (i-ésima dimensión) que representa a una palabra en particular. Esta representación de palabras es particularmente ineficiente en un vocabulario de gran tamaño, aunado con la naturaleza de los dispersos en la cual no se logra captar información acerca de las relaciones entre las palabras.

De forma ideal, se desearía que la representación de las palabras capturen cierta información de ellas, por ejemplo agrupar por naturaleza de las palabras en sustantivos, verbos, plurales, etc. De esta forma los modelos podrían generalizar mejor dada la similaridad de las palabras.

Una de las formas de representación de palabras relativamente reciente, son los modelos basados en Redes Neuronales para embeber palabras, es decir, representar cada una de las palabras en un espacio continuo y de baja dimensionalidad (cientos de dimensiones en lugar de decenas o cientos de miles), que adicionalmente captura cierta información de similaridad de las palabras.

Estos modelos para embeber las palabras en este nuevo espacio, utilizan Redes Neuronales para que a través del entrenamiento crean nuevas representaciones de las palabras, construido en las neuronas de la capa oculta de las Redes Neuronales. El objetivo de éstos modelos puede ser diferente unos de otros, generalmente se busca la agrupación de palabras similares de tal forma que su nueva representación esté organizada de cierta forma que aporte información. Un ejemplo de estás representaciones puede resultar en la posibilidad discriminar aquellas palabras divididas por género, plurales, sustantivos, adjetivos, entre otros.

En los modelos de redes Neuronales para embeber palabras, tanto la información de entrada como la de salida se representan usualmente en el formato *one-hot-encoding*, siendo la nueva representación de las palabras en el espacio continuo y de baja dimensionalidad representado por la activación de las neuronas en la capa oculta.

Una de las grandes ventajas de utilizar este tipo de modelos de representación de palabras es que se pueden distribuir estás representaciones, es decir, no se tiene que entrenar un modelo cada vez que se quiera hacer uso de este tipo de representaciones, sino que se pueden reutilizar cada vez. Además, los pesos obtenidos de modelos anteriores se pueden utilizar para inicializar los pesos de algún nuevo modelo y así lograr converger más rápido el nuevo modelo. Las antes mencionados son algunas de las formas en las cuales se puede reutilizar estas representaciones, sin embargo, existe todo una gama de diversas posibilidades para su reutilización, sobre todo al aplicar técnicas de transferencia de conocimiento, el cual se presentará en secciones posteriores.

Las técnicas de incrustación también son útiles en la representación de atributos categóricos, en los cuales tengan una gran cantidad de posibles valores y tengan ciertas similitudes complejas.

Existen diversos modelos de incrustación de palabras usando Redes Neuronales, unos de los más utilizados recientemente son los modelos de bolsa de palabras continua (CBOW, por

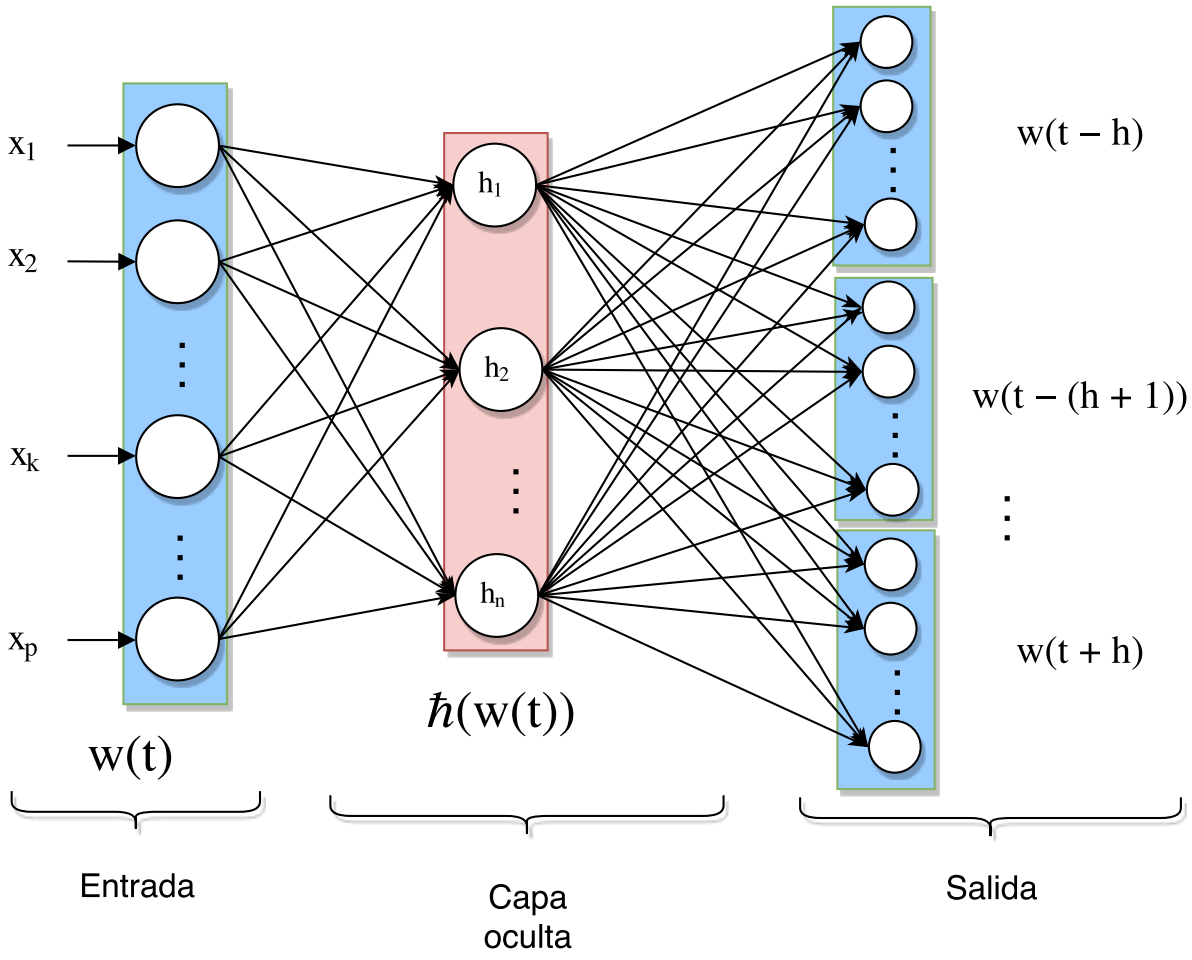

Figura 2.2: Espacio de palabras incrustadas (Skip-gram)

sus siglas en inglés de *Continous Bag of Words*) y Skip-gram propuestos por Mikolov *et al.* (2013a).

El modelo skip-gram tiene el propósito de predecir un contexto de palabras dada una palabra de entrada $w_{context} = (w_{t-i}, \ldots, w_{t-1}, w_{t+1}, \ldots, w_{t+i})$ donde $i \geq 1$ data una palabra $w(t)$ como entrada (figura 2.2). La proyección $\hbar(w(t))$ es una representación en un espacio continuo y de baja dimensionalidad de la palabra $w(t)$.

El modelo CBOW se basa en la misma idea detrás del modelo de incrustación de palabras *skipgram*, sin embargo, a diferencia de éste, tiene el propósito de predecir una palabra $w(t)$ a partir de una entrada que corresponde a un conjunto de palabras que representan el contexto

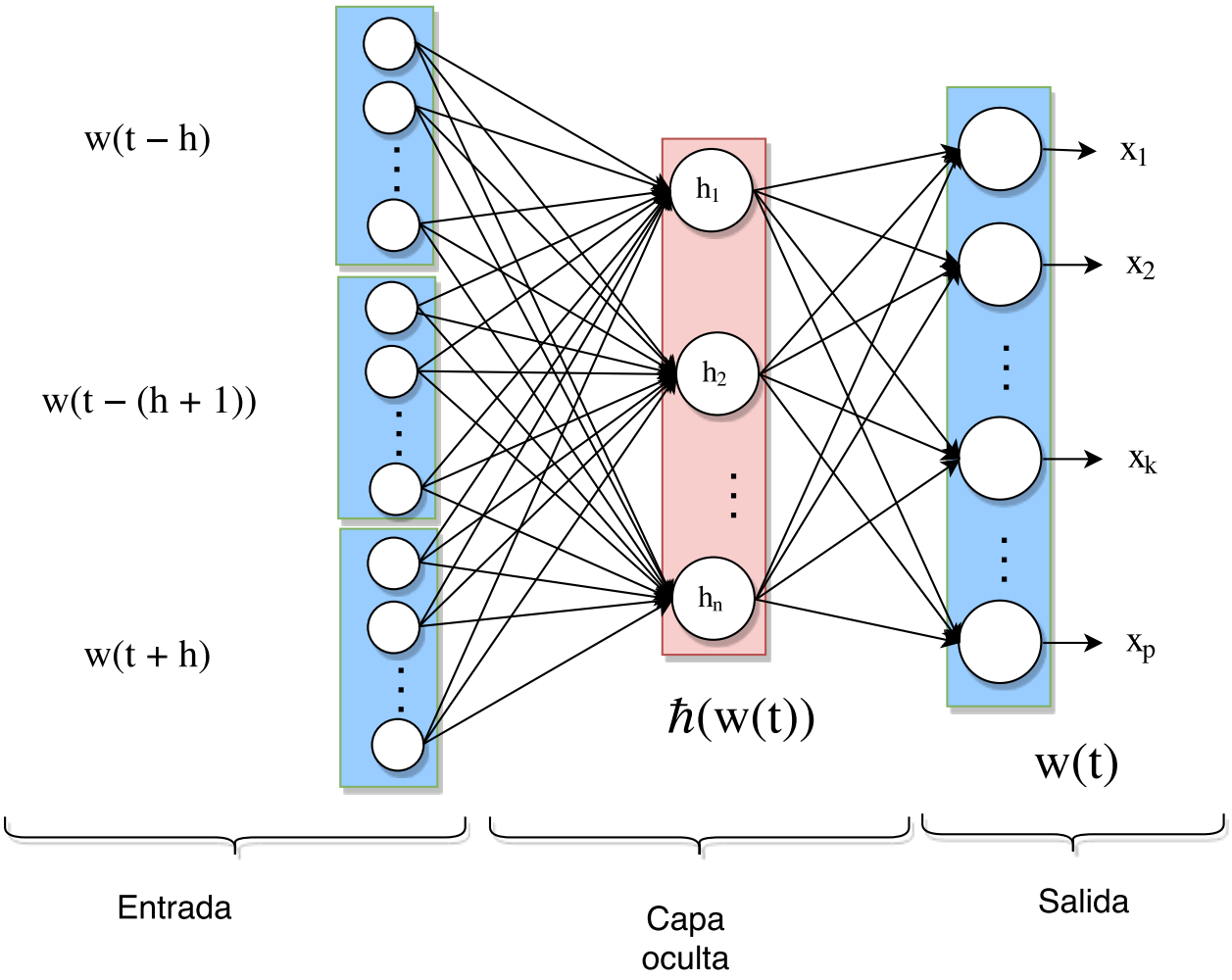

Figura 2.3: Espacio de palabras incrustadas (CBOW)

de la palabra $w(t)$ representador por $w_{context} = (w_{t-i}, \ldots, w_{t-1}, w_{t+1}, \ldots, w_{t+i})$ donde $i \geq 1$ (figura 2.3). La proyección $\hbar(w(t))$ es una representación en un espacio continuo y de baja dimensionalidad de la palabra $w(t)$.

2.6. Trabajos relacionados

Para resolver la problemática expuesta por diversos investigadores se han propuesto la automatización de la adquisición de conocimiento de un dominio dado, específicamente la

extracción de conceptos relevantes de un tema (Atapattu *et al.*, 2012; Conde *et al.*, 2014; Gantayat e Iyer, 2011; Zouaq y Nkambou, 2009). Las metodologías utilizadas para el proceso de automatización de adquisición de conocimiento según la bibliografía revisada en el rango de tiempo 2009 al 2017 son las siguientes: clasificación de documentos usando Naive Bayes (Kohilavani *et al.*, 2009), minería de datos (Nkambou *et al.*, 2009), extracción de información (Pirrone *et al.*, 2010), análisis de lenguaje natural y reglas heurísticas (Conde *et al.*, 2014). Por otra parte en cuanto al tipo de representación de conocimiento utilizado por los autores anteriormente citados fueron en su mayoría sistemas basados en ontologías, con las cuales se logra una fácil representación y distribución del modelo de conocimiento explícitamente referido por Zhiping y Yu (2009).

2.6.1. Sistemas de (semi-)automatización de adquisición de conocimiento

En base a la revisión de literatura descrita anteriormente, se hace un análisis mas profundo de las técnicas utilizadas para extracción de conceptos relevantes del dominio así como las relaciones entre los conceptos extraídos para formar el modelo de conocimiento utilizado principalmente en los Sistemas de Tutoría Inteligente

La generación automática del módulo de dominio a partir de libros electrónicos mediante el sistema DOM-Sortze (Larrañaga *et al.*, 2014), se implementaron técnicas de procesamiento de lenguaje natural (NLP, por sus siglas en inglés de *Natural Language Procesing*) y métodos heurísticos para lograr la adquisición de conceptos relevantes de un dominio a través del análisis en libros electrónicos. En esta esta propuesta se definió el Modelo del Dominio en dos niveles:

El primer nivel es la ontología de aprendizaje del dominio (OAD): Consta de los conceptos principales de un determinado dominio y las relaciones pedagógicas entre ellos. Las relaciones

pedagógicas utilizadas fueron las siguientes:

- Relación de tipo semántica es-un: la relación $X \overrightarrow{\quad}_{es-un} Y$ indica que X es un tipo particular de Y.
- Relación de tipo semántica parte-de: la relación $X \overrightarrow{\quad}_{parte-de} Y$ indica que X es parte de Y.
- Relación del tipo pre-requisito: la relación $X \overrightarrow{\quad}_{pre-requisito} Y$ indica que X debe ser aprendido antes de intentar aprender Y.
- Relación del tipo siguiente: la relación $X \overrightarrow{\quad}_{siguiente} Y$ indica que se recomienda aprender X antes de Y.

El segundo nivel como el conjunto de objetos de aprendizaje (OA): Que se define como recursos didácticos, secciones del documento usadas en las sesiones de aprendizaje (ej. definiciones, ejercicios), reusables y enriquecidos con metadatos. La metodología usada por Larrañaga para el desarrollo del modelo de dominio es el siguiente: Pre-procesamiento del libro de texto: Esta etapa consiste en la preparación del documento para ser posible el proceso de extracción de conocimiento y así formar los dos niveles del módulo del dominio antes descrito. Formación de la OAD: En esta fase los conceptos más importantes del dominio y las relaciones pedagógicas entre estos son identificados y representados en la OAD. Formación de los OA: En esta etapa los OA son identificados y generados. La etapa de formación de la OAD se divide en dos etapas, las cuales son: En la primera de ellas de realiza un análisis del temario. En primera instancia se realiza un análisis básico, en donde los tópicos son extraídos del temario debido a la homogeneidad de su representación, cada elemento indexado se considera un tópico del dominio. En segundo lugar se realiza un análisis heurístico en donde se categoriza las relaciones entre los conceptos resultantes en el análisis básico y se identifican nuevas relaciones. La segunda etapa en la formación de la OAD se realiza un análisis en el

Tabla 2.3: Resultados de DOM-Sortze Larrañaga *et al.* (2014)

Tipo de relación	Exhaustividad	Precisión
parte de	55.81	-
pre-requisito	23.08	-
es un	30.30	56.10
siguiente	61.54	66.67

cuerpo del documento. En esta fase se identifican nuevos conceptos e identificación de relaciones pedagógicas entre éstos mediante el uso de técnicas de procesamiento de lenguaje natural y procesos basados en patrones respectivamente.

La metodología para evaluación del sistema se realizó mediante la comparación de los conceptos y relaciones identificados por personas que hacen el diseño instruccional. En los resultados obtenidos (Tabla 2.3) se puede observar que las relaciones de tipo pre-requisito y es-un tuvieron el porcentaje más bajo en exhaustividad. El autor en sus conclusiones hace la observación que ninguna de las heurísticas propuestas determinaron relaciones del tipo es-un en el temario. Por otro lado las reglas propuestas para identificar las relaciones del tipo siguiente no determinaron ninguna relación en el cuerpo del documento. Además hace mención de que las relaciones del tipo pre-requisito son por mucho las más difíciles de identificar, siendo las relaciones identificadas extraídas del temario únicamente.

En el trabajo de Conde *et al.* (2014), se realizó un experimento con el sistema DOM-Sortze (Larrañaga *et al.*, 2014) en el lenguaje inglés, además de agregar un proceso de elicitación de Wikipedia. Para aumentar la exhaustividad en las relaciones es-un utilizó una mejora en el proceso de elicitación usando Wikipedia. Los resultados obtenidos fueron una mejora del 29.33 % en la exhaustividad de las relaciones es-un, mientras que la precisión apenas se vio afectada. La precisión en las relaciones parte-de fue ligeramente mejorada (5.09 %), siendo que hubo un decremento mínimo en la exhaustividad (2.20 %). El proceso de elicitación utilizando Wikipedia fue el siguiente:

1. Identificación de los grupos de nodos hermanos del OAD extraídos del temario.
2. Selección de grupos en donde las relaciones entre de los nodos hojas fueran del tipo parte-de.
3. Normalización de los nodos.
4. Enlazar cada nodo normalizado hacia artículos de Wikipedia en donde aparezca el nodo normalizado.
5. Realizar un proceso de desambiguación para enlazar cada nodo normalizado a solo un artículo de Wikipedia.
6. Utilización de técnica de taxonomía para buscar ancestros comunes.
7. Inferir las relaciones de tipo es-un en aquellos grupos donde comparten un ancestro común, así hasta llegar al último nivel de la taxonomía.

En este caso, los resultados fueron significativos a fin de identificar más relaciones del tipo es-un. El autor reporta que el uso de Wikipedia es una vía rápida para mejorar el Aprendizaje Ontológico Educacional multilingüe.

En el trabajo de Atapattu *et al.* (2012) se realizó un sistema para automatizar la extracción de tópicos y sus relaciones de un tema a partir de diapositivas en Power Point utilizando técnicas de aprendizaje ontológico y procesamiento de lenguaje natural. La metodología empleada consiste en técnicas de extracción de conceptos y algoritmos de extracción de jerarquías entre conceptos. El proceso de extracción de conceptos consiste de las siguientes cuatro fases:

1. **Pre-procesamiento**: Se normaliza el texto contenido en diapositivas didácticas hechas por profesores. La normalización incluye:

   a) separar las oraciones por punto, coma o punto y coma.

   b) reemplazar símbolos no alfanuméricos

   c) procesamiento de signos de puntuación

   d) expandir abreviaciones

   e) remover espacios en blanco

2. **Etiquetado en procesamiento de lenguaje natural**: en este proceso se divide en dos sub-procesos:

   a) lematización, consiste en la eliminación de partes no esenciales de la palabra como sufijos y prefijos.

   b) etiquetado, consiste en la identificación de sustantivos, adjetivos, adverbios, etc. en frases o sentencias.

3. **Pos-procesamiento**: consiste en la eliminación de palabras comunes que no tienen gran importancia.

4. **Modelo de ponderación**: se realiza la ponderación de los términos obtenidos en la fase anterior con el objetivo de identificar los que son mas relevantes en el dominio. En este proceso se toman en cuenta 4 modelos de ponderación, los cuales son:

   a) frecuencia de término

   b) conteo de n-grama y posicionamiento

   c) análisis tipográfico

   Los modelos de ponderación definidos se basan en los factores de ponderación antes mencionados. La nomenclatura utilizada es N:sin peso, L: peso de logaritmo de frecuencia de término, T:peso del conteo de n-grama, E:peso de probabilidad de enfatizado.

Tabla 2.4: Resultados del sistema (Atapattu *et al.*, 2012)

Modelo	NNN			LNN			LTN			LTE		
Métrica	P	R	F	P	R	F	P	R	F	P	R	F
Muestra 1	0.238	0.876	0.375	0.636	0.157	0.252	0.535	0.426	0.475	0.534	0.528	0.531
Muestra 2	0.242	0.893	0.381	0.35	0.148	0.208	0.355	0.34	0.347	0.278	0.723	0.402
Muestra 3	0.367	0.925	0.526	0.678	0.158	0.256	0.711	0.658	0.683	0.611	0.733	0.666
Muestra 4	0.219	0.847	0.348	0.305	0.239	0.268	0.349	0.63	0.449	0.329	0.63	0.432
Muestra 5	0.244	0.801	0.374	0.583	0.143	0.23	0.477	0.438	0.457	0.471	0.458	0.465
Promedio	**0.262**	**0.8684**	**0.4008**	**0.5104**	**0.169**	**0.2428**	**0.4854**	**0.4984**	**0.4822**	**0.4446**	**0.6144**	**0.4992**

a) **NNN**: No se aplican modelos de ponderación.

b) **LNN**: Este modelo considera el logaritmo de frecuencia para refinamiento de la lista de conceptos extraídos.

c) **LTN**: Este modelo considera el peso acumulado del logaritmo de frecuencia y conteo de n-grama como factor para filtrar los conceptos mas importantes.

d) **LTE**: Este modelo calcula el peso acumulado del logaritmo de frecuencia, conteo de n-grama y la probabilidad de enfatizado como el factor de selección.

El proceso de extracción de relaciones entre conceptos implementado basado en la propuesta de Cimiano (2006), "top-down". Se realiza un análisis de los niveles de aparición de términos (título, subtitulo, "bullet", "sub-bullet") para establecer de forma jerárquica los conceptos identificados. Los resultados obtenidos del sistema (tabla 2.4) muestran que al no aplicar ningún modelo de ponderación para filtrar los conceptos extraídos se obtiene la exhaustividad mas alta sin embargo la precisión mas baja, siendo el modelo LTN y LTE los que obtuvieron mejor resultado en cuanto a promedio (0.48 y 0.519 respectivamente) y desviación estándar (0.008 y 0.086) de las métricas utilizadas.

Los trabajos relacionados descritos en esta sección requieren un conjunto de datos con una estructura definida, lo cual se puede considerar un problema si se desea aplicar los métodos en dominios donde no se tenga acceso a este tipo de recursos. La principal aportación de esta tesis es la metodología para la identificación de relaciones semánticas empleando técnicas de aprendizaje profundo en datos no estructurados, por lo que no se requiere un formato

específico de datos, con lo cual no se limita a un determinado dominio de aplicación.

Capítulo 3

Análisis de relaciones semánticas en vectores de palabras embebidas

La detección de relaciones entre entidades es una tarea importante en la construcción de bases de conocimiento, brinda la posibilidad de razonar y descubrir nuevo conocimiento partiendo del conocimiento actual.

Hay una especial atención en la detección de relaciones entre entidades sin hacer uso de procesos manualmente creados, superando la dependencia y la limitación del dominio de aplicación, lo cual implica estas propuestas. Recursos manualmente creados como WordNet [1] están enfocados en dominios genéricos (Caraballo, 1999; Xiong y Ji, 2016; Rios-Alvarado *et al.*, 2013; Colace *et al.*, 2014), sin embargo, mientras más especializado sea el dominio es más difícil encontrar recursos estructurados útiles para mejorar el proceso de construcción automática de bases de conocimiento. Mientras haya menor dependencia del idioma, características lingüísticas y más métodos automatizados para capturar la semántica de la información textual, se incrementa la efectividad en el proceso de creación automática de

[1] https://wordnet.princeton.edu/wordnet/

bases de conocimiento en dominios especializados. El impacto del proceso de creación de base de datos de forma automática es mayor cuando los recursos disponibles no tienen una estructura definida, por ejemplo corpus de redes sociales, libros de texto, notas de clase, etc.

Una de las nuevas representaciones de palabras esta compuesta por modelos basados en redes neuronales artificiales, llamados palabras incrustadas o embebidas (WE, por sus siglas en inglés de *Word Embeddings*) . Las representaciones de palabras embebidas tienen el objetivo de reducir la dimensionalidad en donde cada palabra es representada, además de usar un espacio real en lugar de uno binario. Los modelos de Redes Neuronales aplicados al lenguaje han sido ampliamente usados en las técnicas de palabras embebidas (Mikolov *et al.*, 2013a,b; Zhao *et al.*, 2015), en particular el modelo word2vec de (Mikolov *et al.*, 2013b), se ha demostrado que muchas regularidades sintácticas y semánticas se capturar en este nuevo espacio de representación de las palabras.

El propósito de este trabajo es el medir que tan bien los vectores de palabras embebidas puede capturar relaciones semánticas entre palabras. Las propuestas actuales para clasificar relaciones semánticas tales como causa-efecto, producto-productor, mensaje-tópico, contenido-contenedor, etc. usan el contexto de las entidades para predecir la clase de relación semántica Qin *et al.* (2016); Komninos (2016). Sin embargo, en este caso proponemos aprender relaciones semánticas más básicas usando solo la representación de palabras embebidas sin el contexto de cada término.

El modelo de clasificación que fue seleccionado es el estado del arte hasta el momento en la detección de relaciones semánticas Qin *et al.* (2016). Hay un consenso que las CNN logran alcanzar buenos resultados. En este trabajo aquellos modelos en el estado del arte son adaptados para recibir como entrada solo dos vectores de palabras embebidas para obtener como resultado la clase a la cual pertenece su relación.

3.1. Trabajo Relacionado

El uso de la representación de palabras en espacios continuos de vectores para identificar relaciones tanto semánticas como sintácticas ha incrementado recientemente (Fu *et al.*, 2014, 2015; Takase *et al.*, 2016; Hyland *et al.*, 2016; Fan *et al.*, 2016). El uso de estas nuevas representaciones cuentan con un gran potencial dado la posibilidad de a partir de datos no estructurados, identificar relaciones sintácticas como semánticas. En el trabajo de Fu *et al.* (2015) se observa que existen regularidades no lineales entre los vectores en donde se muestran relaciones semánticas, como la expresión bien conocida $v(\texttt{king}) - v(\texttt{queen}) \equiv v(\texttt{man}) - v(\texttt{woman})$ no se puede generalizar la relación semántica hiperónimo-hipónimo entre todo el espacio vectorial, es decir, no se puede generalizar que el vector resultante por $v(\texttt{man}) - v(\texttt{king})$ represente la relación es-un, y con ella, inferir nuevas jerarquías en todo el espacio vectorial.

En el trabajo de Fu *et al.* (2015) se analizaron que hay regiones en donde estas regularidades son compartidas en regiones locales o circundantes entre las palabras. Una de las hipótesis que se desprenden de este estudio es que estas regularidades en las regiones locales pueden ser aprendidas mediante modelos de aprendizaje no lineales. En este experimento se seleccionó el estado del arte en la clasificación de relaciones semánticas usando vectores de palabras embebidas representadas en un espacio continuo y de relativamente baja dimensionalidad Chen *et al.* (2015); Qin *et al.* (2016); dos Santos *et al.* (2015). En lo sucesivo de este capitulo se realizará una adaptación de este tipo de modelos para clasificar relaciones semánticas entre palabras en base a su representación en el espacio continuo de baja dimensión o espacio embebido.

3.2. Metodología

La metodología para implementar esta primer propuesta para la extracción de relaciones semánticas se muestra en la figura 3.1. La etapa de preparación de los datos consiste en cuatro pasos:

1. Transformación del corpus: el corpus del dominio seleccionado se transforma a un espacio vectorial continuo y de baja dimensionalidad.

2. Selección de entidades: se identifican y extraen los términos (entidades) que tienen entre sí una relación semántica, ésto solo si existen en el corpus y por ende en el nuevo espacio embebido.

3. Extracción de relaciones: Se extraen de la base de datos de relaciones que hay entre los términos que se encuentran en el corpus.

4. Transformación de los términos al espacio embebido: Los términos extraídos en el paso 2, son convertidos al espacio continuo y de baja dimensionalidad.

3.2.1. Palabras embebidas

El primer paso en nuestra propuesta, la generación de la nueva representación de palabras ahora en un espacio continuo y de baja dimensionalidad, se empleó el modelo skip-gram propuesto por Mikolov *et al.* (2013a) entrenado con artículos de Wikipedia (al año 2016).

El modelo skip-gram tiene el propósito de predecir un contexto de palabras dada una palabra de entrada $w_{context} = (w_{t-i}, \ldots, w_{t-1}, w_{t+1}, \ldots, w_{t+i})$ donde $i \geq 1$ data una palabra $w(t)$ como entrada (figura 3.2). La proyección es una representación en un espacio continuo y de baja dimensionalidad de la palabra $w(t)$.

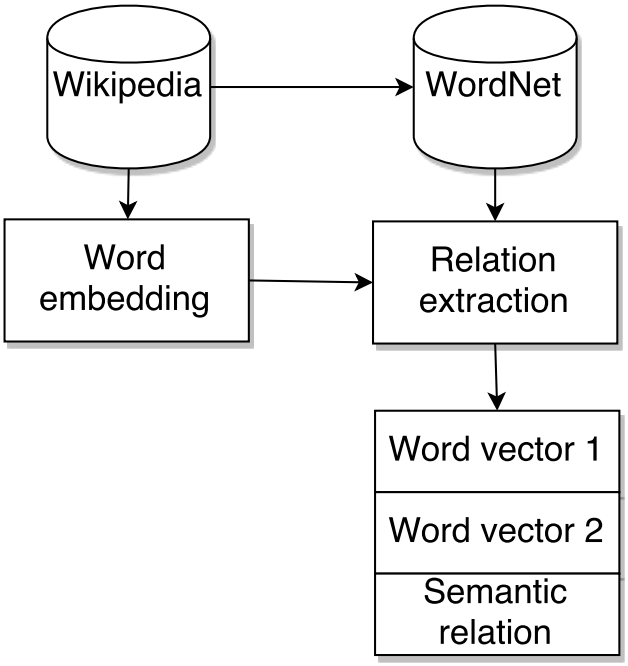

Figura 3.1: Metodología para extracción de relaciones semánticas.

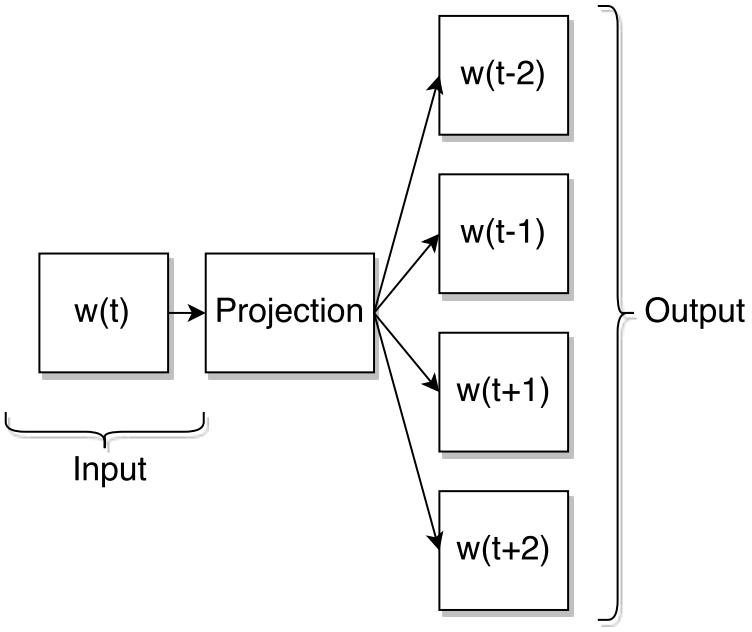

Figura 3.2: Modelo Skip-gram

Relaciones	Representación	Relación inversa
Hiperónimo	$X\overrightarrow{hiperonimo}Y$	$Y\overrightarrow{hiponimo}X$
Hipónimo	$X\overrightarrow{hiponimo}Y$	$Y\overrightarrow{hiperonimo}X$
Holónimo	$X\overrightarrow{holonimo}Y$	$Y\overrightarrow{meronimo}X$
Merónimo	$X\overrightarrow{meronimo}Y$	$Y\overrightarrow{holonimo}X$

Tabla 3.1: Relaciones semánticas

3.2.2. Selección de entidades

Las entidades usadas para obtener las relaciones semánticas entre éstas deben de satisfacer la condición en la cual ambas entidades (e_i y e_2) dada una relación $\mathcal{R}_i = (e_1, e_2, s_i)$ debe de existir en el espacio de palabras embebidas, donde s_i representa el tipo se relación semántica. En otras palabras, ambas entidades deben de existir en el corpus usado en el entrenamiento de la el modelo neural del lenguaje de representación de palabras.

3.2.3. Extracción de relaciones semánticas

Las relaciones semánticas extraídas de la base de datos WordNet están representadas por el conjunto $\mathcal{S} = \{$*hiperónimo*, *hipónimo*, *holónimo* , *merónimo*$\}$.

Las relaciones semánticas son extraídas como sigue, por cada entidad e_i que existen las entidades seleccionadas del paso anterior y que están relacionadas por $\mathcal{S}$ relaciones de la siguiente forma $\mathcal{R}_i = (e_1, e_2, s_i)$, donde $s_i \in \mathcal{S}$.

En la tabla 3.1 se muestran las equivalencias de las relaciones semánticas seleccionadas para predecir en este experimento.

3.2.4. Relaciones semánticas en el espacio de representación de palabras embebidas

Cada una de las relaciones extraídas en tareas previas están representadas como una matriz de dimensiones 2x$\mathcal{D}$, donde $\mathcal{D}$, es la dimensión del espacio de representación embebida, la cuál suele estar en un rango de 100-1000. Todas las entidades en las relaciones están representadas en el espacio vectorial continuo y de baja dimensionalidad asociadas a la categoría que pertenece a la clase de relación $\mathcal{S}$.

3.2.5. Evaluación

Para medir que tan bien los vectores de palabras embebidas capturan las relaciones semánticas en el corpus, se emplearon las métricas empleadas en clasificación, exactitud (ecuación 3.1), precisión (ecuación 3.2), exhaustividad (ecuación 3.3) y la métrica F1 (ecuación 3.4). Para validar el experimento se utilizo validación cruzada k-fold donde k es 8.

Exactitud

$$Exactitud(y, \hat{y}) = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} \hat{y}_i = y_i \tag{3.1}$$

donde, y_i es la clase de salida deseada y $\hat{y}_i$ la salida actual del modelo.

Precisión

$$Precision = \frac{tp}{tp + fp} \tag{3.2}$$

donde, tp son verdaderos positivos y fp falsos positivos.

Exhaustividad

$$Exhaustividad = \frac{tp}{tp + fn} \tag{3.3}$$

donde, tp son verdaderos positivos y fp falsos positivos.

Medida F1

$$Fmeasure = \frac{2 * Precision * Exhaustividad}{Precision + Exhaustividad} \tag{3.4}$$

3.3. Configuración del experimento

3.3.1. Representación de las palabras

La entrada del clasificador es una matriz de tamaño 2x400 formada por dos vectores renglones, cada uno siendo la representación de cada entidad en las relaciones semánticas transformadas en el espació vectorial embebido. La representación de dichos vectores se realizó a través del modelo skip-gram entrenado con todos los artículos de Wikipedia con parámetros de tamaño de ventana de 5 y una dimensión resultante de tamaño 400.

3.3.2. Relaciones semánticas

WordNet es una base de conocimiento manualmente construida para representar relaciones entre entidades. En este proceso se aprovecha esta base de conocimientos para extraer las relaciones semánticas entre entidades que existen en el espacio embebido. De forma aleatoria se seleccionaron 9,000 relaciones existentes entre pares de entidades de una de las clases

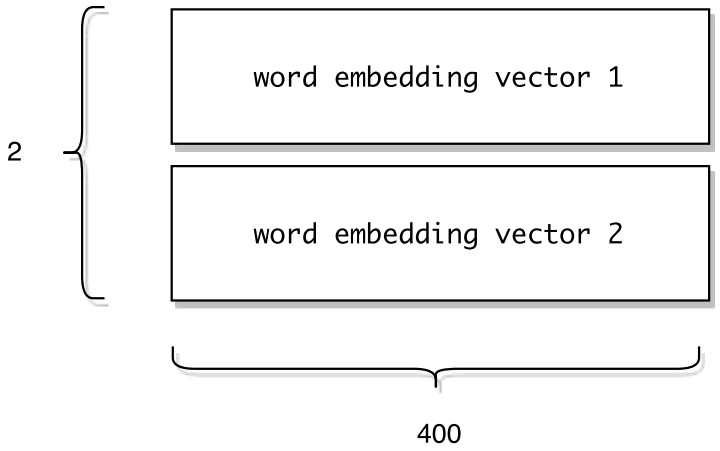

Figura 3.3: Forma de la entrada de datos

(*hiperónimo*, *hipónimo*, *holónimo*, *merónimo*) para mantener un número balanceado de instancias, conformado por un total de 36,000 instancias en todo el conjunto de datos.

3.3.3. Redes Neuronales Convolutivas

La entrada del clasificador es la concatenación de dos vectores de palabras embebidas que fueron transformadas por el modelo skip-gram. Basados en los modelos del estado del arte, se seleccionó una arquitectura de una Red Neuronal Convolutiva (CNN). Dada la naturaleza de este tipo de redes se tenía la hipótesis que en los espacios de características generados por este tipo de red, detectaran las regularidades entre las palabras en el espacio embebido. La vista general de la arquitectura propuesta se muestra en la figura 3.4, ésta arquitectura está basada en la propuesta de Qin *et al.* (2016) para clasificación de relaciones semánticas. El objetivo de este trabajo es el de medir que tan bien las regularidades semánticas pueden capturar los vectores de palabras embebidas sin el conjunto de palabras que conforman el contexto de cada entidad. Debido que el conjunto de palabras del contexto son removidas, se simplifica en gran medida el modelo del estado del arte.

En la tabla 3.2 se muestra la configuración de cada capa usada en la arquitectura de

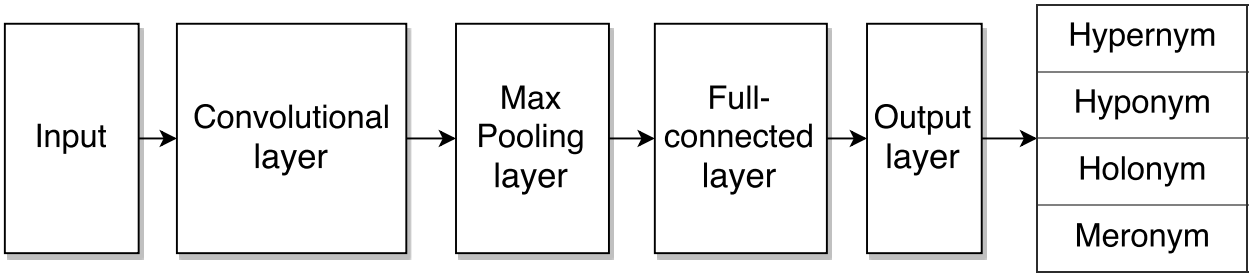

Figura 3.4: Arquitectura de la Red Neuronal Convolutiva

Capas	Dimensión
Entrada	2x400
Convolutiva	100
Agrupamiento	100
Totalmente conectada	800
Salida	4

Tabla 3.2: Dimensiones de las capas de la Red Neuronal Convolutiva

la Red Neuronal propuesta. El tamaño del filtro que se estableció fue de 2x2. En la capa completamente conectada se empleo la función de activación ReLU y un Dropout de 0.4. El algoritmo de aprendizaje utilizado fue Adam Lee y Myung (2017) con una tasa de aprendizaje de $\lambda = 0{,}001$, $\beta_1 = 0{,}9$, $\beta_2 = 0{,}999$, $\epsilon = 1e - 08$.

3.4. Resultados

El resultado se evaluó mediante el uso de validación cruzada K-Fold en donde $k = 8$. La métrica utilizada para su evaluación fue el promedio ponderado de precisión y exhaustividad conocido como F1. El resultado obtenido en la métrica F1 en el promedio de los 8 folds fue de 0.946. En la tabla 3.3 se muestra el resultado en cada segmento del conjunto de datos de prueba. Basándose en los resultados, se observa que las técnicas para embeber palabras en un espacio continuo logran un buen resultado en la captura de relaciones semánticas del tipo es-un y parte-de.

Fold	Exactitud	Precisión	Exhaustividad	F1
1	0.943	0.952	0.943	0.948
2	0.934	0.940	0.933	0.937
3	0.940	0.947	0.943	0.945
4	0.946	0.953	0.943	0.948
5	0.950	0.957	0.947	0.952
6	0.945	0.951	0.950	0.950
7	0.951	0.953	0.950	0.952
8	0.959	0.962	0.961	0.961
Promedio	0.946	0.952	0.946	0.949
σ	0.0073	0.0064	0.0078	0.0069

Tabla 3.3: Resultados del experimento

Tomando en consideración que solamente se emplearon los vectores de palabras embebidas como entrada para detectar las relaciones entre entidades, el resultado de la evaluación fue considerablemente eficaz para clasificar relaciones semánticas.

Tomando en cuenta la transitividad en las relaciones, así como se muestra en la tabla 3.1), se puede aprovechar como un parámetro más para evaluar la consistencia del modelo. Dado una tupla conformada por $\mathcal{T} =< entity1, entity2 >$ y una relación resultante $\mathcal{R}_i$, puede ser evaluado empíricamente si la salida es confiable si al invertir el par de entidades, es decir, formar una tupla con la forma $< entity2, entity1 >$, se debería obtener la relación semántica inversa de $\mathcal{R}_i$. Por ejemplo, $< dog, canine >: hyponym \equiv < canine, dog >: hypernym$.

Capítulo 4

Metodología propuesta para inferencia de relaciones semánticas

4.1. Descripción general de la metodología propuesta

El propósito de esta metodología es la construcción de una ontología del dominio a partir de texto corpus no estructurados a través de técnicas de Aprendizaje Profundo. Esta propuesta consiste de 12 tareas que inicia con la selección de corpus genérico y termina en la generación de la ontología; estas tareas se pueden agrupar en tres fases, las cuales son: i) entrenamiento del modelo para detección de relaciones semánticas en un dominio genérico; ii)adaptación de dominio; y iii) inferencia de relaciones semánticas en un dominio especializado.

A continuación se presentan las fases y tareas propuestas en esta metodología, seguido de ello, se describe de forma detallada las condiciones y variables a considerar en cada uno de ellos.

I. Entrenamiento del modelo para detección de relaciones semánticas en un dominio genéri-

co

a) Selección del corpus genérico.

b) Selección de base de datos para extraer relaciones entre entidades.

c) Construcción automática de base de datos de relaciones semánticas de un dominio genérico, usando la técnica de supervisión distante.

d) Creación de palabras embebidas del corpus de dominio genérico.

e) Construcción de conjunto de datos conformado de relaciones semánticas del dominio genérico.

f) Entrenamiento de modelo de clasificación de relaciones semánticas.

II. Adaptación del dominio

a) Re-entrenamiento de la representación de palabras embebidas con el corpus del dominio específico.

III. Inferencia de relaciones semánticas en un dominio especializado

a) Preparación automática de conjunto de datos del dominio especializado.

b) Predicción de relaciones semánticas existentes en el dominio específico.

c) Creación de ontología basada en las relaciones identificadas.

d) Evaluación de los términos extraídos en la ontología resultante.

e) Evaluación de las relaciones semánticas extraídas en la ontología resultante.

4.1.1. Entrenamiento del modelo para detección de relaciones semánticas en un dominio genérico

Esta fase consiste en la creación de un modelo de aprendizaje para la clasificación de relaciones semánticas dado un par de entidades y el contexto compartido. En este punto se debe considerar la cantidad y calidad de información disponible, ya que las técnicas propuestas para la clasificación de relaciones semánticas no son viables cuando se tienen pocos datos, además, tratándose de procesamiento de texto, al contener una gran cantidad de variables (cada palabra) se requiere, igualmente, una gran cantidad de instancias (sentencias) para detectar patrones entre las palabras.

Como se comprobó en el capítulo anterior (capítulo 3), las regularidades semánticas (relaciones) se pueden generalizar en el espacio de las palabras embebidas, sin embargo, al no considerar el contexto compartido entre las entidades, no es factible la inferencia de relaciones en un dominio especializado debido, principalmente, a la cantidad disponible de información del dominio de interés y a la diferente distribución de datos. Por lo tanto, en esta fase se considera el contexto entre las entidades al entrenar el modelo de aprendizaje, el cual, está basado en el estado del arte en la tarea de la clasificación de relaciones semánticas Zhou *et al.* (2016).

En lo que resta de la sección se describe cada uno de los pasos involucrados en esta fase, que comienza en la selección del corpus y la base de conocimiento, y termina con el modelo de aprendizaje entrenado para la detección de relaciones semánticas entre entidades.

Selección del corpus y base de datos del dominio genérico

La selección del corpus es un paso crucial para el entrenamiento del modelo de aprendizaje para la detección de relaciones semánticas, ya que la calidad del modelo depende de la

calidad de información con el que haya sido entrenado. Hay diversos factores a considerar en este punto, tales como el tamaño del corpus, esto es debido a que los datos textuales son complejos para su procesamiento ya que el significado e intención de las palabras dependen del contexto en el cual se encuentren, por tanto, entre mas contextos se consideren se presume una mejor representación de los términos considerados y posiblemente una mejor generalización del problema. Tomando en cuenta la calidad de la información en el corpus, es importante también considerar el proceso con el cual fue creado, por ejemplo si fue construido de forma automática a través de extracción de sitios web o alguna otra fuente que pueda contener información con ruido, o bien, si fue construida de forma manual.

Por corpus de dominio genérico, se refiere a un corpus que no tenga una tópico en particular, es decir, que abarque la mayor cantidad de tópicos posibles con la idea que en algún punto se considere en el entrenamiento del modelo una distribución de datos similar al que será el dominio objetivo (o dominio especializado) que se desea generar una ontología para su representación.

Por otra parte, en cuanto a la selección de la base de conocimiento que contiene de forma explicita las relaciones entre término o entidades, se deben de considerar factores similares como los descritos en la selección del corpus, como lo son la cantidad de relaciones contenidas, la cantidad de términos o entidades, la forma con la cual fue construido y por último las relaciones que éste contiene, ya que el modelo de aprendizaje será entrenado para la detección las relaciones contenidas en la base de conocimiento, por lo tanto, es esencial considerar este factor.

Para la selección de ambas fuentes de información se debe maximizar la intersección de entidades o términos entre éstos, de tal forma que al generar el conjunto de datos de entrenamiento se obtengan la mayor cantidad de información disponible para incrementar la calidad del modelo resultante. Por lo tanto, el dominio del corpus y el dominio de la base de

conocimiento deben de ser lo más similares, esto con el objetivo de maximizar la probabilidad de un mismo vocabulario en contextos idealmente idénticos o lo más parecidos posible, esto con el objetivo de reducir la ambigüedad de los términos.

Construcción automática de base de datos de relaciones semánticas del dominio genérico

En este punto se hace uso de la técnica de supervisión distante que consiste en la construcción de un conjunto de datos de entrenamiento, mediante la integración de dos recursos, como lo son un corpus y una base de conocimiento donde se representa explícitamente algún tipo de relación semántica entre términos o entidades. La idea general de esta técnica es que si existe una relación entre dos entidades (representadas en una base de conocimiento), dicha relación deberá estar implícita en una oración que contenga ambos términos, por lo tanto, el modelo de aprendizaje podría encontrar un patrón en ese contexto (palabras circundantes) y generalizarlo para otro par de términos.

Uno de los objetivos de esta técnica es el de crear conjuntos de datos de manera automática sin la intervención del humano, lo cual implica un ahorro de esfuerzo y tiempo en el proceso de la construcción de una base de conocimiento. De manera concreta, la técnica de supervisión distante se define en esta metodología con los siguientes pasos:

1. Extracción de los términos o entidades de la base de conocimiento, representados por el conjunto $\{\tau_1, \tau_2, \ldots, \tau_{|\Gamma|}\} \in \Gamma$.

2. Extracción de instancias de relaciones semánticas contenidas en la base de conocimiento, en este punto se considera la base de conocimiento como un grafo dirigido, en el cual cada relación deberá estar conformado por una arista que va desde un vértice origen hasta un vértice destino, representados por la tupla $\langle \tau_i, \tau_j, r_k \rangle$ donde τ_i es el término origen y τ_j el término destino, unidos por la relación semántica $r_k \in \mathcal{R}$, donde $\mathcal{R}$ es

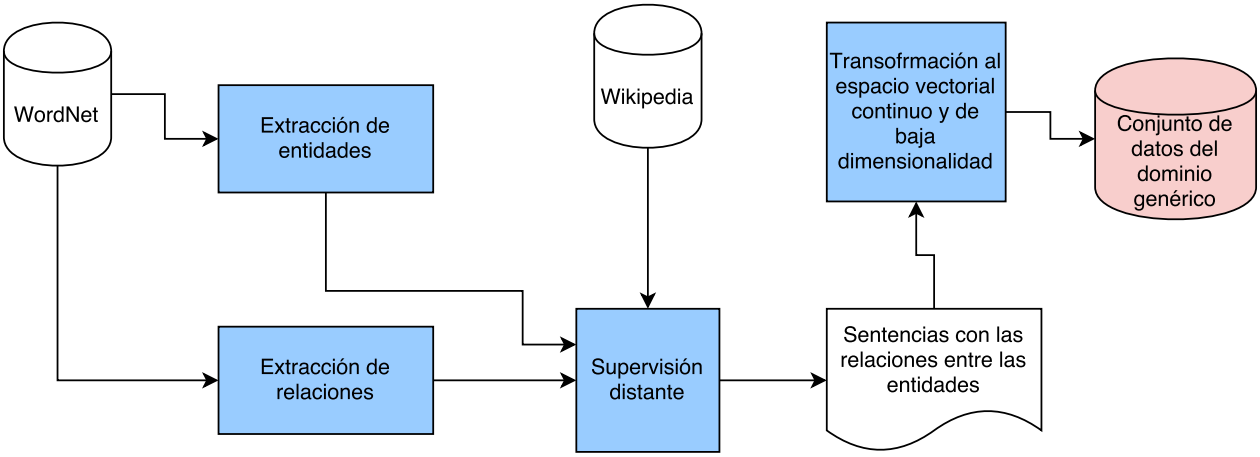

Figura 4.1: Generación de conjunto de datos de dominio genérico usando supervisión distante

el conjunto de relaciones semánticas de interés contenidas en la base de conocimiento (por ejemplo, sinonimia, hiperemia, meronomia, entro otros).

3. Por cada instancia extraída en el paso anterior, se considera los términos o entidades τ_i y τ_j para su búsqueda en el corpus seleccionado, donde ambos términos deben de estar contenidos en la misma sentencia (frase), es decir $\{\tau_i, \tau_j\} \in \rho_{i,j}$ donde $\rho_{i,j}$ es la frase extraída. De ser posible, es recomendable extraer varias sentencias en donde ambos términos estés contenidos, esto con la finalidad de considerar la mayor cantidad de contextos posible.

4. Por último, se forma el conjunto de datos de entrenamiento donde los términos y las palabras en éste se encuentran en su estado natural, representados por caracteres del idioma en cuestión. El conjunto de datos de entrenamiento estará definido por la tupla $\langle \mathcal{X}_S{}^\star, \mathcal{Y}_S \rangle = \{\langle \tau_i, \tau_j, \rho_{i,j}^m \rangle, \langle r_k^m \rangle\}^M$, donde $m = 1, 2, \ldots, M$ representa la cantidad total de instancias construidas, con la restricción que las sentencias $\rho_{i,j}$ no deben repetirse dados un par de términos τ_i y τ_j; el conjunto $\mathcal{X}_S{}^\star$ esta conformado por palabras en lenguaje natural, siendo el conjunto $\mathcal{Y}_S \in \mathcal{R}$ la salida deseada el modelo de entrenamiento.

Creación de palabras embebidas del corpus del dominio genérico

Como se menciona en capítulos anteriores, el uso de vectores de palabras embebidas mejora la representación de éstas para ser procesadas por la computadora, más específicamente para mejorar el rendimiento en modelos de aprendizaje dado la relativa baja dimensionad y el dominio en que se representan.

Existen diversos métodos para crear esta representación de las palabras, sin embargo dados los resultados obtenidos el estudio anterior (Navarro-Almanza *et al.*, 2017b) se recomienda el uso de los modelos *skip-gram* y *CBOW* (Mikolov *et al.*, 2013b), cuya representación es generada a través de un modelo RNA, en donde el proceso para la creación esta nueva representación se concibe desde una perspectiva de un problema de clasificación, sin embargo, la preparación de las entradas es de forma automática, por lo que desde la perspectiva del analista de datos se puede considerar este método como no supervisado.

En el caso del modelo *skip-gram* dada una palabra como entrada w_t) en el modelo RNA se busca obtener un conjunto de palabras con la mayor probabilidad de aparecer en el contexto donde se encuentra la palabra de entrada $w_{context} = (w_{t-i}, \ldots, w_{t-1}, w_{t+1}, \ldots, w_{t+i})$ donde $i \geq 1$ y es un parámetro a considerar llamado *ventana*, el cual se refiere a la cantidad de palabras a considerar como contexto en una oración. Donde cada palabra w_t se puede representar en *one-hot-encoding*, tal que $w(t) \in \{0,1\}^{|V|}$ donde $|V|$ es el tamaño del vocabulario a considerar. En este paso se realiza el entrenamiento del modelo neuronal lingüístico del corpus del dominio genérico (o dominio fuente), definido como $\mathcal{C}_S \in V$, donde V es el vocabulario manejado en este dominio. Una vez entrenado el modelo se genera una hipótesis $h : w_t \rightarrow w_{context}$ de tal forma que la representación de la palabra esta definida por la salida de las neuronas en la capa oculta, cuyo espacio esta representado por $\hbar(w_t) \in \mathbb{R}^D$ donde D es el número de neuronas en la capa oculta de la RNA, usualmente $300 \leq D \leq 1000$. Esta nueva representación de las palabras es de relativa baja dimensionalidad si se compara con la dimensión inicial $|V|$.

Esta representación de las palabras permiten un manejo más eficiente de los datos, se ha demostrado que en el nuevo espacio de baja dimensionalidad se logra capturar relaciones semánticas entre las palabras, por ejemplo, $v(\texttt{king}) - v(\texttt{queen}) \equiv v(\texttt{man}) - v(\texttt{woman})$ (Mikolov *et al.*, 2013b; Qin *et al.*, 2016). La representación de palabras en este espacio permite hacer uso de modelos de aprendizaje sin necesidad de crear los atributos (características o variables) manualmente.

Este paso en la metodología, finaliza con un diccionario de palabras y su representación en el espacio de proyección (o espacio de palabras embebidas), representado por el siguiente conjunto $\{\langle w_i, \hbar(w_i)\rangle\}_{i=1}^{|V|}$.

Construcción del conjunto de datos de relaciones semánticas del dominio genérico

El penúltimo paso para culminar esta primer fase, consiste en preparar el conjunto de datos de entrenamiento para ser alimentado en el modelo de aprendizaje. El tratamiento de datos de entrenamiento $\langle \tau_i, \tau_j, \rho_{i,j}^m \rangle^M$ definido en pasos anteriores, consiste en la transformación de las palabras, que hasta este punto están en su estado natural, al espacio continuo y de baja dimensionalidad generado en el paso anterior. Dado que $\Gamma \subset V$, cada término es transformado por su representación en el espacio embebido $\hbar$, además, cada una de las palabras contenidas en las sentencias ($\rho_{i,j}^m$) donde aparece cada par de términos (τ_i, τ_j) son también transformados al espacio de palabras embebidas, a través de la función $f_\hbar : V \to \mathbb{R}^D$ donde $f_\hbar(\rho) = \{\hbar(w) | \forall w \in \rho\}$. Como resultado de las transformaciones del conjunto de datos de entrenamiento del dominio genérico se obtienen los siguientes conjuntos $\langle \mathcal{X}_S, \mathcal{Y}_S \rangle = \{\langle f_\hbar(\tau_i), f_\hbar(\tau_j), f_\hbar(\rho_{i,j}^m)\rangle, \langle r_k^m \rangle\}_{m=1}^M$, donde cada una de las palabras es sustituida por la representación en el espacio embebido.

Entrenamiento del modelo de aprendizaje

En este último paso de la primer fase consiste en el entrenamiento del modelo de aprendizaje para la detección de relaciones semánticas entre dos términos o entidades, a través del contexto entre ambos.

La entrada del modelo esta representado por el conjunto $\mathcal{X} = \{\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_M\}$, donde cada instancia $\mathcal{X}_m \in \mathbb{R}^{\ell+2\times D}$, ℓ es el número máximo de palabras contenidas cada una de las sentencias extraídas (ρ) y D es la dimensión del espacio embebido; cada instancia en el conjunto de datos de entrada esta compuesta por la concatenación de las matrices de la representación en el espacio embebido de los términos y la sentencia (ecuación 4.1).

$$\mathcal{X}_m = \left[f_\hbar(\tau_i)_{1\times D} \quad f_\hbar(\tau_j)_{1\times D} \quad f_\hbar(\rho^m_{i,j})_{\ell\times D} \right]_{\ell+2\times D} \tag{4.1}$$

Las salidas deseadas esta representado por $\mathcal{Y}$, donde para cada entrada $\mathcal{X}_m$ le corresponde una salida $\mathcal{Y}_m \in \{1, 2, \ldots, K\}$, donde K es el número de clases correspondientes a las relaciones semánticas ($\mathcal{R}$). De tal forma que cada tupla contenida en el conjunto de datos de entrenamiento se representa como el siguiente conjunto $\{\langle \mathcal{X}_m, \mathcal{Y}_m \rangle\}_{m=1}^{M}$ para ser empleado en el modelo de aprendizaje $\mathcal{M} : (\theta, \mathcal{X}) \to \mathcal{Y}$, donde $\theta = \{\theta_1, \theta_2, \ldots, \theta_{|\theta|}\}$ son los parámetros del modelo $\mathcal{M}$ para ajustar los datos de entrada $\mathcal{X}_m$ con los de la salida deseada $\mathcal{Y}_m$. En la etapa de entrenamiento del modelo los parámetros θ son actualizados mediante un algoritmo de aprendizaje (por ejemplo, *gradiente descendente*), hasta llegar a un error deseable $Error(\mathcal{Y}_m, \widehat{\mathcal{Y}}_m) \leq \epsilon$, donde $\widehat{\mathcal{Y}}_m$ es la salida real del modelo $\mathcal{M}$ y ϵ es el error de tolerancia del modelo.

El resultante en este paso es el modelo $\mathcal{M}$ entrenado, con los parámetros θ pseudo-óptimos para la detección de relaciones semánticas entre términos.

4.1.2. Adaptación del dominio

En el proceso de aprendizaje supervisado se entrenan modelos de aprendizaje a través de un conjunto de datos de entrenamiento, los cuales pertenecen a un determinado dominio. Después del proceso de aprendizaje, los modelos al recibir entradas nuevas tienen como salida una respuesta perteneciente al espacio de solución, siempre y cuando las entradas pertenezcan a la distribución del conjunto de datos de entrenamiento. Sin embargo, en ocasiones la evaluación de los modelos se realiza con diferente distribución de datos, por ejemplo si se entrenó un modelo para el reconocimiento de rostros con unas imágenes con cierta intensidad de luz, al evaluar el sistema con imágenes con una intensidad más baja (por ejemplo, capturadas de noche), es probable que el modelo no logre situar la respuesta en el espacio de solución correctamente; en el caso de manejo de un conjunto de texto (corpus) de un cierto dominio (tópico) $\mathcal{D}_{\mathcal{S}}$, al entrenar un modelo de aprendizaje es también probable que no logre otorgar una salida correcta si se aplica en otro dominio $\mathcal{D}_{\mathcal{T}}$, considérese además el uso de palabras que no existen del dominio $\mathcal{D}_{\mathcal{S}}$ en el dominio $\mathcal{D}_{\mathcal{T}}$.

Cuando se presentan este tipo de situaciones en las cuales la distribución entre los datos de entrenamiento y de evaluación son diferentes, se tendría que volver a etiquetar la distribución en donde se va a evaluar el modelo entrenado. Existen diversos escenarios posibles que provoquen un mal funcionamiento de los modelos de aprendizaje, tales como diferentes atributos de los datos entre el dominio $\mathcal{D}_{\mathcal{S}}$ y $\mathcal{D}_{\mathcal{T}}$ ($\mathcal{F}_S \neq \mathcal{F}_S$), diferente distribución de probabilidades marginales entre ambos dominios ($P(X_S) \neq P(X_T)$), mismo espacio de características pero diferente salidas deseadas ($Y_S \neq Y_T$), diferente probabilidad condicional entre los dominios ($P(Y_S|X_S) \neq (Y_T|X_T)$.

En el área de minería de texto es común aplicar técnicas de transferencia de aprendizaje ya que el corpus de entrenamiento generalmente no pertenece a la misma distribución marginal del conjunto de datos de evaluación (Pan *et al.*, 2012). Uno de los procesos más comunes

dentro de la transferencia de aprendizaje aplicado en modelos como las Redes Neuronales son el de re-entrenar la en los datos disponibles del dominio $\mathcal{D}_\mathcal{T}$ con los pesos calculados en el entrenamiento con el conjunto de datos del dominio $\mathcal{D}_\mathcal{S}$.

En esta metodología proponemos aplicar transferencia de aprendizaje mediante la adaptación de dominio, el cual se realiza cuando se ambos dominios $\mathcal{D}_\mathcal{S}$ y $\mathcal{D}_\mathcal{T}$ tienen diferente distribución de probabilidades marginales $P(X_S) \neq P(X_T)$, sin embargo, tienen en común la tarea a realizar. Al realizar una adaptación de dominio, para que el modelo entrenado en el dominio $\mathcal{D}_\mathcal{S}$ pueda ser aplicado en el dominio $\mathcal{D}_\mathcal{T}$, el tipo de transferencia de aprendizaje que se emplea es transductivo.

La entrada del modelo de aprendizaje se define por $\mathcal{X} \in \mathbb{R}^D$ donde D es la dimensión de representación de las palabras en el espacio de vectores embebidos, siendo la salida del sistema definida por $\mathcal{Y} \in \{1, \ldots, K\}$ donde K es el número de clases. Se define la tupla $\langle \mathcal{D}, \mathcal{S}, \mathcal{T} \rangle$ para un problema de clasificación donde $\mathcal{D}$ es la distribución $\mathcal{X} \times \mathcal{Y}$, $\mathcal{S}$ es un conjunto de datos de entrenamiento, de m muestras del fenómeno observado, representado como $\{(x_i, y_i)\}^m$ donde $i = 1, 2, \ldots, m$; y $\mathcal{T}$ es el conjunto de datos empleados para la evaluación tal que $\mathcal{S} \cap \mathcal{T} \equiv \emptyset$. El proceso de aprendizaje consiste en descubrir un modelo matemático $h : \mathcal{X} \rightarrow \mathcal{Y}$ para representar el comportamiento de un fenómeno dado a través de ejemplares $(\mathcal{S})$. En este paso de la metodología se propone la adaptación del dominio de la distribución de datos, unificando la distribución de los datos, es decir una representación común para ambos dominios que se logra incrustando los datos $\mathcal{X}_T$ en el espacio de representación de $\mathcal{X}_S$, procedimiento que se describe a continuación.

Re-entrenamiento de representaciones de palabras embebidas con el corpus del dominio específico

Para la adaptación del dominio entre el conjunto de datos del dominio fuente $\mathcal{X}_S$ al dominio objetivo $\mathcal{X}_T$, se propone realizarlo a través de la incrustación del corpus del dominio específico en el espacio de palabras embebidas creado del corpus del dominio genérico.

Se define el corpus del dominio especializado como $\mathcal{C}_T \in V_T$ en donde $V_T = \{w_1, w_2, \ldots, w_{|V_T|}\}$ es el vocabulario del dominio especializado, debido a la diferencia de dominio entre los corpus, el vocabulario que se emplea no son el mismo $V \neq V_T$. Por lo que se define una función para la transformación del vocabulario del dominio especializado al dominio de las palabras embebidas del dominio genérico $f_{\bar{h}} : V_T \rightarrow \mathbb{R}^D$.

El corpus de dominio específico se extrajeron de libros de texto son pre-procesados para disminuir la cantidad de ruido en el modelo de palabras embebidas. El proceso de limpieza del documento del dominio específico, consiste en los siguientes procesos:

1. Renglones vacíos.
2. Signos de puntuación.
3. Eliminación de caracteres no alfanuméricos.
4. Eliminación de sentencias con menos de N palabras
5. Transformación de las palabras a minúsculas

Una vez que el corpus del dominio específico se le aplican los procedimientos de limpieza antes mencionados se de denota como $\tilde{\mathcal{C}}_T$, y se procede con el re-entrenamiento del modelo de palabras embebidas, agregando el vocabulario del dominio específico al vocabulario existente. De tal forma que se se crea un conjunto que contiene los nuevos términos del dominio

específico transformados en el espacio embebido $\bar{\hbar}$. Es importante destacar que el entrenamiento del modelo neuronal lingüístico fue a partir del de los pesos generados en el proceso construcción del espacio embebido con el corpus del dominio genérico. El conjunto resultante en este paso esta compuesto por el diccionario de los términos que aparecen en el corpus del dominio específico ($\tilde{\mathcal{C}}_T \in V_T$), definido como $\{\langle w_i, \bar{\hbar}(w_i)\rangle\}_{i=1}^{|V_T|}$.

4.1.3. Inferencia de relaciones semánticas en un dominio especializado

En esta fase se realiza la inferencia de relaciones semánticas existentes entre los términos relevantes del dominio específico. La inferencia de las relaciones se realiza a través de la evaluación del modelo $\mathcal{M}$ entrenado con el corpus del dominio genérico $\mathcal{C}_S$, sin embargo, la evaluación se realiza a través del conjunto de datos de entrada $\mathcal{X}_T$, cuyas palabras están representadas en el espacio embebido modificado para el domino especializado ($\mathcal{D}_T$).

Los pasos que constituyen esta fase consiste desde en la preparación de los datos con tenidos en el corpus del dominio específico hasta la inferencia de relaciones inferidas entre los términos relevantes del dominio. En esta fase se emplea el corpus del dominio específico ya pre-procesado definido en pasos anteriores como $\tilde{\mathcal{C}}_T \in V_T$.

Generación automática de conjunto de datos para el dominio específico

Las tareas a realizar en este paso consiste en la identificación de los potenciales términos relevantes del dominio específico ($\mathcal{D}_T$), los cuales se pueden otorgar directamente o ser descubiertos a través técnicas como LDA, por sus siglas en inglés de *Latent Dirichlet Allocation* (Blei *et al.*, 2003); LSA, por sus siglas en inglés de *Latent Semantic Analysis* (Landauer y Dumais, 1997); TextRank (Gantayat e Iyer, 2011); entre otros. Debido al enfoque de esta

propuesta, la extracción de relaciones semánticas, no se limita al uso de una técnica en particular para realizar esta tarea. Sin embargo, se exhorta la utilización de alguna técnica en la cual no se requiera la intervención de un científico de datos, esto es por la idea principal de esta metodología que es la completa automatización de ontología de un dominio específico.

Una vez el conjunto de términos relevantes del dominio especializado son identificados, definidos por $\tilde{\Gamma}$, se emplea el procedimiento para la construcción del conjunto de datos a través de la técnica de supervisión distante.

En este paso se pueden representar los datos como el digrafo completo $\mathbb{K}^{\star} = \langle \tilde{\Gamma}, E \rangle$, es decir cada par de términos está conectado por una arista donde E es el conjunto ordenado de aristas que conectan a los vértices (términos relevantes del dominio $\tilde{\tau} \in \tilde{\Gamma}$), que al ser un digrafo completo contiene $|E| = |\tilde{\Gamma}|(|\tilde{\Gamma}| - 1)$ aristas, ilustrado en la imagen 4.2. Cada par de términos son buscados en el corpus pre-procesado $\tilde{\mathcal{C}}_T$ para la extracción de sentencias donde sean contenidos los términos.

Para cada par de término contenidos en el conjunto $\tilde{\Gamma}$ se realiza un búsqueda en el corpus del dominio específico ($\tilde{\mathcal{C}}_T$) en donde aparezcan en la misma oración, representado por el conjunto $\tilde{\rho} = \{\tilde{\rho}_{i,j} | \forall_{\{\tilde{\tau}_i, \tilde{\tau}_j\}} \in \tilde{\Gamma}, \forall_{(\tilde{\tau}_i, \tilde{\tau}_j)} \in \tilde{\sigma}, \tilde{\tau}_i \neq \tilde{\tau}_j\}$, donde $\tilde{\sigma} \in \tilde{\mathcal{C}}_T$ son las sentencias contenidas en el corpus del dominio específico. Dado que $\tilde{\rho} \subset \tilde{\sigma}$, la estructura de los datos resultantes, se convierte en el grafo dirigido $\mathcal{G}^{\star} = \langle \tilde{\Gamma}, \tilde{E} \rangle$, donde $\tilde{E} = \{\forall_{(\tilde{\tau}_i, \tilde{\tau}_j)} \in \tilde{\rho}\}$, el cual es un grafo dirigido no completo.

El conjunto resultante de este proceso esta definido como $\mathcal{X}_T{}^{\star} = \{\langle \tilde{\tau}_i, \tilde{\tau}_j, \tilde{\rho}^n_{i,j} \rangle\}^N$, donde $n = 1, 2, \ldots, N$; y representa el conjunto de datos de entrada al modelo $\mathcal{M}$, en lenguaje natural.

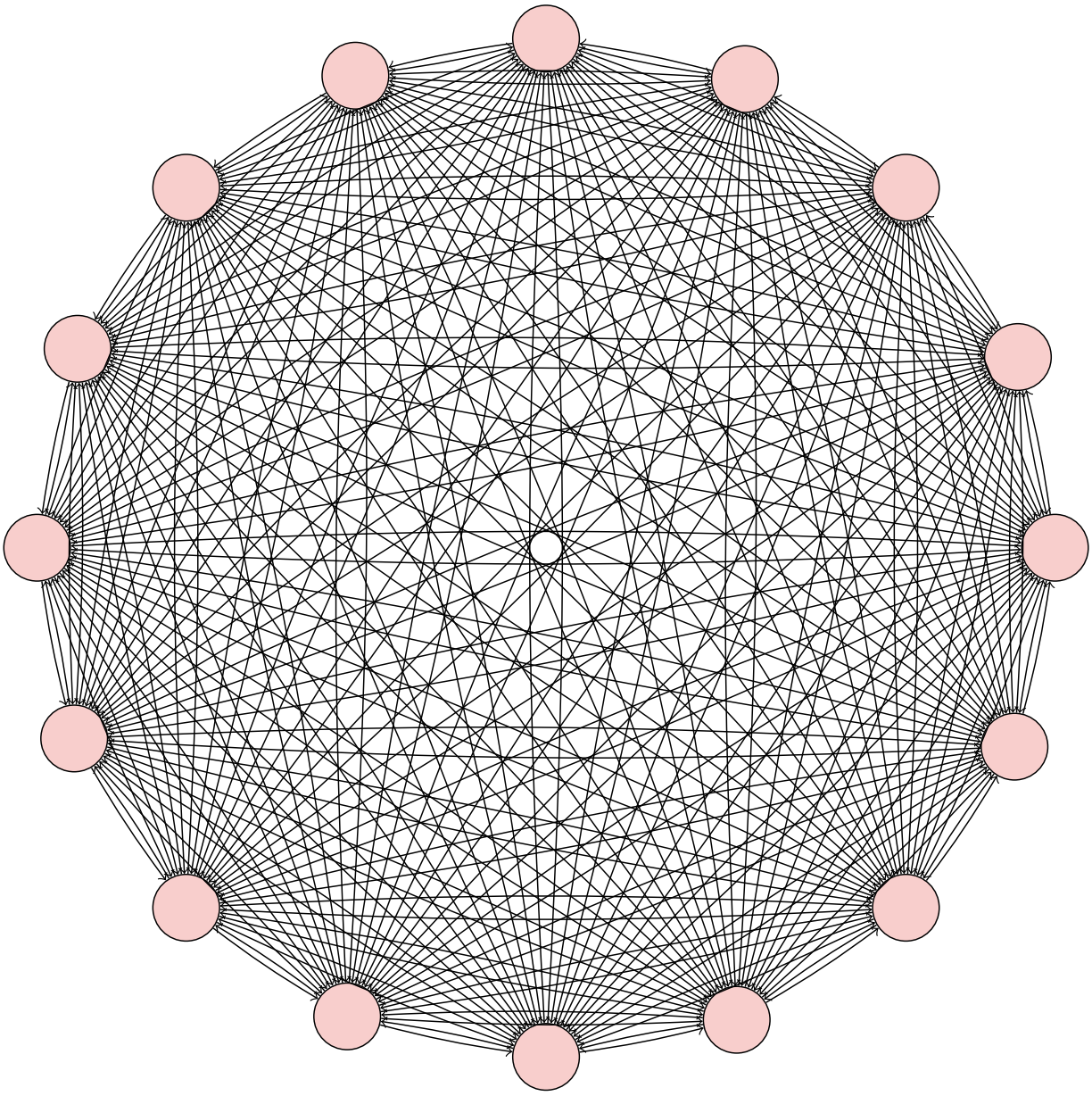

Figura 4.2: Digrafo completo $\mathbb{K}^{\star}_{|\tilde{\Gamma}|}$ considerando 16 términos relevantes del dominio específico.

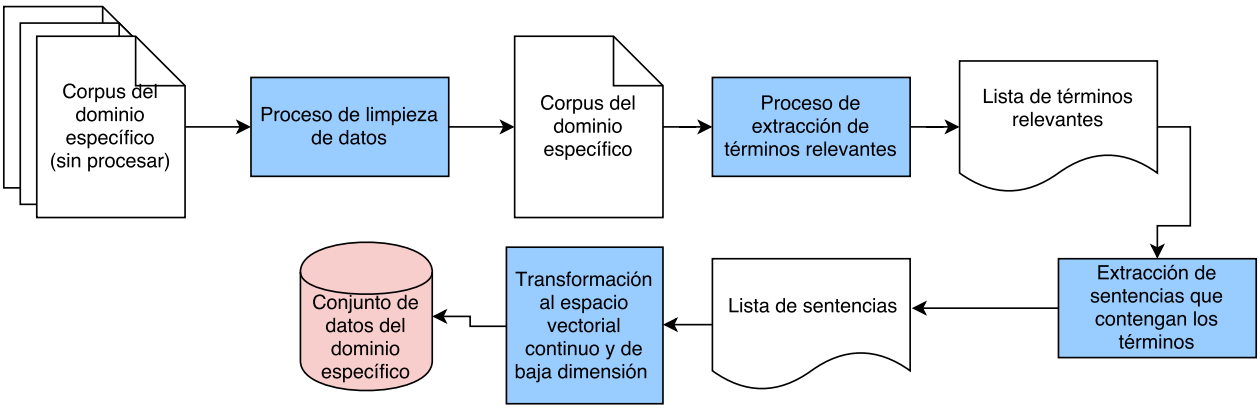

Figura 4.3: Proceso de la construcción del conjunto de datos del dominio específico.

Construcción de conjunto de datos de relaciones semánticas del dominio especializado

El procedimiento para la construcción del conjunto de datos de entrada para la inferencia de relaciones en un dominio especializado es similar al del dominio genérico, salvo por el espacio embebido en el cual se realizará la transformación de las palabras contenidas en el conjunto resultante en el paso anterior.

El tratamiento que se realiza en cada instancia del conjunto $\mathcal{X}_T{}^{\star} = \{\langle \tilde{\tau}_i, \tilde{\tau}_j, \tilde{\rho}^n_{i,j} \rangle\}_{n=1}^{N}$ consiste en la transformación de cada una de las palabras contenidas en el conjunto al espacio embebido creado con el corpus del dominio específico, representado por la función $f_{\bar{\hbar}} : V_T \to \mathbb{R}^D$ donde $f_{\bar{\hbar}}(\tilde{\rho}) = \{\bar{\hbar}(w) | \forall w \in \tilde{\rho}\}$, dado que $\tilde{\Gamma} \in V_T$, cada una de las palabras identificadas como términos relevantes del dominio $\mathcal{T}$ se encuentran definidas en el espacio embebido modificado $\bar{\hbar}$. De tal forma que el conjunto resultante se define como $\mathcal{X}_T = \{\langle f_{\bar{\hbar}}(\tilde{\tau}_i), f_{\bar{\hbar}}(\tilde{\tau}_j), f_{\bar{\hbar}}(\tilde{\rho}^n_{i,j}) \rangle\}_{n=1}^{N}$. Como se puede observar el conjunto de datos $\mathcal{X}_S$ es similar al conjunto $\mathcal{X}_T$, salvo que en éste último las palabras contenidas en el conjunto se establecen por su representación en el espacio de vectores embebidos modificados $\bar{\hbar}$.

Predicción de relaciones semánticas en el dominio específico

En este paso de la fase de la inferencia de relaciones semánticas en el dominio específico, se realiza la evaluación del modelo $\mathcal{M}$ con el conjunto de datos de entrada $\mathcal{X}_T$. Se puede representar, en este paso, al modelo $\mathcal{M}$ como una función $h_{\mathcal{M}} : \{\theta, X_T\} \rightarrow Y_T$ que tiene como entrada el conjunto en el dominio de las características del dominio especializado y los parámetros pseudo-óptimos obtenidos en la etapa de entrenamiento del modelo; y por ende, como salida se obtiene el tipo de relación semántica $r_k \in \mathcal{R}$ que corresponde al par de términos $\{\tilde{\tau}_i, \tilde{\tau}_j\} \in X_{T_m}$.

De forma análoga en el proceso de entrenamiento del modelo $\mathcal{M}$, se realiza el debido procesamiento del conjunto de datos de entrada, de tal forma que ésta se representa en forma de matriz de dimensiones $\ell + 2 \times D$ mismas definidas el proceso de entrenamiento, ℓ representa cantidad máxima de palabras contenidas en cada sentencia (ρ), y D es la dimensión del espacio embebido $\bar{\hbar}$ y $\hbar$.

Cada instancia en el conjunto de datos de entrada $\{\mathcal{X}_{T_1}, \mathcal{X}_{T_2}, \ldots, \mathcal{X}_{T_M}\} \in \mathcal{X}_T$ esta compuesta por la concatenación de las matrices de la representación en el espacio embebido de los términos relevantes $\{\tilde{\tau}_i, \tilde{\tau}_j\} \in \tilde{\Gamma}$ y la sentencia $\tilde{\rho}^n_{i,j}$ (ecuación 4.2).

$$\mathcal{X}_{T_m} = \left[f_{\bar{\hbar}}(\tilde{\tau}_i)_{1\times D} \quad f_{\bar{\hbar}}(\tau_j)_{1\times D} \quad f_{\bar{\hbar}}(\tilde{\rho}^n_{i,j})_{\ell\times D} \right]_{\ell+2\times D} \tag{4.2}$$

Al evaluar el modelo $\mathcal{M}$ a través de la función $h_{\mathcal{M}}$, se obtiene como salida, la predicción de las relaciones semánticas $\mathcal{R}$ representada como $\widehat{\mathcal{Y}}_T \in \mathbb{R}^{|\mathcal{R}|}$, que para obtener la relación $r_k \in \mathcal{R}$, se obtiene el argumento máximo en $\widehat{\mathcal{Y}}_{T_n} \therefore \mathcal{Y}_{T_m} = \underset{r_k}{\text{máx}}\ \widehat{\mathcal{Y}}_{T_n}$.

De manera concreta, una vez se evalúa el modelo en el dominio objetivo, dado un conjunto de entrada $\mathcal{X}_T = \{\langle f_{\bar{\hbar}}(\tilde{\tau}_i), f_{\bar{\hbar}}(\tilde{\tau}_j), f_{\bar{\hbar}}(\tilde{\rho}^n_{i,j})\rangle\}_{n=1}^N$, se obtiene un conjunto con las predicciones del modelo $\mathcal{Y}_T = \{\widehat{\mathcal{Y}}_{T_n}\}_{n=1}^N$. La interpretación del modelo se puede representar por el conjunto

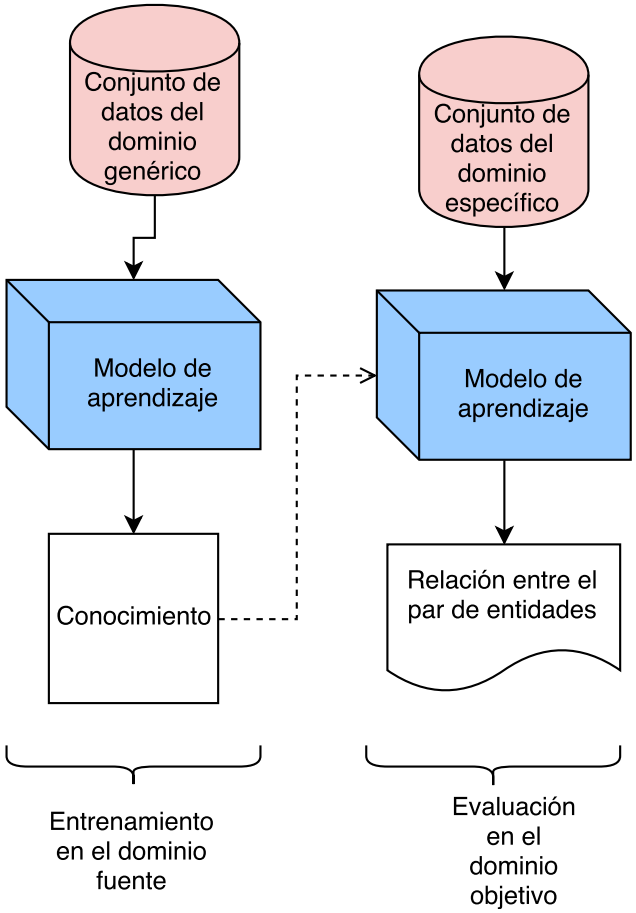

Figura 4.4: Aprendizaje de Ontologías utilizando Transferencia de Aprendizaje.

$\mathcal{Z} = \{\langle \tilde{\tau}_i^n, \tilde{\tau}_j^n, r_k^n \rangle\}_{n=1}^N$, que contiene el par de términos relevantes y la relación obtenida como resultado de la evaluación del modelo.

Fase de construcción de ontologías

Con el conjunto $\mathcal{Z}$ resultante, se forma un grafo dirigido donde cada término corresponde a un nodo (vértice) conectado a otro nodo a través de una arista que representa una relación semántica entre ambos términos, por lo que cada tupla en el conjunto $\mathcal{Z}$ se representa gráficamente de la siguiente forma $\tilde{\tau}_i^n \xrightarrow{r_k^n} \tilde{\tau}_j^n$. Por cada instancia en el conjunto resultante se agrega cada par de nodos con su respectiva arista al grafo dirigido, si se encuentra definido

un nodo, entonces solo se le añade la arista hacia (o desde) el otro nodo (que se agrega al grafo si éste no existe).

De tal forma que se define un grafo $\mathcal{G} = \langle \tilde{\Gamma}, \mathcal{R} \rangle$ formado de los términos del dominio y las relaciones semánticas entre ellos que posteriormente se representará como ontología.

Capítulo 5

Caso de estudio: Ontología del paradigma orientado a objetos

El caso de estudio que se aborda en esta tesis es en la generación automática de una ontología sobre el dominio del paradigma orientado a objetos. Las características generales de la ontología a generar son la extracción de términos relevantes del dominio y las relaciones semánticas entre éstos, en cuanto a relaciones semánticas, se pretenden abarcar únicamente las referentes a la hiperonimia, hiponimia, holonimia, meronimia; las cuales son relaciones básicas que en general se encuentran en las ontologías.

El propósito de este caso de estudio es el de la construcción del modelo de dominio de un ITS, el cual representará los tópicos generales de la materia de programación orientada a objetos, interrelacionados entré sí a través de las relaciones semánticas previamente mencionados. En lo sucesivo de este capítulo se describirán las condiciones que se establecieron para cada uno de los pasos mencionados descritos en la metodología propuesta (capítulo 4). Una vez generada la ontología se evaluará contra una realizada de forma manual a través de de un consenso entre expertos del dominio del paradigma orientado a objetos (Ramírez-Noriega

et al., 2017), los aspectos a evaluar son: extracción de términos relevantes y extracción de relaciones entre los términos relevantes. Sin embargo, es importante mencionar que la propuesta está enfocada en la extracción de relaciones semánticas, el proceso de la extracción de términos relevantes del dominio se contempla en la metodología para cubrir todas las fases de la construcción de ontología pero no se limita al uso de alguna técnica en particular, pero como ya se ha mencionado, se recomienda la aplicación de una técnica de extracción de términos relevantes no supervisada, por motivos de la facilidad de automatizar por completo la implementación de la metodología.

La ontología del paradigma orientado a objetos con la cual se va a comparar nuestra propuesta es en el idioma inglés, contiene 48 clases (términos o entidades), 51 relaciones y una profundidad máxima de 6 niveles. Los términos que se representan en la ontología son los siguientes: *Object orientation, Abstraction, Concurrency, Modularity, Persistence, Hierarchy, Typing, Encapsulation, Abstract class, Class, Interface, Class, Package, Aggregation , Composition , Inheritance, Scope, Modifiers, Hiding, Encapsulation, Attribute, Class modifiers, Constructor, Method, Object, Single inheritance, Multiple inheritance , Subclass, Superclass, Setter, Getter, Abstract, Final, Public, Overloading, Overriding, Protocol, Behavior, Identity, Instance, State, Method modifiers, Parameters, Return, Abstract, Final, Native, Private, Protected, Public, Static, Synchronized*, datos obtenidos del artículo Ramírez-Noriega *et al.* (2017).

Para este caso de estudio se empleará la metodología propuesta en el capítulo anterior (capítulo 4), y se describirá en cada paso cuales fueron las entradas y las salidas obtenidas. El objeto de estudio principal es el del reconocimiento de relaciones semánticas en el dominio del paradigma orientado a objetos, a través de un corpus no estructurado. Tal como se propone en la metodología, esta proceso de divide en tres fases diferentes, las cuales son: *Entrenamiento del modelo para detección de relaciones semánticas en un dominio genérico*, *Adaptación del dominio* e *Inferencia de relaciones semánticas en un dominio especializado*; la primer fase

consiste en la obtención de los parámetros pseudo-óptimos del modelo de aprendizaje en el dominio genérico (o dominio fuente), seguido de la adaptación del dominio que consiste el de preparar los datos con los cuales se van a evaluar el modelo (dominio objetivo), y en la última fase se realiza la inferencia de relaciones semánticas entre los términos relevantes del dominio.

El resulto del capítulo esta estructurado como configuración del experimento, resultados y discusiones. En el apartado de configuración del experimento de describe cuales fueron los datos empleados dentro de la metodología propuesta así como la forma de validación de la base de conocimiento resultante, por último se presentan los resultados seguidos de una breve discusión de los mismos.

5.1. Configuración del experimento

En esta sección se describe a detalle las características de los datos a emplear en cada uno de los pasos de la metodología propuesta, cuyas secciones se llamarán igual que las fases y pasos descritos en la metodología.

El presente experimento se realiza en el ámbito académico, esto debido al objetivo y principal motivación de este proyecto, por lo que el tema como ya se mencionó será el tema del paradigma orientado a objetos; la aplicación de la base de conocimiento resultante será implementada (en trabajos posteriores) en un STI, por lo que se han definido relaciones semánticas básicas necesarias para que este sistema pueda realizar inferencias del orden de aprendizaje de cada tópico de una determinada materia, como son relaciones de hiperonomia, hiponimia, holonomía, meronimia, las cuales son comúnmente conocidas, las primeras dos como relaciones taxonómicas o relaciones *es-un*, siendo las últimas relaciones del tipo *parte-de*. Es importante recalcar que este tipo de relaciones semánticas son relaciones dirigidas por

lo que el orden de los elementos conectados por esta arista refleja una jerarquía entre los datos.

En este experimento se seleccionaron recursos en todas sus fases que fueran asequibles en cualquier dominio, dentro del contexto de la aplicación de este experimento. Por lo que probablemente, se podrían cambiar los recursos descritos a continuación por unos con algún tipo de estructura o pre-procesados y mejorar los resultados. Sin embargo, el enfoque esta en la reutilización de esta metodología en diversos dominios en el contexto de mapas curriculares automáticos para sistemas de enseñanza asistidos por computadora.

5.1.1. Entrenamiento del modelo para detección de relaciones semánticas en un dominio genérico

En esta primer fase del experimento se realiza el entrenamiento de un modelo de aprendizaje para la detección de relaciones semánticas, esto en un dominio genérico, que indistintamente será llamado dominio fuente a lo largo del documento. Este dominio fuente esta constituido por un conjunto de documentos de texto no estructurado correspondiente no solo a un tópico o dominio, sino a varios de estos, con la intención de generalizar su aplicación en dominios específicos.

Selección del corpus y base de datos del dominio genérico

Dada la complejidad del tipo de información inherente en los textos, se requiere una gran cantidad de información para generalizar apropiadamente el área de solución de la detección de relaciones semánticas, por lo que el tamaño del corpus en este paso tiene un rol muy importante, aunado a lo anterior, se debe de considerar un corpus que sea genérico, es decir, que no esté enfocado en un dominio en particular sino que abarque varios dominios.

La selección del corpus fue considerado tomando en cuenta el corpus mas extenso de un dominio genérico, esto es, que el tópico principal no se esté enfocado en un dominio específico. Por lo tanto, la enciclopedia en linea de libre acceso Wikipedia fue seleccionado como corpus de dominio genérico, debido a la facilidad de acceso y la cantidad de información disponible, además de que es continuamente actualizada, agregando nueva información.

Una vez seleccionado el corpus del dominio genérico, se prosigue a la selección de la base de datos o base de conocimiento que contenga de forma explícita relaciones entre clases, términos relevantes o entidades, los cuales deberán aparecer en el corpus seleccionado en el paso anterior. El dominio que abarque esta base de información deberá estar acorde con el corpus para maximizar el conjunto de entidades que se encuentren en ambos.

Para la selección de la base de datos que contiene el conjunto de relaciones de interés para este proyecto (hiperónimo, hipónimo, holónimo y merónimo) fue seleccionada la base de conocimiento WordNet, ya que es construida de forma manual, lo cual reduciría la inclusión de ruido en el proceso de aprendizaje del modelo de clasificación. Además, no está enfocada en un dominio en particular, por lo que lo convierte en un buen complemento al corpus seleccionado.

Construcción automática de base de datos de relaciones semánticas del dominio genérico

La idea principal en este proceso es el de generar suficientes datos de entrenamiento para representar un dominio genérico a través de relaciones entre entidades (conceptos). Las relaciones semánticas extraídas de la base de datos WordNet están representadas por el conjunto $\mathcal{S} = \{hiperónimo, hipónimo, holónimo , merónimo\}$.

La técnica de supervisión distante requiere la búsqueda de cada par de términos en el corpus para extraer sentencias en donde ambos términos aparecieran. En este proceso se

Tabla 5.1: Parámetros del modelo de palabras embebidas

Parámetro	Valor
Dimensión de la palabra	300
Tamaño de ventana	10
Modelo	skip-gram

realizó la búsqueda en la enciclopedia en línea Wikipedia a través del API de Wikipedia. Por cada par de términos se tomaron hasta 10 sentencias diferentes en donde aparecieran ambos términos.

Creación de palabras embebidas del corpus del dominio genérico

La representación de palabras embebidas han sido usadas para representar palabras en un espacio de baja dimensionalidad. Esta representación de las palabras permiten un manejo mas eficiente de los datos, además se ha demostrado que en el nuevo espacio de baja dimensionalidad se logra capturar relaciones semánticas entre las palabras, por ejemplo, $v(\texttt{king}) - v(\texttt{queen}) \equiv v(\texttt{man}) - v(\texttt{woman})$. La representación de palabras en este espacio permite hacer uso de modelos de aprendizaje sin necesidad de crear los atributos manualmente.

En este experimento se utilizó el modelo llamado *skipgram* a través de la herramienta *word2vec* para construir la representación de las palabras en un espacio de baja dimensionalidad. En la tabla 5.2 se muestra la configuración usada para éste experimento.

5.1.2. Construcción del conjunto de datos de relaciones semánticas del dominio genérico

Una vez extraído los términos o entidades contenidas en la base de conocimiento y haber realizado la búsqueda de éstos en el corpus del dominio genérico, representado en la metodo-

logía como $\mathcal{X}_S^\star$, se deben preparar los datos a un dominio numérico para poder ser otorgado a un modelo de aprendizaje.

En este paso se realizó la transformación de los datos generados en el paso anterior, de tal forma que los términos relevantes del dominio específico y las palabras que se encuentran en la sentencia donde aparecen éste par de términos son convertidos a su representación en el espacio continuo y de baja dimensión $\hbar$. La dimensión establecida en la generación del espacio embebido fue de 300, por lo que cada instancia tiene una dimensión de $\ell + 2 \times D$, donde $\ell = 60$ y $D = 300$.

Entrenamiento del modelo de aprendizaje para clasificación de relaciones

En este paso se realiza una preparación del conjunto de datos que representa la entrada y salida deseada de cada instancia.

Como se describe en la metodología, el conjunto de datos de entrada $\mathcal{X}$ se procesa concatenando la representación de ambos términos y la sentencia donde se encuentran éstos (ecuación 4.1). Mientras que el procesamiento de la salidas deseadas cosiste en la representación en $\mathbb{Z}$ de los tipos de relaciones, los cuales se refieren a 0: No existe, 1: Hiperónimo, 2: Hipónimo, 3: Holónimo, 4: Merónimo.

El modelo seleccionado fue usado basado en el estado del arte en la clasificación de relaciones semánticas usado en el conjunto de datos *SemEval task 8*. El modelo en el estado del arte es una Red Neuronal con una Unidad de Compuerta Recurrente Bidireccional (BGRU, por sus siglas en inglés de *Bidirectional Gated Recurrent Unit*) con la técnica de atención. Este tipo de modelos han sido usados en datos secuenciales como vídeos, audio y procesamiento de texto. Los parámetros usados para el entrenamiento fue basado en la propuesta de Zhou *et al.* (2016), que consiste en una capa embebida (e) y una capa de 230 neuronas (h). Se entrenó el modelo hasta $10,000$ iteraciones empleando el algoritmo de aprendizaje Adam

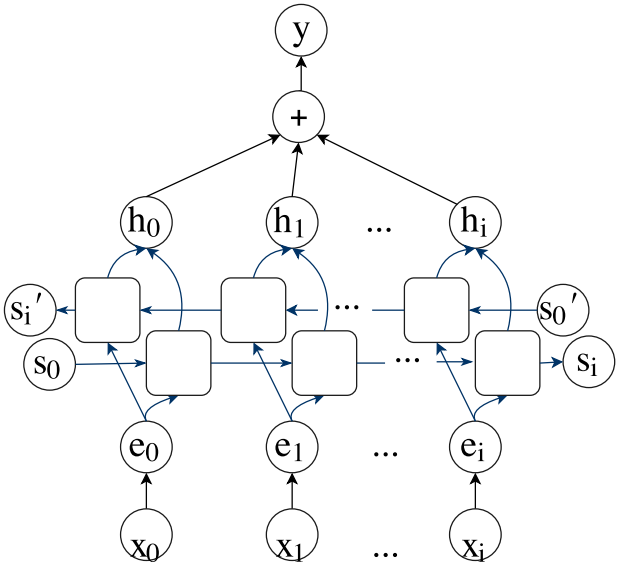

Figura 5.1: Arquitectura del modelo de aprendizaje propuesta por (Zhou *et al.*, 2016)

(Kingma y Ba, 2014) con valor inicial $\eta = 1 \times 10^{-3}$ con mini-lotes de 50 instancias.

5.1.3. Adaptación del dominio

Re-entrenamiento de representaciones de palabras embebidas con el corpus del dominio específico

En este paso se vuelve a entrenar el modelo neuronal para representación de palabras (*word2vec*), sin embargo, se mantienen los pesos establecidos del dominio fuente ($\mathcal{D}_S$) y se añade el nuevo vocabulario del dominio específico para generar la representación de las palabras agregadas al modelo. La idea principal de realizar este procedimiento es el de reducir la diferencia en la distribución de datos del dominio fuente y del dominio objetivo.

El corpus de dominio específico, que consiste en libros de texto sobre el dominio del paradigma orientado a objetos, son pre-procesados para evitar introducir el ruido en el modelo de palabras embebidas. El proceso de limpieza de texto, se baso en remover lo siguiente:

1. Renglones vacíos.

2. Signos de puntuación.

3. Eliminación de caracteres no alfanuméricos.

4. Eliminación de sentencias con menos de 3 palabras

5. Transformación de las palabras a minúsculas

5.1.4. Inferencia de relaciones semánticas en un dominio especializado

Generación automática de conjunto de datos para el dominio específico

Una vez que se modifica la representación de las palabras embebidas, ahora en el dominio específico, se requiere crear el conjunto de datos con esta nueva representación de las palabras. El conjunto de datos empleado para utilizar el modelo aprendido está conformado por pares de palabras, previamente extraídas aquellas con mayor probabilidad de ser representativos del nuevo dominio específico, y de una sentencia en donde el par de palabras aparecen en el corpus de dominio específico.

Para probar la metodología propuesta se introdujeron directamente los términos definidos en la ontología que se consideró como modelo, los cuales se listan a continuación: *scope modularity composition subclass persistence protected final interface parameters constructor superclass setter overriding getter static class hiding hierarchy return inheritance method behavior abstraction abstract state single protocol agreggation orientation modifiers private encapsulation package native typing identity synchronized overloading multiple object instance public methods attribute.*

Construcción de conjunto de datos de relaciones semánticas del dominio especializado

Después que es generado el conjunto de datos del dominio específico, cada palabras que aparece en las sentencias, donde ambos términos son extraídos como términos relevantes, son transformados al espacio embebido. En este experimento se estableció la dimensión de espacio embebido en 300. El tamaño máximo de cada sentencia se estableció en 60 palabras, por lo tanto cada instancia tiene una dimensión de $60 + 2 \times 300$.

Predicción de relaciones semánticas en el dominio específico

Una vez que el conjunto de datos del dominio objetivo fue construido, ya está listo para ser alimentado el modelo entrenado. La entrada del modelo es la representación de vectores de palabras embebidas de cada par de términos clave y la sentencia que los contienen. La salida del modelo es una de las clases con la cual se entrenaron, las cuales son hiperónimo, hipónimo, holónimo y merónimo.

Fase de construcción de ontologías

La lista de conceptos y la relación obtenida por el modelo son utilizados para la construcción del grafo, el cual consiste en las relaciones básicas de una ontología como son relaciones es-un y parte-de y sus relaciones inversas. En esta primer propuesta estas relaciones son suficientes para representar de forma básica la estructura curricular de una materia en el contexto académico.

5.1.5. Evaluación de las relaciones extraídas

Es esta sección se describe la forma de evaluación del modelo de representación de conocimiento obtenido, cuyos resultados serán mostrados y discutidos en las siguientes secciones.

Las emplearon las métricas empleadas para evaluación de esta metodología son las de precisión (expresión 5.1), exhaustividad (expresión 5.2) y la métrica F1 (expresión 5.3).

$$Precision = \frac{tp}{tp + fp} \tag{5.1}$$

donde, tp son verdaderos positivos y fp falsos positivos.

$$Exhaustividad = \frac{tp}{tp + fn} \tag{5.2}$$

donde, tp son verdaderos positivos y fp falsos positivos.

$$Fmeasure = \frac{2 * Precision * Exhaustividad}{Precision + Exhaustividad} \tag{5.3}$$

5.2. Resultados y discusión

Se implementó la metodología, con diversos corpus, creados a partir de cuatro libros de texto pertenecientes al tema de la programación orientada a objetos, los cuales fueron automáticamente transformados a texto plano, sin aplicar algún tipo de corrección salvo los que se mencionan explícitamente en la metodología. Además de los cuatro libros de texto, se creo un corpus a través de la concatenación de éstos documentos, obteniendo un corpus más rico en cuanto a información.

En la aplicación de las métricas para la evaluación del los resultados en la extracción de

relaciones del dominio específico, solo se consideraron las relaciones que contienen términos relevantes que aparecen en el corpus, debido a que el modelo de aprendizaje no puede determinar relaciones semánticas entre conceptos sin la sentencia que contiene tales términos relevantes. Por lo que en la métrica de exhaustividad se considera el número de relaciones relevantes (en el contexto de recuperación de información) como la cantidad de elementos que contiene la intersección de los conjuntos de términos identificados como relevantes del dominio objetivo y las palabras que están contenidas en el corpus o documento de texto.

En la tabla 5.2 se muestran los resultados obtenidos al evaluar el grafo resultante al aplicar la metodología propuesta en los cuatro libros de texto y en el corpus formado de todos ellos. Las evaluaciones mostradas se refiere a la tarea de extracción de relaciones semánticas únicamente, dado que el paso de extracción de términos relevantes se hizo de forma estática ya que la propuesta versa sobre la extracción de relaciones semánticas, sin embargo, en el capítulo anterior se hace mención de algunas técnicas no supervisadas que se pudieran aplicar.

En la tarea de extracción de relaciones se obtuvo una precisión promedio del 62,9 %, una exhaustividad del 80,3 % y un promedio armónico de 70,2 %, con una desviación estándar de 0,1, 0,131, 0,106 respectivamente. Se puede observar una desviación estándar relativamente amplia, sin embargo, estos se debe presumiblemente a la diferencia de longitud del corpus.

Un aspecto interesante de estos resultados, es que en el corpus que representa la unión de los cuatro libros de texto seleccionados se observa un decremento tanto en la precisión como en la exhaustividad de forma significativa. Una de las posibles causas de este comportamiento se puede dar en la etapa de creación del espacio embebido, ya que se realizan más iteraciones dado que éstas dependen del número de sentencias contenidas en el documento, por lo cual es posible la generación del fenómeno de sobre-entrenamiento.

Documento	Precisión	Exhaustividad	F1
1	0.695	0.826	0.754
2	0.625	0.812	0.706
3	0.761	0.95	0. 845
4	0.521	0.84	0.643
Todos	0.545	0.59	0. 566
Promedio	0.629	0.803	0.702
σ	0.1	0.131	0.106

Tabla 5.2: Resultados de la metodología propuesta

Capítulo 6

Conclusiones y trabajo futuro

6.1. Conclusiones

La metodología propuesta muestra resultados prominentes considerando que el proceso es totalmente automatizado y se puede prescindir de la intervención humana. Esto se debe a que una vez que el modelo está entrenado en el dominio fuente, solo se requiere entrenar las palabras embebidas del nuevo corpus del dominio específico, identificar de palabras relevantes para el dominio, y por último, la construcción de conjunto de datos para ser evaluados por el modelo. Los pasos necesarios para cambiar de dominio específico son automáticos, lo cual evita que se haga un cuello de botella en el proceso de generación de ontologías.

Cada uno de los procesos propuestos se puede mejorarse fácilmente debido a su flexibilidad, ya que cada fase en esta metodología no depende del proceso anterior sino solo de la estructura de la información requerida. Por lo tanto, mientras se estructuren los datos correctamente no habría mayores complicaciones.

Este estudio se enfocó en responder las siguientes preguntas de investigación:

- ¿Cuán bien los modelos empleados en Aprendizaje Profundo pueden aprender relaciones entre términos sin usar técnicas de NLP?
- Al haber una falta de conjunto de datos de entrenamiento para cada dominio específico, ¿será útil la aplicación de la técnica de supervisión distante para la representación de las relaciones entre términos de dominio?
- ¿Cuán factible es aplicar Transferencia de Aprendizaje en el problema de la clasificación de relaciones semánticas cuando el modelo fue entrenado con otro corpus (diferente distribución de datos)?

Debido a falta de información estructurada de los diversos dominios específicos, es difícil el entrenamiento de los modelos para representar ese conocimiento. En el dominio de Aprendizaje de Ontología uno de los mayores problemas es el esfuerzo requerido por parte de las personas para la construcción de las ontologías para representar un cierto dominio. Debido al bajo nivel de intervención del humano y que el proceso de construcción manual de características puede ser obviado, esta metodología puede ser valioso como una herramienta de soporte en la construcción de ontologías.

El resultado de estos experimentos muestran que las técnicas de transferencia de aprendizaje y supervisión distante pueden ser aplicadas en el dominio de Aprendizaje de Ontologías. En el caso de la aplicación técnicas de Aprendizaje Profundo para la representación de texto y clasificación de relaciones semánticas se obtuvo una precisión del 62,9 % , una exhaustividad del 80,3 % y medida $F1$ del 70 %, al comparar estos resultados con los métodos semi-automatizados para la construcción de ontologías empleando técnicas de NLP, como los de Atapattu *et al.* (2012), en cuyos resultados se reportaron los siguientes valores, en precisión se obtuvo un 35 %, exhaustividad un 60 % y F1 se obtuvo 42 %; se aprecia que los resultados obtenidos en este trabajo superan en cuanto a precisión e inclusive en el promedio armónico de precisión y exhaustividad, dado por la métrica F1, que en nuestro caso se obtuvo 70 %, lo

que significa una mejora del 28 %.

Comparando esta propuesta con la propuesta de Larrañaga *et al.* (2014), en el cual tiene el mismo objetivo que el trabajo anterior, semi-automatizar el proceso de la construcción del modelo de dominio de un ITS. La idea principal de su propuesta se basa en la extracción de relaciones a través de patrones entre las palabras, por lo que lo hace dependiente del lenguaje y posiblemente del dominio. En este trabajo se aprovecha la estructura contenida en la tabla de contenidos de libros de texto, en los cuales no se especifica si la identificación y extracción de este es de forma automática o fue construida por un humano. El autor reporta una exhaustividad en las relaciones *parte-de* (holónimos y merónimos) del 55,81 %, en la relaciones taxonómicas (hiperónimos e hipónimos) se reporta una valores del 56,1 % y 30,30 % para la precisión y exhaustividad respectivamente.

La metodología propuesta, objeto de esta tesis, se basa enteramente métodos no supervisados en donde se han comprobado ser no dependiente del idioma, como lo son los métodos para transformar las palabras en el espacio embebido *word2vec* (Mikolov *et al.*, 2013a). Aunque las metodologías propuestas extraen diversas relaciones, además de las que se contemplan en este trabajo, se logra equiparar y en algunos casos superar sus resultados, lo que es notable debido a que en la metodología propuesta no se emplean técnicas de NLP dependientes del lenguaje como etiquetado de palabras, identificación de patrones de palabras mayormente empleadas en determinadas relaciones; aun así, se obtuvieron resultados acordes al estado del arte. Aunado a la falta de utilización de técnicas de NLP, también no se tomó en cuenta la estructura del corpus, es decir, no se contemplaron secciones frecuentemente encontrados en libros de texto, como lo son el título, tabla de contenidos, glosarios, entre otros. Por lo tanto, esta metodología podría emplear otro tipo de recursos o corpus pertenecientes a un dominio especializado.

Es importante mencionar que es basado en la relación obtenida entre los pares de términos

por parte del modelo de aprendizaje, aquellas relaciones que no han sido clasificado en una relación pueden ser ignoradas y descartada como término relevante del dominio. Esto es, el modelo puede discriminar términos ruidosos que sean presentados como entrada.

6.2. Aportaciones

Las mayores contribuciones de este proyecto son las siguientes:

- Metodología para automatizar el proceso de construcción de ontologías de un dominio específico a través de corpus no estructurados.
- Creación de conjunto de datos del dominio genérico a través de la técnica de supervisión distante entre el corpus de Wikipedia y WordNet.
- Técnica de Aprendizaje de Ontologías por medio de clasificación de relaciones entre términos de dominio en el espacio de palabras incrustadas (WE).

Las publicaciones realizados como primer autor durante el periodo de la maestría fueron los siguientes:

1. **Navarro-Almanza R.**, Licea G., Juárez-Ramírez R., Mendoza O. (2017) Semantic Capture Analysis in Word Embedding Vectors Using Convolutional Neural Network. In: Rocha Á., Correia A., Adeli H., Reis L., Costanzo S. (eds) Recent Advances in Information Systems and Technologies. WorldCIST 2017. Advances in Intelligent Systems and Computing, vol 569. Springer, Cham.
2. **Navarro-Almanza R.**, Juárez-Ramírez R., Licea G. (2017) Towards supporting Software Engineering using Deep Learning: A case of Software Requirements Classification.

In: 5th International Conference in Software Engineering Research and Innovation. CONISOFT 2017. DOI: 10.1109/CONISOFT.2017.00027.

Las colaboraciones realizados durante el periodo de la maestría fueron los siguientes:

1. S. Jiménez, R. Juárez-Ramírez, **R. Navarro**, A. Coronel and V. H. Castillo, Architecting an Intelligent Tutoring System with an Affective Dialogue Module, 2016 4th International Conference in Software Engineering Research and Innovation (CONISOFT), Puebla, 2016, pp. 122-129. doi: 10.1109/CONISOFT.2016.27.

2. A. Ramirez-Noriega, R. Juarez-Ramirez, **R. Navarro** and J. Lopez-Martinez, Using Bayesian Networks to Obtain the Task's Parameters for Schedule Planning in Scrum 2016 4th International Conference in Software Engineering Research and Innovation (CONISOFT), Puebla, 2016, pp. 167-174. doi: 10.1109/CONISOFT.2016.33.

3. López-Martínez J., Juárez-Ramírez R., Ramírez-Noriega A., Licea G., **Navarro-Almanza R.** (2017) Estimating User Stories' Complexity and Importance in Scrum with Bayesian Networks. In: Rocha Á., Correia A., Adeli H., Reis L., Costanzo S. (eds) Recent Advances in Information Systems and Technologies. WorldCIST 2017. Advances in Intelligent Systems and Computing, vol 569. Springer, Cham.

4. Alan Ramírez-Noriega, Reyes Juarez-Ramírez, Samantha Jiménez, Sergio Inzunza, **Raúl Navarro**, Janeth López-Martínez (2017) An Ontology of the Object Orientation for Intelligent Tutoring Systems. In: 5th International Conference in Software Engineering Research and Innovation. CONISOFT 2017. DOI: 10.1109/CONISOFT.2017.00027.

6.3. Trabajo Futuro

En esta primer propuesta solo se procesaron términos compuestos por únicamente una palabra, sin embargo, generalmente en dominios especializados se emplean términos que están compuestos por diversas palabras.

En futuras propuestas, se planea aplicar modelos para identificar términos compuestos para aumentar la exhaustividad del modelo resultante. En este contexto pueden ser aplicados diversas representaciones de palabras embebidas como el modelo Doc2Vec (Le y Mikolov, 2014) que puede inferir la representación en el espacio embebido de una palabra nueva, es decir, calcular una representación de una palabra que nunca fue presentado al modelo durante su entrenamiento.

El trabajo futuro de este proyecto son:

- Uso de modelos de n-gramas para soportar términos compuestos por diversas palabras.
- Debido al resultado de aplicación de esta metodología, una ontología por corpus, podría ser interesante integrar o combinar automáticamente diversas ontologías del mismo dominio basado en el resultado cuantitativo del modelo.
- Dado la naturaleza de aplicación de este proyecto, incluir un sistema en donde los alumnos evalúen en tiempo real el resultado del sistema, modificando así el modelo generado del dominio específico.
- Aplicar la metodología propuesta en dominios específicos diferentes al utilizado en esta tesis.
- Integración de diversas ontologías de diversos dominios especializados en base a los términos similares.

- Probar la metodología en corpus en el idioma español, para comprobar la independencia del lenguaje.

- Aplicación de métodos más novedosos para representación de grafos tales *Computadora Neuronal Diferenciable* (DNC, por sus siglas en inglés de *Differentiable Neural Computer*) propuesto por Graves *et al.* (2016).

- Emplear la metodología propuesta en otro dominio de aplicación, por ejemplo, en Ingeniería de Software para detección de relaciones entre requerimientos; como extensión del trabajo alterno realizado durante el periodo de la maestría, titulado *Towards supporting Software Engineering using Deep Learning : A case of Software Requirements Classification* (Navarro-Almanza *et al.*, 2017a).

Bibliografía

Atapattu, T., Falkner, K., y Falkner, N. (2012). Automated extraction of semantic concepts from semi- structured data: supporting computer-based education through the analysis of lecture notes. *International Conference on Database and Expert Systems Applications*, (August):161–175.

Blei, D. M., Ng, A. Y., y Jordan, M. I. (2003). Latent Dirichlet Allocation. *J. Mach. Learn. Res.*, 3:993–1022.

BLOOM, B. S. (1984). The 2 Sigma Problem: The Search for Methods of Group Instruction as Effective as One-to-One Tutoring. *Educational Researcher*, 13(6):4–16.

Bylander, T. y Chandrasekaran, B. (1987). Generic tasks for knowledge-based reasoning: the "right" level of abstraction for knowledge acquisition.

Caraballo, S. A. (1999). Automatic construction of a hypernym-labeled noun hierarchy from text. En *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics* -, ACL '99, pp. 120–126, Stroudsburg, PA, USA. Association for Computational Linguistics.

Chakraborty, S., Roy, D., y Basu, A. (2010). TMRF e-Book Development of Knowledge Based Intelligent Tutoring System. *Sajja & Akerkar)*, 1:74–100.

Chen, C. M. (2009). Ontology-based concept map for planning a personalised learning path. *British Journal of Educational Technology*, 40(6):1028–1058.

Chen, Y.-n. V., Rudnicky, A. I., Chen, Y.-n., Wang, W. Y., y Rudnicky, A. I. (2015). Learning Semantic Hierarchy with Distributed Representations for Unsupervised Spoken Language Understanding for Unsupervised Spoken Language Understanding. (SEPTEMBER).

Cho, K., van Merrienboer, B., Gülçehre, Ç., Bougares, F., Schwenk, H., y Bengio, Y. (2014). Learning Phrase Representations using {RNN} Encoder-Decoder for Statistical Machine Translation. *CoRR*, abs/1406.1.

Cimiano, P. (2006). *Ontology learning and population from text: Algorithms, evaluation and applications*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Cimiano, P., Hotho, A., y Staab, S. (2004). Comparing Conceptual , Divisive and Agglomerative Clustering for Learning Taxonomies from Text. En *Text*, volumen 16, pp. 435–439.

Colace, F., De Santo, M., Greco, L., Amato, F., Moscato, V., y Picariello, A. (2014). Terminological ontology learning and population using latent dirichlet allocation. *Journal of Visual Languages and Computing*, 25(6):818–826.

Conde, A., Larra??aga, M., Arruarte, A., y Elorriaga, J. A. (2014). Testing language independence in the semiautomatic construction of educational ontologies. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8474 LNCS:545–550.

Corcho, O. y Gómez-Pérez, A. (2000). A Roadmap to Ontology Specification Languages. En Dieng, R. y Corby, O., editores, *International Conference on Knowledge Engineering and Knowledge Management*, volumen 1937 de *Lecture Notes in Computer Science*, pp. 80–96. Springer Berlin Heidelberg.

dos Santos, C. N., Xiang, B., y Zhou, B. (2015). Classifying Relations by Ranking with Convolutional Neural Networks. *Acl-2015*, (3):626–634.

Escudero, H. y Fuentes-Gonzalez, R. (2012). Generating ontologies from Intelligent Tutoring System courses . A generic. *Semantic Web – Interoperability, Usability, Applicability*, REJEITADO(0):1–9.

Fan, M., Zhou, Q., Abel, A., Zheng, T. F., y Grishman, R. (2016). Probabilistic Belief Embedding for Large-Scale Knowledge Population. *Cognitive Computation*, 8(6):1087–1102.

Faure, D. y Nedellec, C. (1999). Knowledge acquisition of predicate argument structures from technical texts using machine learning: The system ASIUM. En Fensel, D. y Studer, R., editores, *Knowledge Acquisition, Modeling and Management*, volumen 1621 de *Lecture Notes in Computer Science*, pp. 2–7. Springer Berlin Heidelberg.

Fu, R., Guo, J., Qin, B., Che, W., Wang, H., y Liu, T. (2014). Learning Semantic Hierarchies via Word Embeddings. *Acl*, pp. 1199–1209.

Fu, R., Guo, J., Qin, B., Che, W., Wang, H., y Liu, T. (2015). Learning semantic hierarchies: A continuous vector space approach. *IEEE Transactions on Audio, Speech and Language Processing*, 23(3):461–471.

Gantayat, N. e Iyer, S. (2011). Automated building of domain ontologies from lecture notes in courseware. *Proceedings - IEEE International Conference on Technology for Education, T4E 2011*, pp. 89–95.

Geller, J. (2003). John Sowa, Knowledge Representation: Logical, Philosophical, and Computational Foundations, Brooks/Cole, 2000, 512 pp., $70.95 (cloth), ISBN 0-534-94965-7. *Minds and Machines*, 13(3):441–444.

Gligora, M., Polytechnic, M., y Polytechnic, A. J. (2015). A Prevalence Trend of Characteristics of Intelligent and Adaptive Hypermedia E- Learning Systems. (August).

Gómez-Pérez, A., Corcho-Garc\'\ia, O., y Fernández-López, M. (2003). *Ontology Learning*. Springer.

Graves, A., Wayne, G., Reynolds, M., Harley, T., Danihelka, I., Grabska-Barwińska, A., Colmenarejo, S. G., Grefenstette, E., Ramalho, T., Agapiou, J., Badia, A. P., Hermann, K. M., Zwols, Y., Ostrovski, G., Cain, A., King, H., Summerfield, C., Blunsom, P., Kavukcuoglu, K., y Hassabis, D. (2016). Hybrid computing using a neural network with dynamic external memory. *Nature*, 538:471.

Grimm, S., Abecker, A., Völker, J., y Studer, R. (2011). Ontologies and the Semantic Web. En Domingue, J., Fensel, D., y Hendler, J., editores, *Handbook of Semantic Web Technologies*, pp. 507–579. Springer Berlin Heidelberg.

Gruber, T. R. (1995). D. *International Journal of Human-Computer Studies*, 43(5-6):907–928.

Guarino, N. (1992). Concepts, Attributes and Arbitrary Relations: Some Linguistic and Ontological Criteria for Structuring Knowledge Bases. *Data Knowl. Eng.*, 8(3):249–261.

Guarino, N. (1998). *Formal Ontology in Information Systems: Proceedings of the 1st International Conference June 6-8, 1998, Trento, Italy*. IOS Press, Amsterdam, The Netherlands, The Netherlands, 1st edición.

Harris, Z. S. (1954). Distributional structure. *Word*, 10:146–162.

Hasan, K. S. y Ng, V. (2014). Automatic Keyphrase Extraction: A Survey of the State of the Art. En *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1262–1273.

Heart, M. A. (1992). Automatic Acquisition of Hyponyms ˜ om Large Text Corpora Lexico-Syntactic for Hyponymy Patterns. En *International Conference on Computational Linguistics*, COLING '92, pp. 23–28, Stroudsburg, PA, USA. Association for Computational Linguistics.

Hendrickx, I., Kim, S. N., Kozareva, Z., Nakov, P., Ó Séaghdha, D., Padó, S., Pennacchiotti, M., Romano, L., y Szpakowicz, S. (2010). SemEval-2010 Task 8 : Multi-Way Classification of Semantic Relations Between Pairs of Nominals. En *Computational Linguistics*, número June 2009 en DEW '09, pp. 94–99, Stroudsburg, PA, USA. Association for Computational Linguistics.

Hochreiter, S. y Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Comput.*, 9(8):1735–1780.

Hyland, S. L., Karaletsos, T., y Rätsch, G. (2016). A Generative Model of Words and Relationships from Multiple Sources. *Proceedings of the 30th Conference on Artificial Intelligence (AAAI 2016)*, p. 8.

INEE (2015). Segundo Estudio Internacional sobre la Enseñanza y el Aprendizaje (TALIS 2013): Resultados de México.

Kingma, D. P. y Ba, J. (2014). Adam: {A} Method for Stochastic Optimization. *CoRR*, abs/1412.6.

Kohilavani, S., Mala, T., y Geetha, T. V. (2009). Automatic tamil content generation. *2009 International Conference on Intelligent Agent and Multi-Agent Systems, IAMA 2009*.

Komninos, A. (2016). Dependency Based Embeddings for Sentence Classification Tasks. *Naacl2016*, pp. 1490–1500.

Lage, F. J. (2009). Sistemas tutores inteligentes orientados a la enseñanza para la comprensión. *Revista Electrónica de Tecnología Educativa*, pp. 1–19.

Landauer, T. K. y Dumais, S. T. (1997). A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, 104(2):211–240.

Larrañaga, M., Conde, A., Calvo, I., Elorriaga, J. A., y Arruarte, A. (2014). Automatic generation of the domain module from electronic textbooks: Method and validation. *IEEE Transactions on Knowledge and Data Engineering*, 26(1):69–82.

Le, Q. V. y Mikolov, T. (2014). Distributed Representations of Sentences and Documents. En *International Conference on Machine Learning - ICML 2014*, volumen 32, pp. 1188–1196.

Lee, D. y Myung, K. (2017). Read my lips, login to the virtual world. *2017 IEEE International Conference on Consumer Electronics, ICCE 2017*, pp. 434–435.

Lin, Y., Shen, S., Liu, Z., Luan, H., y Sun, M. (2016). Neural Relation Extraction with Selective Attention over Instances. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2124–2133.

McCulloch, W. S. y Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133.

Mikolov, T., Corrado, G., Chen, K., y Dean, J. (2013a). Efficient Estimation of Word Representations in Vector Space. *International Conference on Learning Representations*, pp. 1–12.

Mikolov, T., Yih, W.-t., y Zweig, G. (2013b). Linguistic regularities in continuous space word representations. *Proceedings of NAACL-HLT*, (June):746–751.

Miller, G. A. (1995). WordNet: A Lexical Database for English. *Commun. ACM*, 38(11):39–41.

Mizoguchi, R. y Canada, G. H. (2000). Using Ontological Engineering to Overcome Common AI-ED Problems. *Journal of Artificial Intelligence and Education*, 11:107–121.

Navarro-Almanza, R., Juárez-Ramírez, R., y Licea, G. (2017a). Towards supporting Software Engineering using Deep Learning : A case of Software Requirements Classification. En *5th International Conference in Software Engineering Research and Innovation. CONISOFT*.

Navarro-Almanza, R., Licea, G., Juárez-Ramírez, R., y Mendoza, O. (2017b). *Semantic Capture Analysis in Word Embedding Vectors Using Convolutional Neural Network*, pp. 106–114. Springer International Publishing, Cham.

Nkambou, R., Fournier-Viger, P., y Nguifo, E. M. (2009). Improving the Behavior of Intelligent Tutoring Agents with Data Mining. *IEEE Intelligent Systems*, 24(3):46–53.

Palka, D. L., Rossman, M. C., VanAtten, A. S., y Orphanides, C. D. (2008). *Effect of pingers on harbour porpoise (Phocoena phocoena) bycatch in the US Northeast gillnet fishery*, volumen 10. Cambridge University Press.

Pan, W., Zhong, E., y Yang, Q. (2012). *Transfer Learning for Text Mining*, pp. 223–257. Springer US, Boston, MA.

Perez, A. G. y Benjamins, V. R. (1999). Overview of Knowledge Sharing and Reuse Components : Ontologies and Problem-Solving Methods. *IJCAI-99 workshop on Ontologies and Problem-Solving Method (KRR5)*, pp. 1–15.

Pirrone, R., Pipitone, A., y Russo, G. (2010). Semantic sense extraction from Wikipedia pages. *3rd International Conference on Human System Interaction, HSI'2010 - Conference Proceedings*, pp. 543–547.

Qin, P., Xu, W., y Guo, J. (2016). An empirical convolutional neural network approach for semantic relation classification. *Neurocomputing*, 190:1–9.

Ramírez-Noriega, A., Juárez-Ramírez, R., Jiménez, S., Inzunza, S., Navarro, R., y López-Martínez, J. (2017). An Ontology of the Object Orientation for Intelligent Tutoring Systems. *2017 5th International Conference in Software Engineering Research and Innovation*, pp. 163–170.

Rector, A., Drummond, N., Horridge, M., Rogers, J., Knublauch, H., Stevens, R., Wang, H., y Wroe, C. (2004). OWL Pizzas: Practical Experience of Teaching OWL-DL: Common Errors & Common Patterns. En Motta, E., Shadbolt, N., Stutt, A., y Gibbins, N., editores, *Proceedings of the 14th International Conference on Knowledge Acquisition, Modeling and Management (EKAW 2004)*, volumen 3257 de *Lecture Notes in Computer Science*, pp. 63–81. Springer Berlin Heidelberg.

Rios-Alvarado, A. B., Lopez-Arevalo, I., y Sosa-Sosa, V. J. (2013). Learning concept hierarchies from textual resources for ontologies construction. *Expert Systems with Applications*, 40(15):5907–5915.

Takase, S., Okazaki, N., e Inui, K. (2016). Modeling semantic compositionality of relational patterns. *Engineering Applications of Artificial Intelligence*, 50:256–264.

Urban, G., Geras, K. J., Kahou, S. E., Aslan, O., Wang, S., Caruana, R., Mohamed, A., Philipose, M., y Richardson, M. (2016). Do Deep Convolutional Nets Really Need to be Deep and Convolutional?

Vanlehn, K. (1987). Student modelling. *Medical & biological illustration*, 19:19.

Xiong, S. y Ji, D. (2016). Exploiting flexible-constrained K-means clustering with word embedding for aspect-phrase grouping. *Information Sciences*, 367-368:689–699.

Zeng, D., Liu, K., Chen, Y., y Zhao, J. (2015). Distant Supervision for Relation Extraction via Piecewise Convolutional Neural Networks. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, (September):1753–1762.

Zhao, H., Lu, Z., y Poupart, P. (2015). Self-adaptive hierarchical sentence model. *IJCAI International Joint Conference on Artificial Intelligence*, 2015-Janua:4069–4076.

Zhiping, L. Z. L. y Yu, S. Y. S. (2009). Ontology-Based Knowledge Management in Intelligent Tutoring Systems. *2009 International Conference on Management and Service Science*, pp. 1–4.

Zhou, P., Shi, W., Tian, J., Qi, Z., Li, B., Hao, H., y Xu, B. (2016). Attention-Based Bidirectional Long Short-Term Memory Networks for Relation Classification. En *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 207–212.

Zouaq, A. y Nkambou, R. (2009). Evaluating the generation of domain ontologies in the knowledge puzzle project. *IEEE Transactions on Knowledge and Data Engineering*, 21(11):1559–1572.

Zouaq, A. y Nkambou, R. (2010). A Survey of Domain Ontology Engineering: Methods and Tools. *Advances in Intelligent Tutoring Systems*, pp. 103–119.