UNIVERSIDAD DE MURCIA

FACULTAD DE INFORMÁTICA

Interfaces del Lenguaje Natural para la Consulta y Recuperación de Información de Bases de Conocimiento Basadas en Ontologías

D. Mario Andrés Paredes Valverde

2017

UNIVERSIDAD DE MURCIA

Facultad de Informática

Tesis Doctoral

Interfaces de lenguaje natural para la consulta y recuperación de información de bases de conocimiento basadas en ontologías.

Mario Andrés Paredes Valverde

2017

Directores:

Dr. Rafael Valencia García

Dr. Miguel Ángel Rodríguez García

Agradecimientos

A mi madre y padre por sus consejos, valores y motivación constante que me han permitido alcanzar mis metas, pero sobre todo por su amor.

A mi hermano por su apoyo en todo momento y por esas pláticas a distancia siempre tan ocurrentes.

A Pilar por su gran paciencia, comprensión y por creer más en mí que yo mismo.

A mi amigo Omar por su comprensión y confianza.

A Rafa por darme la oportunidad de realizar el doctorado bajo su tutoría y sobre todo por la amistad brindada. A Ricardo Colomo por permitirme colaborar con él durante mi estancia en Noruega. A Miguel Ángel por su amistad y por la co-dirección de este trabajo.

A mis amigos y amigas del laboratorio, Jojo, Manu, Ginés, Mari Carmen, Fran, Lucía y Astrid. Ha sido un placer compartir estos años con ustedes.

A Marcelino, Asunción y Paty por ese gran apoyo en momentos de debilidad y por hacerme sentir cerca de México durante todas esas tardes.

A mi abuela Irma que dejó en mí y mi familia un hermoso sentimiento.

A mi abuelo, mis tías, tíos, primas y primos por todo el cariño otorgado.

A todos y todas ustedes…

¡GRACIAS!

Índice de contenido

Índice de contenido i

Índice de tablas v

Índice de figuras vi

Lista de acrónimos ix

Capítulo 1. Introducción 1

Capítulo 2. Estado del arte 5

2.1 Introducción 5

2.2 Web Semántica 6

2.2.1 Definición 6

2.2.2 Arquitectura de la Web Semántica 7

2.2.3 Ontología 9

2.2.4 Lenguajes de consulta 21

2.2.5 Linked data 23

2.3 Procesamiento de lenguaje natural 27

2.3.1 Definición 27

2.3.2 Antecedentes 28

2.3.3 Niveles de procesamiento de lenguaje natural 30

2.3.4 Aplicaciones del procesamiento de lenguaje natural 41

2.4 Sistemas de búsqueda de respuestas 49

2.4.1 Definición 49

2.4.2 Antecedentes 50

2.4.3 Esquema básico de un sistema de búsqueda de respuestas 50

2.4.4 Principales métodos 57

2.5 Interfaces de lenguaje natural 63

2.5.1 Introducción 63

2.5.2 Arquitectura genérica de una interfaz de lenguaje natural....64
2.5.3 Interfaces de lenguaje natural orientadas a bases de datos relacionales..66
2.5.4 Interfaces de lenguaje natural orientadas a bases de conocimiento....73
2.5.5 Esfuerzos de investigación para el desarrollo de NLIKB....77
2.6 Conclusiones....87
Capítulo 3. Objetivo de esta tesis doctoral....89
3.1 Motivación....89
3.2 Objetivos....91
3.3 Metodología....91
3.3.1 Estudio del estado del arte.....91
3.3.2 Formalización de la propuesta.....92
3.3.3 Implementación de la propuesta.....92
3.3.4 Validación de la propuesta....92
Capítulo 4. Interfaz de lenguaje natural para bases de conocimiento basadas en ontologías 93
4.1 Introducción....93
4.2 Arquitectura....94
4.2.1 Modelo ontológico de la pregunta....96
4.2.2 Preprocesamiento de la base de conocimiento....100
4.2.3 Procesamiento de la pregunta....105
4.2.4 Clasificación de la pregunta....120
4.2.5 Generación de la consulta en lenguaje formal....123
4.3 Resumen....129
Capítulo 5. Validación de la interfaz de lenguaje natural para bases de conocimiento basadas en ontologías....131
5.1 Introducción....131
5.2 Bases de conocimiento....132
5.2.1 Dbpedia....132

5.2.2 MusicBrainz ... 135
5.3 Metodología de validación ... 138
5.4 Recolección del corpus de preguntas ... 139
5.4.1 Proceso de obtención del corpus ... 139
5.4.2 Corpus de preguntas de DBpedia ... 140
5.4.3 Corpus de MusicBrainz ... 141
5.5 Generación de consultas SPARQL ... 142
5.5.1 Representación de los corpus en formato XML ... 143
5.6 Medidas de evaluación ... 143
5.6.1 Definición de las métricas de evaluación estándar ... 144
5.6.2 Adaptación de las métricas de evaluación ... 145
5.7 Evaluación de la interfaz de lenguaje natural para bases de conocimiento basadas en ontologías ... 146
5.7.1 Resultados obtenidos en el dominio de DBpedia ... 146
5.7.2 Resultados obtenidos en el dominio de MusicBrainz ... 149
5.7.3 Resultados generales ... 152
5.7.4 Discusión de resultados ... 153
5.8 Resumen ... 157
Capítulo 6. Conclusiones y líneas futuras ... 159
6.1 Conclusiones ... 159
6.2 Aportaciones ... 161
6.3 Limitaciones y trabajo a futuro ... 162
Capítulo 7. Contribuciones científicas ... 165
7.1 Publicaciones JCR ... 165
7.2 Publicaciones en revistas ... 165
7.3 Capítulos en libro ... 166
7.4 Congresos internacionales ... 166
Capítulo 8. Summary ... 167

8.1 Introduction 167
8.2 Aims of the thesis 168
8.3 State of art 169
8.4 Results 170
8.4.1 Architecture 170
8.4.2 Question ontology model 171
8.4.3 Knowledge base pre-processing 173
8.4.4 Question processing 173
8.4.5 Question classification 174
8.4.6 Query construction and execution 175
8.5 Evaluation 177
8.5.1 Evaluation results obtained in the DBpedia's domain 178
8.5.2 Evaluation results obtained in the MusicBrainz's domain. 179
8.6 Conclusions and future work 180
Referencias 183

Índice de tablas

Tabla 2-1. Axiomas OWL DL....19

Tabla 2-2. Conjuntos de datos de Linking Open Data por tópico....27

Tabla 2-3. Niveles del lenguaje natural y herramientas PLN....31

Tabla 2-4. Taxonomía de la pregunta (D. Moldovan et al. 2000)....52

Tabla 2-5. Tabla comparativa de interfaces de lenguaje natural para bases de conocimiento.....85

Tabla 4-1. Categorías gramaticales utilizadas por el etiquetador POS.....106

Tabla 4-2. Lista de resultados de una consulta SPARQL para DBpedia.....111

Tabla 4-3. Ordenación por distancia de Levenshtein.....112

Tabla 4-4. Ejemplo de sinónimos obtenidos de WordNet.....113

Tabla 4-5. Dependencias utilizadas.....115

Tabla 4-6. Individuos de la base de conocimiento para el término Paris.....119

Tabla 4-7. Clasificación de preguntas y respuestas.....121

Tabla 4-8. Ejemplos más frecuentes de patrones de pregunta obtenidos.....122

Tabla 4-9. Elementos de la base de conocimiento identificados en la pregunta.....123

Tabla 4-10. Plantillas de tripletas RDF.....125

Tabla 4-11. Extracto de los resultados de DBpedia para una consulta SPARQL.....128

Tabla 5-1. Instancias por clase de la ontología de DBpedia.....135

Tabla 5-2. Descripción del corpus de preguntas para el dominio de DBpedia.....140

Tabla 5-3. Descripción del corpus de preguntas para el dominio de MusicBrainz.....141

Tabla 5-4. Resultados de evaluación en DBpedia.....148

Tabla 5-5. Resultados obtenidos en MusicBrainz.....151

Table 8-1. Question and answers classification.....175

Table 8-2. RDF-based templates.....176

Índice de figuras

Figura 2-1. Arquitectura de la Web Semántica. 7

Figura 2-2 Clasificación de Ontologías establecida por Guarino (N. Guarino 1998). 11

Figura 2-3. Ejemplo de un grafo RDF. 13

Figura 2-4. Representación gráfica de la relación entre los lenguajes y perfiles OWL. 17

Figura 2-5. Arquitectura de un sistema de representación de conocimiento basado en lógica descriptiva. 18

Figura 2-6. Ejemplo de consulta SPARQL. 21

Figura 2-7. Nube de Linking Open Data de agosto 2014 26

Figura 2-8. Etapas de análisis en el procesamiento de lenguaje natural. 31

Figura 2-9. Análisis de constituyentes sintácticos. 35

Figura 2-10. Análisis sintáctico de dependencias. 36

Figura 2-11. Ejemplo de semántica de la oración 40

Figura 2-12. Arquitectura básica de un sistema de búsqueda de respuestas. 51

Figura 2-13. Proceso de análisis de la pregunta de un sistema QA. 51

Figura 2-14. Ejemplo de representación lógica. 62

Figura 2-15. Arquitectura genérica de una interfaz de lenguaje natural 64

Figura 2-16. Arquitectura NLIDB basada en la coincidencia de patrones. 69

Figura 2-17. Arquitectura NLIDB basada en la sintaxis. 70

Figura 2-18. Arquitectura NLIDB basada en gramática semántica. 70

Figura 2-19. Arquitectura de NLIDB basada en un lenguaje de representación intermedio. 71

Figura 2-20. Componentes de alto nivel de un sistema de pregunta-respuesta para Linked Data. 74

Figura 4-1. Arquitectura de la NLIKB. 95

Figura 4-2. Modelo ontológico de la pregunta. 97

Figura 4-3. Elementos de la URI. 101

Figura 4-4. Extracto del gazetteer generado en la fase de preprocesamiento. 104

Figura 4-5. Módulo de procesamiento de la pregunta..... 105

Figura 4-6. Ejemplo de tokenización. 106

Figura 4-7. Ejemplo de etiquetador POS. 107

Figura 4-8. Ejemplo de lematización. 107

Figura 4-9. Ejemplo de reconocimiento de entidad nombrada. 108

Figura 4-10. Ejemplo de reglas de RegexNER de Stanford NLP. 109

Figura 4-11. Consulta SPARQL para la recuperación de datos de acuerdo a similitud de cadena..... 110

Figura 4-12. Ejemplo de análisis de una pregunta..... 114

Figura 4-13. Análisis de dependencias. 116

Figura 4-14. Reglas RegexNER para diferenciar palabras con significado verbo y sustantivo..... 118

Figura 4-15. Análisis de la pregunta: Who is the mayor of Paris?. 119

Figura 4-16. Ejemplo de consulta SPARQL con cláusula ASK. 120

Figura 4-17. Análisis de preguntas en lenguaje natural. 122

Figura 4-18. Ejemplos de análisis de la pregunta..... 123

Figura 4-19. Tripletas RDF generadas para una relación sintáctica nsubjpass. 126

Figura 4-20. Tripletas RDF generadas para una relación sintáctica nmodAgent..... 126

Figura 4-21. Tripletas RDF generadas para una relación sintáctica nmodIn. 126

Figura 4-22. Tripletas RDF generadas para la relación sintáctica acl. 126

Figura 4-23. Dependencias que comparten elementos..... 127

Figura 4-24. Tripletas generadas para las relaciones sintácticas nsubjpass y nmod:agent. 127

Figura 4-25. Tripletas generadas para las relaciones sintácticas acl y nmod:in. 127

Figura 4-26. Tripletas RDF resultantes tras la unión de dos grupos generales de tripletas RDF. 128

Figura 4-27. Consulta SPARQL final. 128

Figura 5-1. Ejemplo de artículo de Wikipedia. 134

Figura 5-2. Corpus de preguntas para DBpedia. 141

Figura 5-3. Corpus de preguntas para MusicBrainz. 142

Figura 5-4. Registro XML de una pregunta en el corpus de preguntas de DBpedia. 143

Figura 5-5. Resultados de evaluación en DBpedia. 147

Figura 5-6. Tipos de resultado en DBpedia. 149

Figura 5-7. Resultados de evaluación en MusicBrainz. 150

Figura 5-8. Tipos de resultado en MusicBrainz. 151

Figura 5-9. Resultados promedio en ambos dominios. 152

Figura 5-10. Tipos de resultados en ambos dominios. 153

Figure 8-1. NLI's architecture. 171

Figure 8-2. Question model. 172

Figure 8-3. Evaluation results obtained in the DBpedia's domain. 178

Figure 8-4. Evaluation results obtained in the MusicBrainz's domain. 179

Lista de acrónimos

API	Application Programming Interface
DAG	Directed Acyclic Graph
DBMS	Database Management System
DL	Description Logics
DPC	Domain Processing Components
EI	Extracción de Información
FOL	First Order Logic – Lógica de primer orden
GUI	Graphical User Interfaz
IA	Inteligencia Artificial
IDF	Inverse Document Frequency
KR	Knowledge Representation
LOD	Linking Open Data
MRL	Meaning Representation Language
MT	Machine Translation
MUC	Message Understanding Conference
NER	Named Entity Recognition
NLG	Natural Language Generation
NLI	Natural Language Interfaz
NLIDB	Natural Language Interfaces to Databases
NLIKB	Natural Language Interfaces to Knowledge Bases
NLU	Natural Language Understanding
NS	Namespace
OWL	Web Ontology Language
PLN	Procesamiento de Lenguaje Natural
QA	Question Answering

RDF	Resource Description Framework
RDFS	Resource Description Framework Schema
RI	Recuperación de Información
RNE	Reconocimiento de Nombres de Entidades
SPARQL	SPARQL Protocol and RDF Query Language
SQL	Structured Query Language
SRL	Semantic Role Labeling
SVM	Support Vector Machine
URL	Uniform Resource Locator
URI	Universal Resource Identifier
W3C	World Wide Web Consortium
WSD	Word Sense Disambiguation
XML	eXtensible Markup Language

Capítulo 1. Introducción

La Web Semántica (Berners-Lee et al. 2001) es considerada como una extensión de la Web actual, donde la información tiene un significado bien definido y entendible no solo por humanos, sino también por computadoras, permitiendo así que estas últimas puedan automatizar, integrar y reutilizar información de alta calidad a través de distintas aplicaciones. Tim Berners-Lee, quien es considerado el padre de la Web, estableció una arquitectura basada en capas para implementar la Web Semántica. De tal arquitectura, una de las tecnologías más sobresalientes son las ontologías, las cuales son consideradas como uno de los pilares de la Web Semántica[1]. Como será descrito en el capítulo 2 de esta tesis, existen diversas definiciones formales del concepto ontología. Sin embargo, una de las más extendidas es la provista por Studer y colaboradores (Studer, Benjamins, and Fensel 1998) quienes la definen como una especificación formal y explícita de una conceptualización compartida. En otras palabras, una ontología permite representar formalmente y de manera explícita estructuras de conocimiento a través de conceptos, sus propiedades, sus atributos, las relaciones con otros conceptos y los axiomas relacionados con estos (Legaz García 2015).

Dadas las ventajas de la Web Semántica descritas en el párrafo anterior. En los últimos años, un gran número de individuos y organizaciones en diversos dominios han adoptado el enfoque basado en ontologías para publicar su información. Esto ha provocado un exponencial crecimiento de información disponible en la Web e intranets representada a través de RDF (RDF Working Group 2016). Tal fenómeno también ha dado paso a la necesidad de contar con mecanismos que permitan el acceso a esta información a todo tipo de usuarios. Actualmente, existen herramientas enfocadas en resolver tal necesidad, una de ellas son los lenguajes formales de consulta como SPARQL. Sin embargo, este enfoque resulta complicado para usuarios ocasionales (Kaufmann, Bernstein, and Fischer 2007), ya que es necesario que este cuente con cierto grado de conocimiento y experiencia en tecnologías de la Web Semántica tales como RDF, OWL, generación de consultas en

[1] http://semanticweb.org/wiki/Ontology.html

lenguaje formal e incluso conocer la estructura de datos de la base de conocimiento a consultar (OWL Working Group 2016).

La necesidad de hacer accesible la información disponible bajo el enfoque de la Web Semántica a todo tipo de usuarios, sean expertos u ocasionales, demanda el desarrollo de mecanismos de recuperación intuitivos y fáciles de usar. Tomando en cuenta esto, el paradigma de recuperación de información basado en lenguaje natural es generalmente considerado como el más intuitivo desde un punto de vista de uso (Cimiano et al. 2008), pues permite al usuario emplear todo el poder de expresividad del lenguaje natural en vez de un lenguaje poco natural o limitado. Además, este paradigma oculta la estructura de la base de conocimientos, así como del lenguaje de consulta.

Lo expuesto anteriormente ha sido la principal motivación para llevar a cabo el trabajo de investigación que se describe en esta tesis doctoral. Cuyo objetivo principal es desarrollar soluciones basadas en tecnologías de procesamiento natural y Web Semántica que permitan reducir la brecha existente entre el usuario y las bases de conocimiento basadas en ontologías a través del lenguaje natural. Para cumplir con este objetivo, se ha seguido una metodología compuesta por cuatro partes principales que son:

- **Estudio del estado del arte**. En esta parte se llevó a cabo un análisis de todos aquellos desarrollos de última tecnología en los contextos de Web Semántica, Procesamiento del Lenguaje Natural (PLN) e interfaces de lenguaje natural. Con respecto a este último punto se hizo hincapié en analizar enfoques existentes orientados a bases de conocimiento basadas en ontologías.
- **Formalización de la propuesta**. Esta parte contempla la definición y formalización de una interfaz de lenguaje natural para bases de conocimiento basadas en ontologías, la cual estará basada en tecnologías de PLN y Web Semántica.
- **Implementación de la propuesta**. Esta etapa consiste en la implementación de la interfaz de lenguaje natural propuesta por medio de herramientas de PLN y Web semántica.
- **Validación de la propuesta**. Finalmente, esta etapa consistió en la validación de la interfaz propuesta en dos bases de conocimiento diferentes, a saber, DBpedia y MusicBrainz. El objetivo de esta etapa es evaluar la efectividad de la interfaz para proveer la respuesta correcta dada una pregunta en lenguaje natural. Además, la validación en una segunda base de conocimiento persiguió el objetivo de validar la portabilidad de la interfaz.

Cada uno de las tareas especificadas en la metodología han sido llevadas a cabo consiguiendo los resultados que se presentan en esta tesis, la cual está constituida por una serie de capítulos que se describen a continuación.

En el siguiente capítulo, capítulo 2, se provee una descripción del estado actual de las tecnologías involucradas en la presente investigación, que son Web Semántica, PLN e interfaces de lenguaje natural. Concretamente, este capítulo describe en primer lugar, la arquitectura de la Web Semántica resaltando las tecnologías de ontologías, lenguajes de consulta y Linked Data. Posteriormente, se describen los niveles de PLN, así como algunas de las aplicaciones de esta tecnología. En cuanto a las interfaces de lenguaje natural, se proporciona una descripción de algunas arquitecturas utilizadas en el desarrollo este tipo de sistemas. Finalmente, se provee una comparación de interfaces de lenguaje natural orientas a bases de conocimiento basadas en ontologías.

En el capítulo 3 discute la principal motivación para llevar a cabo el presente trabajo de investigación. En esta sección se proveen tanto el objetivo general como los objetivos específicos establecidos. Además, se describe la metodología seguida durante el desarrollo de este proyecto.

El capítulo 4 presenta la interfaz de lenguaje natural para bases de conocimiento basadas en ontologías que se propone en esta tesis. Aquí, se describe su arquitectura y se describe su funcionamiento general. También, este capítulo proporciona una descripción detallada de una de las principales contribuciones de este trabajo, es decir, del modelo ontológico de la pregunta que permite describir su estructura sintáctica y su contexto.

El capítulo 5 describe los experimentos de evaluación realizados para medir la efectividad de la interfaz propuesta. La evaluación se basa en proveer la respuesta correcta a una pregunta en leguaje natural a partir de una base de conocimiento. Tales experimentos se llevaron a cabo en dos bases de conocimiento diferentes, a saber, DBpedia y MusicBrainz, con el objetivo adicional de comprobar la portabilidad de la interfaz.

El capítulo 6 agrupa conclusiones finales, discusión de las principales contribuciones y limitaciones del trabajo realizado, así como las posibles vías futuras que permitirán direccionar estas últimas.

Las contribuciones científicas y contribuciones a congreso derivadas de este trabajo se presentan en el capítulo 7.

Finalmente, en el capítulo 8 se incluye un breve resumen de la tesis en inglés.

Capítulo 2. Estado del arte

2.1 Introducción

El presente capítulo provee una descripción del estado actual de las tecnologías involucradas en este trabajo de tesis doctoral que son: Web Semántica, PLN e interfaces de lenguaje natural.

En el siguiente apartado se define el concepto de Web Semántica y se provee una descripción de cada uno de las tecnologías que componen su arquitectura, la cual fue propuesta por Tim Berners-Lee, quien es considerado el padre de la Web. Después se describe a detalle uno de los componentes principales de la Web Semántica, que es de gran importancia para este trabajo, las ontologías. Acerca de esta tecnología, se presentan sus tipos, elementos y los diferentes lenguajes utilizados para su desarrollo. Finalmente, se presenta Linked Data, un conjunto de buenas prácticas presentes en la Web Semántica para publicar y enlazar datos estructurados en la Web.

La tercera parte de este apartado provee una descripción de PLN, tecnología clave en el desarrollo de este trabajo. Esta inicia con la definición de la tecnología, y se presentan los antecedentes históricos de la misma. Después se analizan los diferentes niveles de PLN, donde se hace hincapié en el análisis sintáctico, el cual juega un papel fundamental en la interfaz de lenguaje natural propuesta. Finalmente, se presentan algunas aplicaciones del PLN tales como recuperación de información, minería de datos, generación automática de resúmenes y análisis de sentimientos.

Una de las aplicaciones del PLN altamente relacionada con este trabajo corresponde a los sistemas de búsqueda de respuestas. Por tal motivo, se describe a detalle dicha tecnología, iniciando con su definición y con una breve historia de esta. Posteriormente, se presenta la arquitectura típica de un sistema de búsqueda de respuestas y se detallan cada uno de sus componentes. A continuación, se describen los principales métodos utilizados en este tipo de sistemas. Por ejemplo, la bolsa de palabras, el análisis morfo-sintáctico, la clasificación del tipo de respuesta esperado o la traducción a un lenguaje estructurado, entre otros.

La quinta sección aborda el campo de las interfaces de lenguaje natural, principal tecnología que aborda el presente trabajo. En esta sección se describe la arquitectura genérica de una interfaz de lenguaje natural. Posteriormente, se provee un estudio de las interfaces de lenguaje natural orientadas a bases de datos relacionales, que incluye sus

antecedentes, ventajas y desventajas, así como las principales arquitecturas utilizadas en el desarrollo de este tipo de aplicaciones, tales como coincidencia de patrones, basada en la sintaxis, basada en la gramática semántica y lenguajes de representación intermedia. Posteriormente, se describe el desarrollo de interfaces de lenguaje natural orientadas a bases de conocimiento, donde se hace énfasis en las interfaces orientadas a Linked Data. En este sentido, se presentan los componentes de un sistema de pregunta respuesta para Linked Data y las diferentes campañas de evaluación de este tipo de sistemas. Finalmente, se presenta un estudio de interfaces de lenguaje natural para bases de conocimiento existentes en la literatura, las cuales son comparadas con la interfaz presentada en esta tesis doctoral a través de seis criterios que son: tipo de interfaz de lenguaje natural, portabilidad, dominio, técnicas de PLN, lenguaje y enfoque para la representación intermedia de la pregunta.

2.2 Web Semántica

2.2.1 Definición

La mayor parte del contenido que existe actualmente en la Web está diseñada únicamente para ser entendida por los humanos y no por computadoras. Estas últimas podrían procesar tal información en mucho menor tiempo y sobre todo de manera significativa para el humano. Con el objetivo de dar solución al problema antes planteado surge la Web Semántica, la cual es considerada una extensión de la Web actual, en la cual se le asigna un significado bien definido a la información permitiendo a computadoras y humanos trabajar de manera cooperativa (Unger, Freitas, and Cimiano 2014). En otras palabras, la Web Semántica trata de proveer al contenido actual de la Web de una estructura entendible no solo por los humanos, sino también por agentes de software, con el objetivo de que estos sean capaces de procesar y comprender la información y con ello resolver las necesidades específicas de información de los usuarios.

Para que los agentes de software cumplan con el objetivo antes mencionado. La información contenida en la Web debe poseer una estructura, es decir, debe estar representada a través de un conjunto predefinido de atributos, valores y relaciones (Hsieh and Shipman 2002). Además, es necesario un conjunto de reglas de inferencia que permita a los agentes inteligentes llevar a cabo tareas de razonamiento automático. Estas reglas no son más que formas lógicas que consisten de una función que toma premisas, las analizan y obtiene una conclusión o conclusiones.

De esta manera, el principal reto de la Web Semántica es proveer un lenguaje que exprese tanto datos como reglas de razonamiento y que permita que las reglas de cualquier sistema de representación de conocimiento puedan ser exportadas a la Web (Unger, Freitas, and Cimiano 2014). Así, Tim Berners-Lee propuso una primera arquitectura por capas para implementar la Web Semántica la cual se describe a continuación.

2.2.2 Arquitectura de la Web Semántica

En 2006, Tim Berners-Lee define la Web Semántica como una arquitectura en capas, donde cada capa contiene diferentes tecnologías semánticas (ver Figura 2-1). El orden de estas capas responde al nivel de abstracción de cada una de ellas. En un nivel inferior se encuentran las tecnologías encargadas de la identificación y representación de los recursos y en el nivel superior las tecnologías futuras que permitirán tener una Web Semántica inteligente. A continuación, se describe cada una de las capas de la arquitectura de la Web Semántica.

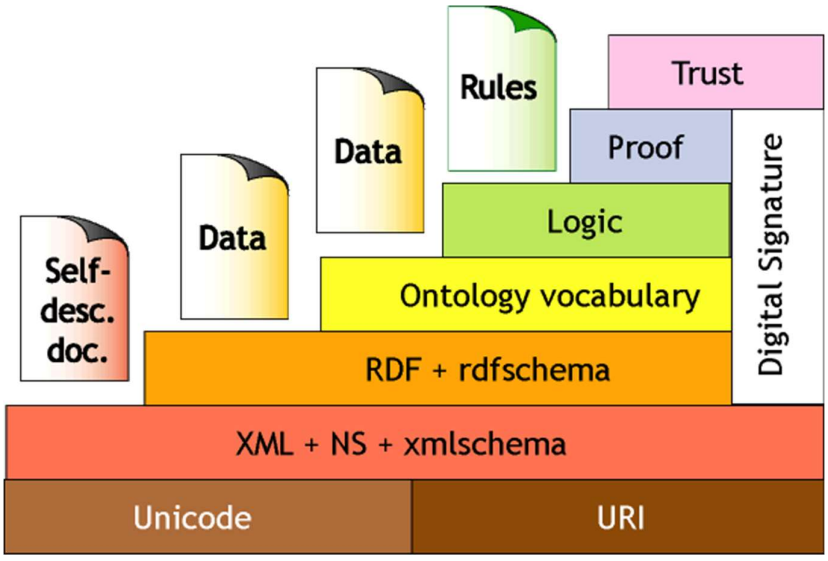

Figura 2-1. Arquitectura de la Web Semántica.

- **Unicode y URI**. En el nivel más bajo se encuentran tecnologías Unicode y URI. Por un lado, Unicode (Unicode Staff 1991) es un estándar de codificación de caracteres que permite que todos los lenguajes humanos puedan ser utilizados en la Web de una manera estandarizada. Por otro lado, URI (Masinter, Berners-Lee, and Fielding 2005) provee un mecanismo de identificación uniforme de los recursos. En conjunto, estas tecnologías permiten identificar de manera

inequívoca los recursos existentes en la Web, así como representarlos correctamente en cualquier idioma.

- **XML + NS + XML Schema**. La tecnología XML (*eXtensible Markup Language*) (World Wide Web Consortium 2016a) es un formato de texto simple y flexible que juega un papel importante en el intercambio de información tanto en la Web como en otros sitios. XML NS (*namespaces*) permite proveer un nombre único a los elementos y atributos contenidos en un documento XML. Finalmente, XML Schema (World Wide Web Consortium 2016b) permite describir la estructura de un documento XML a través de la definición de los bloques del documento tales como los elementos y atributos, los tipos de datos para esos elementos, así como los valores de los mismos. El objetivo de estas tecnologías es asegurar la integración de la definición de la Web Semántica con los demás estándares basados en XML.
- **RDF + RDF Schema**. El lenguaje RDF (*Resource Description Framework*) (RDF Working Group 2016) permite representar todos los recursos contenidos en la Web Semántica. En este contexto, un recurso es cualquier cosa que puede tener datos asociados, sea este una entidad del mundo real o abstracta. RDF está basado en tripletas de la forma sujeto-predicado-objeto. Estas tripletas representan una relación entre dos objetos, siendo estos dos el sujeto y el objeto. El elemento restante representa la naturaleza de la relación entre dichos elementos. Los tres elementos que conforman la tripleta se identifican inequívocamente a través de URI. RDF permite representar los recursos en forma de grafos dirigidos etiquetados, donde el sujeto y objeto son los grafos y el predicado el arco que conecta ambos nodos. Por otro lado, RDF Schema (RDFS) (Brickley and Guha 2016) es una extensión del vocabulario RDF que permite describir los recursos como clases, organizar estas clases en jerarquías, definir las relaciones entre las clases, definir las relaciones entre clases como propiedades, así como definir sus dominios y rangos.
- **Ontology vocabulary** (Vocabulario ontológico). Esta capa provee un conjunto de reglas de inferencia que mejoran la funcionalidad y expresividad de la capa anterior. Además, este vocabulario proporciona nuevos conceptos, relaciones y propiedades que permiten conceptualizar un dominio concreto.
- **Capa lógica**. La capa lógica permite definir reglas de inferencia que posibilita a los agentes de software procesar y relacionar información, así como inferir nuevo conocimiento sobre el ya disponible. Además, estas reglas permiten procesar la información a nivel semántico de manera automática.

- **Capa de prueba**. La capa de prueba ejecuta y evalúa las sentencias de la capa lógica (aserciones) para que, en conjunto, con la capa de confianza, determinen la confiabilidad de las fuentes de información.
- **Capa de confianza**. Esta capa tiene como objetivo evaluar las pruebas ofrecidas por la capa anterior y con ello comprobar de manera exhaustiva la fiabilidad de las fuentes.
- **Firma digital**. El objetivo de esta capa es definir el ámbito de confianza para las capas de Prueba y Web Semántica. La firma digital se lleva a cabo mediante mecanismos de criptografía tales como la firma digital que permite a los ordenadores y agentes de software verificar la seguridad de la información, así como la confiablidad de la fuente.

Con el objetivo de mejorar la comprensión del presente trabajo de tesis, en las siguientes secciones se describen más a detalle las tecnologías de la Web Semántica utilizadas a lo largo de la investigación reportada en este documento.

2.2.3 Ontología

2.2.3.1 Definición

Uno de los componentes principales de la Web Semántica son las ontologías. El término ontología fue tomado del contexto de la filosofía en el cual se le define como: "*una teoría sobre la naturaleza de la existencia*". Sin embargo, en el contexto de IA (Inteligencia Artificial) y la Web, una ontología es: "*un documento o archivo que define formalmente las relaciones entre los términos*" (Unger, Freitas, and Cimiano 2014). Continuando en el contexto de IA, Robert Neches y colaboradores (Neches et al. 1991) establecen que: "*una ontología define los términos básicos y relaciones que conforman el vocabulario de un área específica, así como las reglas para combinar dichos términos y las relaciones para definir extensiones de vocabularios*".

En el contexto de ciencias de la información, existen diversas definiciones formales del concepto ontología. Por ejemplo, Guarino (Guarino 1998) define: "*una ontología representa una visión común, compartible, y reutilizable del conocimiento de un dominio de aplicación*". Más tarde, el mismo autor en conjunto con otros colaboradores (Guarino, Oberle, and Staab 2009) establecen que "*una ontología es un tipo de objeto de información o un artefacto computacional que permite modelar formalmente la estructura de un sistema, es decir, las entidades y relaciones que emergen de su observación, y las cuales son útiles para nuestros propósitos*".

Una de las definiciones más extendida es la provista por Thomas Robert Gruber (Gruber 1993), quién la define como "*una especificación explícita de una conceptualización*". Años más tarde, Borst (Borst, Akkermans, and Top 1997) la define como "*una especificación formal de una conceptualización compartida*". La diferencia entre estas definiciones radica en que la segunda hace hincapié en el concepto *compartida*, pues se prioriza que tal conceptualización sea una visión consensuada en vez de representar una visión particular. Posteriormente, y tomando como base las definiciones antes mencionadas, Studer y colaboradores (Studer, Benjamins, and Fensel 1998) definen a la ontología como "*una especificación formal y explícita de una conceptualización compartida*".

Como podemos notar, existen una gran cantidad de definiciones para el concepto de ontología que coinciden en las características de formalidad, explicitud y consensual. De esta manera, podemos considerar a una ontología como un modelo que representa formalmente y de manera explícita estructuras de conocimiento a través de conceptos, sus propiedades, sus atributos, las relaciones con otros conceptos y los axiomas relacionados con estos (Legaz García 2015).

En las siguientes secciones se describirán cada uno de los conceptos que componen a una ontología, así como aquellos lenguajes que permiten representarlas, tales como RDFS y OWL (McGuinness and Harmelen 2016). Pero antes, se presentan una serie de clasificaciones de ontologías que han sido propuestas en la literatura.

2.2.3.2 Tipos de ontologías

De acuerdo con (Steve, Gangemi, and Pisanelli 1997) existen tres tipos fundamentales de ontologías que son:

- **Ontologías de dominio**. Este tipo de ontologías representa el conocimiento especializado relacionado con un dominio o subdominio en particular, tales como la medicina (Mustaffa, Ishak, and Lukose 2012), oncología (Kumar and Smith 2005), biología (Schulz et al. 2006), Computación en la Nube (Rodríguez-García et al. 2014), entre otros.
- **Ontologías genéricas**. A través de este tipo de ontologías se representan conceptos generales y fundacionales del conocimiento tales como las estructuras parte/todo, la cuantificación, así como los procesos o los tipos de objetos. Algunos ejemplos de este tipo de ontologías son las presentadas en (Sowa 1995) y (Borgo, Guarino, and Masolo 1996).
- **Ontologías representacionales**. Estas ontologías especifican las conceptualizaciones que subyacen a los formalismos de representación del

conocimiento (Guarino and Boldrin 1993), debido a lo cual, este tipo de ontologías suelen denominarse también como meta-ontologías. Un ejemplo claro de este tipo de ontologías es la ontología presentada en (Gruber 1993).

Por su parte, (Guarino 1998) establece una clasificación de ontologías de acuerdo a su nivel de dependencia respecto a una tarea en particular o un punto de vista. Esta clasificación se aprecia en la Figura 2-2.

Figura 2-2 Clasificación de Ontologías establecida por Guarino (N. Guarino 1998).

- **Ontologías de alto nivel**. Estas describen conceptos generales tales como espacio, tiempo, objetos, eventos, acciones, entre otros, los cuales no dependen del dominio o problema en particular.
- **Ontologías de dominio y de tarea**. Estas ontologías describen el vocabulario relacionado con un dominio en particular o una tarea en específico. Estas especializan los términos especificados por las ontologías de alto nivel.
- **Ontologías de aplicación**. Estas describen los conceptos dependientes tanto del dominio en particular como de la tarea en específico. Es decir, como se aprecia en la Figura 2-2, este tipo de ontologías son a menudo especializaciones de las ontologías de dominio y de tarea.

Es importante mencionar que las clasificaciones mostradas anteriormente no son las únicas que existen en la literatura. Sin embargo, se encuentran dentro de las clasificaciones más extendidas.

2.2.3.3 Elementos de la ontología

En general, una ontología es expresada a través de cinco elementos fundamentales (Rodríguez García 2014): conceptos, atributos, individuos, relaciones y axiomas.

- **Conceptos**. Un concepto, también conocido como clase, término o tipo, representa cualquier entidad o *cosa* dentro de un dominio. Estos proveen un mecanismo de abstracción que permite agrupar recursos que cuenten con características similares. Este puede poseer diferentes atributos, así como establecer relaciones con otros conceptos.
- **Atributos**. Estos elementos representan la estructura interna de los conceptos. Existen dos tipos de atributos atendiendo a su origen: específicos y heredados. Los atributos específicos son los propios del concepto al que pertenecen. Los atributos heredados vienen dados por las relaciones taxonómicas en las que el concepto desempeña el rol de hijo, por lo que hereda los atributos de la clase padre.
- **Individuos**. También conocidos como instancias, estos representan elementos concretos de un dominio, los cuales están descritos en términos de sus conceptos. Algunos autores enfatizan la diferencia entre clase e individuo, mientras la primera hace referencia a lo que es general en la realidad, el segundo hace referencia a lo que es particular en la realidad, es decir, a aquello que existe en tiempo y espacio (B. Smith 2004).
- **Relaciones**. Estos elementos describen las interacciones entre conceptos de un dominio. Estas se definen formalmente como cualquier subconjunto de un producto de *n* conjuntos, esto es: *R: C1 x C2 x … x Cn*. Las relaciones generalmente suelen ser binarias y pueden ser expresadas directamente entre individuos, entre conceptos o entre ambos. Algunos ejemplos de relaciones binarias son la especialización (es-un) o la de composición (parte-todo).
- **Axiomas**. Estas representan expresiones que siempre son ciertas y son usadas para restringir los valores de clases o instancias. Este tipo de elementos son incluidos en la ontología con propósitos tales como la definición del significado de los componentes ontológicos, así como de restricciones complejas sobre los valores de los atributos, argumentos de relaciones, entre otros. Además, los axiomas permiten verificar la corrección de la información especificada en la ontología o deducir nueva información. Cabe mencionar que las ontologías que incluyen axiomas son denominadas ontologías pesadas, mientras que las que no contienen son llamadas ontologías ligeras.

2.2.3.4 Lenguajes para el desarrollo de ontologías

Existen diversos lenguajes que disponen de la capacidad de desarrollar ontologías. Ejemplos de estos lenguajes son SHOE, RDF, RDFS y OWL, siendo los dos últimos los más utilizados actualmente.

2.2.3.4.1 SHOE

SHOE (Luke et al. 1997) es un lenguaje basado en marcos con una sintaxis XML que puede ser embebido en documentos HTML. SHOE permitía definir clases, relaciones entre clases, así como reglas de inferencia expresadas en fórmulas de cláusulas de Horn (Heflin, Hendler, and Luke 1998). Además, este lenguaje utiliza referencias URI para los nombres de recursos. Sin embargo, SHOE contaba con una serie de limitaciones tal como la carencia de mecanismos que permitiesen expresar negaciones o disyunciones.

2.2.3.4.2 RDF

RDF permite expresar el significado de la información a través de un conjunto de tripletas compuestas por dos nodos (sujeto y objeto) unidas por un arco (predicado) (Klyne, Carrol, and McBride 2016). Estas tripletas expresan afirmaciones tales como: *un escritor* (sujeto) *es el autor de* (predicado) *un libro* (objeto). En RDF, los tres elementos anteriores son identificados por una URI (*Universal Resource Identifier*), es decir, por una cadena de caracteres que sigue un estándar y que permite identificar los recursos de una red de manera unívoca. La Figura 2-3 presenta un ejemplo de grafo RDF. En dicha imagen, se aprecian dos nodos, el de la izquierda representa un libro cuyo título es *XLST Quickly*, mientras que el de la derecha representa un autor cuyo nombre y apellido es *Bob Duchame*. Estos elementos están relacionados a través del predicado *creator* (creador).

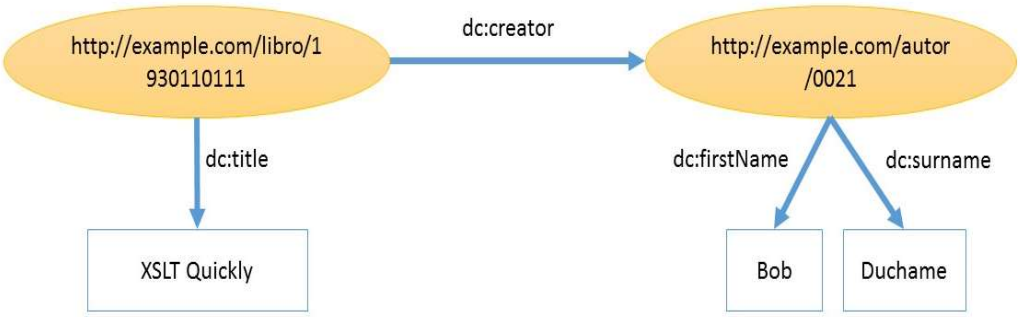

Figura 2-3. Ejemplo de un grafo RDF.

2.2.3.4.3 RDFS

RDFS es una extensión de RDF que define un vocabulario con elementos tales como *rdfs:Class*, *rdfs:Resource* y *rdf:Property*, los cuales permiten definir clases, recursos y propiedades, respectivamente. Gracias a RDFS es posible definir relaciones de pertenencia entre clases, así como dominios y rangos para las propiedades. Para ello provee el siguiente conjunto de propiedades:

- **rdfs:range**. Esta propiedad se utiliza para establecer que los valores de una propiedad son instancias de una o más clases.
- **rdfs:domain**. Este recurso se utiliza para establecer que cualquier recurso que tenga una determinada propiedad es una instancia de una o más clases.
- **rdf:type**. Este elemento se usa para establecer que un recurso es una instancia de una clase.
- **rdfs:subClassOf**. Esta propiedad permite modelar las jerarquías de clases.
- **rdfs:subPropertyOf**. Esta permite modelar las jerarquías de propiedades.
- **rdfs:label**. Este elemento puede ser utilizado para proveer una versión entendible por el humano del nombre del recurso.
- **rdfs:comment**. Esta propiedad puede ser utilizada para proveer una descripción del recurso en lenguaje natural entendible por el humano.

RDFS es considerado un lenguaje ontológico demasiado primitivo (Antoniou and Harmelen 2004), ya que básicamente permite la organización de vocabularios en jerarquías y no permite establecer restricciones de cardinalidad o expresar clases disjuntas, ni la combinación booleana de clases, ni expresar características de las propiedades tales como transitividad, simetría, unicidad, propiedad inversa, entre otros. Sin embargo, es importante mencionar que RDFS sí permite lleva a cabo tareas de razonamiento automáticas para inferir nuevas relaciones sobre una base de conocimiento dada (Rodríguez García 2014).

2.2.3.4.4 OWL

El lenguaje OWL fue desarrollado por el Web Ontology Working Group formado por la W3C (World Wide Web Consortium) y en febrero de 2004 se convirtió en una recomendación de la W3C. En 2009 se publicó la versión OWL 2, la cual representa la última versión hasta el momento.

El lenguaje OWL puede ser usado para representar explícitamente el significado de términos en vocabularios y las relaciones entre esos términos (OWL Working Group 2016). Este lenguaje está diseñado para que agentes de software puedan procesar y

explotar el contenido de la información de manera más significativa, ya que este proporciona vocabulario adicional que permite describir clases y propiedades junto con una semántica formal. En OWL, al igual que ocurre en RDFS, los elementos básicos son las clases, propiedades e individuos. El primer elemento define grupos de individuos que comparten propiedades. Los individuos representan las instancias de las clases. Con respecto a las propiedades, en OWL se distinguen dos tipos de estas, que son *owl:ObjectProperty* y *owl:DatatypeProperty*. La primera permite establecer relaciones entre individuos. Por ejemplo, la relación *tieneHijo* relaciona una instancia de la clase Persona con otra instancia de la misma clase. El segundo tipo de propiedad relaciona individuos con un valor literal, por ejemplo, la relación *tieneEdad* relaciona una instancia de la clase Persona con un valor numérico.

El lenguaje OWL provee tres sub-lenguajes que van incrementando su nivel de expresividad y cuyo uso está orientado a comunidades o usuarios específicos.

- **OWL Lite**. Este sub-lenguaje es el menos expresivo. Este fue diseñado principalmente para representar jerarquías de clasificación con restricciones simples tales como restricciones de rango local, existenciales y de cardinalidad simple (valores de 0 o 1). Además, este permite establecer propiedades como la inversa, transitiva y simétrica.
- **OWL DL**. Este sub-lenguaje proporciona la mayor expresividad posible garantizando que todas las conclusiones son computables y finalizarán en un tiempo finito, es decir, garantiza la decibilidad, la cual, en el contexto de lógica, se refiere a la existencia de un método efectivo para determinar si un objeto es miembro de un conjunto de fórmulas. OWL DL incluye todos los constructores del lenguaje OWL. Sin embargo, solo pueden ser usados bajo ciertas restricciones tal como la condición de separación en el tipo de recurso. El nombre de este lenguaje viene dado por su correspondencia con la lógica descriptiva (*Description Logics*) (Baader and Nutt 2003), un campo de la investigación que ha estudiado las lógicas que forman la base formal de OWL.
- **OWL Full**. Este sub-lenguaje provee la máxima expresividad y la libertad sintáctica de RDF gracias a la posibilidad de utilizar todos los constructores y primitivas definidas en OWL. Sin embargo, tal nivel de expresividad y libertad se traducen en la no garantía de decidibilidad. OWL Full incorpora los niveles Lite y DL y permite mezclar libremente OWL y RDF.

La versión OWL 2 incorpora nuevas funcionalidades que mejoran la expresividad del lenguaje. Entre estás característica destacan la posibilidad de definir las claves en las

clases, cadenas de propiedades, tipos de datos y rangos de datos más complejos, restricciones de cardinalidad cualificadas, así como la definición de propiedades asimétricas, reflexivas y disjuntas. Otro cambio sobresaliente en esta versión consiste en la inclusión de tres perfiles para OWL DL, siendo cada uno de ellos más restrictivos que OWL DL. Estos perfiles son:

- **OWL 2 EL**. Este perfil es cercano a la lógica descriptiva EL++ (Baader, Brandt, and Lutz 2005), la cual permite realizar tareas básicas de razonamiento en tiempos polinómicos. Este perfil es adecuado para aplicaciones que requieren ontologías muy largas y donde se priorice un mejor rendimiento y no un nivel más alto de expresividad.
- **OWL 2 QL**. Este perfil está orientado a aplicaciones que utilizan ontologías relativamente ligeras, que requieran grandes volúmenes de instancias de datos y donde la consulta de información a través de consultas relacionales (SQL) es de gran utilidad, ya que permite responder a consultas de un modo formal y completo en un tiempo razonable, computacionalmente hablando.
- **OWL 2 RL**. Este perfil permite implementar algoritmos con tiempo de razonamiento polinomial a través de tecnologías de bases de datos basadas en reglas que operan directamente en tripletas RDF. OWL 2 RL está orientado a aplicaciones que requieran razonamiento escalable sin sacrificar demasiado el poder de expresividad (Motik et al. 2016). Este perfil es adecuado para aplicaciones que hagan uso de ontologías relativamente ligeras, que organicen un gran número de individuos y donde es de gran importancia manejar directamente datos en forma de tripletas RDF. Respecto al acrónimo RL, este viene dado por el hecho de que el razonamiento en este perfil puede implementarse a través de un lenguaje de reglas.

La Figura 2-4 representa de manera gráfica las relaciones existentes entre las dos versiones de OWL y los sub-lenguajes y perfiles descritos.

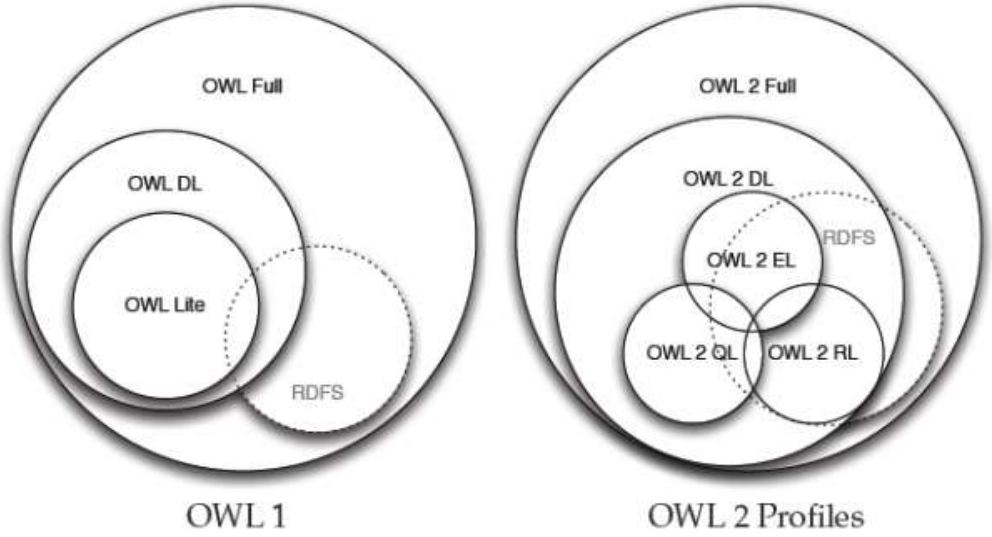

Figura 2-4. Representación gráfica de la relación entre los lenguajes y perfiles OWL

2.2.3.5 Lógica y razonamiento

De acuerdo con (Sowa 2000), la representación del conocimiento o KR (*Knowledge Representation*) es la aplicación de la lógica y las ontologías para poder construir modelos computables sobre algún dominio. KR aplica teorías y técnicas de tres campos: la lógica, las ontologías y la computación (García Moreno 2015). La lógica proporciona la estructura y reglas de inferencia formal a la representación del conocimiento. Las ontologías definen el tipo de cosas que existen en el dominio de la aplicación, haciendo que esta representación no sea confusa. La computación provee soporte para su implementación por programas de ordenador.

La lógica (o estudio del razonamiento correcto) es un elemento fundamental para que la representación del conocimiento no sea vaga (Shapiro 1995). El razonamiento permite inferir conocimiento representado implícitamente del conocimiento que está explícitamente contenido en la base de conocimiento (Baader and Nutt 2003).

La lógica descriptiva o DL (Description Logic) es una familia de lenguajes de representación del conocimiento basados en la lógica que puede ser utilizada para representar el conocimiento terminológico de un dominio de aplicación de manera estructurada (Baader, Horrocks, and Sattler 2008). Un sistema KR basado en DL provee facilidades para llevar a cabo la configuración de la base de conocimiento, razonar acerca de su contenido y manipularlo. En la Figura 2-5 se muestran los componentes principales

de un sistema KR basado en DL. Como se puede observar, la base de conocimiento está compuesta por el TBox, que introduce el vocabulario de un dominio de aplicación tales como clases, propiedades y restricciones, y el ABox, que contiene las declaraciones concernientes a los individuos en términos de su vocabulario (Baader and Nutt 2003).

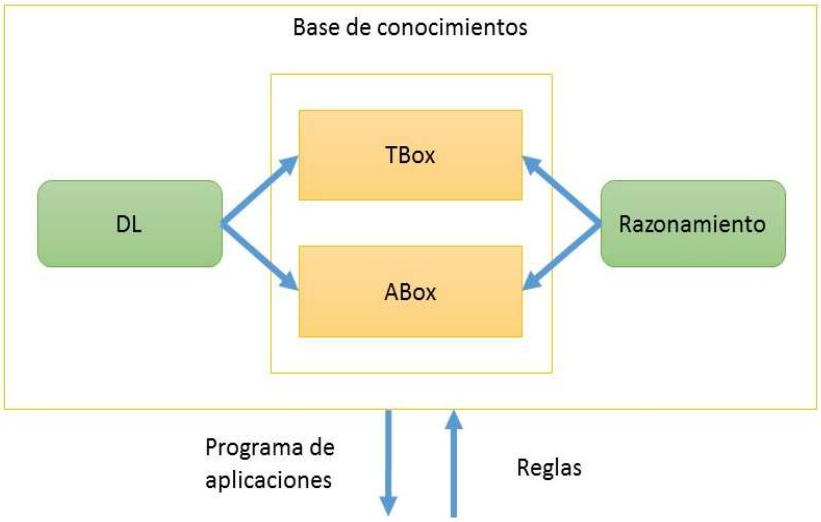

Figura 2-5. Arquitectura de un sistema de representación de conocimiento basado en lógica descriptiva.

El lenguaje OWL DL, que como recordaremos, proporciona la máxima expresividad a la vez que garantiza completitud computacional y razonamiento en un tiempo finito, representa una variante sintáctica de la lógica descriptiva {SHOIN(D)} (Horrocks 2005). Por esta razón, un razonador OWL debe proporcionar como mínimo el siguiente conjunto de servicios de inferencia de lógica descriptiva.

- **Chequeo de consistencia**. Este servicio se asegura de que una ontología no contiene ninguna definición contradictoria.
- **Satisfacibilidad de clase**. Esta comprueba si es posible que una clase tenga instancias. En caso de que una clase no pueda tener instancias, la definición de instancias para esa clase provoca que la ontología sea inconsistente.
- **Clasificación**. Este se encarga de computar las relaciones de subclase entre todas las clases con el objetivo de crear la jerarquía de clases completa.
- **Realización**. Este encuentra la clase más específica a la que un individuo pertenece. Este servicio solo se puede ejecutar después de haberse llevado a cabo el servicio de clasificación, ya que el tipo de un individuo se define respecto a la jerarquía de clases.

OWL-DL provee diversos axiomas que son importantes para el razonamiento. Estos axiomas se aplican desde nivel de clases hasta el de individuos. La Tabla 2-1 muestra un extracto de tales axiomas.

Tabla 2-1. Axiomas OWL DL

Sintaxis abstracta	Sintaxis DL
Clase	
EnumeratedClass(A o_1...o_n)	$A \equiv \{o_1,\ldots,o_n\}$
DisjointClasses(C_1...C_n)	$C_i \sqcap C_j = \bot$, $i \neq j$
EquivalentClasses(C_1...C_n)	$C_1 \equiv \ldots \equiv C_n$
SubClassOf(C_1 C_2)	$C_1 \sqsubseteq C_2$
Datatype(D)	
Property	
SubPropertyOf(U_1 U_2)	$U_1 \sqsubseteq U_2$
EquivalentProperties(U_1...U_2)	$U_1 \equiv \ldots \equiv U_n$
Annotation	
AnnotationProperty(S)	
Individuo	
SameIndividual(o_1...o_n)	$o_1 \equiv \ldots \equiv o_n$
DifferentIndividual(o_1...o_n)	$o_i \neq o_j$, $i \neq j$

De entre los axiomas presentados en la Tabla 2-1 podemos destacar, a nivel de clase, al axioma *subClassOf*, el cual se conoce como la condición necesaria, es decir, una condición que se debe cumplir para que un individuo pertenezca a una clase, pero no es suficiente por sí sola. Otros axiomas de interés son *DisjointClasses* y *EquivalentClasses*, los cuales permiten especificar que una colección de descripciones (clases) sean disjuntas entre ellas o tengan las mismas instancias, respectivamente. Esta misma condición puede aplicarse a individuos a través de los axiomas *DifferentIndividual* y *SameIndividual*.

2.2.3.5.1 Razonadores OWL DL

Un razonador es un programa que permite inferir consecuencias lógicas a partir de un conjunto de hechos explícitos o axiomas y, normalmente, provee soporte automatizado para tareas de razonamiento tales como la clasificación, depuración y consulta. Actualmente, existe una gran cantidad de razonadores que varían, entre otras cosas, en las capacidades expresivas que proporcionan, que van de menor expresividad (RDFS) a mayor expresividad (OWL Full).

Los razonadores OWL DL proveen capacidades de inferencia propias de las características expresivas de OWL DL, las cuales permiten la implementación de aplicaciones a partir de ellos. Entre estos razonadores destacan FaCT++ (Horrocks 1998),

RACER (Haarslev, Möller, and Turhan 2001), KAON2 (Motik and Studer 2005), Elephant (Sertkaya 2013), ELK (Kazakov, Krötzsch, and Simancik 2012) y Pellet (Sirin et al. 2007).

- **FaCT++**. Razonador basado en OWL DL y escrito en C++. Este permite comprobar la consistencia de una ontología, la satisfacción de un concepto o grupo de conceptos, deducciones de relaciones entre conceptos y la creación de taxonomías.
- **RACER**. RACER (*Renamed ABox And Concept Expression Reasoner*) provee servicios de inferencia optimizados, permitiendo desarrollar aplicaciones sofisticadas (García Moreno 2015). Este proporciona: razonamientos basados en reglas y restricciones, procesado de consultas expresivas y servicios de persistencia de datos basados en AllegroGraph (Aasman 2006).
- **KAON2**. Este es un marco de trabajo basado en Java para trabajar con ontologías basadas en OWL DL. Sin embargo, solo proporciona soporte parcial, ya que no soporta, por ejemplo, clases enumeradas. KAON2 provee una API enfocada al desarrollo de aplicaciones para la gestión de ontologías y un motor de inferencia que permite procesar consultas basadas en SPARQL.
- **Elephant**. Razonador basado en consecuencias que soporta parte del fragmento OWL 2 EL para la clasificación, consistencia y realización de tareas de razonamiento. Su objetivo es proporcionar un razonamiento ligero y de alto rendimiento para el perfil OWL 2 EL el cual, a pesar de limitar la expresividad, permite implementar algoritmos de razonamiento rápidos que pueden manipular grandes conjuntos de datos (Croset, Overington, and Rebholz-Schuhmann 2013).
- **ELK**. Este es un razonador especializado para el lenguaje OWL EL. ELK combina alto rendimiento y soporte integral para características del lenguaje. En su núcleo, este razonador emplea un motor de razonamiento basado en consecuencias que puede aprovechar los sistemas multi-núcleo y multi-procesador. La arquitectura modular de ELK permite que este pueda ser usado como aplicación autónoma, plugin de Protégé o biblioteca de programación.
- **Pellet**. Este razonador provee soporte para la expresividad total de OWL DL (incluidas clases enumeradas). Este proporciona servicios estándar de inferencia tales como: comprobación de consistencia, satisfacción de conceptos, clasificación y comprensión. Además, provee ciertas capacidades de OWL Full tales como soporte a propiedades funcionalmente inversas o al compartimiento de vocabulario entre individuos, clases o propiedades.

2.2.4 Lenguajes de consulta

2.2.4.1 SPARQL

SPARQL (*SPARQL Protocol and RDF Query Language*) (Prud'Hommeaux, Seaborne, and others 2008) es un lenguaje estandarizado para la consulta de grafos RDF. Este lenguaje permite obtener tripletas de datos de repositorios de datos RDF o repositorios que proporcionan vistas RDF.

Para expresar consultas SPARQL es necesario crear un conjunto de patrones de tripletas que formen un patrón de grafo básico. La consulta devuelve un subgrafo del grafo RDF consultado que es equivalente al patrón de grafo básico con las variables reemplazadas por términos RDF del subgrafo de los datos. Consideremos el ejemplo mostrado en la Figura 2-6.

```
PREFIX foaf:<http://xmlns.com/foaf/0.1/>
SELECT ?name ?mbox
WHERE {
  ?x foaf:name ?name .
  ?x foaf:mbox ?mbox .
}
```

Figura 2-6. Ejemplo de consulta SPARQL.

La primera línea define el prefijo del espacio de nombres (*namespace*). Un espacio de nombres es una recomendación W3C para proporcionar elementos y atributos con nombre único en un archivo XML. Una consulta SPARQL puede contener diversos espacios de nombres con el objetivo de incluir elementos procedentes de más de un vocabulario. El uso de estos prefijos permite resolver la ambigüedad existente entre elementos que tengan el mismo nombre (homonimia) en diferentes vocabularios. En la consulta anterior, se establece que los elementos con prefijo *foaf* hacen referencia al espacio de nombres especificado entre los símbolos menor que (<) y mayor que (>).

Las líneas cuatro y cinco son tripletas RDF que forman el patrón de grafo a ser obtenido. En estas tripletas el sujeto, predicado y objetos pueden ser variables, las cuales están identificadas con el símbolo *?* al inicio de ellas. En esta consulta, se busca el recurso *?x* que participa en tripletas RDF con los predicados *foaf:name* y *foaf:mbox*. De acuerdo con la especificación del vocabulario FOAF (Brickley and Miller 2012), *name* indica el nombre de una cosa, y *mbox* indica el buzón de Internet asociado con un propietario. La cláusula SELECT, al igual que en lenguajes de consulta como SQL, indica la lista de valores de variables a obtener.

Una consulta SPARQL además puede contener restricciones para valores a través de la cláusula FILTER. Por ejemplo, podemos restringir el valor de un recurso para que coincida con una cadena de texto a través de la siguiente expresión *FILTER regex(?mbox, "company")*. Con esto indicamos que el recurso identificado por la variable *?mbox* debe coincidir con la cadena de texto *company*. Otra restricción que podemos agregar es la correspondiente al valor de un número. Por ejemplo, podemos restringir que la consulta devuelva aquellos recursos cuyo precio sea menor a 20 a través de la siguiente expresión *FILTER (?price < 20)*. Además de las ya mencionadas restricciones, existen otras que permiten verificar si una variable es una URI (isIRI), o un literal (isLiteral). Una mayor descripción de estos y otras restricciones vienen definidas en la especificación del lenguaje SPARQL (Prud'Hommeaux, Seaborne, and others 2008).

En SPARQL además es posible modificar la secuencia del resultado utilizando las siguientes palabras clave:

- **ORDER BY**. Ordenar por el valor de una variable.
- **DISTINCT**. Resultados únicos solamente.
- **LIMIT**. Limita el número de resultados retornados por la consulta SPARQL.

Otra de las características importantes en SPARQL son las formas de resultados de consultas, es decir, además de la posibilidad de conseguir la lista de valores también es posible construir el grafo RDF o confirmar si algún resultado ha sido encontrado o no. Existen cuatro formas de resultados de consultas SPARQL.

- **SELECT**. Este devuelve la lista de valores de variables establecidas en el patrón de consulta.
- **CONSTRUCT**. Este devuelve un grafo RDF construido por variables de sustitución en el patrón de consulta.
- **DESCRIBE**. Devuelve un grafo RDF que describe los recursos que fueron encontrados.
- **ASK**. Devuelve un valor booleano que indica si el patrón de consulta coincide o no, es decir, ninguna información respecto a las variables que coinciden es devuelta.

SPARQL ha sido diseñado para su uso en los datos globales de la Web, por lo que permite hacer consultas sobre las fuentes de datos dispersas (Malik, Goel, and Maniktala 2010).

2.2.5 Linked data

Linked Data es un conjunto de buenas prácticas presentes en la Web Semántica para publicar y enlazar entre sí datos estructurados que se encuentran distribuidos en la Web. Este conjunto de buenas prácticas, conocidos como principios de Linked Data, fue propuesto por Tim Berners-Lee (Berners-Lee 2011):

1. **Usar URIs como nombres para las cosas**. Al nombrar los recursos mediante URIs, se ofrece una forma estándar y unívoca para referirnos a cualquier recurso.
2. **Usar URIs HTTP para que las personas pueden buscar esos nombres**. El uso de URIs sobre HTTP asegura que cualquier recurso pueda ser buscado y accedido en la Web.
3. **Cuando alguien busque una URI, proveer información útil a través de estándares tales como RDF o SPARQL**. Una vez que se accede al recurso identificado a través de una URI HTTP, se debe obtener información útil acerca de dicho recurso. Tal información se representa mediante descripciones estándares en RDF.
4. **Incluir enlaces a otros URIs para que desde un recurso se puedan descubrir otros**. Este principio es necesario para conectar los datos de tal forma que no se queden aislados y se puedan compartir con fuentes externas, así como permitir que otros sitios puedan enlazar datos propios.

Respecto al tercer principio de Linked Data, se debe enfatizar el hecho de llegar a un acuerdo sobre el formato estándar de este contenido, el cual permita a un amplio rango de aplicaciones procesar contenido Web. Así, cuando se publican datos siguiendo los principios de Linked Data, tales datos son representados utilizando RDF, el cual provee un modelo de datos simple y apropiado para la arquitectura Web. Los datos RDF publicados en la Web como Linked Data pueden ser serializados en diferentes formas, siendo los formatos más usados RDF/XML (Beckett and McBride 2004) y RDFa (Adida and Birbeck 2008).

De acuerdo con (Heath and Bizer 2011) el uso del modelo de datos RDF en Linked Data provee los siguientes beneficios:

1. Utiliza URIs HTTP como identificadores únicos globales para elementos de datos, así como para el vocabulario de términos. Por lo que RDF está inherentemente diseñado para ser utilizados a escala global y permite a cualquiera hacer referencia a cualquier recurso.

2. Los clientes pueden buscar cualquier URI en un grafo RDF en la Web para obtener información adicional. Así, cada tripleta RDF es parte de la Web global de datos y puede ser usada como punto de partida para explorar este espacio de datos.
3. El modelo de datos permite establecer enlaces RDF entre datos de diferentes fuentes.
4. La información proveniente de diferentes fuentes puede combinarse fácilmente uniendo los dos conjuntos de tripletas en un único grafo.
5. RDF permite representar información que esta expresada a través de un esquema diferente en un único grafo, lo que significa que es posible combinar términos de diferentes vocabularios para representar datos.
6. El modelo de datos RDF, en combinación con lenguajes de esquema tales como RDFS y OWL, permite estructurar los datos tanto como se quiera, lo que significa que los datos fuertemente estructurados, así como los semiestructurados pueden ser representados.

La recomendación RDF (Klyne, Carrol, and McBride 2016) especifica un conjunto de características que no han alcanzado una amplia adopción dentro de la comunidad Linked Data. Específicamente, las características que deben ser evitadas en el contexto de Linked Data son:

1. **RDF reificación**. La reificación o cosificación consiste en asignar una URI a una tripleta y utilizar esta como sujeto y objeto en otra tripleta. Esta práctica debe ser evitada ya que resulta más compleja de consultar mediante SPARQL. Como alternativa a esta característica esta la publicación de metadatos acerca de las declaraciones RDF individuales.
2. **Colecciones y contenedores RDF**. Estas características son problemáticas si los datos necesitan ser consultados mediante SPARQL. Entonces, en casos donde el orden relativo de los elementos de un conjunto no es significante, es recomendable el uso de múltiples tripletas con el mismo predicado.
3. **Nodos en blanco (*blank nodes*)**. El alcance de los nodos en blanco está limitado al documento en el que aparecen, lo que significa que no es posible crear enlaces RDF a ellos desde documentos externos, disminuyendo así el potencial de interconexión entre diferentes fuentes Linked Data.

2.2.5.1 La Web de datos

Un gran número de individuos y organizaciones han adoptado el enfoque Linked Data para publicar sus datos. El resultado de esto es un espacio global de datos llamado la Web de datos (Web of Data) (Bizer, Heath, and Berners-Lee 2009). Esta contiene billones de declaraciones RDF provenientes de múltiples fuentes y que cubren tópicos tales como localizaciones geográficas, gente, compañías, libros, publicaciones científicas, películas, música, programas de radio y televisión, genes, proteínas, medicamentos y ensayos clínicos, datos estadísticos, resultados de censos, comunidades en línea y comentarios.

El origen de la Web de datos recae en el proyecto Linking Open Data (LOD), cuyo propósito era inicializar la Web de datos a través de la identificación de los conjuntos de datos (*datasets*) disponibles bajo licencias abiertas, convertirlos a RDF de acuerdo con los principios de Linked Data y publicarlos en la Web (Heath and Bizer 2011). La Figura 2-7 muestra la cantidad de conjuntos de datos publicados en la Web siguiendo los principios de Linked Data actualizada hasta el 30 de agosto de 2014. En este diagrama, cada nodo representa un conjunto de datos distinto publicado como Linked Data. Los arcos indican la existentica de enlaces entre elementos de los conjuntos de datos. La dirección de las flechas indica el conjunto de datos que contiene los enlaces. Por ejemplo, una flecha de *A* a *B* indica que el conjunto de datos *A* contiene tripletas RDF que utilizan identificadores del conjunto de datos *B*. La flecha bidireccional indica que el fenómeno anterior ocurre en ambos sentidos, es decir, tanto de *A* a *B*, como de *B* a *A*. Los arcos más gruesos corresponden a un mayor número de enlaces.

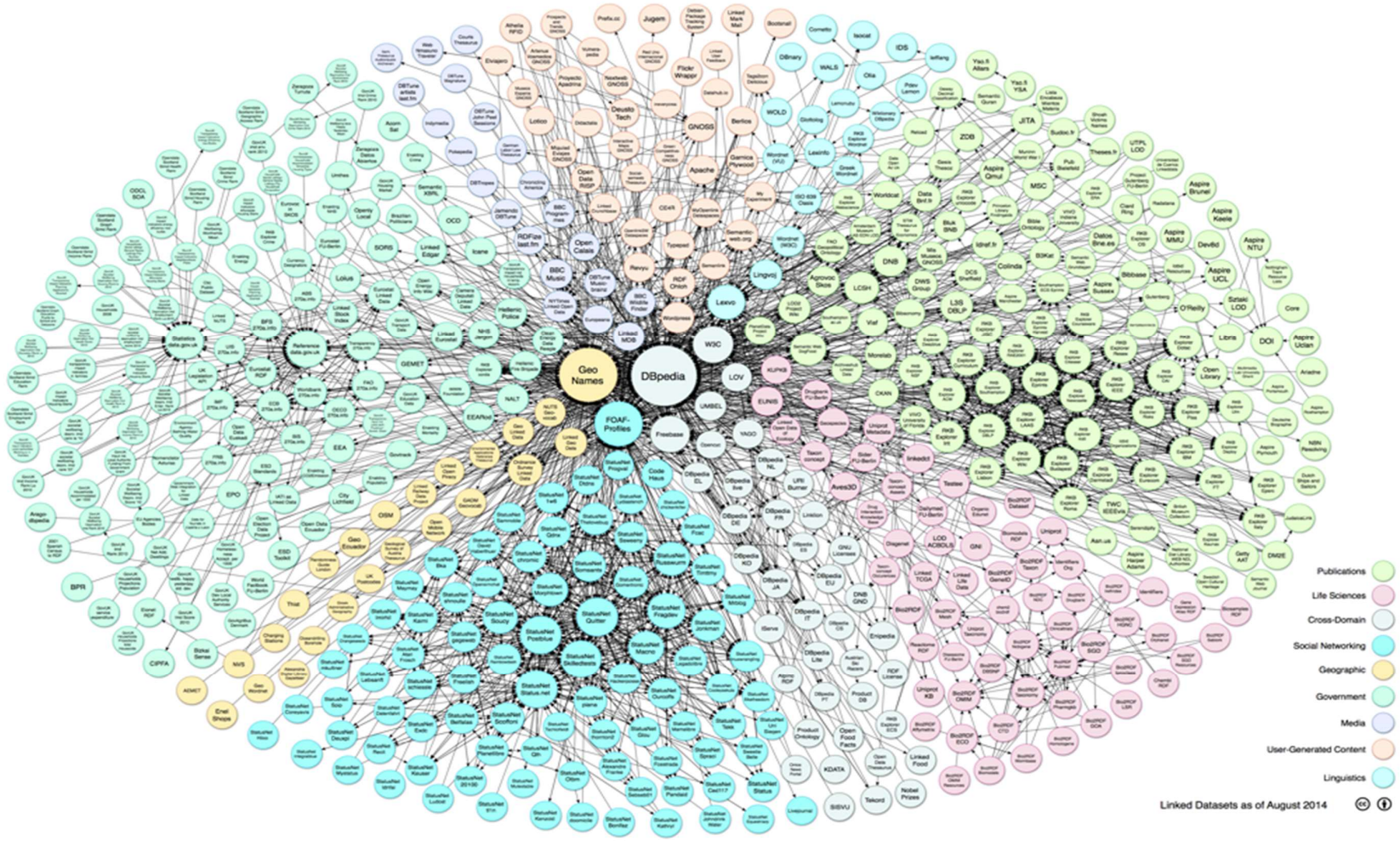

Figura 2-7. Nube de Linking Open Data de agosto 2014. [Recuperado de http://linkeddata.org/]

La Tabla 2-2 muestra una vista general de los 1014 conjuntos de datos por tópico disponibles en Linked Data al 30 de agosto de 2014 (Schmachtenberg, Bizer, and Paulheim 2014).

Tabla 2-2. Conjuntos de datos de Linking Open Data por tópico

Tópico	Conjuntos de datos	%
Gobierno	183	18.05
Publicaciones	96	9.47
Ciencias de la vida	83	8.19
Contenido generado por el usuario	48	4.73
Multidominio	41	4.04
Medios de comunicación	22	2.17
Geografía	21	2.07
Web social	520	51.28
Total de conjuntos de datos	1014	

En la Web de datos existen diversos conjuntos de datos que abarcan múltiples dominios. Este tipo de conjuntos de datos juegan un papel fundamental en la Web de datos, ya que ayudan a conectar conjuntos de datos con un dominio específico en un espacio de datos único e interconectado evitando la fragmentación en conjuntos de datos aislados.

2.3 Procesamiento de lenguaje natural

2.3.1 Definición

En lingüística, un lenguaje natural es cualquier lengua o idioma que ha sido generado en un grupo de hablantes con el propósito de comunicarse. Los lenguajes naturales pueden tomar diferentes formas tales como el habla, señas o la escritura. El PLN es un área de investigación de la Inteligencia Artificial (IA) que emplea un conjunto de tecnologías computacionales para analizar y generar de manera automática textos expresados en lenguaje natural. En la literatura existe un sinnúmero de definiciones de PLN. Por ejemplo, de acuerdo con (Sosa 1997), el concepto de PLN hace referencia a las "*técnicas de tratamiento de lenguaje y su aplicación en diversas áreas por medio de métodos computacionales*". (Liddy 2001) define al PLN como: "*un conjunto de técnicas computacionales teóricamente motivadas para analizar y representar naturalmente textos de origen natural en uno o más niveles de análisis lingüísticos para lograr el propósito de procesar el lenguaje humano por una serie de tareas o aplicaciones*". En (Chowdhury 2003) lo definen como: "*un área de investigación que explora cómo las computadoras pueden*

utilizarse para entender y manipular texto escrito en lenguaje natural o del habla para hacer operaciones útiles".

A pesar de las diferencias existentes entre las definiciones antes citadas, todas ellas cuentan con algo en común: el uso de técnicas o métodos computacionales que permitan analizar y entender el significado del lenguaje que utilizan los humanos para comunicarse en su vida cotidiana, y con ello facilitar la comunicación entre seres humanos y máquinas.

2.3.2 Antecedentes

Las primeras investigaciones en el campo de PLN datan de la década de 1940, donde su interés fundamental era la traducción automática. En 1947, Warren Weaver mencionó por primera vez la posibilidad de utilizar computadoras digitales para traducir documentos entre lenguajes humanos naturales. De ahí que se le considere como uno de los pioneros de la traducción automática. En esta misma década y hasta finales de los años 50 se llevó a cabo un trabajo intenso sobre dos paradigmas fundamentales: los autómatas y los modelos probabilísticos. Los autómatas dieron lugar a trabajos tales como el de (Shannon 2001) quien definió la teoría de autómatas y aplicó la teoría de la probabilidad de procesos de Markov para definir sistemas discretos, semejantes a los autómatas finitos, que procesaran el lenguaje humano. Basado en la idea de Shannon, (Chomsky 1956) consideró las máquinas de estado finito como un mecanismo para caracterizar una gramática, y definió un lenguaje de estado finito como un lenguaje generado por una gramática de estado finito. Estos modelos dieron paso a la teoría de lenguaje formal, la cual utiliza el álgebra y la teoría de conjuntos para definir lenguajes formales como secuencias de símbolos. En lo que concierne a los modelos probabilísticos, Shannon realizó contribuciones tales como el uso de la entropía como una manera de medir la capacidad de información de un canal, o el contenido de información de un canal, y llevó a cabo la primera medida de entropía del idioma inglés usando técnicas probabilísticas.

Durante el periodo de 1957-1970, el procesamiento del lenguaje y habla se dividió en dos paradigmas: simbólico y estocástico (Jurafsky and Martin 2014). El paradigma simbólico se originó a partir de dos líneas de investigación. La primera línea se basó en las obras de Chomsky y colegas, dando como resultado trabajos tales como el TDAP (Transformations and Discourse Analysis Project) (Joshi and Hopely 1996) uno de los primeros sistemas de análisis completos, el cual fue implementado en la Universidad de Pennsylvania. La segunda línea de investigación fue la IA cuyo principal foco de investigación es el estudio de algoritmos estadísticos y estocásticos que incluían modelos probabilísticos y redes neuronales. En esta etapa, se desarrollaron los primeros sistemas

de entendimiento del lenguaje natural, que trabajaban en dominios simples principalmente a través de una combinación de coincidencia de patrones y búsqueda por palabras clave con heurística simple y técnicas de pregunta-respuesta. El paradigma estocástico fue seguido por departamentos de estadística e ingeniería eléctrica. Bajo este paradigma, el método bayesiano comenzó a aplicarse a problemas de reconocimiento óptico de caracteres. Ejemplos de la aplicación de este método son el de (Bledsoe and Browning 1959) donde se describe un sistema bayesiano de reconocimiento de texto, y el de (Khamis, Mosteller, and Wallace 1966), donde se aplicaron métodos bayesianos para el problema de la atribución de autoría en artículos.

Durante la década de 1960 se desarrollaron los primeros modelos psicológicos de procesamiento de lenguaje humano basados en la gramática transformacional, así como la primera línea de corpus: el corpus Brown del inglés americano (Kucera et al. 1967), una colección de un millón de palabras obtenidas de 500 textos escritos de diferentes géneros tales como periódicos, novelas, académicos, entre otros.

Para el periodo de 1970 a 1983 se desarrollaron nuevos paradigmas de investigación: estocástico, basado en la lógica, entendimiento del lenguaje natural y modelado del discurso (Jurafsky and Martin 2014). El paradigma estocástico desempeñó un rol fundamental en el desarrollo de algoritmos de reconocimiento del habla, particularmente el uso del modelo oculto de Markov y los teoremas de codificación de canal ruidoso y decodificación. El paradigma basado en la lógica desarrollado por Colmerauer (Colmerauer 1975) representa el primer formalismo gramatical basado en las cláusulas de Horn. Con respecto al entendimiento del lenguaje natural, destaca el surgimiento del sistema SHRDLU (Winograd 1972), el cual simulaba un robot integrado en un mundo de bloques que aceptaba comandos de texto en lenguaje natural. El paradigma basado en la lógica y el de entendimiento del lenguaje natural fueron utilizados a la vez en sistemas que usaron la lógica de predicados como representación semántica, tal es el caso del sistema de pregunta-respuesta LUNAR (Woods 1979). El modelado del discurso se enfocó en áreas tales como la estructura del discurso y el foco del discurso (Grosz and others 1977), (Sidner 1986), y la resolución de referencia automática (Hobbs 1978).

Entre los años de 1983 y 1993, las investigaciones se centraron en el modelo de estados finitos y el retorno del empirismo. La primera de ellas recibió atención después de su inserción en el estudio de la fonología de estados finitos (Kaplan and Kay 1981) y la morfología de los modelos de estados finitos de la sintaxis de Church (Church 1980). En cuanto al retorno del empirismo, comenzaron a surgir métodos y enfoques probabilísticos de reconocimiento de voz y de procesamiento del lenguaje influenciados principalmente

por el trabajo desarrollado en el Centro de Investigación de la IBM Thomas J. Watson, el cual se centró en modelos probabilísticos de reconocimiento de voz.

En los últimos años, el campo del PLN ha cambiado inmensamente. En primer lugar, modelo probabilísticos y conducidos por los datos se han vuelto estándares a lo largo del PLN. Algoritmos de análisis, etiquetado gramatical, resolución de referencia y procesamiento del discurso integran ya modelos probabilísticos y emplean metodologías de evaluación en el reconocimiento de voz y recuperación de la información. En segundo lugar, el incremento del nivel de procesamiento y rapidez de los ordenadores, ha permitido la explotación de diferentes áreas del PLN, tales como el reconocimiento de voz y la detección de errores ortográficos y gramaticales. Finalmente, el auge de la Web ha hecho hincapié en la necesidad de innovación en los procesos de recuperación y extracción de información de los sistemas de almacenamiento (Jurafsky and Martin 2014).

2.3.3 Niveles de procesamiento de lenguaje natural

El PLN generalmente se divide en un número de etapas enfocadas en los tres aspectos o dimensiones que constituyen la teoría lingüística o desde un punto más general, la teoría semiótica, las cuales son la sintaxis, la semántica y la pragmática (Indurkhya and Damerau 2010). La sintaxis especifica las reglas de acuerdo a las cuales una expresión, como lo puede ser una oración, está *bien formada*. La semántica especifica las reglas según las cuales dicha expresión es *portadora de un significado*, es decir, que es interpretable en relación con alguna situación posible. Por último, la pragmática está orientada a la formulación de las reglas según las cuales un acto verbal es apropiado en relación con un contexto (Dijk and Mayoral 1987). La pragmática a menudo se le relaciona con el discurso, es decir, con el conjunto de enunciados con que se expresa un pensamiento, razonamiento, sentimiento o deseo. Mientras tanto, la sintaxis y la semántica generalmente se les relaciona con las cuestiones oracionales.

En (Indurkhya and Damerau 2010), los autores establecen un conjunto de cinco etapas de análisis en las cuales se descompone el PLN (ver Figura 2-8) siendo la entrada de este proceso un texto, y la salida el significado deseado del hablante.

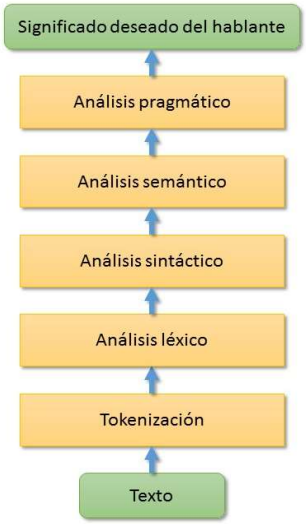

Figura 2-8. Etapas de análisis en el procesamiento de lenguaje natural.

De acuerdo con (Liddy 2001) y (Feldman 1999) el mejor método para entender qué es lo que realmente pasa dentro de un sistema de PLN es a través del enfoque de niveles del lenguaje natural. La Tabla 2-3 muestra los siete niveles del lenguaje que utilizan los humanos para extraer el significado del lenguaje natural. Además, se muestra su ámbito de actuación, las herramientas de procesamientos que podrían ser utilizadas para llevar a cabo el procesamiento de manera automática y las estructuras lingüísticas obtenidas en cada nivel.

Tabla 2-3. Niveles del lenguaje natural y herramientas PLN.

Nivel de PLN	Ámbito de actuación	Herramientas utilizadas	Resultados del procesamiento
Fonético	Sonidos	Corpus de aprendizaje Modelos acústicos Diccionarios de unidades de síntesis	Fonemas
Léxico	Formas	Lematizador Etiquetado POS Lexicón computacional	Palabras
Morfológico	Formas	Lematizador Etiquetado POS Lexicón computacional	Palabras
Sintáctico	Estructuras	Bases de datos sintácticas Treebank	Frases
Semántico	Significados	Bases de datos semánticas Lexicón computacional Ontologías	Relaciones
Discurso	Comunicación	Bases de datos semánticas Ontologías	Textos
Pragmático	Comunicación	Bases de datos semánticas Ontologías	Textos

Como se mencionó previamente, los humanos extraen el significado del lenguaje oral o escrito en al menos los siete niveles presentados en la Tabla 2-3. Para entender el PLN, es importante distinguir entre cada uno de ellos, ya que no todos los sistemas de PLN usan cada nivel (Feldman 1999). Sin embargo, un sistema de PLN será más capaz a nivel que ocupe más niveles del lenguaje (Liddy 2001).

Es de suma importancia resaltar el hecho de que las fases o niveles de procesamiento del lenguaje no son niveles aislados, sino que se encuentran interrelacionados. Es decir, existen niveles que, para llevar a cabo su proceso de análisis, demandan el conocimiento de niveles anteriores, o incluso posteriores. En los siguientes apartados se describen cada uno de los niveles del lenguaje según (Liddy 2001) y (Feldman 1999).

2.3.3.1 Nivel fonológico

El nivel fonológico se ocupa de la interpretación de los sonidos del habla dentro y a través de las palabras. En el análisis fonológico se utilizan tres tipos de reglas: 1) reglas fonéticas, enfocadas al sonido dentro de las palabras; 2) reglas fonológicas, encargadas de las variaciones existentes en la pronunciación cuando las palabras son habladas juntas; y 3) reglas prosódicas, las cuales se enfocan en la fluctuación del acento y la entonación de los humanos cuando pronuncian una oración. Este nivel es de crucial importancia para la comprensión del lenguaje hablado y en sistemas de reconocimiento de voz. Cuando un sistema de PLN acepta como entrada la voz humana, las ondas sonoras son analizadas y codificadas en una señal digital que permita su interpretación a través de diversas reglas o mediante la comparación con el modelo de lenguaje particular que se esté utilizando.

Algunas de las herramientas utilizadas en este nivel son los corpus de aprendizaje, los modelos acústicos y los diccionarios de unidades de síntesis. El primero de ellos está diseñado para recolectar un amplio conjunto de muestras que permitan obtener un amplio margen de variabilidad fonética en las realizaciones alofónicas, es decir, en cada uno de los sonidos que en un idioma dado se reconoce como un determinado fonema. Los modelos acústicos representan la pronunciación en un formato legible por una máquina. Este modelo usa el hecho de que las palabras habladas están compuestas de sonidos al igual que las palabras escritas están compuestas de letras. Por último, los diccionarios de unidades de síntesis se refieren a un inventario completo de unidades acústicas que se utilizan en una lengua determinada, ejemplo de unidades de síntesis son las frases, palabras, difonemas, trifonemas, entre otros.

2.3.3.2 Nivel morfológico

Este nivel se ocupa de la naturaleza composicional de las palabras, las cuales están compuestas de morfemas. Un morfema hace referencia a la unidad más pequeña de la lengua que tiene significado léxico o gramatical, y que unido a un lexema modifica su definición. De esta manera, los sistemas de procesamiento morfológicos transforman cada secuencia de caracteres en una secuencia de morfemas, todo ello a través de técnicas tales como *stemming*, lematización, o etiquetadores POS (*Part-Of-Speech*).

La técnica de *stemming* permite reducir una palabra a su raíz o stem. Esta técnica consiste en un proceso de extracción y sustitución de sufijos de palabras para llegar a una forma común de la raíz de la palabra. Un ejemplo de *stem* es *proces* para las palabras *procesamiento* y *procesar*. La técnica de lematización permite obtener el lema de una palabra, es decir, la forma que por convenio se acepta como representante de todas las formas flexionadas de una misma palabra. En otras palabras, el lema de una palabra es la palabra que se encuentra como entrada en un diccionario tradicional. Por ejemplo, la palabra *decir* es el lema de las palabras *dije*, *diré* y *dijéramos*. Finalmente, un etiquetador POS permite asignar a cada una de las palabras de un texto su categoría gramatical, tal como sustantivo, pronombre, adjetivo, verbo, adverbio, entre otras.

En resumen, con estás técnicas es posible determinar aspectos como tiempo, género, número, grado, etc., así como clasificar las unidades lingüísticas en las categorías gramaticales antes mencionadas.

2.3.3.3 Nivel Léxico

En este nivel, tanto humanos como sistemas PLN, interpretan el significado de las palabras de manera individual. En este nivel existen varios tipos de procesamiento que contribuyen a la comprensión a nivel de palabra. Una de las técnicas más populares se basa en la asignación de etiquetas individuales a cada palabra contenida en el texto. En este proceso de etiquetado, existen palabras que pueden tener más de una etiqueta. Sin embargo, se selecciona aquella cuya probabilidad sea mayor de acuerdo al contexto en que ocurre la palabra.

A nivel léxico puede ser necesario el uso de un lexicón que puede variar su naturaleza y extensión de la información codificada dependiendo del enfoque particular tomado por el sistema PLN. Un lexicón puede ser simple, con solo las palabras y sus correspondientes etiquetas o puede ser cada vez más complejo, al incluir información sobre la clase semántica de la palabra, sus argumentos, las limitaciones semánticas de esos argumentos.

La definición del sentido en la representación semántica utilizada en el sistema en particular e incluso el contexto semántico en el que se usa cada sentido de una palabra polisémica.

2.3.3.4 Nivel sintáctico

La sintaxis es el estudio de las relaciones formales entre palabras (Jurafsky and Martin 2014), es decir, se encarga de estudiar cómo las palabras son agrupadas en clases llamadas *parts-of-speech* (tales como sustantivos, verbos, adjetivos, preposiciones, adverbios, conjunciones, entre otros), cómo ellas se agrupan con su palabras vecinas formando frases, y la manera en que las palabras dependen de otras palabras contenidas en la oración. El resultado de este nivel de procesamiento es una representación de la estructura de la oración que pone de manifiesto las relaciones de dependencia estructural de las palabras. Dicha estructura se puede representar de diferentes formas, dependiendo del enfoque lingüístico adoptado. Sin embargo, existen dos formas populares que son: el análisis de constituyentes y el análisis de dependencias.

En el análisis de constituyentes, la oración es dividida en partes denominadas constituyente sintáctico, el cual puede ser una palabra o secuencia de palabras, que funcionan en conjunto como una unidad dentro de la estructura jerárquica de una oración. Cada constituyente es a su vez dividido en partes más pequeñas hasta llegar al nivel de palabra. Por ejemplo, podemos tener un conjunto de palabras denominado sintagma nominal que actúa como una unidad, este sintagma nominal puede contener una única palabra tal como *ella* o *María*, o frases tales como *la casa* o *la Universidad de Murcia*. Antes de llevar a cabo este análisis, es necesario llevar a cabo el análisis léxico y morfológico con el objetivo de obtener información individual de cada una de las palabras contenidas en la oración, la cual ayudará a determinar cómo se agrupan o relacionan las palabras en el análisis sintáctico. La Figura 2-9 muestra el análisis de constituyentes de la frase *Enrique estudia en la Universidad de Murcia*, el cual se representa a través de árboles sintácticos de gramáticas libres de contexto (*context-free grammars*) (Chomsky 1956). Es importante mencionar que, para propósitos de este ejemplo, el análisis sintáctico se llevó a cabo mediante Freeling (Padró and Stanilovsky 2012), una suite de herramientas de análisis de lenguaje de código abierto.

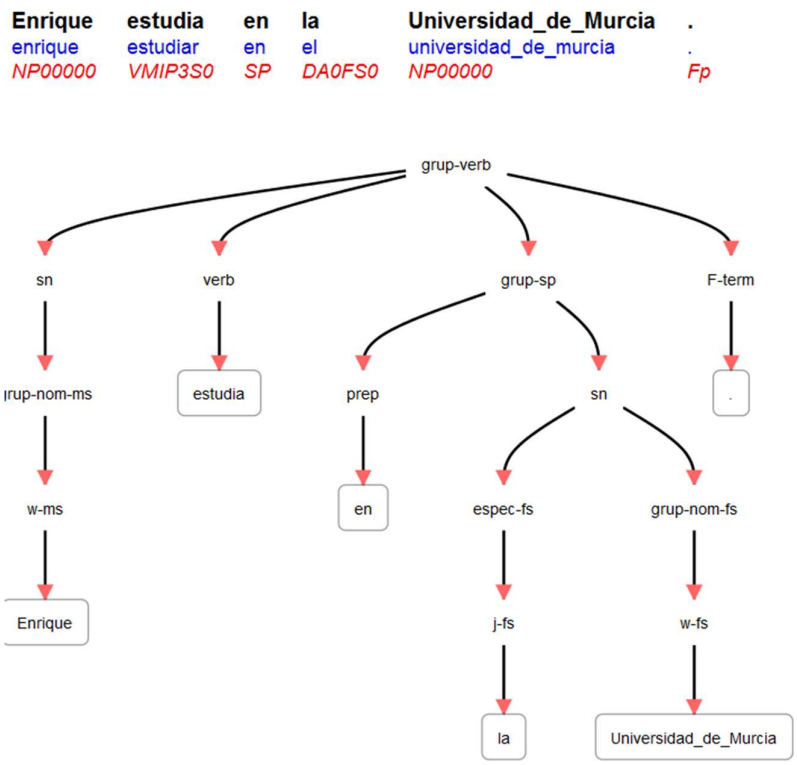

Figura 2-9. Análisis de constituyentes sintácticos.

La primera parte de la Figura 2-9 muestra el análisis morfológico de la oración, en la cual se aprecia que los elementos *Enrique* y *Universidad de Murcia* son nombres propios (NP) y la palabra *estudia* es el verbo principal indicativo (VMI). La segunda parte de la imagen muestra la representación del análisis de constituyentes sintácticos, en la cual se aprecian todas las relaciones existentes entre las palabras contenidas en la oración. En este análisis, destacan el nodo raíz (*grup-verb*) el cual indica un fragmento verbal, es decir, un constituyente que tiene como núcleo un verbo. En la parte izquierda se encuentra el sintagma nominal (sn), es decir, un constituyente cuyo núcleo es un nombre o sustantivo. En la parte derecha, se muestra otro sintagma nominal (sn) correspondiente a la *Universidad de Murcia*.

El análisis de dependencias se encarga de determinar las relaciones gramaticales o funciones sintácticas que existen entre las palabras de la oración. Ejemplos de este tipo de funciones son sujeto, objeto directo, objeto indirecto, complementos, entre otros. Dicho esto, el análisis de dependencias para el ejemplo anterior se muestra en la Figura 2-10.

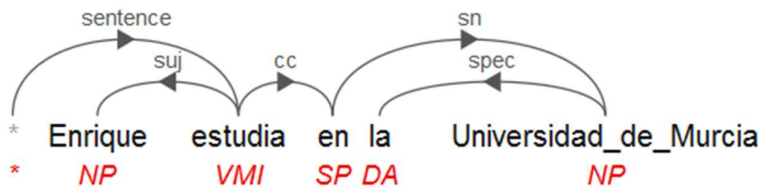

Figura 2-10. Análisis sintáctico de dependencias.

En la Figura 2-10 podemos observar que el verbo principal indicativo (VMI). En este caso la palabra *estudia*, tiene como sujeto (suj) al elemento *Enrique*. También se aprecia que el elemento *en* es el adyuvante del verbo principal. El adyuvante especifica generalmente el tiempo, lugar, modo, etc., del verbo. Finalmente, tenemos al elemento *Universidad de Murcia* como el sintagma nominal (sn) del adyuvante *en*. Con todas estas relaciones, es posible determinar que quien estudia es Enrique y que lo hace en un lugar específico, en este caso, en la Universidad de Murcia.

2.3.3.5 Análisis semántico

La semántica es el estudio del significado de las expresiones lingüísticas (Jurafsky and Martin 2014), tales como morfemas, palabras, frases y oraciones. Este significado puede ser capturado mediante estructuras formales las cuales requieren el uso de un amplio rango de fuentes de conocimiento y técnicas de inferencia. Entre las fuentes de conocimiento que son usadas regularmente están los significados de las palabras, los significados asociados con estructuras gramaticales, conocimiento acerca de la estructura del discurso, conocimiento acerca del contexto en el cual el discurso está ocurriendo y conocimiento de sentido común acerca del tema en cuestión.

En el análisis semántico del lenguaje se pueden distinguir la semántica léxica, la cual estudia el significado de las palabras y las relaciones entre ellas, y la semántica oracional, la cual estudia el significado de unidades sintácticas más largas que una palabra.

2.3.3.5.1 Semántica léxica.

La semántica léxica está basada en el principio de *composicionalidad*, según el cual el significada de una oración se puede componer a partir del significado de sus partes. Este enfoque asigna significado a las entradas con base únicamente en el conocimiento estático del léxico y la gramática, es decir, este significado es independiente del contexto y libre de inferencia, característica que lo hace bastante limitado en su alcance.

Uno de los problemas existentes en la semántica léxica es la ambigüedad a que da lugar la polisemia y la homonimia. La primera de ellas hace referencia a las palabras que tienen

varios significados, como por ejemplo la palabra *gato*, la cual puede hacer referencia a un animal de la familia de los felinos, o a una herramienta para levantar objetos pesados. La homonimia es la cualidad de dos palabras, de distinto origen y significado, que tienen la misma categoría y forma gráfica. Es posible distinguir dos tipos de homónimos, las palabras homógrafas, que coinciden en su escritura, aunque no necesariamente en pronunciación, y las palabras homófonas, que coinciden en pronunciación, aunque no necesariamente en su escritura.

Para llevar a cabo la asociación de los sentidos correspondientes a cada una de las palabras de un texto, existe la desambiguación semántica automática (WSD, *Word Sense Disambiguation*), la cual puede ser llevada a cabo bajo dos grandes enfoques generales, los cuales no son totalmente excluyentes: el enfoque cualitativo y el cuantitativo (López 2001).

- **Cualitativo**. Este enfoque se basa en reglas para seleccionar el sentido asociado con cada palabra. De esta manera, ante una entrada de una palabra con varias acepciones, se selecciona de manera determinista el significado para el cual las reglas se satisfacen. Bajo este enfoque las reglas específicas fallan frecuentemente en la selección ante entradas excepcionales, por otro, las reglas genéricas corren el riesgo de seleccionar sentidos incorrectos.
- **Cuantitativo**. Este enfoque computa valores escalables para cada uno de los sentidos candidatos de una palabra y selecciona como sentido el de valor máximo. Este enfoque emplea grandes bases de conocimiento y corpus de textos. Este enfoque es más robusto ante entradas excepcionales.

Existen distintos enfoques de desambiguación atendiendo al tipo de fuentes de conocimiento léxico utilizados.

- **Fuentes de información estructurada**. Conocido también como desambiguación basada en fuentes de conocimiento, entre las que se encuentran los diccionarios o lexicones, tesauros, ontologías o redes semánticas y lexicones generativos. Este tipo de sistemas de desambiguación semántica asignan los sentidos a las palabras a partir de reglas heurísticas que relacionen la palabra que se quiere desambiguar con sus descripciones, definiciones o ejemplos contenidos en la fuente de conocimiento. Por ejemplo, si la fuente de conocimiento es un diccionario, se pueden emplear métodos como: (a) sentido más frecuente, que consiste en asociar a las palabras polisémicas el primer sentido de su entra en el diccionario, o (b) definición, donde las definiciones de los diccionarios se usan como bolsas de palabras que caracterizan un sentido.

Cuando la fuente de conocimientos es una red semántica como WordNet (Miller 1995), se utilizarán medidas de similitud tales como: (a) extensión del camino, donde se mide la longitud del camino *p* más corto entre los posibles sentidos de las dos palabras a través de la relación *es-un*, o (b) distancia semántica (Sussna 1993), donde se asignan pesos a los enlaces (links) de WordNet en base al tipo de relación (sinonimia, homonimia, etc.) y se define una métrica que tiene en cuenta el número de arcos del mismo tipo que parten de un nodo y la profundidad de la sección o límite (*edge*) en el árbol general.

- **Fuentes de información no estructurada**. También conocido como desambiguación basada en corpus. En este enfoque es necesario un conjunto amplio y sustancial del uso de esas palabras, el cual se denomina corpus. Gracias a los avances tecnológicos de hoy en día, existen corpus cada vez más grandes, lo que ha dado paso al desarrollo de métodos de desambiguación estadísticos supervisados y no supervisados.
    - **Método supervisado**. En este método se conoce el estado real (en este caso, la etiqueta del sentido de la palabra) para cada conjunto de datos sobre los que se hace el entrenamiento, es decir, se cuenta con un corpus desambiguado (corpus anotado semánticamente). Este método se puede ver a menudo como una tarea de clasificación (Manning and Schütze 1999). A partir de los ejemplos contenidos en el corpus, se determinan los valores de los atributos relevantes para cada categoría, lo que equivale a establecer un modelo de las categorías y a partir de esta información el sistema debe producir un procedimiento para la correcta categorización de futuros ejemplos.
    - **Método no supervisado**. En este no se conoce la clasificación de los datos de la muestra del entrenamiento, es decir, el algoritmo de desambiguación no está provisto de un conjunto de sentidos a priori al análisis del corpus, sino que éstos se infieren del texto a posteriori. En consecuencia, este tipo de aprendizaje se puede considerar a menudo como una tarea de agrupación (*clustering*).

A pesar de que existe cierta variabilidad en la literatura respecto a la clasificación antes presentada. Generalmente se suele hablar de desambiguación semántica supervisada si se entrena con un corpus etiquetado con sentidos y de desambiguación semántica no supervisada si el corpus de entrenamiento tiene otro tipo de anotación o no tiene ninguna (Resnik and Yarowsky 1999).

2.3.3.5.2 Semántica oracional.

Como ya se mencionó, la semántica léxica está basada en el principio de *composicionalidad.* Sin embargo, análisis más profundos concluyen que el significado de una oración no se basa solamente en las palabras que lo forman, sino también en el orden, agrupación y relaciones entre palabras de la oración (Jurafsky and Martin 2014).

Uno de los mayores problemas a los que se enfrentan los analizadores semánticos es el hecho de que patrones sintácticos similares pueden corresponder distintas interpretaciones semánticas y significados similares pueden ser realizados sintácticamente en muchas formas diferentes (Jurafsky and Martin 2014). Para lidiar con este problema, es necesario establecer relaciones entre la sintaxis y la semántica (Shi and Mihalcea 2005). Para tratar con este hecho, la asignación de roles semánticos a los diferentes argumentos verbales de una oración es una tarea clave.

La semántica oracional consiste en identificar las relaciones semánticas (rol temático o semántico) que existen entre los predicados (principalmente el verbal) y sus complementos. Lo antes dicho implica el establecimiento, para cada predicado, de la correspondencia entre constituyentes sintácticos y roles temáticos. Un rol temático es el término utilizado para describir el papel desempeñado por una entidad concreta en un evento (O'Grady, Dobrovolsky, and Aronoff 1993). Algunos ejemplos de roles temáticos son:

- **Agente**. La entidad que lleva a cabo deliberadamente una acción.
- **Tema**. La entidad que se someterá a un cambio de estado o transferencia.
- **Fuente**. El punto de partida de una transferencia.
- **Objetivo**. El punto final de una transferencia.
- **Experimentador**. La entidad que percibe algo.
- **Ubicación**. El lugar en el que se encuentra una entidad o acción.
- **Estímulo**. La entidad percibida.
- **Instrumento**. La entidad usada para llevar a cabo una acción.
- **Actor**. La entidad que produce algún acontecimiento o estado.
- **Medida**. La extensión de una dimensión (tamaño, tiempo, precio).

El proceso a través del cual se determina el papel que los argumentos de los verbos juegan en una oración, recibe el nombre de anotación de roles semánticos (SRL - *Semantic Role Labeling*). El objetivo en SRL es identificar para cada uno de los verbos de una oración, todos los constituyentes que juegan algún papel semántico determinando el rol concreto de cada uno de ellos respecto al verbo (Moreda 2009). En la Figura 2-11

podemos observar como a una misma estructura sintáctica *sujeto* pueden corresponder distintos roles temáticos.

Juan lee el periódico.
Juan llegó a las siete.

Figura 2-11. Ejemplo de semántica de la oración.

Como podemos observar, *Juan* es el sujeto sintáctico en todas las oraciones. Este hecho nos dice muy poco del significado de la oración ya que se ve claramente una gran diferencia en la relación de dicho sujeto con los predicados verbales. Así, en la primera oración *Juan* es el *agente* de leer, mientras que en la segunda es el tema de *llegar*.

En (Moreda 2009) concluyen que en una oración cada rol semántico es asignado a un único constituyente y cada constituyente juega un rol único. Por tanto, aunque cambie el orden de los constituyentes o incluso la voz o el tiempo verbal de la oración, los roles semánticos de los argumentos se mantienen. Todo ello hace de SRL una tarea clave para sistemas de PLN, tales como los sistemas de búsqueda de respuestas dado que con la definición de roles se puede responder a preguntas tales como quién, cuándo, dónde, etc. Por ejemplo, consideremos la primera frase mostrada anteriormente, donde el rol agente *Juan*, respondería a la pregunta ¿Quién lee el periódico?

2.3.3.6 Nivel del discurso o contextual

El nivel del discurso de PLN trabaja con unidades de texto más largas que una sentencia. Este nivel no interpreta las sentencias de un texto como solo sentencias concatenadas, sino que se enfoca en las propiedades del texto como un todo que transmiten un significado a través del establecimiento de conexiones entre las sentencias que lo componen. Existen diversos tipos de procesamiento del discurso, siendo la resolución anafórica y el reconocimiento de la estructura del texto/discurso los más conocidos.

- **Resolución anafórica**. Este se basa en remplazar palabras tales como pronombres, las cuales son semánticamente vacías, con la entidad apropiada a la que ellos hacen referencia.
- **Reconocimiento de la estructura del texto/discurso**. Este determina las funciones de las sentencias contenidas en el texto, las cuales, a su vez agregan una representación significativa al texto. De esta manera, artículos de periódicos pueden ser descompuestos en componentes del discurso tales como: historia principal, eventos anteriores, evaluación, citas, entre otros.

2.3.3.7 Nivel pragmático

Los significados comunicados a través del lenguaje son de dos tipos: el significado convencional y el significado intencional. El primero de ellos es estudiado en la semántica, mientras que el segundo corresponde a la pragmática. El análisis pragmático se ocupa del estudio de lo comunicado por el escritor y lo que es interpretado por el lector. Este análisis se enfoca en el modo en que el contexto influye en la interpretación del significado. Por ello, utiliza información del contexto por encima de los contenidos para comprender el significado del texto. El análisis realizado en este nivel se encuentra relacionado con los factores extralingüísticos que condicionan el uso del lenguaje en situaciones comunicativas concretas, es decir, en todos aquellos factores que no pueden ser analizados por los niveles anteriores, tales como la intención comunicativa, el contexto verbal y la situación o conocimiento del mundo. Para llevar a cabo este análisis, los sistemas de procesamiento pragmático disponen de bases de conocimiento y módulos de inferencia que permiten interpretar las intenciones, los planes y los objetivos de un texto.

2.3.4 Aplicaciones del procesamiento de lenguaje natural

El campo del PLN puede ser aplicado en aplicaciones tales como recuperación y extracción de información, traducción automática, minería de datos, generación de resúmenes, análisis de sentimientos y sistemas de búsqueda de respuestas, entre otras. A continuación, se explican estas aplicaciones. Sin embargo, para propósitos del presente trabajo de tesis, el concepto de sistemas de búsquedas de respuestas será descrito con mayor detalle en la sección 2.4.

2.3.4.1 Recuperación de información

La recuperación de información (RI) es el proceso de encontrar material (usualmente documentos) de naturaleza no estructurada (usualmente texto) que satisfaga una necesidad de información, dentro de grandes colecciones (usualmente almacenada en computadoras) (Manning, Raghavan, and Schütze 2008).

La RI puede también cubrir otros tipos de datos y problemas de información diferentes a los especificados en la definición anterior. El término *información no estructurada* es un tipo de información que no tiene un modelo de datos predefinido (Chen, Chiang, and Storey 2012). La información estructurada son los datos que están perfectamente definidos y sujetos a un formato concreto, el principal ejemplo de este tipo de datos son las bases de datos relacionales. Además, existe la información semiestructurada que, a pesar

de no residir en una base de datos relacional, presentan una organización interna que facilita su tratamiento, tales como documentos XML o HTML.

El campo de RI también provee soporte a actividades tales como la navegación o filtrado de una colección de documentos o el procesamiento de un conjunto de documentos recuperados. Otra actividad es el agrupamiento (*clustering*) de documentos basado en su contenido, la clasificación de los documentos de acuerdo a un conjunto de temas, necesidades de información u otras categorías.

Los sistemas de RI pueden diferenciarse por la escala de datos sobre la cual operan: a) sistemas de búsqueda web; b) sistemas RI personales y c) sistemas empresariales, institucionales y de dominio específico. El nivel más alto corresponde a la búsqueda Web, en la cual el sistema tiene que llevar a cabo la búsqueda sobre millones de documentos almacenados en millones de ordenadores; en el otro extremo de los niveles, se encuentran los sistemas RI personales, cuyos ejemplos claros son los incluidos en sistemas operativos como Mac OS o Windows que permiten la búsqueda y manejo de una amplia gama de tipos de documentos; en un nivel intermedio se encuentran los sistemas de RI de dominio específico, donde la recuperación puede ser llevada a cabo para colecciones de documentos internos de una organización, los cuales típicamente están almacenados en un sistema de archivos centralizados y una o varias máquinas dedicadas que permiten llevar a cabo búsquedas sobre dicha información.

2.3.4.2 Extracción de información

La extracción de información (EI) se define como una tecnología basada en el análisis del lenguaje natural para extraer fragmentos de información. El proceso toma como entrada textos y produce un formato fijo de datos inequívocos como salida (Cunningham 2005). En otras palabras, este proceso consiste en obtener las partes que interesan en el texto para pasarlas a un formato estructurado (Hernández and Gómez 2013). Para ello, el proceso extrae fragmentos de texto con significado relevante ignorando los fragmentos irrelevantes que se emplean para estructurarlos. De esta manera, el ordenador es capaz de entender y almacenar la informaci'on extraida en un sistema de almacenamiento, como una base de datos, para su futura explotación (Cowie and Lehnert 1996).

De acuerdo con (Cowie and Lehnert 1996) un sistema de EI está compuesto por los siguientes elementos básicos que funcionan a diferentes niveles del texto:

- **Filtrado**. Este elemento funciona a nivel de texto y determina la relevancia del texto o partes del texto basado en estadísticas de la palabra o la ocurrencia de patrones particulares.

- **Etiquetador POS**. Este elemento funciona a nivel de palabra y es el encargado de asignar o etiquetar a cada una de las palabras de un texto su correspondiente categoría gramatical.
- **Etiquetador semántico**. Este funciona a nivel de sintagma nominal, es decir, sintagma que tiene como núcleo o elemento principal un nombre o sustantivo. El término *nombre de entidad* es ampliamente utilizado en este contexto y fue acuñado en la sexta edición de las conferencias MUC (*Message Understanding Conference*) (Grishman and Sundheim 1996), en la cual fueron reconocidos los tipos de entidades más estudiadas, como los nombres propios, lugares y organizaciones. El concepto Reconocimiento de Nombres de Entidades (RNE o NER, *Named Entity Recognition*) se define como la tarea que se encarga de clasificar cada palabra de un documento en un conjunto de categorías predefinidas (Zhou and Su 2002).
- **Análisis sintáctico**. Este elemento funciona a nivel de sentencia y se encarga del estudio de las relaciones formales entre palabras, es decir, cómo se agrupan en clases (sustantivos, verbos, adjetivos, etc.), cómo estas se agrupan con sus palabras vecinas y la manera en que las palabras dependen de otras palabras contenidas en la oración.
- **Discurso**. Este funciona a nivel inter-sentencia. Su misión se centra en solapar y fusionar las estructuras producidas por el analizador. Además, reconoce y unifica las expresiones referenciales.
- **Generación de salida**. Este elemento funciona a nivel de plantilla y le da formato a la salida de acuerdo al formato predefinido de esta.

2.3.4.3 Traducción automática

La traducción automática (TA) o MT (*Machine Translation*), es un área de la lingüística computacional que investiga el uso de software para traducir texto o habla de un lenguaje a otro. De acuerdo con (Tripathi and Sarkhel 2010) el proceso general de traducción tiene dos niveles:

- **Traducción literal**. La traducción literal significa traducir *palabra por palabra*. En este nivel, el texto traducido puede no transmitir el significado del texto original, lo que significa que en ocasiones la semántica puede diferir del texto original.
- **Parafraseo**. Este nivel hace referencia a la equivalencia dinámica, es decir, el texto traducido contiene la esencia del texto original pero no necesariamente contiene una traducción *palabra por palabra*.

Algunos métodos para traducción automática son:

- **Traducción basada en diccionarios**. Este método se basa en las entradas de un diccionario, de esta forma, la equivalencia de una palabra es utilizada para desarrollar la versión traducida.
- **Traducción basada en reglas**. En este método, además de utilizar diccionarios bilingües, se construyen reglas lingüísticas sobre la información morfológica, sintáctica y semántica. Este método convierte las estructuras del lenguaje fuente a las estructuras del lenguaje destino. Además, es capaz de lidiar con una amplia variedad de fenómenos lingüísticos, es extensible y mantenible. Sin embargo, excepciones en la gramática agregan dificultades al sistema.
- **Traducción basada en bases de conocimiento**. Conocido como KBMT (*Knowledge Based Machine Translation*). Este método se basa en bases de conocimiento. Un ejemplo de traductor basado en este enfoque es el proyecto KBMT-89 (Nirenburg 1989), el cual utiliza una ontología de cerca de 1500 conceptos para llevar a cabo la traducción del inglés al japonés y viceversa.
- **Traducción basada en corpus**. Este método es de los más ampliamente explorados debido a su alto nivel de precisión alcanzado. Esta cuenta con los siguientes enfoques principales:
    - **Traducción estadística automática.** En este enfoque se utilizan corpus bilingües. Este enfoque se divide en tres enfoques: a) modelo basado en palabras, donde se requieren algoritmos relacionados con el alineamiento de las palabras; b) modelo basado en la frase, donde se desarrolla una secuencia de palabras en el lenguaje fuente y destino; y c) modelo basado en la sintaxis, el cual utiliza reglas de traducción que consisten en una secuencia de palabras y variables en el lenguaje fuente, un árbol sintáctico en el lenguaje destino y un vector de valores que describe la probabilidad de pares.
    - **Traducción basada en ejemplos.** Este método encuentra ejemplos análogos de las combinaciones de idiomas. Así, dado un conjunto de sentencias en el lenguaje fuente y sus correspondientes traducciones en el lenguaje destino, estos son utilizados para traducir sentencias semejantes.
    - **Traducción basada en el contexto.** Este se basa en un modelo de traducción ligero que utiliza un diccionario bilingüe y un decodificador de largo alcance a través de n-gramas y solapamiento en cascada. El proceso de traducción es mejorado mediante una sustitución de *tokens* y

frases, tanto para el lenguaje origen como destino, cuando los mejores candidatos no pueden ser resueltos por el decodificador (Carbonell et al. 2006). Esta sustitución utiliza un generador de sinónimos implementado como un proceso de aprendizaje no supervisado basado en corpus. El decodificador requiere de un amplio corpus del lenguaje destino y la substitución de un corpus separado del lenguaje fuente.

2.3.4.4 Minería de datos

La minería de datos involucra el uso de herramientas de análisis de datos sofisticadas para descubrir patrones válidos y relaciones previamente desconocidos en grandes conjuntos de datos (Edelstein 1998). Entre estas herramientas se encuentran modelos estadísticos, algoritmos matemáticos y métodos de máquinas de aprendizaje. De esta manera, la minería de datos consiste no solo en la recopilación y gestión de datos, sino que también incluye tareas de análisis y predicción.

Esta tecnología puede ser empleada sobre información representada de múltiples formas tales como cuantitativa, textual o multimedia. Para ello, estas aplicaciones usan una variedad de parámetros para examinar los datos tales como: asociación (patrones donde un evento está conectado a otro), secuencia (patrones donde un evento conduce a otro evento), clasificación (identificación de nuevos patrones), *clustering* (encontrar grupos de hechos previamente desconocidos) y previsión (descubrimiento de patrones a partir de los cuales es posible realizar predicciones respecto a las actividades futuras) (Seifert 2004).

La minería de datos utiliza un enfoque de descubrimiento en el cual los algoritmos pueden ser utilizados para examinar diversas relaciones de datos multidimensionales simultáneamente, identificando aquellas que son únicas o frecuentemente representadas. Para llevar a cabo este proceso con éxito, es importante contar con una clara formulación del problema a resolver y con el acceso a los datos pertinentes (Makulowich 1999).

Gracias a los avances tecnológicos, actualmente la minería de datos es utilizada con una amplia gama de propósitos tanto en sectores públicos como privados. Por ejemplo, en el contexto médico, la minería de datos ayuda a predecir la efectividad de un medicamento o un procedimiento médico. En el sector público, la minería de datos ayudó al gobierno federal a recuperar millones de dólares en pagos fraudulentos (Cahlink 2000).

2.3.4.5 Generación de resúmenes automática

El objetivo de la generación de resúmenes es tomar una fuente de información, extraer contenido de esta y presentar el contenido más importante al usuario de forma condensada y sensible a sus necesidades o de la aplicación (Mani 2001). De acuerdo con (Nenkova, Maskey, and Liu 2011), existen diversos tipos de resúmenes:

- **Resúmenes extractivos**. Estos son producidos mediante la concatenación de diversas sentencias que son tomadas tal cual aparecen en el texto a resumir.
- **Resúmenes abstractos**. Estos pueden reusar frases o cláusulas del documento a resumir, pero la mayoría del resumen esta expresado en palabras del autor del resumen.
- **Nivel de documento**. Los sistemas producen el resumen de solo un documento.
- **Resumen multidocumento**. Los cuales proveen un breve compendio de muchos documentos.
- **Resumen indicativo**. Este permite al lector identificar el tema y puede proveer características tales como longitud, estilo de escritura, entre otras.
- **Resumen informativo**. Este puede ser leído en lugar del documento, es decir, este incluye hecho reportados en el documento original.
- **Resumen de palabras clave**. Este consiste en un conjunto de palabras o frases indicativas contenidas en el documento original.
- **Titular**. El o los documentos de entrada con resumidos por una sola sentencia.
- **Resumen genérico**. Este realiza pocas suposiciones respecto a la audiencia u objetivo del resumen, ni del género o dominio de los materiales que necesitan ser resumidos. Generalmente, asume que la audiencia es solo una, es decir, que cualquier persona puede leer el resumen.
- **Resumen enfocado en la consulta**. Su objetivo es resumir sólo la información que es relevante para la consulta específica de un usuario.
- **Actualización de resumen**. Este es un resumen multidocumento sensitivo al tiempo, es decir, transmite el desarrollo de un evento importante más allá de lo que el usuario ya ha visto.

La generación de resúmenes ha sido dividida en tres principales etapas (Jones and others 1999): 1) la interpretación del texto fuente para obtener una representación de texto; 2) transformación de la representación de texto en un resumen de la representación; y 3) la generación del resumen del texto a partir de la representación anterior. La generación de resúmenes requiere un análisis explícito y detallado de factores

del contexto (Lloret 2008). Jones y colaboradores distinguen tres clases de factores del contexto(Jones and others 1999):

- **Factores de entrada**. Estos factores son: forma del texto (por ejemplo, la estructura de este), tipo de tema (común, especializado o restringido) y unidad (uno o múltiples documentos).
- **Factores de propósito**. Estos son los más importantes y se dividen en: situación (el contexto dentro del cual son usados), audiencia (lectores del resumen) y uso (objetivo de este).
- **Factores de salida**. En esta clase se agrupan factores tales como material (contenido), formato y estilo.

Existen diversos enfoques para la generación de resúmenes tales como el presentado en (Alonso et al. 2004). A continuación, presentamos la propuesta de (Mani and Maybury 1999):

- **Nivel superficial**. Este enfoque se basa en características superficiales y en una combinación selectiva de éstas con el objetivo de obtener una función de relevancia para extraer la información. Entre estas características se encuentran: a) temáticas, las cuales recaen en estadísticas de ocurrencia de palabra; b) ubicación, la cual se refiere a la ubicación de las sentencias a incluir en el resumen, dentro del texto, párrafo o sección en particular; y c) palabras clave tales como *importante*, *en este trabajo*, etc.
- **Nivel de entidad**. Este construye una representación del texto a través del modelado de las entidades y relaciones del texto. Estas relaciones incluyen: a) similitud, por ejemplo, que palabras compartan el mismo *stem*; b) proximidad, la cual refiere a la distancia entre unidades de texto; c) relaciones de tesauro, tales como sinónimos; d) relaciones lógicas tales como acuerdo, contradicción, vinculación y consistencia; e) relaciones sintácticas; f) relaciones basadas en el significado, por ejemplo, las relaciones predicado-argumento.
- **Nivel de discurso**. El objetivo de este nivel es modelar la estructura global del texto y sus relaciones. Dentro de la información que puede ser explotada en este nivel se encuentra: a) formato del documento; b) discusiones de temas; y c) la estructura retórica del texto. Esto permite construir la estructura de la coherencia de un texto de manera que unidades centrales del texto reflejen su importancia.

2.3.4.6 Análisis de sentimientos

Las opiniones son una parte fundamental de las actividades que lleva a cabo el ser humano ya que estas influyen en el comportamiento del mismo. Por ejemplo, cuando necesitamos tomar una decisión respecto a la compra o consumo de un producto o servicio, necesitamos conocer las opiniones de otros. También, diversas organizaciones y negocios quieren siempre conocer las opiniones públicas y de sus consumidores respecto a sus productos o servicios. La gran explosión que han tenido los medios sociales tales como foros, blogs y redes sociales, han dado paso a que tanto individuos como organizaciones utilicen el contenido en estos medios para tomar decisiones. Sin embargo, encontrar y monitorizar las opiniones contenidas en estos medios resulta una tarea complicada debido a la gran cantidad de información contenida. Este problema ha dado paso a la necesidad de contar con sistemas de análisis de sentimientos automáticos.

El análisis de sentimientos, también conocido como minería de opiniones, es el campo de estudio que analiza las opiniones, sentimientos, evaluaciones, actitudes y emociones de la gente hacia entidades tales como productos, servicios, organizaciones, individuos, cuestiones, eventos, temas, y sus atributos (Liu 2012). De manera general, el análisis de sentimientos se puede llevar a cabo en tres niveles principales:

- **Nivel de documento**. El objetivo de este nivel es clasificar si un documento de opinión completo expresa un sentimiento positivo o negativo (Pang, Lee, and Vaithyanathan 2002). Este nivel asume que cada documento expresa opiniones de una sola entidad.
- **Nivel de sentencia**. Este nivel determina si una sentencia expresa una opinión positiva, negativa o neutral, donde neutral usualmente significa que no hay opinión (Liu 2012).
- **Nivel de aspecto o entidad**. Este nivel lleva a cabo un análisis de grano fino. En muchas aplicaciones, los objetos de opinión son descritos por entidades y sus diferentes aspectos. Por ejemplo, se puede opinar de un restaurante a través de entidades tales como servicio, menú, entre otras. Así, el objetivo de este nivel es descubrir el sentimiento sobre esas entidades y sus aspectos.

De acuerdo con (Yusof, Mohamed, and Abdul-Rahman 2015) los principales enfoques de análisis de sentimientos pueden ser clasificados en dos categorías:

- **Clasificación basada en lexicones**. Este enfoque se divide en dos categorías, basada en diccionarios y basada en corpus. En la primera categoría, el sentimiento es identificado utilizando diccionarios tales como SentiWordNet

(Esuli and Sebastiani 2006). En la segunda categoría se identifican las palabras de opinión en base en una lista de palabras. Esta categoría se puede dividir en estadístico y semántico. En el enfoque estadístico, se calculan las coocurrencias de palabras para identificar el sentimiento. En el enfoque semántico, los términos son representados en un espacio semántico para descubrir la relación entre términos.

- **Clasificación basada en máquinas de aprendizaje**. Este enfoque es básicamente un algoritmo de clasificación, el cual utiliza un corpus etiquetado con el fin de reconocer las características que utilizará para clasificar el sentimiento (Giannakopoulos et al. 2012). Este enfoque puede ser supervisado, semisupervisado y no supervisado. Debido a que el análisis de sentimientos es una tarea de clasificación, es más conveniente adoptar un enfoque supervisado (Yusof, Mohamed, and Abdul-Rahman 2015). El proceso de análisis de sentimientos en este enfoque se divide en dos fases: extraer características de datos de entrenamiento y convertirlos en vectores de características, entrenar el clasificador con el vector de características y aplicar el clasificador para casos que no se ven (G. Wang et al. 2014).

2.4 Sistemas de búsqueda de respuestas

2.4.1 Definición

La búsqueda de respuestas, llamado en inglés Question Answering (QA), puede ser definido como "*un proceso capaz de entender preguntas formuladas en lenguaje natural como el inglés y responder exactamente con la información solicitada*" (Indurkhya and Damerau 2010). Un sistema QA debe ser capaz de determinar la necesidad de información expresada en una pregunta, localizar la información requerida, extraerla, generar una respuesta y presentarla.

Debemos enfatizar la diferencia entre un sistema de recuperación de información (RI) y un sistema QA. Por un lado, un sistema de RI tiene como objetivo devolver, dada una consulta planteada por un usuario, los documentos más relevantes de acuerdo a la consulta provista. Estos documentos pueden pertenecer a una colección o biblioteca digital o ser localizados por algún buscador de Internet. Por otro lado, un sistema QA es un tipo de recuperación de información en el que se parte de una consulta expresada en lenguaje natural y debe devolver no ya un documento que sea relevante (es decir que contenga la respuesta), sino la propia respuesta (normalmente un hecho) (Martínez Barco et al. 2007).

2.4.2 Antecedentes

La investigación en el campo de QA se ha desarrollado desde las perspectivas de IA y RI. En etapas tempranas de IA, los sistemas QA respondían a preguntas utilizando como primera fuente de conocimiento la información almacenada en bases de datos. Este tipo de sistemas fueron denominadas interfaces de lenguaje natural para bases de datos (NLIDB, *Natural Language Interfaces to DataBases*). Algunos ejemplos de este tipo de sistemas son LUNAR (Woods 1973) y BASEBALL (Green Jr et al. 1961). Estos sistemas traducen las preguntas expresadas en lenguaje natural en una serie de consultas entendibles por la base de datos, con el objetivo responder a la pregunta provista. Otro ejemplo, es el sistema QUALM (Lehnert 1977) que responde preguntas a partir de una base de conocimientos construida previamente a partir de documentos textuales.

Desde una perspectiva de RI, en la década de 1990, surgieron un conjunto de formas de evaluación de sistemas QA tales como la conferencia TREC (Voorhees 2001), CLEF (Vallin et al. 2005) y NTCIR (Kando 2005). En estos enfoques, los sistemas QA se enfocan en encontrar fragmentos de texto que contengan la respuesta dentro de una amplia colección de documentos. Además, estos combinan técnicas de RI y PLN que son generalmente independientes del dominio de la aplicación. En otras palabras, estas investigaciones se enfocan en sistemas QA basadas en texto y de dominio abierto. Por ejemplo, el sistema MURAX (Kupiec 1993), el cual es considerado el primer sistema QA de dominio abierto ya que combina técnicas de RI con técnicas básicas de PLN (etiquetado POS y comparación de patrones sintácticos).

Ambos enfoques, IA y RI, se han desarrollado en paralelo y representan a su vez dos enfoques que pueden denominarse como sistemas QA basados en bases de conocimiento estructurada y sistemas QA basados en texto libre, respectivamente. Los primeros son aptos para aplicaciones que utilicen consultas complejas en un ambiente de información estructurado. Los segundos son probablemente aptos para aplicaciones de amplio propósito que tratan con preguntas sobre hechos simples (Indurkhya and Damerau 2010).

2.4.3 Esquema básico de un sistema de búsqueda de respuestas

La arquitectura típica de un sistema de búsqueda de respuestas basado en texto libre está compuesta por tres módulos principales: (1) módulo de análisis de la pregunta, (2) módulo de recuperación de información, y (3) módulo de extracción de la respuesta. Estos elementos funcionan de manera conjunta con el objetivo de procesar preguntas y documentos en diferentes niveles, hasta obtener una respuesta final. La Figura 2-12 muestra la arquitectura antes mencionada.

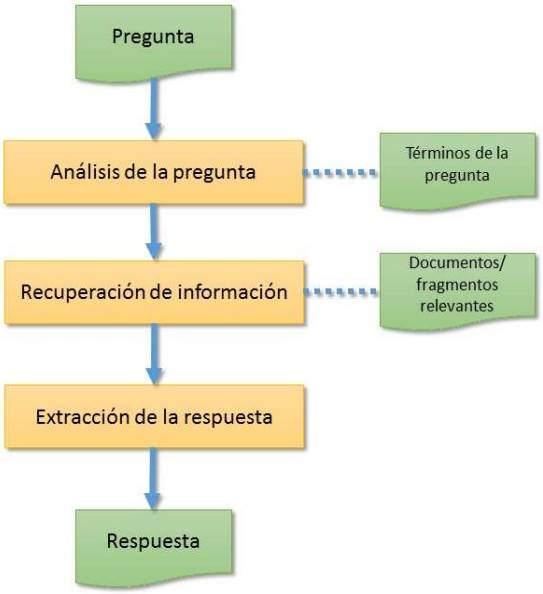

Figura 2-12. Arquitectura básica de un sistema de búsqueda de respuestas.

2.4.3.1 Análisis de la pregunta

En este módulo, las preguntas provistas al sistema son procesadas con el objetivo de detectar y extraer información que pueda ser de utilidad para los siguientes módulos. Este módulo está compuesto de dos fases principales (ver Figura 2-13): clasificación de la pregunta y formulación de consultas (Athenikos and Han 2010). La clasificación de la pregunta determina el tipo de pregunta proporcionada por el usuario y el tipo de respuesta esperado por éste. La formulación de consultas consiste en la generación de una consulta que será la entrada a un motor de recuperación de documentos, transformando la pregunta en alguna forma canónica. El proceso de análisis de la pregunta es muy importante ya que el desempeño de los siguientes módulos dependerá en gran medida de la calidad de la información extraída de la pregunta.

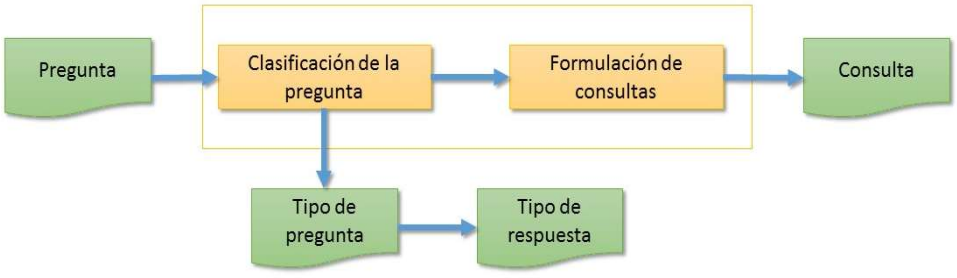

Figura 2-13. Proceso de análisis de la pregunta de un sistema QA.

2.4.3.1.1 Clasificación de la pregunta

Una parte importante del análisis de la pregunta es determinar la clase de la pregunta (qué, quién, etc.), así como el tipo de respuesta esperada por el usuario (lugar, persona, cantidad, etc.). Por ejemplo, para la pregunta ¿Quién es el presidente de México?, el tipo de presunta es ¿Quién?, mientras que lo que se espera como respuesta es el nombre de una persona.

Los tipos de pregunta y respuestas son categorizados en taxonomías. Una de las primeras taxonomías fue propuesta por Moldovan y colaboradores para el sistema Lasso (D. I. Moldovan et al. 1999). Esta taxonomía, mostrada en la Tabla 2-4, está basada en el dominio relacionado con las noticias de TREC 1999 y define una jerarquía de preguntas de dos niveles.

Tabla 2-4. Taxonomía de la pregunta (D. Moldovan et al. 2000)

Clase	Subclase	Tipo de respuesta
what	basic what	money | number | definition | title | noun phrase | undefined
	what-who	person | organization
	what-when	date
	what-where	location
who		person | organization
how	basic how	manner
	how-many	number
	how-long	time | distance
	how-much	money | price
	how-much-<modifier>	
	how-far	distance
	how-tall	number
	how-rich	undefined
	how-large	number
where		location
when		date
which	which-who	person
	which-where	location
	which-when	date
	which-what	noun phrase | organization
name	name-who	person | organization
	name-where	location
	name-what	title | noun phrase
why		reason
whom		person | organization

El primer nivel de clasificación identifica el tipo básico de la pregunta, el cual, para este caso, se basa en la partícula interrogativa. El segundo nivel se centra más aún el tipo de pregunta añadiendo información sobre el contexto de la pregunta y determinando el tipo de respuesta esperado. Otras taxonomías más complejas las podemos encontrar en (X. Li and Roth 2002), la cual clasifica cerca de 50 tipos de respuestas incluyendo colores, religiones e instrumentos musicales o la taxonomía de ISI (E. H. Hovy et al. 2000) con 140 diferentes tipos de respuestas.

Para llevar a cabo la clasificación de preguntas existen dos enfoques principales: basado en reglas y basado en máquinas de aprendizaje.

- **Basado en reglas**. En este enfoque existen un conjunto de reglas que mapean patrones de preguntas en tipos de preguntas. Estos patrones son expresados a través de expresiones regulares. La identificación del tipo de respuesta usualmente se lleva a cabo a través del análisis de los términos interrogativos de la pregunta (*wh*-términos, como *what, where, which*, etc.). De esta manera, para la pregunta: *Where is the Big Ben?* el término *where* indica que el usuario está buscando un lugar, tal como se aprecia en la taxonomía presentada en la Tabla 2-4. Este enfoque puede dar resultados aceptables para una taxonomía de preguntas de grano grueso. Sin embargo, el desarrollo de tales reglas puede demandar mucho tiempo cuando la taxonomía de la pregunta es de grano fino o cuando se requiere una muy alta precisión. Además, un cambio en el dominio de la aplicación, requerirá un nuevo conjunto de reglas de clasificación.
- **Basado en máquinas de aprendizaje**. En este enfoque, la clasificación de la pregunta se lleva a cabo con métodos estadísticos de clasificación. Para ello, es necesario un corpus de preguntas con anotaciones que indiquen la clase correcta de cada pregunta. En (Zhang and Lee 2003) los autores comparan el uso de árboles de decisión, vecino más cercano, clasificador Bayesiano ingenuo, y SVM (Support Vector Machine) para la clasificación de la pregunta, obteniendo este último los mejores resultados. Un factor clave en este enfoque es la selección de características que representan las preguntas. Por ejemplo, en (X. Li and Roth 2002) usan características tales como palabras, etiquetas POS, entidades nombradas y n-gramas. Por otra parte en (Zhang and Lee 2003) incorporan características basadas en información sintáctica.

2.4.3.1.2 Generación de consultas

La arquitectura de los sistemas QA tienen en su núcleo un sistema de recuperación de información (RI). Debido a esto, el sistema QA extrae información específica de la pregunta

que permita la generación de consultas, que procesadas por un sistema RI, facilitan la selección de la información que da soporte a la generación de la respuesta. La generación de este tipo de consultas se lleva a cabo mediante dos enfoques: selección de palabras clave y la generación de patrones de respuesta (Indurkhya and Damerau 2010), los cuales no son mutuamente excluyentes.

- **Selección de palabras clave**. Este enfoque consiste en la selección de términos contenidos en la pregunta que indican la existencia de una respuesta candidata en el texto circundante. Por ejemplo, en la pregunta: ¿Qué países limitan con el sur de México?, el conjunto de palabras clave estaría conformado por los términos *limitan*, *sur* y *México*.
- **Generación de patrones de respuesta**. En este enfoque, las consultas constituyen una o varias combinaciones de los términos de la pregunta, de tal manera que cada consulta resultante expresa una forma a través de la cual es posible hallar la respuesta. Para estos casos, el sistema RI extraerá fragmentos de texto literal que contengan cualquiera de las expresiones candidatas a respuesta asociadas con cada tipo de pregunta (Hermjakob, Echihabi, and Marcu 2002) (Soubbotin and Soubbotin 2002).

Los enfoques antes descritos están orientados a preguntas formuladas en lenguaje natural libre. Existen otros ejemplos de sistemas que restringen el lenguaje de entrada de tal manera que las preguntas son expresadas mediante un vocabulario controlado, es decir, mediante un vocabulario con sintaxis y vocabulario restringido.

2.4.3.2 Recuperación de información

Este módulo utiliza la información extraída por el módulo de análisis de la pregunta para llevar a cabo la selección inicial de información candidata a ser respuesta. Este proceso varía de acuerdo al tipo de acceso a los datos contenedores de la respuesta.

- **Recuperación de información sobre datos estructurados**. En este caso la recuperación de información se transforma en un acceso a la fuente de información mediante un lenguaje de consulta. Para lo cual se deben establecer mecanismos de traducción de la pregunta al lenguaje de consulta tal como SQL.
- **Recuperación de información sobre datos no estructurados o semiestructurados**. En este caso la información susceptible de contener la respuesta debe ser analizada y relacionada con la pregunta. Para ello, técnicas de RI son utilizadas en vez de técnicas NLP, debido al requerimiento de velocidad de procesamiento (Roberts and Gaizauskas 2004). Entre estas

técnicas se encuentran los métodos de clasificación, aunque las características de motores Booleanos los hacen más aptos para sistemas QA (D. I. Moldovan et al. 1999) (Tellex et al. 2003). Estas técnicas reducen el número de documentos a un pequeño subconjunto que incluyen textos que pueden contener la respuesta. Después, un componente filtra o penaliza aquellos textos que no contengan instancias del tipo de respuesta esperada.

Debido a que un mismo concepto puede ser expresados a través de diferentes, pero equivalentes, términos, formas o expresiones, el conjunto de palabras clave contenidas en la pregunta es extendido a través de alternativas morfológicas, léxicas o semánticas, ya sea a través de técnicas estadísticas (Chu-Carroll et al. 2006) o bases de conocimiento semántico como WordNet (Miller 1995). Con eso se logra extraer información que contenga los conceptos clave que no están expresados tal cual en la pregunta.

2.4.3.3 Extracción de la respuesta

Este módulo analiza detalladamente los textos relevantes seleccionados por el módulo anterior con el objetivo de localizar y extraer la respuesta. Este compara entre sí la representación de la pregunta y la representación de los textos relevantes para obtener el conjunto de respuestas candidatas. Finalmente, estas respuestas candidatas se clasifican en función de la probabilidad de que sea la respuesta correcta. Esta clasificación puede ser llevada a cabo a través de la combinación de los siguientes métodos:

- **Similitud**. Provee el rango más alto a las respuestas candidatas que están en un contexto similar a la pregunta.
- **Popularidad**. Provee el rango más alto a las respuestas candidatas que aparecen más frecuentemente.
- **Patrones**. Provee el rango más alto a las respuestas candidatas que coinciden con patrones específicos de la pregunta.
- **Validación de la respuesta**. Provee el rango más alto a las respuestas candidatas que tienen valores aceptables.

2.4.3.3.1 Similitud

Cuando una sentencia es similar a la pregunta y esta contiene una cadena compatible con el tipo de respuesta esperada, es razonable concluir que la respuesta candidata es la respuesta correcta. Para determinar la similitud se puede tomar en cuenta el número de palabras en común o utilizar métodos basados en información sintáctica o semántica. Trabajos tales como (Oard et al. 2000) toma en cuenta la similitud entre estructuras sintácticas de la pregunta y de las respuestas candidatas. En (Attardi et al. 2002) y

(Litkowski 2002) usan tripletas semánticas para representar sentencias relevantes. Una tripleta semántica está formada por una entidad discurso, su rol semántico en el texto y el término con el cual está relacionado. Otro ejemplo es el presentado en (Lee et al. 2001) donde utilizan 361 patrones léxico-semánticos para localizar la respuesta.

2.4.3.3.2 Popularidad

Un método simple para clasificar respuestas candidatas es contar cuantas veces ellas aparecen. Sin embargo, es necesario determinar si dos cadenas de textos hacen referencia a la misma entidad, lo cual está relacionado con la tarea de correferencia, término lingüístico definido como la referencia al mismo ente de dos o más expresiones lingüísticas en el mismo texto.

Un método que ha ganado popularidad es la explotación de la información disponible en la Web. Esta información produce redundancia de datos, lo que provoca que la respuesta a una pregunta puede aparecer diversas veces en distintos contextos. De esta manera, este método se basa en el supuesto de que como el número de variaciones de las formas para justificar una respuesta crecen, la probabilidad de encontrar una justificación simple incrementa (Indurkhya and Damerau 2010). Por ejemplo, en (Brill et al. 2001) utilizan este enfoque para producir una serie de consultas Web por cada pregunta obteniendo con ello información cercana a las palabras de la consulta y etiquetando cadenas de texto que aparecen frecuentemente en los fragmentos como posibles respuestas.

2.4.3.3.3 Patrones

Este método se basa en el desarrollo de un conjunto de patrones que indican formas típicas de responder preguntas específicas. Por ejemplo, en (Soubbotin and Soubbotin 2002) se construyen manualmente una serie de patrones, denominados patrones indicativos, mediante el estudio de expresiones que son usualmente respuestas de cierto tipo de preguntas. El problema de este enfoque es el tiempo que consume el desarrollo del conjunto de patrones. En este sentido, diversos sistemas tales como (Ravichandran and Hovy 2002) han experimentado el uso de técnicas de máquinas de aprendizaje para desarrollar estos patrones. En este trabajo reúnen el corpus a través de la Web, basados en el supuesto de que específicos patrones de preguntas son respondidos con específicos patrones de respuesta. Finalmente, sistemas tales como (Hermjakob, Echihabi, and Marcu 2002) combinan técnicas de aprendizaje automático con una revisión manual de los patrones resultantes.

2.4.3.3.4 Validación de la respuesta

Este método es utilizado para determinar si una respuesta candidata es una buena respuesta. Uno de los métodos para este enfoque consiste en determinar si la respuesta es una entidad nombrada compatible con el tipo de respuesta esperada. Sin embargo, en ocasiones es necesario llevar a cabo una validación más detallada. Existen métodos basados en la lógica que convierten la pregunta y la sentencia que contiene la respuesta en formas lógicas. Estas son procesadas por un módulo de lógica para determinar si es la respuesta correcta.

Green (Green 1969) utiliza técnicas de resolución para encontrar respuestas a las preguntas. En este enfoque la negación de la pregunta es introducida y el módulo de lógica, basado en un conjunto de axiomas y proposiciones, necesita encontrar una contradicción para probar la pregunta. Sistemas QA actuales producen los axiomas automáticamente a partir de la sentencia que contiene la respuesta. Además, los axiomas que dan soporte a los enfoques de validación de la respuesta basados en la lógica necesitan ser extendidos con información que rara vez está explícita en el texto, es decir, que generalmente se debe asumir. A este proceso se le denomina abducción (Hobbs et al. 1988). Para evitar el uso de información falsa, los axiomas que son abducidos están directamente conectados con la información específica de la pregunta y respuesta (D. Moldovan, Clark, et al. 2003) tales como: información lingüística (nombres compuestos, conjunciones, etc.,), entidades nombradas (vinculación de instancias de entidades nombradas con sus respectivos tipos), y recursos léxicos (cadenas léxicas, WordNet).

2.4.4 Principales métodos

A continuación, se describen los principales métodos en sistemas QA (Kolomiyets and Moens 2011) para analizar las necesidades de información (consultas o preguntas) expresadas en lenguaje natural.

2.4.4.1 Bolsa de palabras (Bag-of-words)

Este enfoque es el más simple ya que considera a la pregunta como una bolsa de palabras, es decir, como un conjunto de palabras, posiblemente sin palabras vacías (artículos, preposiciones, etc.) y sin tomar en cuenta ninguna característica estructural de la sentencia o gramaticalidad del discurso o información de las posiciones de las palabras.

El caso más simple es cuando tanto la pregunta como los objetos de información están representados como una bolsa de palabras, para lo cual utilizan un modelo booleano para obtener las respuestas candidatas. Existen otros métodos, tal como el modelo de espacio

vectorial, en el cual un objeto de información y la consulta en lenguaje natural son representadas como un vector de términos en un espacio de *p* dimensiones, donde *p* es el número de términos en el vocabulario. El documento y el vector de la consulta son comparados a través de su similitud (por ejemplo, mediante el coseno del ángulo entre los vectores) o su distancia. Comparado con el método booleano, este enfoque devuelve respuestas aún si las restricciones planteadas por la pregunta se cumplen parcialmente, lo que a su vez reduce la precisión.

Existen métodos probabilísticos que ofrecen una forma natural de integrar relaciones probabilísticas entre los términos de un modelo. Entre estos enfoques se encuentra el modelo del lenguaje (Croft and Lafferty 2013), en el cual un modelo de contenido probabilístico es construido a partir del objeto de información (documento o sentencia), y posiblemente de la información de la consulta. De esta manera, se compara la información contenida en la consulta usando modelos de información tales como el Kullback-Leibler (Kullback and Leibler 1951). En todos los modelos, la tarea de reemplazar palabras por sus sinónimos puede mejorar las posibilidades de encontrar correspondencias entre la pregunta y la información.

Los modelos antes descritos carecen de la precisión necesaria en sistemas QA, incluso para preguntas factuales simples que requieren la recuperación de una entidad como persona, fecha, etc. (D. Moldovan, Paşca, et al. 2003).

2.4.4.2 Análisis morfo-sintáctico de oraciones en lenguaje natural

En el análisis morfológico, las técnicas de *stemming* y *lematización* pueden incrementar la oportunidad de encontrar coincidencias entre la pregunta y posibles respuestas. Para el caso de análisis sintáctico, los etiquetadores POS y la técnica de análisis sintáctico superficial o *chunking* permiten detectar la clase sintáctica de cada palabra (tales como sujetos o verbos) y detectar sintagmas nominales o frases verbales, respectivamente. Además, también es posible utilizar el análisis de constituyentes y el análisis de dependencias.

Técnicamente, los núcleos de los árboles sintácticos son utilizados para obtener la similitud entre dos árboles de dependencia de una sentencia (Collins and Duffy 2001), donde los nodos palabra son enriquecidos con etiquetas POS y posible información semántica. De esta manera, tales núcleos pueden ser utilizados para obtener la relación entre la consulta y la respuesta candidata. Otro enfoque utilizado son los modelos de edición de árboles, los cuales reconocen vinculaciones de texto y paráfrasis, y pueden

encontrar similitudes semánticas entre las preguntas y las respuestas a través de secuencias de transformación de árboles (Heilman and Smith 2010).

El análisis morfo-sintáctico permite capturar las relaciones entre las palabras de la pregunta, lo que se traduce en una representación enriquecida de la pregunta. Sin embargo, se incrementa la complejidad computacional del procesamiento, debido a que tal análisis es llevado a cabo en tiempo real y el análisis de documentos es llevado a cabo off-line, a expensas de representaciones de indexación más ricas. Por ejemplo, a través de arquitecturas MapReduce (J. Lin and Dyer 2010).

2.4.4.3 Clasificación semántica del tipo de respuesta esperada

Una pregunta en lenguaje natural provee información adicional respecto al tipo de información que se espera como respuesta. Esto permite identificar la clase semántica de la respuesta esperada en la pregunta y la correspondiente clase semántica de la respuesta candidata en la estructura de la información.

Motivados por las evaluaciones TREC y por los parámetros de la tarea, las preguntas fueron categorizadas como factuales, lista, definición, hipotética, causal, procedimiento y confirmación. Para las preguntas factuales, los tipos de respuestas fueron organizados en taxonomías. Un sinnúmero de taxonomías de respuestas han sido propuestas, por ejemplo, en (Srihari and Li 2000) definen clases inspiradas por la conferencia MUC (*Message Understanding Conference*). En (Mahesh, Nirenburg, and others 1995) presentan 27 categorías de respuestas basados en la ontología Microkosmos. (Harabagiu et al. 2000) proponen una taxonomía basada en WordNet, la cual contiene 18 categorías y 15 hojas de las categorías superiores. Otro ejemplo es la taxonomía propuesta en (D. Moldovan et al. 2000) la cual fue presentada en la Tabla 2-4.

Existen diversos métodos para llevar a cabo la identificación del tipo de respuesta esperada. Algunos utilizan reglas escritas manualmente posiblemente en combinación con una gramática (E. Hovy, Hermjakob, and Ravichandran 2002). Otros utilizan técnicas de máquinas de aprendizaje supervisado, los cuales entrenan un modelo de clasificación a partir de ejemplos manualmente anotados (preguntas con sus correspondientes tipos de respuestas). Estudios comparativos del uso de máquinas de aprendizaje para determinar el tipo de respuesta esperada están reportados en (Radev et al. 2005), (Zhang and Lee 2003) y (X. Li and Roth 2006). Con respecto a la fuente de información, el análisis de las respuestas candidatas se lleva a cabo mediante técnicas tales como NER (*Named Entity Recognition*) la cual provee etiquetas semánticas a los elementos contenidos en la respuesta candidata.

En cuanto a la obtención de las respuestas candidatas, existen dos enfoques principales: determinístico y probabilístico. En el primer enfoque, una vez que el tipo de respuesta ha sido identificada, la lista de respuestas candidatas puede ser filtrada por la clase semántica de la respuesta esperada. En el segundo enfoque, la puntuación de confianza del tipo de respuesta esperada o las puntuaciones de unas mejores hipótesis pueden ser incorporadas a un modelo de recuperación probabilístico (Moens 2006).

El problema con el presente enfoque es que existen diferentes tipos de preguntas y respuestas en forma de listas planas o jerarquías. Estas listas facilitan mucho el establecimiento de relaciones entre preguntas y respuestas. Sin embargo, a menudo son muy específicas de la aplicación.

2.4.4.4 Clasificación semántica de todos los constituyentes de las preguntas o sentencias en lenguaje natural

Un etiquetado semántico más completo de la pregunta completa y de la respuesta candidata pueden mejorar el desempeño de un sistema QA tal y como se prueba en (Shen and Lapata 2007). Aquí, las etiquetas son asignadas en el contexto de una situación denotada por un verbo principal en la sentencia. Esta última puede ser representada mediante una estructura predicado-argumento que describe un caso general de composición de roles semánticos: quién hizo qué para quién, cuándo y dónde, para que propósito, por qué medios.

Los analizadores de dependencias han sido utilizados para representar automáticamente consultas como tripletas y de esta manera desambiguar roles sintácticos tales como Sujeto, Objeto y Objeto indirecto (Nivre, Hall, and Nilsson 2006), (D. Lin 2003). También, sistemas de clasificación de marcos y roles semánticos producen una valiosa descomposición de expresiones en lenguaje natural. El reconocimiento de: a) roles semánticos y marcos; b) información, relaciones temporal y espacial más específicas; es comúnmente llevado a cabo a través de técnicas de máquinas de aprendizaje supervisado y más específicamente por medio de algoritmos de clasificación dependientes del contexto tales como campos aleatorios condicionales (McCallum, Schultz, and Singh 2009).

En lo concerniente a la recuperación de posibles respuestas, la semántica (en este caso la etiqueta semántica) puede restringir el mapeo entre la pregunta y respuestas candidatas. Además, esta puede ser usada para mapear sin necesidad de las instancias léxicas, ampliando de esta manera la búsqueda. Otra posibilidad es considerar un modelo lógico de recuperación e inferir la respuesta a una consulta. Por ejemplo, por medio de razonamiento espacial o temporal (Saquete et al. 2009).

2.4.4.5 Identificación de las relaciones del discurso necesarias

Las necesidades de información pueden ser expresadas a través de distintas preguntas. Por ejemplo, preguntas subsecuentes pueden refinar o ampliar la búsqueda original. En estos casos, es de suma importancia detectar como los datos en las preguntas se relacionan entre sí. Por otro lado, las respuestas a una pregunta no necesariamente están localizadas en una sola sentencia. Por ejemplo, para las preguntas cuya respuesta es una lista, las respuestas pueden estar esparcidas a lo largo de la colección de documentos. En estos casos es importante identificar como los datos están conectados a lo largo de los documentos. Una de las relaciones importantes es la de equivalencia, para la cual dos o más menciones a una entidad (por ejemplo, una persona), una acción o a un evento hacen referencia a la misma cosa, en el contexto de la situación descrita en el discurso. Cuando se trata con entidades individuales, a este fenómeno se le denomina resolución de correferencia del sintagma nominal. Este enfoque ha sido empleado en trabajos tales como (Morton 1999), (Zheng 2002) y (Hickl et al. 2007). Además de la equivalencia, otras relaciones pueden ser definidas, incluyendo relaciones que hacen referencia a la hiperonimia (términos que tienen un significado de gran extensión y, por tanto, incluyen otros más concretos o específicos) e hiponimia (palabras de significado restringido con las que se puede concretar la realidad a la que hacen referencia otras de significado más amplio) o referencias temporales y espaciales.

De acuerdo con (Kolomiyets and Moens 2011) hay una escasez de métodos de recuperación que incorporen conocimiento de las relaciones de discurso. Sin embargo, esta es una parte esencial de sistemas QA con fuentes de datos de poca redundancia.

2.4.4.6 Traducción a y recuperación con un lenguaje estructurado

Componentes tales como el etiquetado de roles semánticos permiten la traducción de una pregunta expresada en lenguaje natural a un lenguaje de consulta estructurado. El método más común para consultar bases de datos relacionales es a través del lenguaje SQL (*Structured Query Language*). En el contexto RI a partir de documentos, lenguajes de consulta tales como XPath, XML Path y XQuery han sido desarrollados para manipular información de documentos basados en XML (Fuhr et al. 2008). Algunos ejemplos de traducción de una consulta en lenguaje natural a un lenguaje estructurado utilizando un conjunto de reglas simbólicas se presentan en (Frank et al. 2007) y (Ferrández et al. 2009) o a través de un mapeo (Popescu, Etzioni, and Kautz 2003).

Una vez que se ha traducido la pregunta en un formato estructurado, los modelos de bases de datos clásicos pueden ser aplicados para recuperar la información siguiendo un

modelo determinístico, donde los datos son recuperados solo cuando estos cumplan con las condiciones establecidas en la consulta.

Este enfoque carece de un marco de trabajo que permita modelar la incertidumbre en las traducciones o en la representación de datos. Además, no provee respuestas o una lista de respuestas ponderadas cuando las respuestas cumplen solo ciertas restricciones impuestas por la consulta.

2.4.4.7 Traducción a y razonamiento con una representación lógica

Las preguntas e información pueden ser representados siguiendo un enfoque de lógica de primer orden, también llamada lógica predictiva, la cual permite estudiar la inferencia en los lenguajes de primer orden. Estos lenguajes cuentan con cuantificadores que alcanzan variables de un individuo con predicados y funciones cuyos argumentos son sólo constantes o variables de un individuo. Una variante de este enfoque es el lenguaje MRL (*Meaning Representation Language*) (Blackburn and Bos 2005) que permite representar sentencias expresadas en lenguaje natural a través de lógica predictiva siguiendo un formalismo gramatical basado en marcos de casos. De esta manera, la pregunta: ¿Quién compró WhatsApp? tendría la siguiente representación:

```
COMPRAR (x, WhatsApp)
```

Figura 2-14. Ejemplo de representación lógica.

En esta representación, el argumento de la primera posición (x) denota al comprador y el segundo denota al objeto. Así, un programa lógico buscará en la fuente de información el predicado COMPRAR con el valor conocido del segundo argumento, es decir *WhatsApp*. La traducción de las expresiones en lenguaje natural a estructuras lógicas predicado-argumento a menudo integra el reconocimiento de roles semánticos.

Una vez que la consulta y las respuestas candidatas son representadas utilizando formalismo lógico, la relevancia de una respuesta respecto a la consulta puede ser deducida a través de modelos de prueba de teoremas (Girle and Fitting 1998), siendo estos últimos proposiciones lógicamente deducibles de los axiomas. Además, también es posible utilizar enfoques probabilísticos para determinar la probabilidad de que una respuesta sea la correcta. Por ejemplo, ciertas respuestas pueden ser probadas incluso si todos los patrones no coinciden, resultando en una clasificación probabilística de las respuestas.

Por otro lado, la información semántica puede ser representada como grafos, los cuales ofrecen distintas posibilidades para ponderar respuestas potenciales que puedan ser

relevantes para una consulta. Cuando los grafos son no dirigidos, técnicas de coincidencia de sub-grafos pueden ser aplicados (Mollá 2006). Por otra parte, cuando los grafos son dirigidos, se pueden incorporar los grafos de la consulta y de las respuestas potenciales en una red Bayesiana para realizar inferencias probabilísticas basadas en tal red (Pearl 2014).

2.5 Interfaces de lenguaje natural

2.5.1 Introducción

Uno de los factores limitantes en la usabilidad de las computadoras corresponde a la usabilidad de las interfaces (R. W. Smith 2006). En el inicio, la habilidad de manipular los datos y programas disponibles en una computadora estaba restringido a individuos capaces de dominar una sintaxis y un vocabulario específico a través de interfaces de línea de comandos. Un avance en la usabilidad de las interfaces son las interfaces gráficas de usuario o GUI (*Graphical User Interface*). El uso de este tipo de interfaces se lleva a cabo mediante dispositivos de entrada tales como ratones y teclados, e incluso a través de la pantalla táctil con la que cuentan actualmente un gran número de dispositivos. El uso de las GUI se lleva a cabo principalmente a través de funciones de apuntar y hacer clic, las cuales permiten llevar a cabo tareas tales como seleccionar archivos, seleccionar información de esos archivos, así como manipular diversas aplicaciones. A pesar de que las GUI han hecho que la interacción con la computadora o dispositivos móviles sea más fácil para un gran número de personas, éstas requieren que el usuario tenga conocimiento de cada una de las opciones que le ofrece la interfaz, así como de la ubicación de cada una de estas funciones. Además, en ocasiones las interfaces gráficas varían dependiendo del dispositivo desde el cual se esté utilizando, así como de la versión de la aplicación, misma que sufre de cambios ante constantes actualizaciones por parte del proveedor de la misma. Por el contrario, las interfaces de lenguaje natural (NLI) no requieren que el usuario cuente con un conocimiento especializado, ya que le permite usar todo el poder del lenguaje que ya posee en lugar de verse forzado a utilizar un modo de comunicación poco natural y limitante como lo son las GUI. Así, el objetivo de las NLI es superar la brecha existente entre el rendimiento lingüístico del usuario y la competencia lingüística del sistema computacional subyacente (Manaris 1998).

A pesar de que existen sistemas basados en lenguaje natural enfocados a múltiples tareas tales como la interacción con robots (Khayrallah, Trott, and Feldman 2015), (Stenmark and Nugues 2013), la transcripción y dictado (Vivancos-Vicente et al. 2016) o el control de dispositivos en el internet de las cosas (Noguera-Arnaldos et al. 2015). En esta

sección nos enfocaremos en NLI orientadas a bases de datos relacionales y a bases de conocimiento. De acuerdo con (Perrault and Grosz 1988) la adopción del lenguaje natural en estos últimos tipos de sistemas se debe principalmente a las siguientes razones:

- Este provee un vocabulario inmediato para hablar acerca de los contenidos de la base de datos.
- Este provee los medios para acceder a la información en la base de datos independientemente de su estructura y comunicación.
- Protege al usuario del lenguaje de acceso formal del sistema subyacente.
- Está disponible con un mínimo de entrenamiento tanto para usuarios novatos como ocasionales.

A continuación, se presenta la arquitectura genérica de una NLI y se provee una descripción más detallada de las NLI orientadas a bases de datos, conocidas como NLIDB (*Natural Language Interfaces to Databases*), y de las NLI para bases de conocimiento conocidas como NLIKB (*Natural Language Interfaces to Knowledge Bases*).

2.5.2 Arquitectura genérica de una interfaz de lenguaje natural

Smith (R. W. Smith 2006) propone una arquitectura genérica para una NLI enfocada a algún tipo de aplicación funcional, como lo puede ser un gestor de bases de datos. Esta arquitectura se muestra en la Figura 2-15.

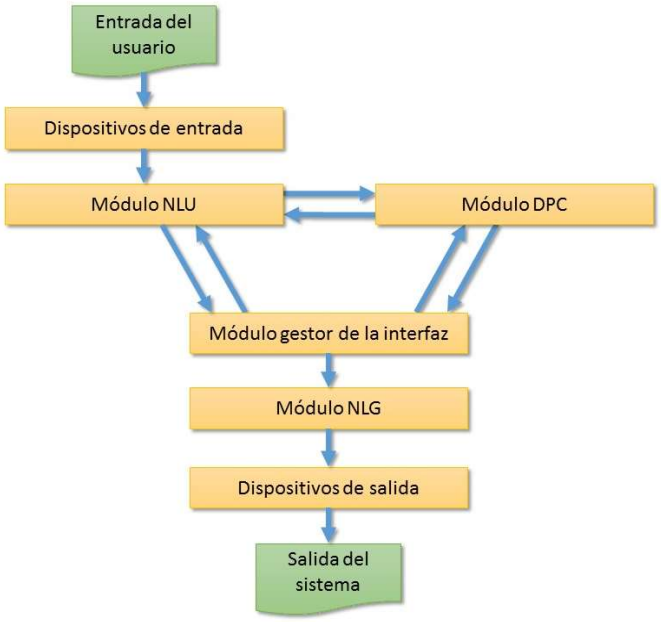

Figura 2-15. Arquitectura genérica de una interfaz de lenguaje natural

Los elementos que componen la arquitectura mostrada en la figura anterior son:

- **Dispositivos de entrada**. Esta comprende una gran variedad de dispositivos tales como el teclado, para interfaces basadas en texto, o el micrófono, para interfaces basadas en el habla.
- **Módulo NLU** (*Natural Language Understanding*). Este analiza la consulta provista por el usuario. En el contexto de IA y lingüística computacional, NLU es un campo de PLN que se ocupa de la comprensión de texto (Ovchinnikova 2012), es decir, se encarga de interpretar el fragmento de texto de entrada. Este proceso puede ser entendido como una traducción del texto expresado en lenguaje natural a una representación en un lenguaje formal no ambiguo. La representación generada por este módulo es usada por los módulos posteriores para llevar a cabo sus respectivas tareas. El módulo NLU puede en ocasiones requerir de información específica del dominio de la aplicación a desarrollar con el objetivo de completar su tarea. Para ello, interactúa con el módulo DPC tal como se aprecia en la Figura 2-15.
- **Módulo DPC** (*Domain Processing Components*). Este componente obtiene la información específica del dominio, es decir, interactúa directamente con la fuente de información de la cual se espera extraer los datos que ayuden a dar respuesta a las solicitudes del usuario.
- **Módulo gestor de la interfaz**. Este módulo comprende todos aquellos componentes que mantienen información acerca de la interacción constante entre la NLI y el humano. Debido a que el lenguaje natural presenta características tales como la ambigüedad, es necesario que la NLI cuente con mecanismos que le permitan intercambiar información con el humano para lograr que este último valide las diferentes interpretaciones que ha dado la NLI a su petición. En otras palabras, para que el usuario pueda aprobar, rechazar e incluso elegir de entre varias interpretaciones aquella que encaje con la petición realizada. Un ejemplo de este tipo de mecanismos de retroalimentación son los diálogos de clarificación los cuales, en caso de ambigüedades, piden al usuario que desambigüe el significado de su petición a partir de sugerencias provistas por el sistema para mejorar la precisión del sistema (Damljanović and Bontcheva 2010). Una vez que el proceso de entendimiento se completa y se crea una representación en lenguaje formal de la petición del usuario, este módulo formula las correspondientes respuestas basado en la información provista por el módulo DCP y la retroalimentación obtenida por el usuario para finalmente enviar esta información al componente NLG.

- **Módulo NLG** (*Natural Language Generation*). La generación de lenguaje natural es el proceso mediante el cual el pensamiento se presenta en lenguaje (McDonald 2010). Sin embargo, en el contexto de IA y lingüística computacional, la NLG se encarga de generar texto o voz en lenguaje natural a partir de representaciones de datos estructurados y procesables por la máquina tales como las bases de conocimiento (Vicente Moreno et al. 2015). En este sentido, este módulo será el encargado de generar una respuesta ya sea solo en lenguaje natural o en combinación con otras modalidades tales como gráficos.
- **Dispositivos de salida**. Estos dispositivos permiten al usuario visualizar los resultados obtenidos por la NLI ante la petición realizada. Ejemplos de este tipo de dispositivos son los monitores, impresoras o sintetizadores de voz.

2.5.3 Interfaces de lenguaje natural orientadas a bases de datos relacionales

2.5.3.1 Antecedentes

Una NLIDB es un sistema que permite al usuario acceder a la información almacenada en una base de datos a través de la escritura de peticiones expresadas en algún lenguaje natural como lo es el inglés (Androutsopoulos, Ritchie, and Thanisch 1995). El desarrollo de los primeros prototipos de NLIDB datan de finales de la década de 1960 y principios de la década de 1970. Uno de los primeros y más conocidos sistemas de este tipo es LUNAR (Woods, Kaplan, and Nash-Webber 1972), el cual fue diseñado como una interfaz enfocada a una base de datos con información relacionada con análisis químicos de rocas de la luna. LUNAR contaba con tres elementos principales, un analizador, una rutina de interpretación semánticas y un interpretador de consultas.

Otras interfaces que aparecieron a finales de los años setenta fueron RENDEZVOUS (Codd 1974) y LADDER (Hendrix et al. 1978). La primera de ellas involucra al usuario a través de diálogos que les ayudan a formular sus consultas. La segunda interfaz podía ser configurada para diferentes sistemas de administración de bases de datos. Además, esta interfaz utilizó gramáticas semánticas, una técnica que entrelaza procesamiento sintáctico y semántico. Esta técnica permitía la implementación de sistemas con características sobresalientes, pero cuya portabilidad de dominio era difícil, ya que demandaba la elaboración de una gramática diferente para cada dominio.

En los años ochenta surgió CHAT-80 (Warren and Pereira 1982) una NLIDB implementada en Prolog que transformaba preguntas en inglés en expresiones Prolog que eran ejecutadas en la base de datos Prolog. En esa misma década se desarrollaron múltiples NLIDB entre las que destacan ASK (Thompson and Thompson 1983) y JANUS

(Resnik 1989). ASK era capaz de interactuar con múltiples bases de datos externas, además de permitirle a usuarios finales enseñarle al sistema nuevas palabras y conceptos durante el proceso de interacción; JANUS permitía interactuar con múltiples bases de datos, para lo cual los sistemas subyacentes podían participar en la evaluación de la solicitud en lenguaje natural ocultando al usuario la heterogeneidad del sistema completo.

Años más tarde, distintas NLIDBs disponibles comercialmente aparecieron y han evolucionado adoptando los grandes avances en el campo de PLN e integrando lenguaje y gráficos para la explotación de las ventajas de ambas modalidades. Algunos de estos sistemas comerciales fueron INTELLECT (Harris 1984), IBM's LANGUAGE ACCESS (Ott 1992), Q&A de Symantec (Hendrix 1986) y LOQUI (Binot et al. 1991).

2.5.3.2 Ventajas y desventajas

El acceso a la información contenida en una base de datos generalmente se lleva a cabo utilizando un lenguaje formal de consulta como SQL. Sin embargo, existe otros mecanismos de acceso a esta información tales como las interfaces basadas en formularios y las interfaces gráficas. Bajo un enfoque de interfaces basadas en formularios el usuario rellena la información que ya es conocida en los correspondientes campos y el sistema completa los campos restantes a través de la consulta de la base de datos subyacente. Cuando existen más de dos respuestas, el sistema responde generando una pila de formularios rellenos, uno por cada posible respuesta. En este contexto existe un método importante, denominado *query by example* (Zloof 1975), que permite al usuario combinar un número arbitrario de formularios, donde cada uno refleja la estructura de la tabla de la base de datos. Por otro lado, en el enfoque de interfaces gráficas, el usuario primero selecciona las tablas de la base de datos a ser utilizadas en la consulta, cada una de las cuales aparece en pantalla como un marco consistente de ranuras de atributos que pueden ser llenados por el usuario, a través del teclado uniendo atributos de los diferentes marcos, a través del ratón o imponiendo restricciones sobre los atributos utilizando el ratón y las opciones de menú.

En la literatura se han mencionado diversas ventajas y desventajas de las NLIDB. Entre las ventajas de este tipo de sistemas podemos mencionar:

- **Sin lenguaje artificial**. Una de las ventajas de las NLIDB es el hecho que el usuario no tiene la necesidad de aprender un lenguaje de comunicación artificial, los cuales pueden ser difíciles de aprender y dominar, al menos para usuarios sin conocimientos especiales en computación.

- **Mejor para unas preguntas**. Existen algunos tipos de preguntas que pueden ser expresadas fácilmente en lenguaje natural, pero que pueden ser difícil de expresar a través de interfaces gráficas o basadas en formularios. Por ejemplo, las preguntas que incluyen negación, tal como: ¿Cuáles departamentos no tienen programadores?, o las que incluyen universalidad como: ¿Cuáles compañías suministran a todos los departamentos?
- **Discurso**. Permiten el uso de expresiones anafóricas mediante el uso de preguntas breves donde el significado de cada una de las preguntas es complementado por el contexto del discurso.

Algunas de las desventajas de las NLIDB mencionadas en la literatura son las siguientes:

- **Cobertura lingüística no obvia**. Una de las quejas frecuentes respecto a las NLIDB es el hecho de que las capacidades lingüísticas del sistema no son obvias para el usuario (Tennant et al. 1983), (Cohen 1992). En otras palabras, los usuarios encuentran difícil entender y recordar a que tipos de preguntas puede o no hacer. Por ejemplo, si el sistema responde de manera correcta a un tipo de pregunta, el usuario puede asumir que el sistema es capaz de dar respuesta a todas las preguntas de ese tipo, hecho que no es cierto. Además, si el sistema provee una respuesta equivocada a un tipo de pregunta, el usuario puede asumir que todas las preguntas de ese tipo no serán respondidas por el sistema.
- **Fallas lingüísticas vs conceptuales**. Cuando una NLIDB es incapaz de interpretar una pregunta, el usuario no sabe si esto se debe a que la pregunta esta fuera de la cobertura lingüística del sistema o si esta fuera de la cobertura conceptual del mismo (Tennant et al. 1983). En ocasiones cuando el usuario piensa que el problema se debe a la limitada cobertura lingüística, este intenta reformular la pregunta utilizando diferentes conceptos que el sistema desconoce. En otras ocasiones, el usuario no intenta reformular la pregunta, pues no nota que la pregunta formulada de esa manera no puede ser respondida por el sistema, aunque una reformulación de la misma podría ser interpretada correctamente por el sistema.
- **Los usuarios asumen inteligencia por parte del sistema**. A menudo, los usuarios de NLIDB asumen que el sistema es inteligente. Por ejemplo, si el sistema provee acceso a través de lenguaje natural a cierta información de la base de datos, los usuarios tienden a creer que el sistema puede deducir otros hechos a partir de esa información, hechos que, a pesar de no estar

explícitamente codificados, resultan obvios para cualquiera con sentido común (Hendrix 1982).

2.5.3.3 Arquitecturas

En (Androutsopoulos, Ritchie, and Thanisch 1995) se presentan algunas arquitecturas utilizadas en el desarrollo de NLIDB las cuales varían respecto al tipo de información aplicada y la forma en que es aplicada. A continuación, se provee una descripción de cada una de ellas.

2.5.3.3.1 Coincidencia de patrones

Algunas de las primeras NLIDB utilizaron técnicas de coincidencia de patrones para responder a las preguntas provistas por el usuario. El conjunto de patrones son definidos utilizando un conjunto de reglas definidas manualmente, de esta manera, la consulta en lenguaje natural es mapeada con el conjunto definido de reglas. La principal ventaja de esta arquitectura es su simplicidad, ya que no requiere la existencia de módulos de análisis e interpretación. En la Figura 2-16 podemos ver los principales componentes de esta arquitectura.

Figura 2-16. Arquitectura NLIDB basada en la coincidencia de patrones.

Los sistemas NLIDB basados en este tipo de arquitectura no tienen que estar basados necesariamente en la técnica discutida anteriormente. Por ejemplo, el sistema SAVVY (Johnson 1984) utiliza técnicas de coincidencia de patrones similares a las aplicadas en el procesamiento de señales.

2.5.3.3.2 Basadas en la sintaxis

Bajo este enfoque, la pregunta provista por el usuario se analiza sintácticamente y el árbol de análisis resultante se mapea a una expresión en un leguaje de consulta de bases de datos. Un ejemplo de NLIDB que utiliza este enfoque es LUNAR (Woods, Kaplan, and Nash-Webber 1972) del cual se habló en secciones anteriores. En la Figura 2-17 se presentan los elementos de la arquitectura en cuestión.

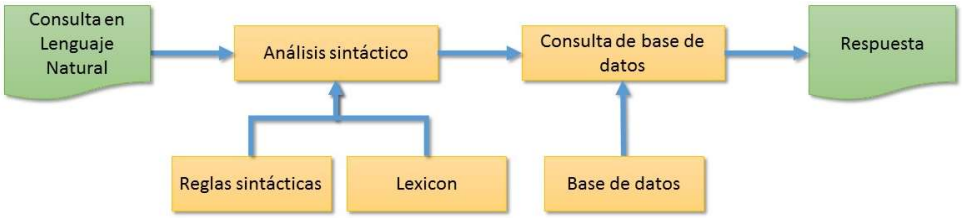

Figura 2-17. Arquitectura NLIDB basada en la sintaxis.

Los sistemas NLIDB basados en la sintaxis utilizan una gramática que describe las posibles estructuras sintácticas de las preguntas del usuario. Utilizando esta gramática, la NLIDB puede mapear el árbol sintáctico de la pregunta en una consulta de base de datos para ser evaluada por el subyacente sistema de base de datos. Este mapeo puede ser llevado a cabo mediante reglas o puede estar completamente basado en la información sintáctica del árbol sintáctico de la pregunta.

Las NLIDB basadas en sintaxis normalmente están enfocadas a sistemas específicos de la aplicación que proporcionan lenguajes de consulta diseñados para facilitar el mapeo entre el árbol sintáctico obtenido a una consulta de base de datos. Sin embargo, esta tarea resulta difícil.

2.5.3.3.3 Basadas en la gramática semántica

En las NLIDB basadas en la gramática semántica, el proceso de pregunta-respuesta se lleva a cabo, al igual que en la arquitectura anterior, mediante al análisis sintáctico de la consulta del usuario y el mapeo del árbol sintáctico a una consulta de base de datos, con la diferencia de que las categorías gramaticales (los nodos que no son hoja en el árbol sintáctico) no necesariamente corresponden con conceptos sintácticos tales como un sintagma nominal o un sustantivo. En este enfoque, la información semántica del dominio de conocimiento (por ejemplo, el hecho de que una pregunta pueda hacer referencia a alumnos o maestros) es conectada en la gramática semántica. Las categorías de la gramática semántica son generalmente elegidas para hacer cumplir restricciones semánticas. En la Figura 2-18 se muestran los elementos de esta arquitectura.

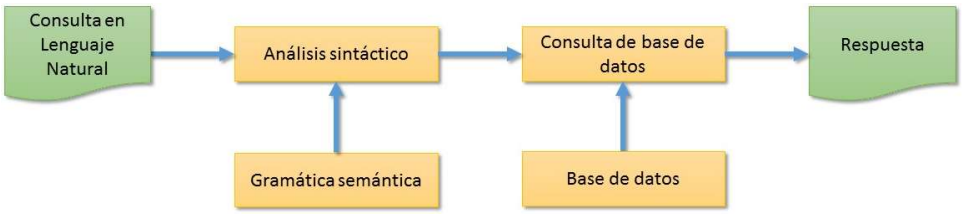

Figura 2-18. Arquitectura NLIDB basada en gramática semántica.

Las categorías gramaticales pueden ser utilizadas para facilitar el mapeo del árbol sintáctico a objetos de la base de datos. Por ejemplo, la existencia de un nodo *pregunta_de_alumnos* le indica al sistema que se debe consultar a la tabla que contiene la información de los alumnos, en vez de tablas relacionadas con maestros o departamentos.

La gramática semántica fue introducida como una metodología de ingeniería que permite al conocimiento semántico ser fácilmente incluido en el sistema. Sin embargo, debido a que esta contiene conocimiento fuertemente enlazado con el dominio de conocimiento, los sistemas basados en esta arquitectura cuentan con una baja portabilidad a otros dominios, ya que para poder hacerlo es necesario desarrollar una nueva gramática semántica enfocada al nuevo dominio.

2.5.3.3.4 Lenguajes de representación intermedia

En esta arquitectura, la NLIDB transforma la pregunta expresada en lenguaje natural en una consulta lógica intermedia, expresada en algún lenguaje interno de representación del significado. La consulta lógica intermedia expresa el significado de la pregunta del usuario a través de conceptos de alto nivel independientes de la estructura de la base de datos. Finalmente, la consulta lógica es traducida a una consulta en un lenguaje de consulta de bases de datos, la cual será evaluada en función de la base de datos. En (Androutsopoulos, Ritchie, and Thanisch 1995) se presenta una posible arquitectura para un sistema NLIDB basado en el uso de lenguajes de representación intermedia, misma que se presenta en la Figura 2-19.

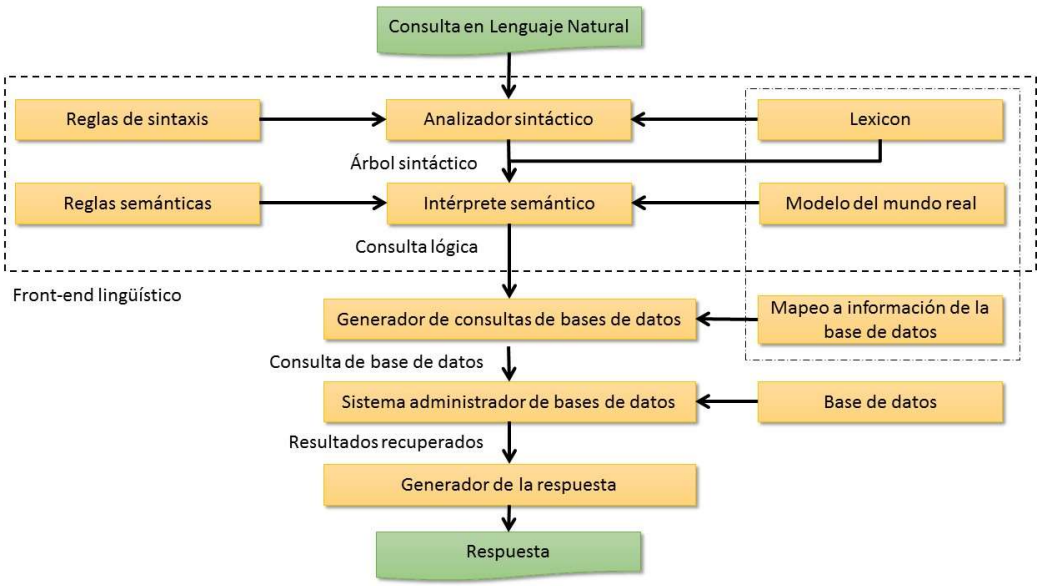

Figura 2-19. Arquitectura de NLIDB basada en un lenguaje de representación intermedio.

Como se puede observar en la figura anterior, la consulta en lenguaje natural es procesada por el analizador sintáctico, el cual consulta un conjunto de reglas de sintaxis, y genera un árbol sintáctico. Después, el intérprete semántico transforma el árbol sintáctico a una consulta lógica intermedia utilizando reglas semánticas similares a las reglas de mapeo y expresiones lógicas de las palabras contenidas en el lexicón. Además, el intérprete semántico consulta un modelo del mundo real que describe la estructura del mundo circundante. Generalmente, este modelo contiene una jerarquía de clases de objetos del mundo real, así como restricciones sobre los tipos de argumentos que cada predicado lógico puede tener. Esta jerarquía es típicamente utilizada en combinación con restricciones sobre los predicados, las cuales especifican las clases a las que pueden pertenecer los argumentos de un predicado lógico.

La consulta lógica generada por los módulos anteriores expresa el significado de la pregunta del usuario a través de conceptos lógicos de alto nivel, es decir, la consulta lógica no hace referencia a objetos de la base de datos tales como tablas o columnas. Con el objetivo de recuperar la información solicitada por el usuario, la consulta lógica es transformada en una consulta expresada en un lenguaje de consulta de bases de datos. Esta tarea es llevada a cabo por el generador de consultas de bases de datos apoyado de un mapeo a información de la base de datos, el cual especifica cómo los predicados lógicos se relacionan con objetos de la base de datos. Una vez generada la consulta de base de datos, esta es ejecutada por el sistema gestor de base de datos o DBMS (*Database Management System*) y pasa la información recuperada al módulo generador de la respuesta, quién devuelve la información recuperada al usuario.

Dentro de las ventajas de la arquitectura basada en un lenguaje de representación intermedia podemos mencionar las siguientes: (1) la parte del sistema encargada de generar la consulta lógica es independiente del DBMS subyacente, lo que permite a la NLIDB ser portada a un diferente DBMS, reescribiendo el módulo generador de la consulta de base de datos; (2) el conocimiento dependiente del dominio está claramente separado del resto de los módulos encargados de generar la representación lógica de la pregunta del usuario, lo que permite la portabilidad del dominio del conocimiento; (3) permite agregar módulos de razonamiento como un módulo intermedio entre el intérprete semántico y el generado de consultas de bases de datos, de tal manera que el sistema NLIDB pueda llevar a cabo razonamiento basado en la información contenida en la base de datos. Para llevar a cabo esta última tarea, el sistema NLIDB debe contar con acceso a experiencia del dominio (la cual puede ser expresada a través de reglas) que le indiquen como llevar a cabo el razonamiento basado en los datos en bruto contenidos en la base de datos.

2.5.4 Interfaces de lenguaje natural orientadas a bases de conocimiento

2.5.4.1 Introducción

De acuerdo con (Davies, Fensel, and Van Harmelen 2003), uno de los principales beneficios obtenidos de la emergencia de la Web Semántica es la posibilidad de acceder a los datos de manera más eficiente a través del uso de ontologías. La consulta a tal información se lleva a cabo mediante lenguajes tales como SeRQL o SPARQL. Sin embargo, la sintaxis de estos lenguajes puede resultar demasiado compleja, especialmente para usuarios no familiarizados con este tipo de lenguajes. Con el objetivo de minimizar la curva de aprendizaje y proveer acceso a dicha información de manera simple tanto a usuarios expertos como ocasionales, los principios de los sistemas NLIDB, quienes mantienen su información en bases de datos relacionales, han sido adaptados a la Web Semántica resultando en lo que se conoce como interfaces de lenguaje natural para bases de conocimiento o de acuerdo con (Jubilson et al. 2016) como NLIKB (*Natural Language Interfaces to Knowledge Bases*).

Existen diversos mecanismos para el acceso a bases de conocimiento tales como las interfaces gráficas, donde los usuarios pueden navegar por los datos, o sistemas que ofrecen una interfaz basada en formularios, tal es el caso de la plataforma KIM (Kiryakov et al. 2004). Además, existen los sistemas que proveen una simple caja de texto donde el usuario puede expresar su consulta a través de palabras clave o preguntas completas y el sistema devuelve la respuesta en una forma entendible por el usuario. En este sentido, es importante mencionar que de acuerdo a la evaluación llevada a cabo en (Kaufmann and Bernstein 2007) los sistemas que proveen interfaces de lenguaje natural son considerados por los usuarios como los más aceptables. Esta conclusión es obtenida a partir de un estudio de usabilidad que comparaba cuatro tipos de interfaces de lenguaje de consulta para bases de conocimiento y que involucró 48 usuarios con conocimientos generales. Para ser más específicos, la opción que permitía realizar búsquedas a través de preguntas completas fue el mecanismo preferido por los usuarios. Por otro lado, en (Linckels and Meinel 2007) presentan una comparación entre interfaces de lenguaje natural basadas en palabras clave y preguntas completas. Aquí, el 69% de los usuarios indicaron que aceptarían usar preguntas completas siempre y cuando éstas permitieran obtener mejores resultados.

2.5.4.2 *Interfaces de lenguaje natural orientadas a Linked Data*

Al tiempo que la cantidad de información disponible en Linked Data crece día a día, también lo hace la necesidad de proveer acceso a esta información a usuarios tanto expertos como ocasionales. En este sentido, las interfaces de lenguaje natural, y en específico los sistemas de pregunta respuesta, están recibiendo especial atención (Lopez et al. 2011) debido a que proveen un mecanismo intuitivo de acceso a la información a la vez que oculta a los usuarios aspectos técnicos relacionados con la estructura subyacente de la información, los vocabularios y los diferentes lenguajes de consulta utilizados.

2.5.4.3 *Componente de un sistema de pregunta respuesta para Linked Data*

A pesar de que existen diferentes sistemas de pregunta respuesta con ciertas particularidades respecto a su arquitectura, existen funcionalidades de alto nivel y componentes que están presentes en la mayoría de ellos. Unger y colaboradores (Unger, Freitas, and Cimiano 2014) presentan los componentes de alto nivel de un sistema de pregunta respuesta para Linked Data, los cuales se presentan en la Figura 2-20.

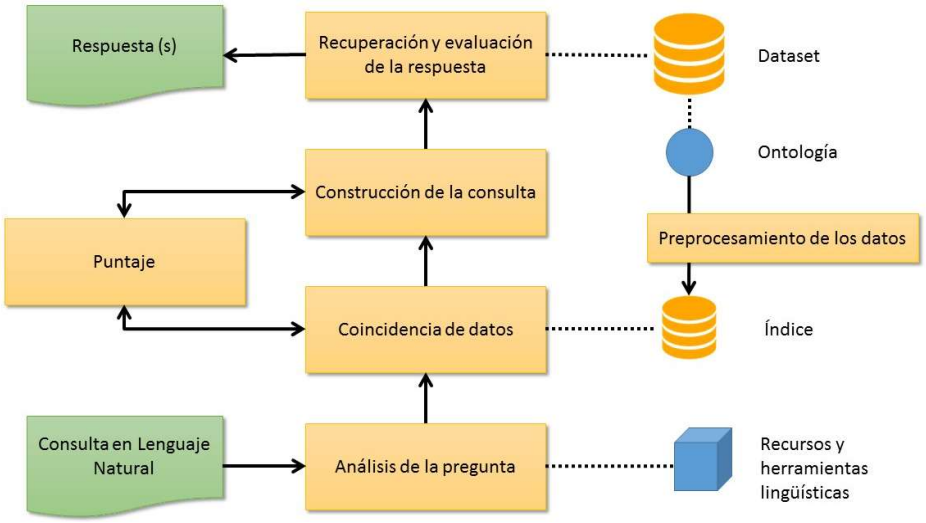

Figura 2-20. Componentes de alto nivel de un sistema de pregunta-respuesta para Linked Data.

A continuación, se describen cada uno de los componentes presentados en la figura anterior.

- **Preprocesamiento de los datos**. Este componente procesa previamente la información contenida en los conjuntos de datos con el objetivo de reducir el tiempo de respuesta del sistema. Por ejemplo, los NLI para Linked Data a

menudo utilizan un índice del conjunto de datos para relacionar expresiones en lenguaje natural con elementos del vocabulario.

- **Análisis de la pregunta**. El primer paso en los sistemas de pregunta respuesta consiste en el análisis lingüístico de la pregunta, el cual puede incluir el análisis sintáctico y semántico de la misma, a través de herramientas tales como etiquetadores POS, analizadores sintácticos y el reconocimiento de nombres de entidades (NER). El análisis de la pregunta puede comprender además la detección del tipo de pregunta, el foco de la misma y el tipo de respuesta esperada.
- **Coincidencia de datos**. Este componente relaciona los elementos contenidos en la pregunta del usuario expresada en lenguaje natural con los elementos contenidos en el conjunto de datos.
- **Construcción de la consulta**. Este componente transforma la pregunta del usuario en una consulta expresada en un lenguaje formal de consulta. Este proceso de traducción se basa en la información obtenida por los dos componentes antes mencionados.
- **Puntaje**. Tanto el componente de coincidencia de datos como el de construcción de la consulta pueden proveer diferentes elementos candidatos que necesitan ser puntuados y ordenados. Algunos criterios para llevar a cabo estas tareas pueden ser similitud de cadenas de texto, relaciones semánticas, frecuencia de términos y la coherencia de los candidatos y sus combinaciones con el esquema de datos.
- **Recuperación y evaluación de la respuesta**. Una vez construida la consulta, esta debe ser ejecutada sobre la base de conocimiento con el objetivo de extraer la respuesta correspondiente. Además, este componente puede llevar a cabo un proceso de evaluación de la respuesta cuyo objetivo sea verificar si el tipo de la respuesta obtenida coincide con el tipo de respuesta esperado.
- **Presentación de la respuesta**. Este elemento, como su nombre lo indica, es el encargado de presentar la respuesta al usuario. Sin embargo, su principal objetivo es representar la respuesta en un formato que sea comprensible por usuarios que no estén familiarizados con el lenguaje de la Web Semántica, ya sea a través de lenguaje natural o como un gráfico visual.

2.5.4.4 Campañas de evaluación de sistemas pregunta-respuesta para Linked Data

Existen algunas campañas de evaluación de sistemas de pregunta-respuestas orientados a Linked Data las cuales proveen puntos de referencia (*benchmarks*) para evaluar y comparar sistemas de este tipo en base a dos criterios principales:

- **Correctitud**. Esta se refiere a si la respuesta es correcta.
- **Completitud**. Este determina si la respuesta es completa, especialmente cuando la respuesta consiste de una lista o de una pregunta de definición.

Entre las campañas de evaluación más sobresalientes se encuentran:

- **QALD**. El objetivo de QALD (*Question Answering over Linked Data*) (Lopez et al. 2013) es evaluar sistemas de pregunta-respuesta multilingües. Este reto proporciona un conjunto de datos RDF y un conjunto de preguntas en lenguaje natural en diversos lenguajes con sus respectivas respuestas. De esta manera, los sistemas son evaluados contra este conjunto de preguntas a través de las métricas de precisión, exactitud y el valor-F.
- **INEX Linked Data**[2]. El objetivo de este reto es investigar técnicas de recuperación sobre una combinación de información textual y estructurada, con el objetivo de reducir la brecha entre las búsquedas a través de palabras clave y técnicas de razonamiento disponibles para la Web Semántica.
- **BioASQ**. Este reto (Tsatsaronis et al. 2012) incluye una amplia variedad de tareas de diferentes áreas tales como clasificación jerárquica de textos, máquinas de aprendizaje, recuperación de información, resumen de múltiples documentos y sistemas de pregunta-respuesta a partir de información textual y estructurada, todas ellas enfocadas a información biomédica.
- **CLEF Question Answering Track**[3]. Este reto reúne a los tres mencionados anteriormente con el objetivo de fomentar la visión general de que los sistemas de pregunta respuesta pueden encontrar, procesar e integrar información de todas las fuentes de datos disponibles sin importar la diversidad de estas.

[2] http://inex.mmci.uni-saarland.de/tracks/dc/index.html

[3] http://nlp.uned.es/clef-qa/

2.5.5 Esfuerzos de investigación para el desarrollo de NLIKB

Un gran número de investigadores han enfocado sus esfuerzos en desarrollar NLIKB enfocadas a diversos dominios y aplicando diversas técnicas de procesamiento de lenguaje. A continuación, se presentan las más sobresalientes.

Aqualog (Lopez, Pasin, and Motta 2005) es una interfaz que traduce la pregunta del usuario en una representación basada en tripletas. Esta representación se basa en la información sintáctica de la pregunta y un mapeo de los términos lingüísticos con elementos de la ontología. Aqualog ha sido evaluada en dos dominios diferentes (académico y vinos) donde, de acuerdo a los autores, el tiempo de configuración de cambio de dominio es corto. Una de las características sobresalientes de este trabajo es la incorporación de un mecanismo que le permite aprender la jerga del dominio y de esta manera relacionar estos términos con la ontología en preguntas futuras.

Querix (Kaufmann, Bernstein, and Zumstein 2006) emplea un conjunto reducido de herramientas de PLN ya que ante problemas de ambigüedades pregunta al usuario a través de diálogos de clarificación. Esta interfaz utiliza árboles sintácticos para relacionar los términos de la pregunta con elementos de la ontología a través de patrones heurísticos. Finalmente, los patrones identificados son transformados en un conjunto de consultas SPARQL ponderadas, de las cuales el usuario selecciona la más apropiada.

PANTO (C. Wang et al. 2007) (*Portable nAtural laNguage interface to Ontologies*) se basa en los sintagmas nominales identificados en la pregunta para formar representaciones intermedias denominadas *QueryTriples*. Estos componentes son mapeados a *OntoTriples* los cuales son representados en términos de la ontología. Finalmente, los *OntoTriples* junto con los modificadores identificados son transformados en consultas SPARQL.

ORAKEL (Cimiano et al. 2008) utiliza la ontología de la base de conocimiento para guiar el proceso iterativo de generación del lexicón llevado a cabo por un ingeniero con el objetivo de adaptar el sistema a un dominio en particular. Esta interfaz permite a este ingeniero agrupar diversas representaciones semánticas para expresar una pregunta. ORAKEL incorpora un motor de inferencia que ayuda a proporcionar una respuesta a través de la explotación de la información almacenada en la base de conocimientos. Para ello, el sistema traduce la pregunta del usuario en una forma lógica, en específico FOL (*First-Order Logic*), que pueda ser evaluada por dicho motor. Esta representación es traducida a un lenguaje de consulta para que sea ejecutada con respecto a la base de conocimiento.

QACID (Ferrández et al. 2009) (*Question Answering to CInema Domain*) analiza diversas preguntas para un dominio en específico y genera grupos que contienen diferentes formas de solicitar la misma información. Cada grupo es manualmente asociado con una consulta SPARQL. QACID integra un motor de vinculación que lleva a cabo deducciones semánticas para relacionar una nueva pregunta con el correspondiente grupo y, por ende, con su correspondiente consulta SPARQL, que le permitirá obtener la información de la base de conocimiento.

FREyA (Damljanovic, Agatonovic, and Cunningham 2010) utiliza la información descrita por la ontología para identificar POCs (*Potential Ontology Concepts*), es decir, las preguntas que contengan adjetivos, sustantivos y sintagmas nominales. Para cada POC sugiere elementos de la ontología con los que relacionarlos con base en el árbol sintáctico y en similitud de cadenas, obteniendo así elementos denominados OCs (*Ontology Concepts*). A partir de estas sugerencias el usuario selecciona la más apropiada, y así el sistema genera las correspondientes consultas SPARQL.

QALL-ME (Ferrández et al. 2011) considera el contexto espacial y temporal de la pregunta, el cual es determinado por algoritmos o puede ser indicado explícitamente por el usuario. La relación entre la pregunta y consultas formales la lleva a cabo mediante RTE (*Recognizing Text Entailment* – Reconocimiento de Vinculación Textual) cuya idea principal es predefinir un conjunto finito de patrones de preguntas con patrones de consulta. Donde los patrones de la pregunta contienen variables que corresponde a conceptos de la ontología.

SWIP (Pradel, Haemmerlé, and Hernandez 2012) permite expresar consultas a través de un lenguaje pivote que consiste básicamente en palabras clave que deben hacer referencia a elementos de la ontología, sean clases, relaciones o instancias. A partir de esta expresión, SWIP selecciona patrones predefinidos, pudiendo ser uno o más, por lo que el sistema las ordena de acuerdo a la cobertura del patrón. Estos patrones son mostrados en lenguaje natural para que el usuario seleccione el más adecuado para construir la consulta SPARQL.

SWSNL (Habernal and KonopíK 2013) integra un módulo de comprensión de lenguaje natural basado en un modelo estadístico y aprendizaje supervisado, para anotar una pregunta mediante MLE (*Maximun Likelihood Estimation*). Una vez obtenida la representación semántica, SWSNL utiliza un conjunto de reglas heurísticas que corresponden a fragmentos de consultas SPARQL a través de las cuales se formula la consulta completa.

REHABROBO-CNL (Dogmus, Patoglu, and Erdem 2014) permite al usuario expresar consultas de información a través de un lenguaje controlado. A partir de la consulta generada por el usuario, el sistema obtiene una representación basada en árboles denominada QDT (*Query Description Tree*). Una vez obtenido el árbol QDT, el sistema genera una representación basada en lógica descriptiva de cada uno de los conceptos identificados. Finalmente, cada concepto DL es transformado en un concepto SPARQL que permitirá construir la consulta general.

En (S. Li, He, and Wu 2014) los autores presentan una interfaz de lenguaje natural que genera una representación semántica en forma de árbol a partir de la consulta de información mediante un analizador de dependencias. Cada nodo del árbol semántico corresponde con un elemento de la ontología, sea este una clase, un individuo o una propiedad. De esta manera, el árbol semántico resultante es traducido a una consulta SPARQL compuesta de cláusulas WHERE para obtener la respuesta de la base de conocimientos.

AutoSPARQL (Unger et al. 2012) procesa la pregunta mediante un etiquetador POS a partir de las cuales se generan entradas léxicas mediante un conjunto de heurísticas. Estas entradas, junto con entradas léxicas predefinidas e independientes del dominio generan una representación semántica de la pregunta. Esta representación es transformada en una plantilla SPARQL a la que pueden corresponder diversas consultas SPARQL candidatas. Finalmente, las consultas con mayor ponderación, basada en similitud de cadenas y patrones de lenguaje natural, son ejecutadas contra la base de conocimiento.

MYAutoSPARQL (Sharef, Noah, and Murad 2015) genera tripletas lingüísticas a partir de la pregunta que están basadas en los conceptos de la ontología identificados en la pregunta. Esta interfaz establece diez variables observables que se relacionan con expresiones superlativas, de orientación, agregación, composición, comparativas y de negación, las cuales son importantes para la generación de la respectiva consulta en lenguaje formal. Además, propone doce patrones lingüísticos que permiten relacionar preguntas de tipo comparativas, superlativas, negación y agregación, con su respectiva consulta SPARQL.

TR-Discover (Song et al. 2015) emplea un conjunto de reglas gramaticales para analizar la pregunta y de esta manera obtener una representación FOL (*First Order Logic*) de la misma. La representación FOL es traducida a SPARQL en dos pasos. El primero consiste en generar un árbol a través de ANTLR (Parr 2013). El segundo paso consiste en recorrer ese árbol, y extraer las condiciones atómicas lógicas para crear las restricciones de consultas en SPARQL.

En (Hamon, Grabar, and Mougin 2016) se presenta una NLIKB para el dominio biomédico. Esta interfaz lleva a cabo la anotación semántica de la pregunta mediante patrones generados manualmente que reflejan las dependencias sintácticas básicas. Después, construye una abstracción de la pregunta a partir de reglas que permiten relacionar los elementos obtenidos en el paso anterior con entidades del contexto. Hecho esto, conecta las entidades identificadas para construir patrones de grafos a ser obtenidos. A partir de estos patrones la interfaz genera la consulta SPARQL que será enviada a un SPARQL-*endpoint* para obtener la respuesta.

2.5.5.1 Comparación de las NLIKB existentes en la literatura

Como se mencionó en la sección 2.5.4.4, existen diversas campañas de evaluación de sistemas de pregunta-respuesta que proveen puntos de referencia para evaluar y comparar este tipo de sistemas en base a su correctitud y completitud. Sin embargo, gran parte de los trabajos presentados en la sección anterior no consideraron esas campañas dentro de sus experimentos de evaluación. Esto se pudo deber a los tiempos en que dichas campañas fueron promovidas o, en mayor medida, a que el lenguaje y dominio al cual se enfocan las investigaciones varían a los establecidos en las campañas. Por tales motivos, es difícil determinar cuál de los trabajos citados provee los mejores resultados. Sin embargo, en esta tesis doctoral se emplean seis criterios a través de los cuales se lleva a cabo una comparativa de las NLIKB descritas en la sección anterior, que son:

- Tipo de NLI. Este criterio especifica si la interfaz acepta preguntas expresadas completamente en lenguaje natural o a través de un lenguaje controlado.
- Portabilidad. Este criterio determina si la NLIKB puede ser aplicada en múltiples dominios.
- Dominio. Este criterio indica el dominio al cual están orientadas las interfaces y, por ende, en el cual se llevaron a cabo los experimentos de evaluación.
- Técnicas PLN utilizadas. Este criterio se refiere a las técnicas PLN utilizadas por la NLIKB para la interpretación de la pregunta del usuario.
- Lenguaje. Este criterio hace referencia al lenguaje para el cual fue desarrollada la NLIKB.
- Enfoque. Este criterio se refiere al método o técnica utilizada para representar la pregunta en lenguaje natural previo a la generación de consultas en un lenguaje formal.

En la Tabla 2-5 se presenta un resumen de la comparación llevada a cabo a partir de las características antes mencionadas. A continuación, se discute a detalle los resultados presentados en dicha tabla.

2.5.5.1.1 Tipo de NLI

En lo que se refiere al tipo de NLI bajo el cual han sido desarrolladas las NLIKB analizadas en este trabajo, nos encontramos con tres diferentes enfoques: (1) lenguaje natural, el cual acepta preguntas expresadas a través de términos libremente elegidos; (2) lenguaje controlado, que representa un subconjunto del lenguaje natural con un vocabulario y gramática restringida; y (3) lenguaje pivote, un enfoque inspirado por la consulta por palabras clave en el cual se construyen consultas en términos de gráficos conceptuales. Como se puede observar en la Tabla 2-5 la mayoría de los trabajos descritos emplean el enfoque basado en lenguaje natural, con excepción de SWIP (Pradel, Haemmerlé, and Hernandez 2012) y REHABROBO-CNL (Dogmus, Patoglu, and Erdem 2014) quienes utilizan un enfoque basado en lenguaje pivote y lenguaje controlado respectivamente. Estos últimos trabajos proveen interfaces que incluyen elementos tales como listas desplegables, campos de autocompletado y vistas de acordeón, entre otros. Este tipo de elementos pueden afectar la facilidad, naturalidad y eficacia de los usuarios para expresar sus preguntas dentro de las limitaciones impuestas por el sistema. En este sentido, la aproximación presentada en este trabajo de tesis adopta un enfoque de lenguaje natural a través del cual los usuarios puedan expresar sus preguntas utilizando solo palabras clave o mediante una pregunta expresada en términos libremente elegidos, evitando con ello la necesidad de aprender cualquier lenguaje adicional o el uso de un nuevo software.

2.5.5.1.2 Portabilidad

Respecto a la característica de portabilidad ofrecida por las NLIKB analizadas, se puede apreciar una clara tendencia hacia el desarrollo de interfaces que puedan funcionar para distintas bases de conocimiento correspondientes a diferentes dominios. Sin embargo, trabajos tal como REHABROBO-CNL (Dogmus, Patoglu, and Erdem 2014) establece una gramática y patrones de consulta SPARQL un tanto ligados a la ontología del dominio. Por su parte, QACID (Ferrández et al. 2009), establece un conjunto de patrones de preguntas en lenguaje natural compatibles únicamente con la fuente de información subyacente. Además, QACID demanda la recopilación de preguntas del dominio específico, tarea que demanda un esfuerzo considerable y la habilidad para asegurar una relación significativa entre las preguntas recopiladas y el dominio. Trabajos tales como Aqualog (Lopez, Pasin, and Motta 2005) y ORAKEL (Cimiano et al. 2008) demandan un proceso de configuración

para que puedan ser aplicados a otros dominios. En el caso de Aqualog, este requiere la generación de nuevas tripletas, denominadas *Onto-triple*, con el fin de relacionar la pregunta con los elementos de la ontología. Por su parte, ORAKEL demanda la generación manual de un lexicón, hecho que, aunque permite controlar directamente la calidad y la cobertura del dominio, demanda un considerable esfuerzo por parte del experto. Dicho esto, la NLIKB propuesta en este trabajo de tesis opta por un enfoque que le otorgue un nivel alto de independencia del dominio. Para ello, esta interfaz contempla la generación de un lexicón de manera automática a partir de los elementos descritos en la ontología, de manera más específica, a través de la extracción de la propiedad *rdfs:label* de las clases y propiedades, la cual provee una versión entendible por el humano del nombre del recurso. Además, nuestra aproximación integra técnicas de PLN tales como el reconocimiento de entidades nombradas, lematización y el uso de sinónimos para relacionar términos de la pregunta con elementos de la base de conocimiento, lidiando con palabras declaradas en diferentes tiempos verbales, en su forma plural, y con palabras que no están descritas literalmente en la base de conocimiento, pero que son sinónimos de algunos de sus elementos. Finalmente, el presente trabajo contempla el uso de un conjunto de plantillas de consulta SPARQL que son independientes del dominio, por lo que la traducción de la pregunta a una consulta en lenguaje formal, no estará ligada a elementos específicos de la base de conocimiento.

2.5.5.1.3 Dominio

Las NLIKB analizadas abordan dominios específicos que van desde cine y música, hasta el de robótica de rehabilitación y biomedicina. Sin embargo, trabajos tales como AutoSPARQL (Unger et al. 2012) y FREyA (Damljanovic, Agatonovic, and Cunningham 2010) abordan un dominio más general que es el de DBpedia, un esfuerzo comunitario para extraer conocimiento estructurado y multilingüe de Wikipedia y hacerlo disponible de manera gratuita en la Web a través de tecnologías de la Web Semántica y Linked Data (Lehmann et al. 2015). En este sentido, la NLIKB presentada en este trabajo intenta proveer una manera intuitiva de acceso a información publicada bajo el enfoque Linked Data, el cual ha sido adoptado por un gran número de individuos y organizaciones para publicar sus datos. Lo antes dicho se ve reflejado en la Web de Datos la cual contiene billones de declaraciones RDF que cubren diversos tópicos. Concretamente, la NLIKB propuesta ha sido evaluada con los conjuntos de datos de DBpedia en su versión en inglés y en MusicBrainz, una fuente de información para el dominio de la música.

2.5.5.1.4 Técnicas PLN utilizadas

Las técnicas de PLN utilizadas por las NLIKB incluyen la tokenización, etiquetado POS, reconocimiento de entidades nombradas, y el uso de sinónimos. Trabajos tales como Aqualog (Lopez, Pasin, and Motta 2005), Querix (Kaufmann, Bernstein, and Zumstein 2006), PANTO (C. Wang et al. 2007), ORAKEL (Cimiano et al. 2008) y AutoSPARQL (Unger et al. 2012) llevan a cabo el análisis sintáctico de la pregunta, bajo el enfoque de constituyentes (ver sección 2.3.3.4) el cual representa poca información semántica (S. Li, He, and Wu 2014) comparado con el enfoque de análisis de dependencias, que de acuerdo con (Kübler, McDonald, and Nivre 2009) es más adecuada para lenguajes con un orden de palabras libre y flexible. Con base en lo antes dicho, nuestra aproximación procesa la pregunta bajo un enfoque de análisis de dependencias, cuyos árboles generados guardan cierto grado de similitud con respecto a las tripletas RDF que forman el patrón de grafos a ser obtenido de la base de conocimiento. En otras palabras, cada nodo del árbol de dependencias correspondería con una clase, instancia o valor de una propiedad, que, dentro de la tripleta RDF representan el sujeto y objeto del predicado o el dominio y rango de la propiedad. Mientras que el arco entre los nodos corresponde con una propiedad (*object* o *datatype*) de la ontología, es decir, al predicado de la tripleta. Este enfoque ayuda en gran medida a la transformación de la pregunta en lenguaje natural a una consulta SPARQL.

2.5.5.1.5 Lenguaje

En cuanto a lenguaje se refiere, la mayoría de las interfaces analizadas están orientadas al lenguaje inglés, exceptuando QACID (Ferrández et al. 2009) que se enfoca en el español, al presentado en (S. Li, He, and Wu 2014) que se enfoca en el chino, y por último QALL-ME (Ferrández et al. 2011) y SWSNL (Habernal and KonopíK 2013) que proveen un soporte multilingüe, en específico para el inglés, alemán, italiano y español, en el caso de QALL-ME, e inglés y checo en el caso de SWSNL. En este apartado, la presente NLIKB se enfoca en el idioma inglés como una primera aproximación, por nuestra parte, al desarrollo de este tipo de interfaces para Linked Data, además de que este idioma se encuentra dentro de los más hablados en el mundo.

2.5.5.1.6 Enfoque

Como ya se mencionó, este criterio se refiere al método o técnica utilizada para representar la pregunta previo a la generación de consultas en lenguaje formal. Como se puede observar en la Tabla 2-5, trabajos tales como Aqualog (Lopez, Pasin, and Motta 2005), PANTO (C. Wang et al. 2007) y FREyA (Damljanovic, Agatonovic, and Cunningham

2010) utilizan un enfoque basado en tripletas cuyos elementos (sujeto – predicado – objeto) son obtenidos mediante el proceso de análisis sintáctico. Por otro lado, interfaces tales como TR-Discover (Song et al. 2015), AutoSPARQL (Unger et al. 2012), MYAutoSPARQL (Sharef, Noah, and Murad 2015) y el presentado en (Hamon, Grabar, and Mougin 2016) emplean patrones para mapear la pregunta y finalmente obtener las correspondientes consultas formales. En este trabajo de tesis se propone un novedoso enfoque basado en un modelo ontológico independiente del dominio que permite representar tanto la estructura sintáctica de la pregunta, como la información relacionada con el contexto de la misma. En cuanto a la estructura sintáctica, la ontología contempla elementos tales como verbos, modificadores, y entidades nombradas, a los que se les suma información tal como su lema, categoría gramatical y sus sinónimos. Además, contempla las relaciones sintácticas entre dichos elementos que son obtenidas a través del análisis de dependencias del que se habló previamente. Con respecto a la información del contexto, la ontología define elementos que permiten referenciar a aquellos elementos de la base de conocimiento, sean de TBox o ABox, con los que los términos de la pregunta tengan una relación, ya sea a través de similitud de cadenas o pertenezcan al dominio y rango de propiedades identificadas. A partir de toda la información almacenada en el modelo, el sistema es capaz de determinar el tipo de pregunta provista, el tipo de respuesta esperada por el usuario, así como generar la correspondiente consulta SPARQL, a través de un conjunto de plantillas independientes del dominio.

Tabla 2-5. Tabla comparativa de interfaces de lenguaje natural para bases de conocimiento.

Nombre	Enfoque	Tipo de NLI	Portabilidad	Dominio	Técnicas PLN utilizadas	Lenguaje
Aqualog (Lopez, Pasin, and Motta 2005)	Basada en tripletas	Lenguaje Natural	Independiente del dominio (Requiere adaptación de módulos)	Académico y Enología	Análisis sintáctico, expresiones regulares, similitud de cadenas, sinónimos	Inglés
Querix (Kaufmann, Bernstein, and Zumstein 2006)	Diálogos de clarificación	Lenguaje Natural	Independiente del dominio	Información geográfica	Árboles sintácticos, sinónimos	Inglés
PANTO (C. Wang et al. 2007)	Basada en tripletas	Lenguaje natural	Independiente del dominio	Información geográfica, Información de restaurantes, Anuncios de empleos	Árboles sintácticos, sinónimos, similitud de cadenas	Inglés
ORAKEL (Cimiano et al. 2008)	Basado en patrones	Lenguaje natural	Independiente del dominio (Demanda la participación de personas para la generación del lexicón)	Base de conocimiento con hechos relacionados a Alemania y publicaciones de la librería digital de British Telecom	Análisis sintáctico, sinónimos	Inglés
QACID (Ferrández et al. 2009)	Agrupación de preguntas	Lenguaje natural	Específico del dominio	Cine	Análisis morfológico, NER	Español
FREyA (Damljanovic, Agatonovic, and Cunningham 2011)	Basada en tripletas	Lenguaje natural	Independiente del dominio	Información geográfica, DBpedia	Análisis sintáctico, sinónimos, similitud de cadenas	Inglés
QALL-ME (Ferrández et al. 2011)	Reconocimiento de vinculación textual	Lenguaje natural	Independiente del dominio	Eventos cinematográficos en el dominio del turismo	Reconocimiento de vinculación textual	Multilingüe (inglés, alemán, italiano, español)

Nombre	Enfoque	Tipo de NLI	Portabilidad	Dominio	Técnicas PLN utilizadas	Lenguaje
SWIP (Pradel, Haemmerlé, and Hernandez 2012)	Basada en patrones	Lenguaje pivote	Específico del dominio	Cine, música	Coincidencia de patrones	Inglés
SWSNL (Habernal and KonopíK 2013)	Aprendizaje supervisado	Lenguaje natural	Independiente del dominio	Opciones de alojamiento, transporte público	Etiquetado POS, NER, análisis semántico, similitud de cadenas	Multilingüe (inglés y checo)
REHABROBO-CNL (Dogmus, Patoglu, and Erdem 2014)	Árbol de descripción de la consulta	Lenguaje controlado	Específico del dominio	Robótica de rehabilitación	Lenguaje natural controlado	Inglés
(S. Li, He, and Wu 2014)	Árbol semántico	Lenguaje natural	No especificado	Música	Tokenización, etiquetado POS, análisis sintáctico, sinónimos	Chino
AutoSPARQL (Unger et al. 2012)	Patrones de consulta	Lenguaje natural	Independiente del dominio	DBpedia	Etiquetador POS, análisis sintáctico, NER, sinónimos	Inglés
MYAutoSPARQL (Sharef, Noah, and Murad 2015)	Patrones	Lenguaje natural	No especificado	Información geográfica	Tokenización, etiquetador POS	Inglés
TR-Discover (Song et al. 2015)	Patrones	Lenguaje natural	Independiente del dominio	Farmacéutico, Biomédico	Reglas gramaticales libres del contexto	Inglés
(Hamon, Grabar, and Mougin 2016)	Patrones	Lenguaje natural	No especificado	Biomédico	Tokenización, etiquetador POS, lematización	Inglés
Nuestra aproximación	**Basada en ontologías**	**Lenguaje natural**	**Independiente del dominio**	**DBpedia, MusicBrainz**	**Lematización, NER, análisis de dependencias, sinónimos**	**Inglés**

2.6 Conclusiones

En este capítulo se presentó un detallado estudio del arte de la Web Semántica, PLN e interfaces de lenguaje natural, tecnologías centrales en este trabajo de tesis doctoral. En lo que respecta a la Web Semántica, se abordó el campo de las ontologías, y más concretamente, se describieron sus elementos fundamentales mediante los cuales es posible representar formalmente estructuras de conocimiento, a saber: conceptos, atributos, individuos, relaciones y axiomas. En este mismo contexto, se presentó el lenguaje de consulta SPARQL, qué es utilizado por la interfaz aquí propuesta para recuperar información de la base de conocimiento que permita dar respuesta a la pregunta provista por el usuario. Además, se detalló la estructura de una consulta SPARQL, información que será de vital importancia para comprender el proceso de generación de consultas implementado en nuestro trabajo. Finalmente, se presentó Linked Data, dominio hacia el cual va dirigida nuestra propuesta.

En cuanto al PLN, destaca la descripción provista de los niveles de PLN, y en específico, del análisis sintáctico. En cuanto a este último, se presentaron dos enfoques para llevarlo a cabo, que son el análisis de constituyentes y el análisis de dependencias. En la explicación provista se aprecia que el análisis de dependencias guarda una estrecha relación con el formato de las tripletas RDF que forman el patrón de grafos a ser obtenido de la base de conocimiento, lo que ayuda a la transformación de la pregunta en lenguaje natural a consultas en lenguaje formal, en este caso SPARQL. Con respecto a los sistemas de búsqueda de respuestas, destaca la descripción provista de la arquitectura típica de un sistema de este tipo basado en texto libre, la cual se compone de tres módulos principales: el módulo de análisis de la pregunta, el módulo de recuperación de información y el módulo de extracción de la respuesta. Además, la sección relacionada con la clasificación de la pregunta resultó de gran importancia, pues la interfaz aquí propuesta adopta una de las taxonomías de preguntas descritas en esa sección.

Finalmente, se abordó el campo de las interfaces de lenguaje natural, donde se presentan arquitecturas que sirvieron como base para el establecimiento de la arquitectura correspondiente a nuestro trabajo de tesis doctoral. Además, se presentó un análisis de los esfuerzos de investigación orientados al desarrollo de interfaces de lenguaje natural para bases de conocimiento. En este análisis se describen las características más sobresalientes de nuestro trabajo con respecto los demás, de las cuales destaca el uso de análisis sintáctico basado en relaciones de dependencia y el establecimiento de un modelo ontológico independiente del dominio para la representación de la estructura y contexto de la pregunta.

Capítulo 3. Objetivo de esta tesis doctoral

3.1 Motivación

El exponencial crecimiento de información disponible en la Web e intranets ha dado paso a la necesidad de contar con mecanismos capaces de procesar y comprender dicha información y con ello resolver necesidades específicas. Para ello, esta información debe contar con una estructura que permita a las computadoras procesarla en mucho menor tiempo y sobre todo de manera significativa para el humano. Ante esta situación surge la Web Semántica, la cual añade a la información de la Web actual una estructura bien definida a través de un conjunto de atributos, valores y relaciones, para lo cual emplea una de las tecnologías más sobresalientes de su arquitectura, que son las ontologías.

Existe un gran número de individuos y organizaciones de diversos dominios que han adoptado el enfoque basado en ontologías para publicar su información. Entre estos dominios podemos mencionar el de medicina (Ruiz-Martínez et al. 2012), finanzas (Salas-Zárate et al. 2016), servicios en la nube (Rodríguez-García et al. 2014) y sistemas de recomendación (Colombo-Mendoza et al. 2015), entre muchos otros. Hoy en día existen diversos mecanismos para acceder a este tipo de información. Uno de los enfoques más extendidos consiste en el uso de un lenguaje formal de consulta tal como SPARQL. Sin embargo, este enfoque resulta complicado para usuarios ocasionales (Kaufmann, Bernstein, and Fischer 2007), ya que demanda que el usuario cuente con un alto nivel de conocimiento en tecnologías como RDF y expresiones de lenguaje de consulta, así como el conocimiento previo de la estructura de datos de la base de conocimiento subyacente.

La necesidad de hacer accesible la información de la Web Semántica a todo tipo de usuarios, sean expertos u ocasionales, demanda el desarrollo de nuevos mecanismos de recuperación de información. En este sentido, el paradigma de recuperación de información basado en lenguaje natural es generalmente considerado como el más intuitivo desde un punto de vista de uso (Cimiano et al. 2008), ya que permite ocultar al usuario la formalidad de una base de conocimientos basada en ontologías así como el lenguaje de consulta ejecutable, permitiendo a los usuarios emplear todo el poder comunicativo del lenguaje natural en lugar de verse forzados a utilizar un lenguaje limitado. Además, de acuerdo con (Elbedweihy, Wrigley, and Ciravegna 2012), este

paradigma ofrece una mejor experiencia al usuario que enfoques tales como la búsqueda basada en formularios.

Actualmente, existen diversos esfuerzos de investigación por proveer interfaces de lenguaje natural para bases de conocimiento de diversos dominios y lenguajes. Algunas de estas propuestas, permiten expresar al usuario consultas de información mediante un vocabulario y una gramática restringida, denominado vocabulario controlado o a través de un enfoque inspirado en la búsqueda por palabras clave. Por otra parte, existen trabajos que proveen una interfaz de lenguaje natural cuya adaptación a un nuevo dominio demanda la participación de un experto en tareas tales como la recolección de preguntas del dominio o la generación de un lexicón. En cuanto a técnicas de PLN, destaca una tendencia hacia el uso de análisis sintáctico basado en constituyentes, el cual, de acuerdo con (S. Li, He, and Wu 2014) representa menos información semántica comparado con el enfoque basado en dependencias. Finalmente, una gran cantidad de las interfaces existentes en la literatura adoptan el enfoque basado en tripletas y el basado en patrones para representar la consulta de información de manera formal. Dicha representación se traduce a una consulta en lenguaje formal mediante la cual recuperar la información solicitada de la base de conocimiento.

En este trabajo de tesis doctoral se proponen una solución basada en lenguaje natural y ontologías para la consulta y recuperación de información de bases de conocimiento. La solución propuesta aprovecha la tecnología de la Web semántica de dos maneras. La primera de ellas consiste en procesar la ontología de la base de conocimiento para generar un vocabulario que le permita conocer los términos comúnmente utilizados por los usuarios en el dominio modelado, y de esta manera poder relacionar los elementos contenidos en la pregunta del usuario con aquellos descritos en la base de conocimiento. La segunda, consiste en utilizar un modelo ontológico independiente del dominio para representar tanto la estructura sintáctica de la pregunta, como el contexto de la misma en términos de la base de conocimiento. Para obtener tal representación, se aplican técnicas de PLN, entre las que destaca el análisis de dependencias. A través de esta técnica se obtiene una representación sintáctica de la pregunta que guarda una estrecha relación con las tripletas RDF que forman el patrón de grafos a ser obtenido de la base de conocimiento. Este hecho ayudará en gran medida en generar las consultas SPARQL respectivas con base en un conjunto de plantillas de consulta independientes del dominio. Por último, cabe mencionar que la interfaz de lenguaje natural propuesta en este trabajo se compone de módulos independientes de la base de conocimiento, con el objetivo de reducir el esfuerzo de crear una interfaz de lenguaje natural para una aplicación dada.

3.2 Objetivos

El objetivo principal de esta tesis es desarrollar soluciones basadas en tecnologías de procesamiento del lenguaje natural y Web Semántica que permitan reducir la brecha existente entre el usuario y las bases de conocimiento a través del lenguaje natural. Para lograr lo antes mencionado es necesario cumplir con los siguientes objetivos específicos:

- Diseño e implementación de un modelo ontológico independiente del dominio para la representación de la estructura sintáctica y contexto de la pregunta en lenguaje natural.
- Diseño de la arquitectura de una interfaz de lenguaje natural para bases de conocimiento basadas en ontologías.
- Diseño e implementación de un proceso de análisis de preguntas basado en técnicas de procesamiento de lenguaje natural y Web semántica.
- Diseño e implementación de un proceso de generación de consultas SPARQL a partir de una representación semántica de la pregunta en lenguaje natural.
- Validación de los resultados obtenidos por medio de bases de conocimiento basadas en Linked Data.

3.3 Metodología

La metodología seguida durante el desarrollo de este proyecto de tesis se divide en cuatro partes principales: en la primera se desarrolla un estudio del estado del arte que permita conocer los esfuerzos de investigación más relevantes dentro de las áreas de interés del proyecto; la segunda parte consiste en la formalización de los métodos propuestos en este trabajo para interfaces de lenguaje natural enfocadas a bases de conocimiento; en la tercera etapa, se lleva a cabo la implementación de los métodos propuestos; finalmente, en la cuarta parte se lleva a cabo la validación de la propuesta en las bases de conocimiento de DBpedia y MusicBrainz.

3.3.1 Estudio del estado del arte.

En esta parte de la metodología se llevó a cabo un análisis de todos aquellos desarrollos de última tecnología realizados en los contextos de Web Semántica, PLN e interfaces de lenguaje natural, principales tecnologías involucradas en este trabajo de tesis.

- **Web Semántica**. Estudio de los componentes de la arquitectura de la Web Semántica enfatizando las ontologías. Respecto a esta tecnología, se analizan sus tipos, elementos y los diferentes lenguajes utilizados para su desarrollo.

Finalmente, se aborda el enfoque de Linked Data, un conjunto de buenas prácticas para publicar y enlazar datos estructurados en la Web.

- **Procesamiento de lenguaje natural**. Análisis de los diferentes niveles de PLN, así como las diversas aplicaciones de esta tecnología.
- **Interfaces de lenguaje natural**. Análisis de las principales arquitecturas utilizadas en el desarrollo de este tipo de aplicaciones y de los esfuerzos de investigación más sobresalientes enfocados en proveer soluciones de este tipo.

3.3.2 Formalización de la propuesta.

Esta parte de la metodología contempla el desarrollo de una ontología enfocada a la descripción de la estructura sintáctica de la pregunta, así como de su contexto en términos de la base de conocimiento del dominio. Toda la información a almacenar en esta ontología será obtenida mediante el proceso de análisis de preguntas en lenguaje natural diseñado en este trabajo, el cual estará basado en técnicas tales como el análisis de dependencias, lematización y la búsqueda de sinónimos. Finalmente, la creación de consultas SPARQL a partir de una representación semántica de la pregunta, permitirá generar el patrón de grafo a ser obtenido de la base de conocimiento para dar respuesta a la pregunta provista.

3.3.3 Implementación de la propuesta.

Esta etapa consiste en la implementación de la interfaz de lenguaje natural propuesta por medio de herramientas de PLN y Web Semántica. La interfaz implementada agrupa el modelo ontológico independiente del dominio para la representación de preguntas en lenguaje natural, el proceso de análisis de preguntas basado en técnicas de PLN y el proceso de generación de consultas SPARQL a partir de la representación semántica de la pregunta en lenguaje natural.

3.3.4 Validación de la propuesta

Finalmente, esta parte de la metodología contempla la validación de la interfaz de lenguaje natural implementada en bases de conocimiento basadas en Linked Data. En concreto, la interfaz desarrollada se aplicará sobre el conjunto de datos de DBpedia, un esfuerzo comunitario para extraer conocimiento estructurado y multilingüe de Wikipedia, y el conjunto de datos de MusicBrainz, una fuente de información ampliamente utilizada en el dominio de la música.

Capítulo 4. Interfaz de lenguaje natural para bases de conocimiento basadas en ontologías

4.1 Introducción

La Web semántica aporta un significado semántico a la información con el objetivo de que esta sea fácil de utilizar y sobre todo que esta pueda ser explotada por aplicaciones avanzadas en tareas de búsqueda, intercambio e integración de información. Las ontologías representan la tecnología semántica base de la Web semántica, pues es a través de esta que los usuarios definen una vista común, compartible y reutilizable del dominio de aplicación a través de un vocabulario formal. Este enfoque de representación del conocimiento facilita en gran medida el uso y acceso a la información. Esto ha ocasionado un exponencial crecimiento del número de individuos y organizaciones de diversos dominios que han adoptado tal enfoque para publicar su información. Quizá el ejemplo más representativo de este tipo de información es la denominada Web de datos, la cual sigue los principios de Linked Data para publicar y enlazar entre sí datos estructurados y distribuidos en la Web.

Hoy en día, el acceso a bases de conocimiento basadas en ontologías se lleva a cabo generalmente a través de un lenguaje formal de consulta como SPARQL. Este mecanismo restringe el acceso a la información a solo aquellos usuarios con conocimientos y experiencia en tecnologías tales como RDF y en la generación de consultas en un lenguaje formal de consulta. Además, requiere conocer la estructura de datos de la base de conocimiento en cuestión. Ante esta situación, el paradigma de acceso a la información basado en lenguaje natural es generalmente considerado el más intuitivo desde un punto de vista de uso (Cimiano et al. 2008), incluso ofrece una mejor experiencia de usuario que enfoques tales como la búsqueda basada en formularios (Elbedweihy, Wrigley, and Ciravegna 2012).

En este capítulo se presenta nuestro esfuerzo de investigación por proveer una interfaz de lenguaje natural que busca reducir la brecha existente entre los usuarios, ocasionales o expertos, y las bases de conocimiento semánticas. Este capítulo explica el funcionamiento

de la interfaz a través de la descripción de cada uno de los componentes que conforman su arquitectura, los cuales combinan tecnologías de PLN y Web Semántica.

La interfaz descrita en este apartado recibe una pregunta expresada en lenguaje natural a partir de la cual genera una consulta SPARQL para recuperar la información pertinente de la base de conocimiento. El proceso de generación de la consulta SPARQL se basa en un modelo ontológico independiente del dominio que recoge toda la información que describe la estructura sintáctica y el contexto de la pregunta en términos de la base de conocimiento de la aplicación. Esta información es obtenida a través de un proceso de análisis de la pregunta basado en técnicas de PLN y Web Semántica. A continuación, se describe a detalle cada uno de los módulos que conforman la arquitectura de la interfaz.

4.2 Arquitectura

La arquitectura de la NLIKB propuesta en este trabajo de tesis toma como base dos arquitecturas propuestas en la literatura. Por un lado, la arquitectura genérica para NLI enfocada a aplicaciones funcionales propuesta por Smith (R. W. Smith 2006), de la cual considera los módulos de comprensión del lenguaje natural y el módulo de procesamiento del dominio. Por otro lado, los componentes de alto nivel de un sistema de pregunta respuesta para Linked Data propuestos por Unger y colaboradores (Unger, Freitas, and Cimiano 2014), de los cuales toma en cuenta el módulo de coincidencia de datos y de construcción de la consulta. De esta manera, la arquitectura resultante se compone de cuatro módulos principales que son: preprocesamiento de la base de conocimiento, procesamiento de la pregunta, clasificación de la pregunta, generación y ejecución de la consulta en lenguaje formal. A estos módulos, se añade un modelo ontológico independiente del dominio que describe tanto la estructura de la pregunta provista por el usuario, como el contexto de la misma en términos de la base de conocimiento del dominio. Un esquema general de esta arquitectura se muestra en la Figura 4-1.

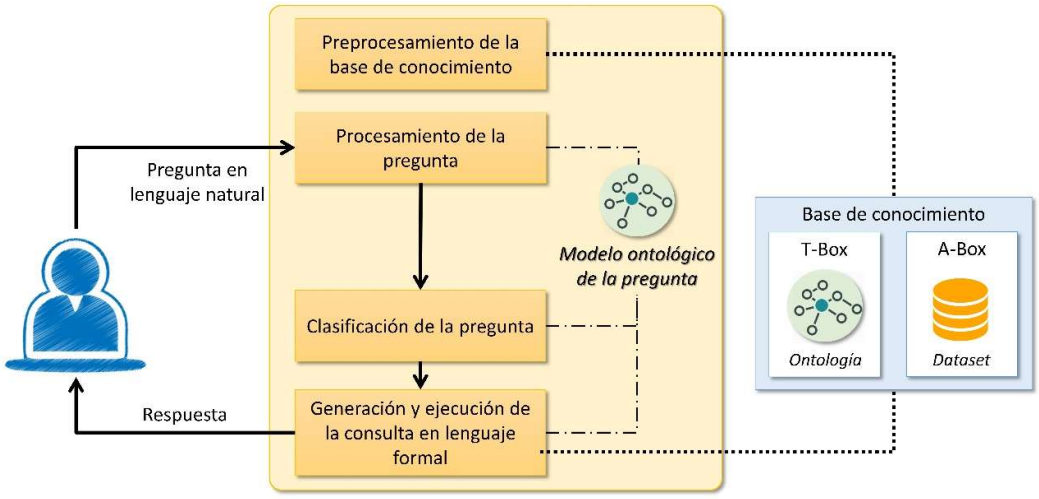

Figura 4-1. Arquitectura de la NLIKB.

De manera general, la NLIKB propuesta en esta tesis doctoral funciona de la siguiente manera. Antes de que el usuario interactué con la interfaz, la NLIKB necesita contar con un vocabulario que le permita relacionar los elementos contenidos en la pregunta del usuario con aquellos descritos en el TBox de la base de conocimiento. Este vocabulario se obtiene por el módulo de preprocesamiento de la base de conocimiento mediante la extracción de los conceptos y las propiedades descritas en la ontología de la base de conocimiento. Hecho esto, cuando el usuario provee una pregunta expresada en lenguaje natural, el módulo de procesamiento de la pregunta la analiza con el objetivo de obtener información de interés referente a su estructura y contexto en términos de la base de conocimiento. La información recopilada es almacenada en el modelo ontológico de la pregunta. A partir de esta información, el sistema determina el tipo de pregunta provista por el usuario apoyado de un conjunto de reglas establecidos. Además, el sistema identifica el tipo de respuesta esperada por el usuario, de acuerdo a una clasificación de preguntas y respuestas previamente establecidas. El siguiente paso consiste en generar la correspondiente consulta en un lenguaje formal (SPARQL), a partir de un conjunto de plantillas independientes del dominio. Esta consulta es ejecutada en la base de conocimiento para obtener la información que responda a la pregunta proporcionada y finalmente mostrar los resultados al usuario.

La interfaz de lenguaje natural propuesta en esta tesis provee las siguientes contribuciones al dominio de NLIKB:

- Un modelo ontológico independiente del dominio que permite representar tanto la estructura de la pregunta como el contexto de la misma. Gracias a esto, el

sistema puede determinar el tipo de respuesta esperado por el usuario lo que reduce en gran medida el espacio de búsqueda.

- Una clasificación de preguntas adaptada al contexto de bases de conocimiento.
- Un conjunto de plantillas de tripletas RDF independientes del dominio que permiten generar consultas en un lenguaje formal (SPARQL) a partir de la información descrita por el modelo ontológico de la pregunta.

En las siguientes secciones se describe de manera más detallada cada uno de los módulos que componen la NLIKB propuesta en esta tesis.

4.2.1 Modelo ontológico de la pregunta

De acuerdo con (R. W. Smith 2006) uno de los principios para el diseño de NLI consiste en desarrollar representaciones semánticas del sentido de la entrada del usuario, la cual sea consistente con la representación semántica usada en la aplicación del dominio. En este sentido, se propone un modelo ontológico independiente del dominio que describe tanto la estructura sintáctica de la pregunta como la información relacionada con el contexto de la misma. La estructura de la pregunta será representada a través de conceptos tales como verbos, modificadores, entidades nombradas, entre otros, así como las relaciones sintácticas entre ellos. Con respecto a la información del contexto, el modelo ontológico describirá los elementos de la base de conocimiento (individuos, conceptos y relaciones) con los cuales tengan relación los términos contenidos en la pregunta. Esta información se obtiene mediante técnicas de PLN y Web Semántica implementadas por el módulo de procesamiento de la pregunta que será descrito en la sección 4.2.3. Esta información será utilizada por los subsecuentes módulos que componen la NLIKB para determinar el tipo de pregunta provista por el usuario, el tipo de respuesta esperada y generar las correspondientes consultas en un lenguaje formal. En la Figura 4-2 se muestra un extracto del modelo ontológico de la pregunta propuesto en este trabajo.

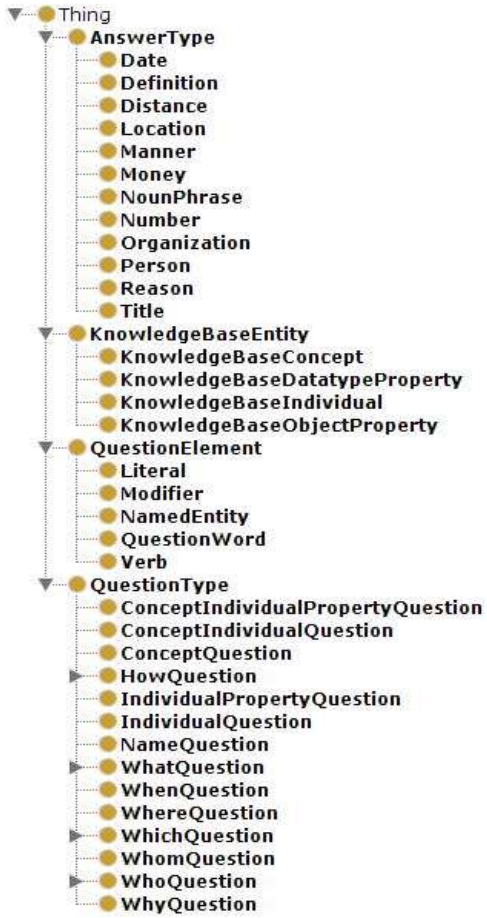

Figura 4-2. Modelo ontológico de la pregunta.

Los elementos del modelo ontológico que permiten describir la estructura sintáctica de la pregunta son:

- ***question:QuestionElement***. Este representa todo aquel elemento contenido en la pregunta provista por el usuario. Los individuos de este concepto tienen las siguientes propiedades sobresalientes:
    - ***question:originalContent***. Esta propiedad representa el término tal cual aparece en la pregunta.
    - ***question:lemma***. Esta propiedad representa el lema del término contenido en la pregunta. El lema de una palabra es una serie de caracteres que forman una unidad semántica y que pueden constituir una entrada en un diccionario tradicional. Por ejemplo, para las palabras *dije*, *diré* y *dijéramos* el lema, es *decir*. Esta propiedad es de suma importancia ya que es posible que la descripción del elemento de la ontología (provista a través de la propiedad *rdfs:label*) no coincida con la

palabra contenida en la pregunta del usuario, ya que esta última puede estar declarada en diferente tiempo verbal o en plural.

- ***question:POS***. Esta propiedad representa la categoría gramatical del elemento contenido en le pregunta. Algunos ejemplos de categorías gramaticales son número cardinal (CD), determinante (DT), adjetivo (JJ), nombre propio (NNP), entre otras.
- ***question:synonym***. Esta propiedad representa un sinónimo del elemento contenido en la pregunta. Cada elemento puede tener más de un sinónimo. Esta propiedad permite incrementar la efectividad del sistema para relacionar los términos provistos en la pregunta con elementos de la base de conocimiento.

- ***question:Literal***. Este elemento representa valores constantes formados por una secuencia de caracteres tales como números.
- ***question:Modifier***. Este elemento comprende aquellas palabras, frases o cláusulas que actúan modificando el núcleo del sujeto en la pregunta, complementando el sentido de la misma.
- ***question:NamedEntity*.** Este representa un elemento contenido en la pregunta que pertenece a una categoría semántica predefinida tales como nombres de personas, organizaciones, lugares, entre otros. Las categorías consideradas en este trabajo pertenecen al grupo de categorías semánticas acuñadas en la sexta edición de las conferencias MUC (Grishman and Sundheim 1996) que son ubicaciones geográficas (LOCATION), personas (PERSON) y organizaciones (ORGANIZATION). Es importante mencionar que este elemento no representa un individuo contenido en la base de conocimiento, sin embargo, este podrá ser relacionado con uno de ellos.
- ***question:QuestionWord*.** Este elemento representa la partícula interrogativa contenida en la pregunta tales como pronombres interrogativos (*What*, *Which*, *Who* y *Where*). Estos elementos permiten determinar el tipo de respuesta esperada por el usuario, por ejemplo, cuando el usuario provee una pregunta que contiene la partícula interrogativa *Who*, generalmente, el usuario está buscando una persona u organización dentro de un contexto específico. Esto a su vez permite reducir el espacio de búsqueda dentro de la base de conocimiento, lo que incrementa la posibilidad de encontrar la respuesta correcta.
- ***question:Verb***. Este elemento representa el verbo principal de la pregunta.

- ***question:QuestionType***. Este elemento representa el tipo de pregunta provista por el usuario. Este tipo corresponde con alguno de los establecidos en la clasificación de preguntas propuesta en este trabajo y de la que se hablará en la sección 2.4.3.1.1.
- ***question:AnswerType***. Este elemento representa el tipo de respuesta esperada por el usuario tal como fecha, lugar, persona, entre otros.

Por otra parte, los elementos de la ontología de la pregunta que permiten describir el contexto de esta última son:

- ***question:KnowledgeBaseEntity***. Este elemento representa un elemento que la pregunta que concuerda con alguno de los descritos en la ontología del dominio. Las propiedades de esta clase son:
    - ***question:URI***. Este representa el identificador del término contenido en la ontología del dominio con el que coincide el elemento de la pregunta.
    - ***question:comment*.** Este elemento contiene la descripción del recurso de la base de conocimiento entendible por el humano, la cual generalmente está contenido en la propiedad *rdfs:comment*.
    - ***question:score***. Este elemento es un valor numérico que indica el nivel de similitud de cadena existente entre el elemento de la pregunta y todo aquel elemento de la ontología. Esta similitud es calculada utilizando la métrica de distancia de Levenshtein (Levenshtein 1966) la cual permite obtener la distancia de edición o distancia entre palabras que corresponde al número mínimo de operaciones necesarias para transformar una cadena de caracteres en otra.
- ***question:KnowledgeBaseConcept***. Este elemento representa las entidades de conocimiento identificadas en la pregunta que son clases en la ontología del dominio. Por ejemplo, en la pregunta: *Who is the president of the United States?* (¿Quién es el presidente de los Estados Unidos?) el término *president* será relacionado con la clase *President* contenida en la ontología del dominio.
- ***question:KnowledgeBaseIndividual***. Este elemento representa las entidades de conocimiento identificadas en la pregunta que son individuos en la ontología del dominio. Tomando en cuenta el ejemplo anterior, un individuo es *United States*. Una propiedad de este elemento es:
    - ***question:Type*.** Esta propiedad representa la clase a la que pertenece esta instancia dentro del dominio de la aplicación. Esta propiedad

permitirá determinar si este individuo corresponde con el dominio o rango de la propiedad identificada y con la cual está relacionado.

- ***question:KnowledgeBaseObjectProperty***. Este representa las entidades de conocimiento identificadas en la pregunta que son propiedades de tipo *object property* en la ontología del dominio. Por ejemplo, en la pregunta: *Give me all books written by Danielle Steel* (Dame todos los libros escritos por Danielle Steel), la propiedad identificada es *written by*. la cual relaciona el concepto *Book* con la clase *Writer*, a la cual pertenece el individuo *Danielle Steel*.
- **question:KnowledgeBaseDatatypeProperty**. Este elemento representa las entidades de conocimiento detectadas en la pregunta que son propiedades de tipo *dataype* en la ontología del dominio. Por ejemplo, en la pregunta: *What is the homepage of Rafael Nadal?* (¿Cuál es la página principal de Rafael Nadal?), un datatype property es *homepage* la cual es una cadena de texto correspondiente a la página principal de Rafael Nadal.

Es importante resaltar el hecho de que el modelo ontológico propuesto en este trabajo es completamente independiente de cualquier base de conocimiento. Esta característica está motivada por la idea de proveer una interfaz de lenguaje natural donde el cambio de la base de conocimiento no demande un gran esfuerzo para su adaptación.

Por otro lado, el modelo de la pregunta define también elementos que describen las relaciones entre los conceptos antes descritos. Concretamente, se ha establecido un conjunto de *object properties* que corresponden con las relaciones gramaticales utilizadas en el análisis sintáctico de dependencias. Sin embargo, para comprender mejor su funcionamiento estas son descritas en la sección 4.2.3.7.

4.2.2 Preprocesamiento de la base de conocimiento

Uno de los principales retos de las interfaces de lenguaje natural orientados a bases de conocimiento es relacionar los elementos contenidos en la pregunta expresada en lenguaje natural provista por el usuario y los elementos de la base de conocimiento. Con el objetivo de afrontar dicho reto, la NLIKB propuesta en esta tesis aprovecha la ontología de la base de conocimientos para generar un vocabulario que le permita conocer los términos empleados por los usuarios en ese dominio en particular. Además, este proceso provee a la NLIKB un nivel de independencia respecto al dominio sobre el cual será utilizada, ya que su adaptación a otro dominio se basará en gran medida en la provisión de la ontología que modele el nuevo dominio.

Como se mencionó en secciones anteriores, una base de conocimiento está compuesta por dos elementos principales: TBox y ABox. El TBox representa el vocabulario del dominio de la aplicación tales como clases, propiedades y restricciones. El ABox contiene todas las declaraciones relativas a los individuos en términos de su vocabulario, es decir, en términos del TBox. Dicho esto, la NLIKB propuesta incluye un módulo que extrae la propiedad *rdfs:label* de cada uno de los elementos descritos en el TBox, es decir, de cada clase, *object property* y *datatype property,* para generar el vocabulario que represente en gran medida el lenguaje utilizado por los usuarios en ese dominio en particular. La propiedad *rdfs:label* es una instancia de *rdf:property* que la especificación RDF *Schema* (Brickley and Guha 2016) recomienda utilizar para proveer una versión del nombre del recurso que sea entendible por el humano. Además, este módulo obtiene el lema del contenido de la propiedad *rdfs:label* con el objetivo de incrementar la posibilidad de relacionar elementos de la pregunta con elementos de la base de conocimiento, ya que ambos términos pueden expresarse en diferente tiempo verbal o en plural.

Desafortunadamente no todos los recursos, sean clases o propiedades, descritos en las ontologías cuentan con una descripción mediante la cual el usuario pueda llevar a cabo la búsqueda, es decir, en ocasiones la propiedad *rdfs:label* del elemento es omitida, está vacía o contiene caracteres especiales tales como guion medio (-) o guion bajo (_) en lugar de espacios en blanco para separar las palabras que son parte de la descripción del elemento. Para hacer frente a este problema, el módulo de la NKILKB genera una descripción textual del elemento de la ontología a partir de su identificador, es decir, a partir de su URI. Para entender mejor este proceso, en la Figura 4-3 se muestran las partes que componen a una URI y que son de interés para este trabajo.

http:	//www.example.com	/domains/example	#EjemploRecurso
Esquema	*Autoridad*	*Ruta*	*Fragmento*

Figura 4-3. Elementos de la URI.

- **Esquema**. Esta parte representa una especificación para asignar los identificadores y en algunas veces indica el protocolo de acceso al recurso, tal como http, https, entre otros.
- **Autoridad**. Este parte identifica la autoridad de nombres.
- **Ruta**. Esta parte es usualmente organizada de forma jerárquica e identifica al recurso en el ámbito del esquema URI y la autoridad de nombres.

- **Fragmento**. Esta parte permite identificar un fragmento del recurso principal, en este caso, del recurso de la ontología. El comienzo de esta parte viene dado por el carácter "#".

Cabe mencionar que otro componente de la URI es la consulta, cuya estructura es no jerárquica y usualmente contiene pares atributo-valor, sin embargo, este elemento no es de interés para este trabajo. Dicho esto, el módulo de preprocesamiento de la base de conocimientos extrae la parte denominada Fragmento de la URI y la procesa de una de las siguientes formas:

a) El sistema remplaza cualquier guion medio (-) o guion bajo (_) contenido el fragmento y lo reemplaza por espacios en blanco.
b) Cuanto el texto extraído del fragmento está escrito en *CamelCase*, el sistema divide el texto en sus palabras constituyentes. *CamelCase* es un estilo de escritura que se caracteriza por que las palabras van unidas entre sí sin espacios en blanco, y la primera letra de cada término se encuentra en mayúscula, lo que permite hacer más legible el conjunto de palabras.

Existen estilos para el modelado de ontología que en vez de utilizar la parte denominada *Fragmento* para identificar al recurso principal, utilizan la parte final del elemento *Ruta*. Para hacer frente a esta situación, el módulo determina cual es el enfoque utilizado dentro del modelado de la ontología. Cuando no detecta la parte *Fragmento*, toma como identificar del recurso la parte siguiente a la última ocurrencia del carácter /. Para el ejemplo mostrado en la Figura 4-3, se tomaría en cuanta la palabra *example*. Sin embargo, la URI no puede contener espacios en blanco, por lo que es probable que la parte de la URI en cuestión utilice caracteres especiales tales como guion bajo (_) o guion medio (-), en vez de espacios en blanco. También, esta parte puede estar escrita bajo un enfoque de escritura *CamelCase* donde, como se explicó anteriormente, las palabras van unidas entre sí sin espacios en blanco y la primera letra de cada término se encuentra en mayúscula. Para resolver este problema, este módulo lleva a cabo el proceso *a* o *b* descritos anteriormente.

La información obtenida por este módulo es almacenada en un conjunto de listas conocidas como *gazetteers*, las cuales serán utilizadas por el módulo de procesamiento de la pregunta para relacionar las palabras contenidas en la pregunta con entidades descritas por la ontología del dominio. Cada entrada de la lista está compuesta por tres elementos que son:

1. Nombre de la entidad provisto por la propiedad *rdfs:label*.
2. Tipo de elemento de la ontología, que puede ser clase (CONCEPT), *object property* (OBJECT_PROPERTY) y *datatype property* (DATATYPE_PROPERTY).
3. URI del elemento que, como recordaremos, permite identificar de manera unívoca al recurso de la base de conocimiento.

En la Figura 4-4 se muestra un extracto del *gazetteer* generado por el sistema para la ontología de DBpedia. En ella se aprecia la definición de conceptos y propiedades de tipo *object* y *datatype* descritos en la versión 2015 de la ontología de DBpedia.

```
artistic genre    CONCEPT     http://dbpedia.org/ontology/ArtisticGenre
associate star    OBJECT_PROPERTY   http://dbpedia.org/ontology/associateStar
author      OBJECT_PROPERTY   http://dbpedia.org/ontology/author
author of preface OBJECT_PROPERTY   http://dbpedia.org/ontology/prefaceBy
baseball season   CONCEPT     http://dbpedia.org/ontology/BaseballSeason
battle      OBJECT_PROPERTY   http://dbpedia.org/ontology/battle
battle honour     DATATYPE_PROPERTY http://dbpedia.org/ontology/battleHonours
battle honours    DATATYPE_PROPERTY http://dbpedia.org/ontology/battleHonours
beatified place   OBJECT_PROPERTY   http://dbpedia.org/ontology/beatifiedPlace
beatify place     OBJECT_PROPERTY   http://dbpedia.org/ontology/beatifiedPlace
beltway city      OBJECT_PROPERTY   http://dbpedia.org/ontology/beltwayCity
big pool record   DATATYPE_PROPERTY http://dbpedia.org/ontology/bigPoolRecord
birth CONCEPT     http://dbpedia.org/ontology/Birth
birth date  DATATYPE_PROPERTY http://dbpedia.org/ontology/birthDate
birth name  DATATYPE_PROPERTY http://dbpedia.org/ontology/birthName
birth place OBJECT_PROPERTY   http://dbpedia.org/ontology/birthPlace
birth sign  OBJECT_PROPERTY   http://dbpedia.org/ontology/birthSign
birth year  DATATYPE_PROPERTY http://dbpedia.org/ontology/birthYear
birthday    DATATYPE_PROPERTY http://dbpedia.org/ontology/birthDate
book  CONCEPT     http://dbpedia.org/ontology/Book
born in     OBJECT_PROPERTY   http://dbpedia.org/ontology/birthPlace
bowl record DATATYPE_PROPERTY http://dbpedia.org/ontology/bowlRecord
death date  DATATYPE_PROPERTY http://dbpedia.org/ontology/deathDate
death place OBJECT_PROPERTY   http://dbpedia.org/ontology/deathPlace
death year  DATATYPE_PROPERTY http://dbpedia.org/ontology/deathYear
number of employee DATATYPE_PROPERTY http://dbpedia.org/ontology/numberOfEmployees
television director     CONCEPT     http://dbpedia.org/ontology/TelevisionDirector
vice president    CONCEPT     http://dbpedia.org/ontology/VicePresident
zip code    DATATYPE_PROPERTY http://dbpedia.org/ontology/zipCode
```

Figura 4-4. Extracto del gazetteer generado en la fase de preprocesamiento.

4.2.3 Procesamiento de la pregunta

Una vez que el módulo de preprocesamiento de la base de conocimiento ha llevado a cabo su tarea, el NLIKB estará disponible para recibir preguntas por parte del usuario. Así, cuando el usuario lleva a cabo dicha tarea, el módulo de procesamiento de la pregunta analiza la pregunta mediante técnicas de PLN y Web Semántica con el objetivo de obtener toda la información relacionada con su estructura y contexto. La información obtenida se almacena en el modelo ontológico de la pregunta propuesto en este trabajo.

El módulo de procesamiento de la pregunta se divide a su vez en las ocho etapas que se aprecian en la Figura 4-5, y que serán descritas en las siguientes secciones.

Figura 4-5. Módulo de procesamiento de la pregunta.

4.2.3.1 Tokenización

La primera etapa del procesamiento de un texto en lenguaje natural consiste en definir claramente los caracteres, palabras y oraciones contenidas en el documento, ya que estos representan las unidades fundamentales de representación del conocimiento para las etapas de procesamiento siguientes. De esta manera, la primer técnica PLN aplicada por el presente módulo es la *Tokenización*, la cual, en el contexto de PLN puede ser definida como la tarea de dividir un flujo de caracteres en palabras (Habert et al. 1998), es decir, esta técnica tiene como objetivo identificar cada unidad atómica, denominada *token*, contenida en el texto. Donde un *token* puede ser una palabra, un símbolo, o algún otro elemento significativo. En el contexto del modelo de la pregunta, el token corresponde con el elemento *question:QuestionElement*. Un ejemplo de tokenización se muestra en la Figura 4-6, donde cada celda representa un *token* de la pregunta *Who is the mayor of New York City?* (¿Quién es el alcalde de la ciudad de Nueva York?) incluyendo el símbolo *?*.

Who	is	the	mayor	of	New	York	City	?

Figura 4-6. Ejemplo de tokenización.

4.2.3.2 Etiquetador POS

Esta técnica permite identificar la categoría gramatical de cada palabra contenida en la pregunta y la cual, dentro del modelo de la pregunta, es representada por la propiedad *question:POS* de la clase *question:QuestionElement*. En la Tabla 4-1 se muestran algunas de las categorías gramaticales de interés para el presente trabajo.

Tabla 4-1. Categorías gramaticales utilizadas por el etiquetador POS.

Etiqueta	Significado	Ejemplo
CC	Conjunciones coordinantes	and, but, nor, or, yet
CD	Número cardinal	first, second, third
DT	Determinante	Artículos incluyendo: a, an, every, no, the, another, any, some, those
IN	Preposición o conjunción subordinante	in, on, beside, above, after, before
JJ	Adjetivo	good, new, long, bad, young
JJR	Adjetivo - comparativo	Adjetivos con la terminación comparativa "er" y un significado comparativo
JJS	Adjetivo - superlativo	Adjetivos con la terminación comparativa "est"
NN	Sustantivo - singular	mother, grandmother, dog
NNS	Sustantivo - plural	cats, restaurants, beaches
NNP	Nombre propio - singular	Britney, Paris
NNPS	Nombre propio-plural	Americas, Johns, Jeffreys
PP	Pronombre personal	I, you, he, she, it, we, they
RB	Adverbio	carefully, quickly, slowly
RBR	Adverbio - comparativo	more quietly, more slowly
RBS	Adverbio - superlativo	most quietly, most slowly
VDB	Verbo – pasado simple	rang, sang, swam
VBG	Verbo - gerundio	helping, jumping
VBN	Verbo – pasado participio	helped, jumped
VB	Verbo – forma base	help, jump
VBP	Verbo – singular presente	sing, jump, run
VBZ	Verbo – tercera persona singular presente	plays, writes, walks
WDT	*Wh*-determinante	which
WP$	*Wh*-pronombre posesivo	whose
WP	*Wh*-pronombre	what, who, whom
WRB	*Wh*-adverbio	how, where, why, when (sentido temporal)

En la Figura 4-7 se muestra la asignación de la categoría gramatical a cada uno de los tokens de la pregunta ejemplo anterior, de la cual destacan los elementos cuyas categorías gramaticales son *Wh*-pronombre (WP - *Who*), sustantivo (NN - *mayor*) y el nombre propio en singular (NNP - *New York City*). Estas categorías, como se verá más adelante, son algunas de las más importantes dentro del proceso general de la NLIKB.

Who	is	the	mayor	of	New	York	City	?
WP	VBZ	DT	NN	IN	NNP	NNP	NNP	.

Figura 4-7. Ejemplo de etiquetador POS.

4.2.3.3 Lematización

En ocasiones, algunas palabras contenidas en la pregunta del usuario no coinciden con la descripción del elemento de la base de conocimiento provista a través de la propiedad *rdfs:label*. Por ejemplo, la palabra puede estar declarada en un tiempo verbal diferente o en plural. Con el objetivo de afrontar esta situación, el módulo de procesamiento de la pregunta aplica la técnica de lematización para determinar el lema de cada palabra contenida en la pregunta. En este contexto, el lema es la palabra que se encuentra como entrada en un diccionario tradicional. En la Figura 4-8 se muestra, en color verde, el lema de cada uno de los términos de la pregunta: *What is the highest mountain in Spain?* (¿Cuál es la montaña más alta en España?). En esta imagen se puede observar que solo dos palabras varían con respecto a su lema. El verbo *is*, que cambia a su forma base *be*, y el adjetivo superlativo *highest*, que cambia al adjetivo *high*.

What	is	the	highest	mountain	in	Spain	?
WP	VBZ	DT	JJS	NN	IN	NNP	.
What	be	the	high	mountain	in	Spain	?

Figura 4-8. Ejemplo de lematización.

4.2.3.4 Reconocimiento de entidades nombradas

Durante la sexta edición de las conferencias MUC (*Message Understanding Conference*) (Grishman and Sundheim 1996) se estableció la tarea de reconocimiento de entidades nombradas, la cual básicamente consiste en identificar los nombres de personas, organizaciones y ubicaciones geográficas dentro de un texto. El módulo de procesamiento de la pregunta utiliza la herramienta Stanford NLP, específicamente el componente NER (Finkel, Grenager, and Manning 2005). Este componente provee tres modelos para la identificación de entidades nombradas que varían en el número de clases que permite identificar.

- **Modelo de 3 clases**. Este modelo fue entrenado mediante conjuntos de datos y alguna información adicional tal como ACE (*Automatic Content Extraction*) (Doddington et al. 2004) y cantidades limitadas de datos internos. Este modelo permite identificar ubicaciones geográficas (LOCATION), personas (PERSON) y organizaciones (ORGANIZATION),
- **Modelo de 4 clases**. Este modelo fue entrenado en el CoNLL (*Conference on Natural Language Learning*) (Tjong Kim Sang and De Meulder 2003) y permite identificar, además de las tres clases del modelo anterior, elementos misceláneos (MISC, de *miscellaneous*) que comprenden aquellas entidades que no pertenecen a los primeros tres grupos, por ejemplo, nacionalidades.
- **Modelo de 7 clases**. Este modelo fue entrenado con los conjuntos de datos de entrenamiento de las conferencias MUC 6 y MUC 7. Este modelo permite identificar cuatro clases más a las del primer modelo que son dinero (MONEY), porcentaje (PERCENT), fechas (DATE) y hora (TIME).

La NLIKB propuesta en este trabajo hace uso del modelo de siete clases. De esta manera, para la pregunta del ejemplo anterior, este módulo identifica la palabra *Spain* como una ubicación geográfica, tal como se aprecia en la Figura 4-9.

What	is	the	highest	mountain	in	Spain	?
WP	VBZ	DT	JJS	NN	IN	NNP	.
What	be	the	high	mountain	in	Spain	?
						LOCATION	

Figura 4-9. Ejemplo de reconocimiento de entidad nombrada.

De esta manera, cada entidad identificada por este módulo es almacenada en el modelo como una instancia de la clase *question:NamedEntity.* Por último, es importante mencionar que esta tarea es completamente independiente del contexto, ya que no requiere interacción con ningún componente de la base de conocimiento del dominio.

4.2.3.5 Detección de elementos de la base de conocimiento

4.2.3.5.1 Detección de clases y propiedades

Este módulo lleva a cabo un proceso similar al del módulo de reconocimiento de entidades nombradas, solo que, en este caso, en vez de identificar los nombres de personas, organizaciones, ubicaciones geográficas, dinero, porcentaje, fechas y horas. Este módulo es responsable de identificar ocurrencias de las entidades descritas en la ontología de la base de conocimiento a través de la ontología, es decir, de las clases y propiedades *object property* y *datatype* que se corresponden con las clases *KnowledgeBaseConcept*, *KnowledgeBaseObjectProperty* y *KnowledgeBaseDatatypeProperty* del modelo de la pregunta respectivamente.

Para alcanzar el objetivo antes descrito, este módulo se apoya del componente RegexNER provisto por la suite de herramientas de PLN Stanford CoreNLP, y del *gazetteer* generado por el módulo de preprocesamiento de la base de conocimiento. RegexNER es una interfaz basada en patrones, es decir, basada en reglas, que permite llevar a cabo el reconocimiento de entidades nombradas, que en este caso son las clases y las propiedades. En la Figura 4-10 se aprecian tres ejemplos de reglas de RegexNER para la identificación de elementos descritos por la ontología de DBpedia.

```
book   CONCEPT      http://dbpedia.org/ontology/Book
largest city OBJECT_PROPERTY     http://dbpedia.org/ontology/largestCity
postal code  DATATYPE_PROPERTY   http://dbpedia.org/ontology/postalCode
```

Figura 4-10. Ejemplo de reglas de RegexNER de Stanford NLP.

La regla más simple utilizada por RegexNER consiste en únicamente dos campos separados por tabulación por cada línea contenida en un archivo. El primer campo contiene el texto que debe ser relacionado, mientras que el segundo campo indica la categoría de la entidad. Para el ejemplo anterior, las palabras *book*, *largest city* y *postal code* representan los términos a ser relacionados, mientras que los campos OBJECT, OBJECT_PROPERTY y DATATYPE_PROPERTY indican el tipo de entidad dentro de la ontología de la base de conocimiento. Para este trabajo de tesis se utiliza un campo más

que consiste en la URI del elemento de la ontología. Esta propiedad se agrega como una anotación más al texto a ser relacionado.

4.2.3.5.2 Detección de individuos

La detección de individuos tiene como objetivo relacionar las entidades nombradas (*question:NamedEntity* en el contexto del modelo de la pregunta) que fueron identificadas por el proceso descrito en la sección 4.2.3.4 con individuos almacenados en el ABox de la base de conocimiento. Para ello, este módulo lleva a cabo una búsqueda en la base de conocimiento a través de consultas SPARQL. Esta búsqueda se basa en coincidencia de cadenas, por lo que aprovecha el método *regex* de SPARQL que provee la habilidad de realizar consultas del estilo LIKE de SQL para determinar si una cadena de caracteres dada coincide con un patrón especificado. La consulta resultante se asemeja a la mostrada en la Figura 4-11.

```
1  SELECT distinct ?entity ?label ?type ?comment
2  WHERE {
3  ?entity rdfs:label ?label.
4  ?entity rdf:type ?type.
5  ?entity rdfs:comment ?comment.
6  FILTER regex(?label, "Liverpool", "i")
7  }
```

Figura 4-11. Consulta SPARQL para la recuperación de datos de acuerdo a similitud de cadena.

Donde la cláusula SELECT, al igual que en lenguaje de consulta como SQL, indica la lista de valores de variables a obtener. En este caso, el campo *?entity* representa el identificador del elemento, es decir, su URI; el campo *?label* indica la versión en lenguaje natural del nombre del recurso; el campo *?type* indica la clase del elemento, finalmente, el campo *?comment* representa la descripción del recurso de la base de conocimiento entendible por el humano. Las líneas tres a cinco son tripletas RDF que forman el patrón de grafo a ser obtenido. En esta consulta se busca el elemento *?entity* que participe en las tripletas RDF con los predicados *rdfs:label* y *rdf:type*. El predicado *rdfs:label* se utiliza para proveer una versión entendible para el humano del nombre del recurso. Mientras que el predicado *rdf:type* es una instancia de *rdf:property* que es utilizado para indicar que un recurso es una instancia de una clase. Finalmente, en la línea cinco se hace uso del filtro *regex* para especificar que la propiedad *rdfs:label* debe contener la cadena *Liverpool*, por su parte, el argumento *i* indica que la coincidencia de patrón no es sensible al uso de mayúsculas y minúsculas.

Como resultado de la consulta anterior, la base de conocimiento retornará una lista con aquellos elementos cuya propiedad *rdfs:label* coincida con el parámetro enviado. En la

Tabla 4-2 se muestra un extracto de los resultados obtenidos cuando se ejecuta la consulta del ejemplo anterior en el repositorio de datos DBpedia.

Tabla 4-2. Lista de resultados de una consulta SPARQL para DBpedia.

?entity	?label	?type
http://dbpedia.org/resource/Liverpool	Liverpool	http://dbpedia.org/ontology/City
http://dbpedia.org/resource/Liverpool_(album)	Liverpool (album)	http://dbpedia.org/ontology/Album
http://dbpedia.org/resource/Liverpool_F.C.	Liverpool F.C.	http://dbpedia.org/ontology/SoccerClub
http://dbpedia.org/resource/Liverpool_Institute_for_Performing_Arts	Liverpool Institute for Performing Arts	http://dbpedia.org/ontology/Place
http://dbpedia.org/resource/Mersey_Tigers	Liverpool Mersey Tigers	http://dbpedia.org/ontology/BasketballTeam
http://dbpedia.org/resource/Echo_Arena_Liverpool	Echo Arena Liverpool	http://dbpedia.org/ontology/Stadium
http://dbpedia.org/resource/Westfield_Liverpool	Westfield Liverpool	http://dbpedia.org/ontology/ArchitecturalStructure
http://dbpedia.org/resource/Liverpool_Central_railway_station	Liverpool Central railway station	http://dbpedia.org/ontology/Place
http://dbpedia.org/resource/University_of_Liverpool	University of Liverpool	http://dbpedia.org/ontology/University
http://dbpedia.org/resource/Liverpool_Basketball_Club	Liverpool Basketball Club	http://dbpedia.org/ontology/BasketballTeam

Generalmente, cuando se intenta relacionar una entidad nombrada, contenida en la pregunta del usuario con un individuo de la base de conocimiento a través de similitud de cadenas, se obtiene una lista con diversos elementos, tal como ocurre en el ejemplo anterior. Por este motivo, este módulo ordena los individuos obtenidos de acuerdo al grado de similitud existente entre las cadenas de texto referentes a la propiedad *rdfs:label* del individuo y la cadena de caracteres que representa a la entidad nombrada. El algoritmo para calcular el grado de similitud entre cadenas de texto utilizado por este módulo es el propuesto por Levenshtein. Con el objetivo de normalizar el resultado a un valor entre 0 y 1, se empleó la fórmula utilizada en (Rodríguez García 2014):

$$similitud = 1 - \frac{distanciaLeve \quad htein(cadena1, cadena2)}{maximaLongitud(cadena1, cadena2)}$$

Donde *cadena1* y *cadena2* representan las listas de términos de los cuales quiere conocerse el grado de similitud; *distanciaLevenshtein* se refiere a la función que define el algoritmo de Levenshtein utilizado para comparar las cadenas, y *maximaLongitud* representa la función que obtiene la longitud máxima de *cadena1* y *cadena2*. De esta manera, para el ejemplo anterior, la distancia de Levenshtein entre las cadenas *Liverpool* y *Liverpool F.C.* es de 5, ya que se necesitan al menos cinco ediciones elementales para transformar uno en el otro; la máxima longitud (14) corresponde a la segunda cadena. Por lo tanto, la fórmula anterior queda de la siguiente manera: 1- (5/14) obteniendo el valor de 0.642857143. Tomando en cuenta el resultado mostrado en la Tabla 4-2 para la entidad nombrada *Liverpool*, el orden de resultados es el mostrado en la Tabla 4-3.

Tabla 4-3. Ordenación por distancia de Levenshtein.

?label	distanciaLevenshtein	maximaLongitud	Similitud
Liverpool	0	9	1
Liverpool F.C.	5	14	0.642857143
Liverpool (album)	8	17	0.529411765
Westfield Liverpool	10	19	0.473684211
Echo Arena Liverpool	11	20	0.45
Liverpool Mersey Tigers	14	23	0.391304348
University of Liverpool	14	23	0.391304348
Liverpool Basketball Club	16	25	0.36
Liverpool Central railway station	24	33	0.272727273
Liverpool Institute for Performing Arts	30	39	0.230769231

Es importante mencionar que el orden obtenido por la distancia de Levenshtein no determina completamente la instancia de la base de conocimiento con la cual será relacionada la entidad nombrada, ya que esto dependerá en gran medida de su tipo (*rdf:type*), es decir, que este último corresponda con el dominio o rango de alguna de las relaciones (*object property* o *dataype property*) encontrados en la pregunta por el proceso descrito en la sección anterior. La determinación del individuo a relacionar es llevada a cabo por el módulo de desambiguación presentado en la sección 4.2.3.8.

4.2.3.6 Búsqueda de sinónimos

El enfoque presentado en esta tesis lleva a cabo un mapeo entre los términos contenidos en la pregunta del usuario y los elementos contenidos en la base de conocimiento. En la sección anterior se presentó un método para relacionar las entidades nombradas con individuos de la base de conocimiento con base únicamente en la similitud de cadenas

entre estos términos. Sin embargo, debido a la libertad de expresión inherente del uso del lenguaje natural, el usuario puede utilizar una infinidad de palabras para hacer referencia al mismo elemento de la base de conocimiento. Estas palabras son denominadas sinónimos, pertenecen a la misma categoría gramatical y pueden ser definidas como palabras que se escriben diferente pero que tienen un significado total o parcialmente idéntico a otra.

Para abordar este problema, el módulo lleva a cabo la búsqueda de sinónimos de las palabras de interés que fueron detectadas en la consulta de información y almacenadas en la ontología de la pregunta. Esta tarea emplea la base de datos léxica WordNet (Miller 1995). En esta base de datos, los sustantivos, verbos, adjetivos, y adverbios están organizados en conjuntos de sinónimos, cada uno representando un concepto lexicalizado. WordNet define el vocabulario de un lenguaje como un conjunto *W* de pares (*f,s*), donde *f* es una cadena de un alfabeto finito y *s* es un elemento de un conjunto dado de significados. En la Tabla 4-4 se muestra el conjunto de sinónimos obtenidos de WordNet para la palabra *film*.

Tabla 4-4. Ejemplo de sinónimos obtenidos de WordNet.

Sustantivo
S: (n) **movie**, **film**, **picture**, **moving picture**, **moving-picture show**, **motion picture**, **motion-picture show**, **picture show**, **pic**, **flick** (a form of entertainment that enacts a story by sound and a sequence of images giving the illusion of continuous movement) "they went to a movie every Saturday night"; "the film was shot on location"
S: (n) **film**, **cinema**, **celluloid** (a medium that disseminates moving pictures) "theater pieces transferred to celluloid"; "this story would be good cinema"; "film coverage of sporting events"
S: (n) **film**, **photographic film** (photographic material consisting of a base of celluloid covered with a photographic emulsion; used to make negatives or transparencies)
S: (n) **film** (a thin coating or layer) "the table was covered with a film of dust"
S: (n) **film**, **plastic film** (a thin sheet of (usually plastic and usually transparent) material used to wrap or cover things)
Verbo
S: (v) **film**, **shoot**, **take** (make a film or photograph of something) "take a scene"; "shoot a movie"
S: (v) **film** (record in film) "The coronation was filmed"

Una de las características sobresalientes de WordNet es que respeta las categorías sintácticas sustantivo, verbo, adjetivo, y adverbio. Tal como se puede observar en la Tabla 4-4, WordNet provee sinónimos para la palabra *film* agrupándolos de acuerdo a su categoría gramatical, sustantivo y verbo. Además, dentro de cada categoría gramatical, provee diferentes sinónimos de acuerdo a su sentido. El presente módulo aprovecha esta

característica, para elegir los sinónimos de la palabra contenida en la pregunta en base a la categoría gramatical que le fue asignada en fases anteriores.

Consideremos el caso de la clase *Film* descrita en la ontología de DBpedia y cuya propiedad *rdfs:label* es *movie*. La fase de preprocesamiento de la base de conocimiento genera una entrada en el gazetteer que permitirá relacionar la palabra *movie* con la clase *Film*. Por este motivo, para la pregunta mostrada en la Figura 4-12, el módulo de detección de elementos de la base de conocimiento no podrá llevar a cabo dicha relación debido a que *movie* no aparece en la pregunta. Para resolver esta situación, el presente módulo obtiene de WordNet los sinónimos de la palabra *film* con categoría gramatical sustantivo (NNS) dentro de los cuales se encuentra la palabra *movie*, logrando así relacionar tales elementos.

How	many	films	did	Hal	Roach	produce	?
WRB	JJ	NNS	VBD	NNP	NNP	VB	.
how	many	film	do	Hal	Roach	produce	?
				PERSON			

Figura 4-12. Ejemplo de análisis de una pregunta.

4.2.3.7 Análisis de dependencias

Hasta este punto, la interfaz ha identificado todas aquellos conceptos y relaciones contenidos en la pregunta con respecto a la base de conocimiento del dominio. Ahora el reto es determinar la relación entre estos elementos. Para cumplir con este objetivo, el presente módulo lleva a cabo el análisis sintáctico de la pregunta que, tal como se describió en la sección 2.3.3.4, genera una representación de la estructura de la oración que pone de manifiesto las relaciones de dependencia estructural de las palabras.

El supuesto básico del análisis de dependencia es la idea de que la estructura sintáctica consiste esencialmente en palabras enlazadas por relaciones binarias y simétricas llamadas relaciones de dependencia (Kübler, McDonald, and Nivre 2009). Una relación de dependencia es binaria y se da entre una palabra sintácticamente subordinada, denominada dependiente, y otra palabra de la que depende, denominada gobernador, regente o cabeza. Por ejemplo, para la frase *He gave me a book* (El me dio un libro), tenemos las siguientes dependencias, las cuales son expresadas mediante las etiquetas POS y etiquetas frasales especificados en la Penn Treebank (Marcus, Marcinkiewicz, and Santorini 1993):

- **nsubj(gave, he)**. Un sujeto nominal (nsubj) es un nominal que es el sujeto sintáctico. En este caso indica la relación entre el verbo *gave* y su sujeto *he*.
- **iobj(gave, me)**. Esta relación indica el objeto indirecto (iobj) de un verbo, en este caso, del verbo en pasado *gave* (*give*– dar).
- **det(book, a)**. Esta relación se mantiene entre una cabeza nominal (*book*) y su determinante (*a*).
- **dobj(gave, book)**. Esta relación indica la parte de la oración que recibe de manera directa la acción. Generalmente, el objeto directo (dobj) responde a las preguntas: ¿qué? o ¿a quién? En este caso, quien recibe la acción es el libro (*book*).

El módulo descrito en esta sección lleva a cabo su correspondiente tarea mediante el analizador de dependencias provisto por la herramienta Stanford NLP (De Marneffe and Manning 2008). Este analizador contiene aproximadamente 50 relaciones gramaticales. Sin embargo, en la Tabla 4-5 se muestran las relaciones que resultan significativas para la NLIKB de este trabajo de tesis.

Tabla 4-5. Dependencias utilizadas.

Dependencia	Descripción	Ejemplo	Nomenclatura
compound	Esta relación se utiliza para expresiones multipalabra.	*How tall is Claudia Schiffer?*	compound(Schiffer, Claudia)
det	La relación de determinante (det) se mantiene entre una cabeza nominal y su determinante	*Which book do you prefer?*	det(book, which)
dobj	El objeto directo de una frase verbal (VP) es la frase nominal que es el (acusativo) objeto del verbo.	*Which river does the Brooklyn Bridge cross?*	dobj(cross, river)
nmod:agent	Esta relación la utilizan agentes de verbos pasivos.	*Which television shows were created by Walt Disney?*	nmod:agent(created, Disney)
nmod:by	La adición de preposiciones (o información de casos) al nombre de la relación de dependencia no esenciales a menudo hace posible desambiguar su rol semántico. Esta relación se utiliza para la preposición *by*.	*In which films directed by Garry Marshall was Julia Roberts starring?*	nmod:by(directed, Marshall)
nmod:in	Esta relación se utiliza para la preposición *in*.	*In which city did Jhon F. Kennedy die?*	nmod:in(die, city)
nmod:of	Esta relación se utiliza para la	*What is the*	nmod:of(

Dependencia	Descripción	Ejemplo	Nomenclatura
	preposición *of*.	*currency of the Czech Republic?*	currency, Republic)
nmod:poss	Esta se usa para un modificador nominal genitivo/posesivo, expresado ya sea por un nominal en el genitivo o por un determinante posesivo.	*Is Egypt's largest city also its capital?*	nmod:poss(city, Egypt)
nmod:with	Esta relación se utiliza para la preposición *with*.	*Which countries have places with more than two caves?*	nmod:with(places, caves)
nsubj	Un sujeto nominal es un nominal que es el sujeto sintáctico.	*Which river does the Brooklyn Bridge cross?*	nsubj(cross, Bridge)
nsubjpass	Un sujeto nominal pasivo es una frase nominal que es el sujeto sintáctico de una cláusula pasiva.	*Which television shows were created by Walt Disney?*	nsubjpass(created, television)

Es importante mencionar que cada una de estas relaciones de dependencias ha sido definida como un *object property* dentro del modelo de la pregunta con el objetivo de describir las relaciones entre las clases descritas por este último, es decir: *KnowledgeBaseConcept*, *KnowledgeBaseObjectProperty*, *KnowledgeBaseDatatypeProperty, QuestionWord*, entre otros. Por ejemplo, en la Figura 4-13 se muestra el análisis de dependencias para la pregunta: *Which river does the Brooklyn Bridge cross?*

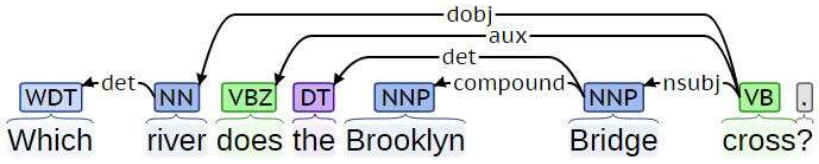

Figura 4-13. Análisis de dependencias.

Como se puede observar en la figura anterior, existe un total de 6 relaciones de dependencia. Sin embargo, el presente módulo solo considera aquellas en las que tanto el gobernador como el dependiente correspondan con *KnowledgeBaseConcept*, *KnowledgeBaseObjectProperty*, *KnowledgeBaseDatatypeProperty*, y *KnowledgeBaseIndividual* identificados en fases previas, o en las cuales aparezca un elemento *question:QuestionWord* cuya propiedad *question:POS* sea WDT (Wh-

determinante), WP (Wh-pronombre) o WRB (Wh-Adverbio). Estos últimos elementos ayudarán al módulo de clasificación de la pregunta a llevar a cabo su respectiva tarea. De esta manera, las relaciones consideradas son:

- **det(river, which)**. La relación de determinante indica al objeto al que se refiere el término *which*. A partir del modelo de la pregunta, la representación semántica queda de la siguiente manera *KnowledgeBaseConcept(river)* ***det*** *questionWord(which)*.
- **compound(Bridge, Brooklyn)**. Esta relación de dependencia se utiliza para expresiones escritas con dos o más palabras separadas, en este caso *Brooklyn Bridge* (el Puente de Brooklyn). Este es almacenado en el modelo de la pregunta como una instancia de *KnowledgeBaseIndividual*.
- **dobj(cross, river)**. El objeto directo (dobj) de una frase verbal (VP) es la frase nominal que es el (acusativo) objeto del verbo. El objeto directo generalmente responde a las preguntas ¿qué? o ¿a quién?, por lo que esta relación indica ¿qué cruza? Su representación en el modelo de la pregunta es *KnowledgeBaseObjectProperty(cross)* ***dobj*** *KnowledgeBaseConcept(river)*.
- **nsubj(cross, Bridge)**. Un sujeto nominal (nsubj) es un nominal que es el sujeto sintáctico de una cláusula. En el ejemplo anterior, *Bridge* (que compone al sujeto *Brooklyn Bridge*) es el sujeto del verbo cruzar (c*ross*). Por lo que podemos deducir que quien cruza es el Puente de Brooklyn. Esta relación se representa como *KnowledgeBaseObjectProperty(cross)* ***nsubj*** *KnowledgeBaseIndividual(Brooklyn Bridge)*.

4.2.3.8 Desambiguación

La ambigüedad se presenta cuando una palabra, un sintagma o una oración puede ser interpretada de más de una forma. En el contexto de este trabajo de tesis, la ambigüedad hace referencia al hecho de cuando una expresión en lenguaje natural puede ser relacionada con más de un elemento del conjunto de datos del dominio. Por ejemplo, en el idioma inglés existen palabras que cuentan con dos significados, uno como verbo y el otro como sustantivo. Consideremos las siguientes preguntas:

1. *What is the last book of The Hunger Games?*
2. *With which travel company did you book your holiday?*

En ambas oraciones aparece la palabra *book*, pero su significado es diferente. En la oración 1, la palabra hace referencia a un objeto (libro) y su categoría gramatical es sustantivo. En la oración 2, el término *book* hace referencia al verbo reservar. Este hecho

cobra importancia en este trabajo debido a que generalmente en el contexto de ontologías, la propiedad *rdfs:label* de una clase es un sustantivo, mientras que para aquellos elementos que son propiedades, y en específico *object properties*, su categoría gramatical es la de verbo. Considerando este hecho, el módulo de detección de entidades y propiedades (ver sección 4.2.3.5.1) necesita conocer cuando la palabra contenida en la pregunta hace referencia a la clase (en el ejemplo anterior, a la clase Libro) y cuando hace referencia a la propiedad o relación (en el ejemplo anterior, reservar). Para ello, este módulo se apoya del módulo de etiquetado POS, el cual indica la categoría gramatical de cada una de las palabras contenidas en la pregunta del usuario. De esta manera, el módulo de detección de entidades y propiedades será capaz de llevar a cabo su tarea a través de reglas como las mostradas en la Figura 4-14.

```
([tag:NN]& /book/)  CONCEPT      http://example.org/ontology/Book
([tag:VB]& /book/)  OBJECT_PROPERTY     http://example.org/ontology/book
```

Figura 4-14. Reglas RegexNER para diferenciar palabras con significado verbo y sustantivo.

El primer campo indica las condiciones que deben cumplirse para que el elemento contenido en la pregunta del usuario se relacione con el elemento de la ontología del dominio. En ambos casos, la secuencia de caracteres que debe coincidir es *book*. Sin embargo, para que este sea considerado como un concepto (CONCEPT) su categoría gramatical dentro de la pregunta debe ser sustantivo (NN). Por el contrario, para que el elemento sea relacionado con la propiedad de tipo *object property*, la categoría gramatical del elemento debe ser verbo (VB).

Por otro lado, un caso de homonimia es posible cuando una de las entidades nombradas identificadas por el sistema puede denotar diferentes individuos. Consideremos la pregunta: *Who is the mayor of Paris?*. De acuerdo con el módulo de detección de individuos (ver sección 4.2.3.5.2), y considerando la base de conocimiento de DBpedia, los individuos con los cuales puede ser relacionada la entidad Paris son los mostrados en la Tabla 4-6.

Tabla 4-6. Individuos de la base de conocimiento para el término Paris.

?label	Tipo	URI	Sim.
Paris	PopulatedPlace	http://dbpedia.org/resource/Paris	1
Edmond Paris	Person	http://dbpedia.org/resource/Edmond_Paris	0.5
Paris Hilton	Person	http://dbpedia.org/resource/Paris_Hilton	0.4166
Eisenberg Paris	Organisation	http://dbpedia.org/resource/Eisenberg_Paris	0.3333
Free Man in Paris	MusicalWork	http://dbpedia.org/resource/Free_Man_in_Paris	0.2941
Gran Prix de Paris	SocialEvent	http://dbpedia.org/resource/Grand_Prix_de_Paris	0.2777
Night Train to Paris	Film	http://dbpedia.org/resource/Night_Train_to_Paris	0.25

Como se pude observar, existe una gran cantidad de entidades que pueden ser relacionadas con el término *Paris*. Además, el tipo o clase de estos elementos es muy variado. Para reducir esta lista, este módulo elimina todos aquellos individuos candidatos cuyo tipo o clase no guarde relación semántica con la propiedad (*object property* o *datatype property*) con la cual guarda una relación sintáctica dentro de la pregunta. En la Figura 4-15 se muestra el análisis de la pregunta en cuestión.

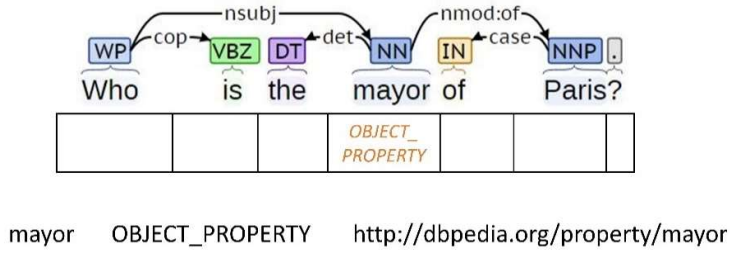

Figura 4-15. Análisis de la pregunta: Who is the mayor of Paris?.

En la Figura 4-15 se muestra la categoría gramatical de cada uno de los términos de la pregunta, las relaciones de dependencia entre ellos, así como los elementos de la base de conocimiento que fueron identificados. En este caso se ha detectado el elemento *KnowledgeBaseObjectProperty(mayor)* con la propiedad *question:URI* igual a *http://dbpedia.org/property/mayor*, y la *NamedEntity(Paris)*. A partir de esta información, el módulo lleva a cabo un proceso que determinará si existe alguna relación entre la propiedad identificada (*mayor*) y los individuos relacionados con la entidad nombrada de la pregunta (ver Tabla 4-6), a través del rango o dominio de la primera. Para ello emplea las consultas SPARQL mostradas en la Figura 4-16.

```
ASK {
?domain <http://dbpedia.org/property/mayor> ?range.
?domain a <http://dbpedia.org/ontology/PopulatedPlace> }

ASK {
?domain <http://dbpedia.org/property/mayor> ?range.
?range a <http://dbpedia.org/ontology/PopulatedPlace> }
```

Figura 4-16. Ejemplo de consulta SPARQL con cláusula ASK.

La cláusula ASK se utiliza para comprobar si un patrón de consulta tiene o no solución. Es importante mencionar que este método no devuelve información acerca de las posibles soluciones de consulta, solo devuelve un valor booleano que indica si existe o no una solución. De igual forma, la primera consulta de la Figura 4-16 es empleada para determinar si existe alguna solución cuando la clase a la que pertenece el individuo es el dominio de la propiedad. Por su parte, la segunda consulta determina si existe solución cuando el tipo o clase a la cual pertenece el individuo es el rango de la propiedad. Si ambas consultas devuelven un *false*, el sistema determina que no existe relación alguna entre los elementos y lo elimina de la lista. Este procedimiento se lleva a cabo para cada uno de los individuos de la lista obtenida.

4.2.4 Clasificación de la pregunta

El tipo de preguntas aceptadas por la presente NLIKB son preguntas factuales, es decir, aquellas que tienen como respuesta un hecho concreto como el nombre de una persona o localidad, la longitud de un objeto o la fecha en que sucedió un evento. Entre estas preguntas se encuentran aquellas que se inician con los llamados *wh*-pronombres tales como *Who*, *What*, *Where*, *Which*, así como expresiones de tipo *How many* para sustantivos contables y *How* seguida de un adjetivo para preguntar acerca de un atributo en particular, por ejemplo, *How tall is the Eiffel Tower?* (¿Qué tan alta es la Torre Eiffel?).

En este trabajo se emplea la taxonomía de la pregunta propuesta por Moldovan en (D. Moldovan et al. 2000) la cual es adaptada para poder utilizarla en el contexto de bases de conocimiento. Esta adaptación consiste en relacionar el tipo de respuesta con tipos de datos establecidos en XML Schema (World Wide Web Consortium 2016b); FOAF (Brickley and Miller 2012), y en la ontología de DBpedia. La clasificación de preguntas y respuestas se muestra en la Tabla 4-7.

Tabla 4-7. Clasificación de preguntas y respuestas.

Clase	Subclase	Tipo de respuesta
what	basic what	money – xsd:double | number – xsd:integer | definition – rdfs:comment, dbo:abstract | noun phrase – foaf:agent, dbo:agent
	what-who	person – foaf:Person, dbo:Person | organization – foaf:Organization, dbo:Organization
	what-when	date – xsd:date
	what-where	location – dbp:Place
who		person – foaf:Person, dbo:Person | organization – foaf:Organization, dbo:Organization
how	how-many	number – xsd:integer
	how-much	money – xsd:double | price - xsd:double
	how-far	distance - xsd:integer, xsd:double
	how-tall	number - xsd:integer, xsd:double
	how-large	number - xsd:integer, xsd:double, dbo:length
where		location – dbp:Place
when		date – xsd:date
which	which-who	person – foaf:Person, dbo:Person
	which-where	location – dbp:Place
	which-when	date – xsd:date
	which-what	noun phrase - foaf:agent, dbo:agent, dbo:Work | organization – foaf:Organization, dbo:Organization
Give/Name/List		noun phrase - foaf:agent, dbo:agent, dbo:Work | organization – foaf:Organization, dbo:Organization | location – dbp:Place

La interfaz propuesta en este trabajo de tesis hace hincapié en la detección de partículas interrogativas las cuales proveen una guía para obtener la respuesta deseada. Por ejemplo, cuando el usuario utiliza la partícula interrogativa *Who* para solicitar información, generalmente él o ella está buscando una persona (*Person*) u organización (*Organization*) dentro de un contexto en específico. Por otro lado, la partícula *Where* indica que el usuario busca información relacionada con un lugar, por lo que el espacio de búsqueda se reduce a solo aquellos individuos que pertenezcan a la clase *Place*, como lo pueden ser ciudades, países, algún edificio, entre otros. Esta misma situación ocurre con el uso de la partícula *When*, la cual hace referencia a un punto en el tiempo. De esta manera, se deduce que el usuario busca alguna fecha, de esta manera, el espacio de búsqueda se ve reducido a aquellos elementos que tengan como rango un valor de tipo *date*.

Para determinar el tipo de pregunta provista por el usuario, este módulo integra un conjunto de reglas basadas en el tipo de relaciones de dependencia existentes entre los elementos de interés contenidos en la pregunta. Algunas de estas reglas se muestran en la Tabla 4-8.

Tabla 4-8. Ejemplos más frecuentes de patrones de pregunta obtenidos.

Pregunta	Regla	Descripción
which	det(WDT, NN)	WDT es *Which*, y NN corresponda con un concepto o propiedad de la ontología.
which	det(WDT, NNS)	WDT es *Which*, y NNS corresponda con un concepto o propiedad de la ontología
what	nsubj(WP, NN)	WP es *What* y NN corresponde con un concepto o propiedad de la ontología
who	nsubj(WP, NN)	WP es *Who* y NN corresponde con un concepto o propiedad de la ontología
how many	advmod(JJ, WRB) and amod(NNS, JJ)	WRB es *How*, JJ *many*, y NNS corresponde con una clase de la ontología.
when	advmod(VBD, WRB) and nsubj(VBD, NN)	WRB es *When*, VBD es un verbo en pasado, y NN es un concepto de la ontología
when	advmod(VBD, WRB) and nsubj(VBD, NNS)	WRB es *When*, VBD es un verbo en pasado, y NNS es un concepto de la ontología

Por ejemplo, para las preguntas mostradas en la Figura 4-17 se utilizan los patrones 1 y 5 respectivamente. En el caso de la primera pregunta se muestran la relación de dependencia determinante (det) entre una entidad de la base de conocimiento (software) y un determinante *WH*. Por su parte, la segunda pregunta cumple con el patrón 3 debido a que la pregunta recoge las dos relaciones de dependencia *advmod* (modificador adverbial) y *amod* (modificador adjetival).

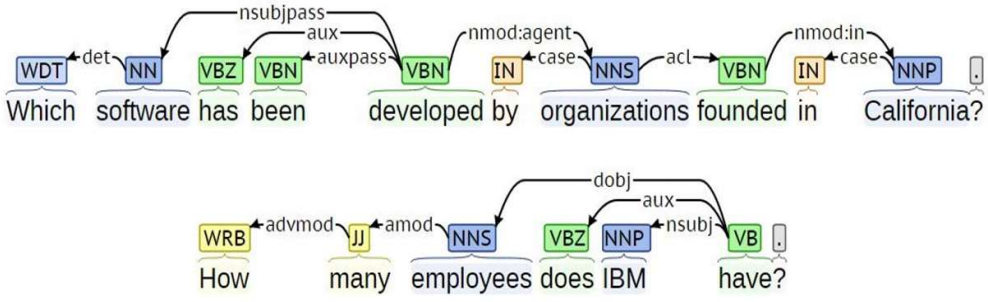

Figura 4-17. Análisis de preguntas en lenguaje natural.

4.2.5 Generación de la consulta en lenguaje formal

Partiendo del conocimiento generado en los módulos anteriores, almacenado en el modelo ontológico de la pregunta. El reto de este componente es generar consultas en un lenguaje formal SPARQL, que puedan ser evaluadas con respecto a la base de conocimiento, con el fin de obtener la información que proporcione respuesta a la pregunta provista. A continuación, se describe el proceso llevado a cabo por el presente módulo a través de un ejemplo que toma en cuenta la base de conocimiento de DBpedia.

Consideremos la pregunta: *Which software has been developed by organizations founded in California?* (¿Qué software ha sido desarrollado por organizaciones fundadas en California?) y cuyo análisis se muestra en la Figura 4-18.

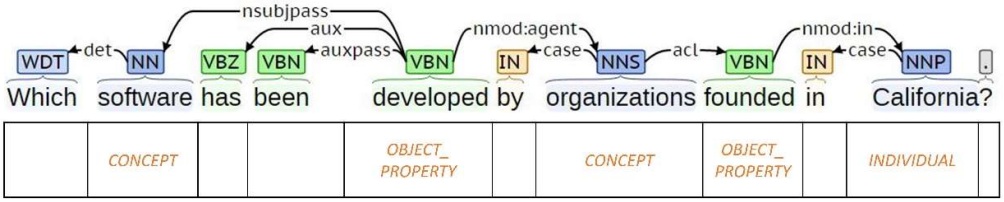

Figura 4-18. Ejemplos de análisis de la pregunta.

Lo primero a destacar en la figura anterior son los términos de la pregunta que fueron relacionados con elementos de la base de conocimiento de DBpedia y que se muestran en la Tabla 4-9.

Tabla 4-9. Elementos de la base de conocimiento identificados en la pregunta.

Término	Modelo de la pregunta	question:URI
software	KnowledgeBaseConcept	*http://dbpedia.org/ontology/Software*
developed	KnowledgeBaseObjectProperty	*http://dbpedia.org/ontology/developer*
organization	KnowledgeBaseConcept	*http://dbpedia.org/ontology/Organisation*
founded	KnowledgeBaseObjectProperty	*http://dbpedia.org/ontology/foundationPlace*
California	KnowledgeBaseIndividual	*http://dbpedia.org/resource/California*

Cabe aclarar que la identificación del elemento *developed* se obtuvo a través del lema del término de la pregunta *developed*, que es *develop*, el cual coincide con el de la propiedad de *developer*. Además, la identificación del elemento *founded* se obtuvo debido a que la *rdfs:label* de la propiedad es *founded in*.

Lo segundo a destacar son las relaciones de dependencia en las que participan los elementos antes descritos, que son:

- nsubjpass(developed, software)
- nmod:agent(developed, organizations)
- nmod:in(founded, California)
- acl(organizations, founded).

La representación semántica de estas relaciones se muestra abajo, donde se resalta en negrita las *object properties* definidas en el modelo de la pregunta:

- KnowledgeBaseObjectProperty(developed) **nsubjpass** KnowledgeBaseConcept(software)
- KnowledgeBaseObjectProperty(developed) **nmodAgent** KnowledgeBaseConcept(organizations)
- KnowledgeBaseObjectProperty(founded) **nmodIn** KnowledgeBaseIndividual(California)
- KnowledgeBaseConcept(organizations) **acl** KnowledgeBaseObjectProperty (founded)

De esta manera, para cada dependencia considerada se generan tripletas RDF considerando el conjunto de plantillas propuestas en este trabajo. En la Tabla 4-10 se presenta un extracto de estas plantillas, donde los elementos en mayúsculas deberán ser sustituidos por la propiedad *question:URI* que representa el identificador del término contenido en la ontología con el que coincide el elemento de la pregunta.

Tabla 4-10. Plantillas de tripletas RDF.

Dependencia	Plantilla de tripleta RDF
KnowledgeBaseConcept acl[4] KnowledgeBaseObjectProperty	?concept a <CONCEPT>
KnowledgeBaseObjectProperty dobj KnowledgeBaseConcept	?var <OBJECTPROPERTY> ?concept. ?concept a <CONCEPT>
KnowledgeBaseObjectProperty dobj KnowledgeBaseIndividual	<INDIVIDUAL> <OBJECTPROPERTY> ?var
KnowledgeBaseObjectProperty nmodAgent KnowledgeBaseConcept	?var <OBJECTPROPERTY> ?concept. ?concept a <CONCEPT>
KnowledgeBaseObjectProperty nmodAgent KnowledgeBaseIndividual	?var < OBJECTPROPERTY> INDIVIDUAL
KnowledgeBaseObjectProperty nmodBy KnowledgeBaseIndividual	?var < OBJECTPROPERTY> INDIVIDUAL
KnowledgeBaseObjectProperty nmodIn KnowledgeBaseConcept	?var <OBJECTPROPERTY> ?concept. ?concept a <CONCEPT>
KnowledgeBaseObjectProperty nmodIn KnowledgeBaseIndividual	<INDIVIDUAL> <OBJECTPROPERTY> ?var
KnowledgeBaseConcept nmodOf KnowledgeBaseIndividual	<INDIVIDUAL> ?relation ?concept. ?relation rdfs:range <CONCEPT>
KnowledgeBaseConcept nmodOf KnowledgeBaseDatatypeProperty	?concept < DATATYPE > ?var
KnowledgeBaseDatatypeProperty nmodOf KnowledgeBaseIndividual	<INDIVIDUAL> < DATATYPE> ?var
KnowledgeBaseObjectProperty nmodOf KnowledgeBaseIndividual	<INDIVIDUAL> <OBJECTPROPERTY> ?var
KnowledgeBaseIndividual nmodPoss KnowledgeBaseObjectProperty	<INDIVIDUAL> <OBJECTPROPERTY> ?var
KnowledgeBaseObjectProperty nmodPoss KnowledgeBaseIndividual	<INDIVIDUAL> <OBJECTPROPERTY> ?var
KnowledgeBaseIndividual nsubj KnowledgeBaseConcept	<INDIVIDUAL> ?relation ?concept. ?concept a <CONCEPT>
KnowledgeBaseIndividual nsubj KnowledgeBaseObjectProperty	<INDIVIDUAL> <OBJECTPROPERTY> ?var
KnowledgeBaseObjectProperty nsubjpass KnowledgeBaseConcept	?concept < OBJECTPROPERTY> ?var. ?concept a <CONCEPT>

[4] El patrón de esta dependencia es utilizado en conjunto con las dependencias nmod:in y nmod:by

Para la relación *KnowledgeBaseObjectProperty(developed) nsubjpass KnowledgeBaseConcept(software)* se generan las tripletas RDF que se muestran en la Figura 4-19.

```
?software <http://dbpedia.org/ontology/developer> ?var.
?software a <http://dbpedia.org/ontology/Software>
```

Figura 4-19. Tripletas RDF generadas para una relación sintáctica nsubjpass.

Para la relación *KnowledgeBaseObjectProperty(developed) nmodAgent KnowledgeBaseConcept(organizations)* se generan las tripletas mostradas en la Figura 4-20:

```
?var <http://dbpedia.org/ontology/developer> ?organization.
?organization a < http://dbpedia.org/ontology/Organisation>
```

Figura 4-20. Tripletas RDF generadas para una relación sintáctica nmodAgent.

Para la relación *KnowledgeBaseObjectProperty(founded) nmodIn KnowledgeBaseIndividual(California)* las tripletas generadas se muestran en la Figura 4-21.

```
?concept < http://dbpedia.org/ontology/foundationPlace>
<http://dbpedia.org/resource/California>
```

Figura 4-21. Tripletas RDF generadas para una relación sintáctica nmodIn.

Finalmente, las tripletas para la relación *KnowledgeBaseConcept(organizations) acl KnowledgeBaseObjectProperty (founded)* se presentan en la Figura 4-22.

```
?organization a < http://dbpedia.org/ontology/Organisation>
```

Figura 4-22. Tripletas RDF generadas para la relación sintáctica acl.

Una vez que se han generado cada una de las tripletas RDF. El siguiente paso es agruparlas en una misma consulta SPARQL, para lo cual este módulo busca relaciones entre los elementos de las dependencias identificados. Como se aprecia en la Figura 4-18, las relaciones *nsubjpass* y *nmod:agent* comparten el elemento *KnowledgeBaseObjectProperty(developed)*, mientras que las relaciones nmod:in y acl comparten el elemento *KnowledgeBaseObjectProperty (founded).* Esto se aprecia de mejor manera en la Figura 2-1.

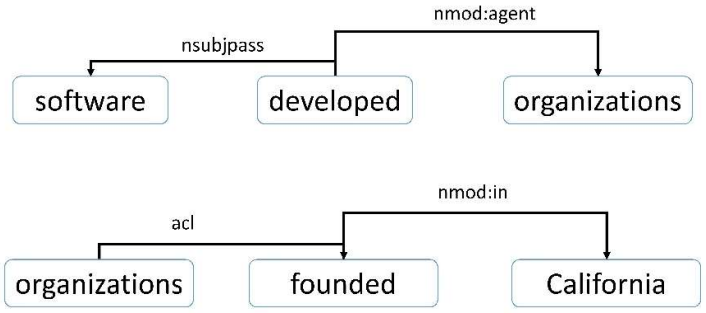

Figura 4-23. Dependencias que comparten elementos.

Cuando dos relaciones binarias, como las relaciones de dependencia, comparten un elemento da lugar a que la unión de ambas relaciones guarde cierto grado de similitud con respecto a las tripletas RDF. Considerando este fenómeno, las tripletas de las relaciones *nsubjpass* (2 tripletas) y *nmod:agent* (2 tripletas) son mezcladas, dando como resultado únicamente 3 tripletas pues se suprime una de las tripletas correspondientes al elemento a través del cual están relacionados, y que se muestran en la Figura 4-24.

```
?software <http://dbpedia.org/ontology/developer> ?organization.
?software a <http://dbpedia.org/ontology/Software>.
?organization a <http://dbpedia.org/ontology/Organisation>
```

Figura 4-24. Tripletas generadas para las relaciones sintácticas nsubjpass y nmod:agent.

Lo mismo ocurre para las dos relaciones restantes, solo que, en este caso la relación *acl* indica que el dominio de la relación *nmod:in* es una organización (*organization*), por lo que las tripletas resultantes son las mostradas en la Figura 4-25.

```
?organization <http://dbpedia.org/ontology/foundationPlace>
<http://dbpedia.org/resource/California>.
?organization a <http://dbpedia.org/ontology/Organisation>
```

Figura 4-25. Tripletas generadas para las relaciones sintácticas acl y nmod:in.

El proceso de buscar elementos en común entre las dependencias debe continuar hasta que se recorran todas las relaciones. Como se observa en la Figura 4-25, las tripletas mostradas comparten un elemento que es *organizations*, por los que las tripletas de ambos grupos generadas en los pasos anteriores se mezclan. En este caso existe una tripleta en común (la que hace referencia a la organización), por lo que una de ellas es eliminada. Esta fue la última relación existente, por lo que las tripletas finales quedan de la forma en que se muestran en la Figura 4-26.

```
?software <http://dbpedia.org/ontology/developer> ?organization.
?software a <http://dbpedia.org/ontology/Software>.
?organization a <http://dbpedia.org/ontology/Organisation>.
?organization <http://dbpedia.org/ontology/foundationPlace>
<http://dbpedia.org/resource/California>
```

Figura 4-26. Tripletas RDF resultantes tras la unión de dos grupos generales de tripletas RDF.

Una vez generadas las tripletas que forman el patrón de grafo a ser obtenido, este módulo genera la cláusula SELECT tomando en cuenta la relación *det(software, Which)*, cuyo patrón coincide con los que se muestran en la Tabla 4-8. A partir de este patrón se genera la consulta SPARQL que obtendrá información relacionada con el tópico software. De esta manera, la consulta SPARQL final se muestra en la Figura 4-27.

```
SELECT ?software
WHERE {
  ?software <http://dbpedia.org/ontology/developer> ?organization.
  ?software a <http://dbpedia.org/ontology/Software>.
  ?organization a <http://dbpedia.org/ontology/Organisation>.
  ?organization <http://dbpedia.org/ontology/foundationPlace>
<http://dbpedia.org/resource/California>
}
```

Figura 4-27. Consulta SPARQL final.

Una vez generada la consulta, el paso final consiste en ejecutar dicha consulta contra la base de conocimiento y mostrar el resultado. Esta consulta devuelve una lista con 144 instancias de la clase *Software*. En la Tabla 4-11 se muestra un extracto de los resultados obtenidos al ejecutar la consulta anterior contra la base de conocimiento de DBpedia.

Tabla 4-11. Extracto de los resultados de DBpedia para una consulta SPARQL.

Dependencia
http://dbpedia.org/resource/Berkeley_DB
http://dbpedia.org/resource/IRIX
http://dbpedia.org/resource/MySQL
http://dbpedia.org/resource/Oracle_Application_Server
http://dbpedia.org/resource/GlassFish
http://dbpedia.org/resource/OpenSolaris
http://dbpedia.org/resource/Oracle_Grid_Engine
http://dbpedia.org/resource/Oracle_Spatial_and_Graph
http://dbpedia.org/resource/The_Lord_of_the_Rings:_Conqu
http://dbpedia.org/resource/Full_Spectrum_Warrior
http://dbpedia.org/resource/Java_Advanced_Imaging

Dependencia
http://dbpedia.org/resource/Oracle_Linux
http://dbpedia.org/resource/Army_Men:_RTS
http://dbpedia.org/resource/TurboTax
http://dbpedia.org/resource/Oracle_Media_Objects
http://dbpedia.org/resource/REX_OS
http://dbpedia.org/resource/Mercenaries_2:_World_in_Flam
http://dbpedia.org/resource/Borland_C++
http://dbpedia.org/resource/Open64
http://dbpedia.org/resource/Oracle_Application_Express
http://dbpedia.org/resource/MySQL_Cluster
http://dbpedia.org/resource/Oracle_BI_Publisher
http://dbpedia.org/resource/PeopleTools
http://dbpedia.org/resource/Jinitiator
http://dbpedia.org/resource/Oracle_iPlanet_Web_Server
http://dbpedia.org/resource/FireChat
http://dbpedia.org/resource/HK2_DI_Kernel
http://dbpedia.org/resource/WebScaleSQL
http://dbpedia.org/resource/Turbo_Assembler
http://dbpedia.org/resource/Tropico_2:_Pirate_Cove

4.3 Resumen

En este capítulo se presentó una interfaz de lenguaje natural para bases de conocimiento semánticas que representa un esfuerzo de investigación por reducir la brecha existente entre este tipo de fuentes de información y usuarios sin experiencia o conocimiento de tecnologías de la Web Semántica.

El capítulo comienza por presentar la arquitectura de la interfaz, seguida por una descripción del funcionamiento general de la misma. A continuación, se describen en detalle cada uno de los módulos que componen dicha interfaz. El modelo ontológico de la pregunta (sección 4.2.1) permite describir la estructura sintáctica de la pregunta y el contexto de la misma en términos de la base de conocimiento del dominio. Toda la información almacenada en este modelo es obtenida a partir de la pregunta provista por el usuario mediante el análisis de la misma aplicando técnicas de PLN. Este proceso de análisis es llevado a cabo por el módulo de procesamiento de la pregunta (sección 4.2.3) a

través de ocho fases que son tokenización, etiquetado POS, lematización, reconocimiento de entidades nombradas, detección de elementos de la base de conocimiento (clases, propiedades e individuos), búsqueda de sinónimos, el análisis de dependencias y desambiguación.

En la sección 4.2.4 se describe el módulo de clasificación de la pregunta que parte de la idea de que toda la información de la estructura y contexto de la pregunta se encuentra ya en el modelo de la pregunta, pues es en base a estos datos, a una clasificación de preguntas y respuestas adaptada al dominio de las bases de conocimiento y a un conjunto de reglas, que le es posible llevar a cabo su tarea. Finalmente, en la 4.2.5 se describe el proceso de generación de consulta en lenguaje formal, concretamente SPARQL, el cual es guiado por las relaciones de dependencia sintáctica existentes entre los elementos contenidos en el modelo de la pregunta, las cuales de igual manera son descritas en este modelo.

Capítulo 5. Validación de la interfaz de lenguaje natural para bases de conocimiento basadas en ontologías

5.1 Introducción

A diferencia de los sistemas de búsqueda y recuperación de información, la interfaz de lenguaje natural para bases de conocimiento semánticas propuesta en este trabajo de tesis doctoral tiene como objetivo obtener la respuesta correcta en vez de proveer solo una lista de documentos relacionados o sus sustitutos. A partir de esta hipótesis, en este apartado se describen los experimentos llevados a cabo con el objetivo de medir la efectividad de la interfaz propuesta para encontrar la respuesta correcta a partir de una base de conocimiento. Los experimentos se llevaron a cabo en dos bases de conocimiento bien diferenciadas, a saber, DBpedia y MusicBrainz.

El primero de los escenarios de validación corresponde a la ontología y conjunto de datos de DBpedia en su versión para el idioma inglés. El segundo escenario de validación corresponde al dominio de la música, para lo cual se empleó el conjunto de datos y la correspondiente ontología de MusicBrainz, una base de conocimiento ampliamente utilizada como fuente de información de URIs relacionadas con la música en la comunidad de Linked Data (Swartz 2002). Uno de los objetivos de este segundo experimento es demostrar la portabilidad de la interfaz propuesta, es decir, determinar si el cambio de dominio afecta el rendimiento del método propuesto.

La ejecución de los experimentos ha requerido la previa recopilación de un corpus de preguntas para cada uno de los dominios establecidos, tarea que demandó la participación de usuarios ajenos al trabajo de investigación. Posteriormente, un grupo de expertos generó de manera manual la correspondiente consulta SPARQL para cada una de las preguntas recolectadas. Después, las preguntas en lenguaje natural se ejecutaron contra la interfaz desarrollada y los resultados obtenidos fueron comparados con los provistos por las consultas SPARQL generadas previamente. Finalmente, para llevar a cabo la validación se utilizaron las métricas de precisión, exhaustividad y su medida armónica conocida como medida-F (Yang and Liu 1999), las cuales son comúnmente aplicadas en

experimentos de recuperación de información (Hripcsak and Rothschild 2005) y por investigadores en el contexto de interfaces de lenguaje natural (Minock, Olofsson, and Näslund 2008).

Este capítulo se encuentra estructurado de la siguiente manera. En la siguiente sección se describen las bases de conocimiento que se utilizarán para evaluar el método propuesto. Posteriormente, se describe como se aplica la metodología de validación. Después se describen los corpus de preguntas utilizados en esta tarea. A continuación, se presentan las métricas de evaluación estándar utilizadas para medir la efectividad de la interfaz para proveer la respuesta correcta. Finalmente, los resultados de evaluación obtenidos en ambos dominios son presentados y discutidos.

5.2 Bases de conocimiento

5.2.1 Dbpedia

El proyecto DBpedia extrae conocimiento de 111 ediciones de Wikipedia en diferentes lenguajes. La versión más amplia de DBpedia es la correspondiente a la edición en inglés de Wikipedia. Esta versión de DBpedia describe 4.58 millones de recursos, de las cuales 4.22 millones están clasificadas en una ontología consistente, de la cual se hablará más adelante.

El proyecto DBpedia genera un conjunto de URIs por cada artículo disponible en Wikipedia. Estas URIs representan los conceptos descritos en una página en particular. Por ejemplo, para el artículo de Wikipedia correspondiente a Rafael Nadal cuya dirección Web es *https://en.wikipedia.org/wiki/Rafael_Nadal,* DBpedia genera la URI *http://dbpedia.org/page/Rafael_Nadal*. Hasta el año 2011, las URIs eran publicadas solo bajo el dominio de http://dbpedia.org. Los principales espacios de nombre eran:

- **http://dbpedia.org/resource/**. El prefijo de este espacio de nombres es *dbr*. Este permite representar los datos del artículo. En DBpedia, a cada recurso que es objeto de un artículo en Wikipedia se le asigna automáticamente una URI DBpedia, la cual se basa en la URI del artículo de Wikipedia. Por ejemplo, el artículo correspondiente a la ciudad de Murcia tiene la siguiente URI *https://en.wikipedia.org/wiki/Murcia*. A partir de esta URI se asigna la siguiente DBpedia URI *http://dbpedia.org/resource/Murcia* o *dbr:Murcia*, haciendo uso del prefijo.
- **http://dbpedia.org/property/**. Su prefijo es *dbp*. Este espacio de nombres permite representar propiedades extraídas de la ficha principal del artículo, la

cual generalmente se encuentra a la derecha de los artículos de Wikipedia. Un ejemplo de este tipo de propiedad es *dbp:populationTotal*.

- **http://dbpedia.org/ontology/**. Utiliza el prefijo *dbo*. Este espacio de nombres permite representar la ontología de DBpedia. Por ejemplo: *dbo:populationTotal*.

Estos espacios de nombres representan el núcleo de Wikipedia. Sin embargo, la integración con diversos lenguajes mostró que este enfoque omitía información muy valiosa (Kontokostas et al. 2012). Por ejemplo, DBpedia extraía artículos en otro idioma que no fuera inglés, solo si este proveía un enlace inter-idioma en inglés y los recursos creados usan el espacio de nombres de DBpedia por defecto. De esta manera, existía solo una relación unidireccional del recurso en otro idioma diferente al inglés al artículo en inglés, y no de manera bidireccional, una característica más apropiada en este contexto (Erdmann et al. 2008). Así, a partir de la versión 3.7 de DBpedia, se generaron dos tipos de conjuntos de datos, que se explican a continuación:

- **Conjuntos de datos localizados**. Estos contienen todas las cosas descritas en un idioma en específico. En este conjunto de datos, las cosas son identificadas a través de URIs específicas tales como *http://<lang>.dbpedia.org/resource/* para los datos de un artículo, y *http://<lang>.dbpedia.org/property/* para las propiedades de datos. Donde *lang* especifica el idioma. Por ejemplo, para el idioma español se tiene la siguiente URI *http://es.dbpedia.org/resource/*.
- **Conjuntos de datos en forma canónica**. Estos conjuntos de datos contienen únicamente cosas por las cuales existe su página correspondiente en la edición de inglés de Wikipedia. En estos conjuntos de datos, las cosas con identificadas con la URI genérica *http://dbpedia.org/resource*.

DBpedia emplea diversos extractores de información para traducir diferentes partes de los artículos de Wikipedia a sentencias RDF. Una de las partes destacables de un artículo de Wikipedia es lo que se denomina como *Ficha*, y aparece generalmente al lado derecho del artículo, tal como se resalta en la Figura 5-1.

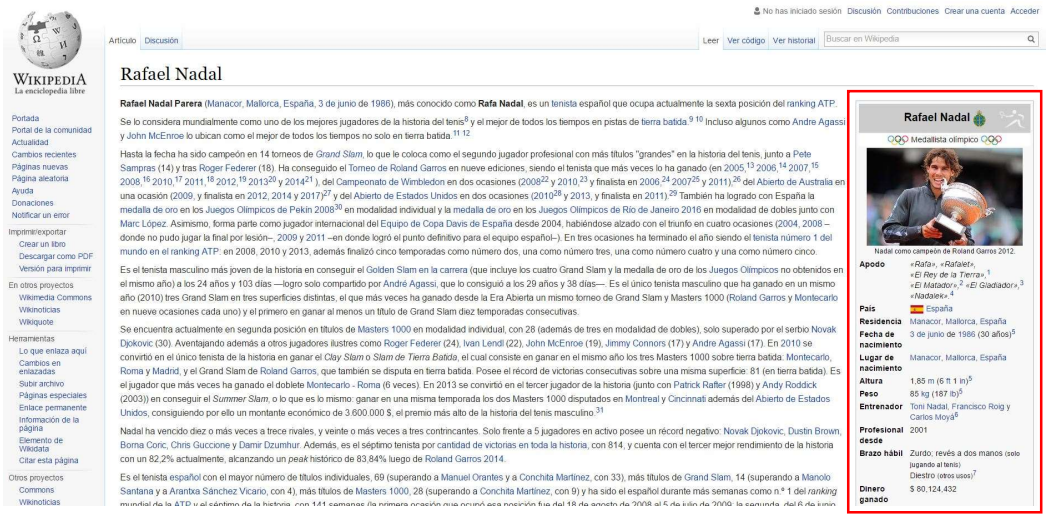

Figura 5-1. Ejemplo de artículo de Wikipedia.

Los extractores de información se pueden dividir en cuatro categorías principales:

- **Extracción de ficha basada en el mapeo**. Esta usa mapeos escritos manualmente que relacionan las fichas del artículo en Wikipedia con los términos de la ontología. La ficha del artículo aparece generalmente en la parte derecha del artículo. Estos mapeos especifican un tipo de datos para cada propiedad contenida en la ficha, lo que ayuda a DBpedia a extraer información de alta calidad.
- **Extracción de ficha en bruto**. Este provee un mapeo directo de la ficha del artículo de Wikipedia a RDF. Aunque la calidad de esta información es de bajo nivel, esta es útil si una ficha aún no ha sido mapeada, por lo que no está disponible en la extracción basada en el mapeo.
- **Extracción de características**. Esta utiliza un número de extractores especializados en una única característica del artículo, tal como la propiedad *label*.
- **Extracción estadística**. Estos extractores están enfocados a tareas de PLN tales como recuperación de información, tareas de desambiguación, extracción de relaciones multilingües, entre otros (Mendes, Jakob, and Bizer 2012). Algunos extractores agregan información de todas las páginas de Wikipedia con el objetivo de proveer datos basados en medidas estadísticas tal como tf-idf (*Term frequency – Inverse document frequency*) con el cual se puede medir cuán relevante es una palabra para un recurso de DBpedia.

5.2.1.1 La ontología de DBpedia

La ontología de DBpedia es una ontología multidominio que ha sido creada de forma manual con base en las fichas de los artículos más utilizados dentro de Wikipedia. Esta ontología contiene 685 clases las cuales están descritas por 2795 propiedades diferentes. Además, esta contiene cerca de 4,233,000 de instancias. La Tabla 5-1 resume la cantidad de instancias por clase de la ontología DBpedia.

Tabla 5-1. Instancias por clase de la ontología de DBpedia.

Clase	Instancias
Recursos en general	4,233,000
Lugar	735,000
Persona	1,450,000
Trabajo	411,000
Especies	251,000
Organización	241,000

A lo largo de su historia, la ontología de DBpedia ha sufrido grandes cambios, por ejemplo, la versión 3.2 introdujo un nuevo método de extracción de fichas de artículo. Este método se basaba en mapeos generados manualmente de las fichas de artículos de Wikipedia a la ontología de DBpedia. A partir de estos mapeos se definieron reglas de grano fino respecto a cómo analizar los valores de las fichas de los artículos. Estos mapeos hicieron hincapié en ciertas debilidades del sistema de la ficha de artículo de Wikipedia, tal como el tener diferentes fichas para la misma clase, utilizar diferentes nombres para la misma propiedad, o no tener claramente definidos los tipos de datos para los valores de las propiedades.

Para la versión 3.5 se introdujo una wiki pública que permite a contribuidores externos la escritura de mapeos de fichas de artículos, así como la edición de la versión actual de la ontología. Finalmente, para la versión más actual de la ontología de DBpedia (3.7) esta es un grafo dirigido acíclico o DAG (*Directed Acyclic Graph*) y no un árbol, por lo que las clases pueden tener múltiples superclases.

5.2.2 MusicBrainz

La música representa un tópico ampliamente utilizado en la vida diaria de mucha gente y organizaciones. Es por ello que existe una necesidad de contar con un mecanismo que permita identificar de manera confiable y no ambigua recursos relacionados con la música permitiendo así a máquinas y humanos interactuar de una manera significativa. Para responder a este problema surge MusicBrainz (Swartz 2002), un esfuerzo comunitario por

proveer una manera de identificación de la música y hacerla disponible al público. MusicBrainz es una enciclopedia abierta de música definida por la NGS (*Next Generation Schema*) que representa el esquema de la base de datos, y la ARs (*Advanced Relationships*), la cual representa tanto las relaciones existentes entre las entidades, como las relaciones existentes entre entidades y recursos fuera de la base de datos de MusicBrainz.

Actualmente, MusicBrainz es ampliamente utilizada en la comunidad de Linked data como una fuente de información de URIs relacionadas con la música (Swartz 2002). Sin embargo, existe una tendencia por parte de usuarios de Linked Data por construir nuevas URIs basadas en los identificadores de MusicBrainz, esto se debe a que Linked Data no los provee directamente. En este sentido, LinkedBrainz (Jacobson, Dixon, and Sandler 2010) surge como un esfuerzo por proveer el contenido de MusicBrainz siguiendo los principios de Linked Data, es decir, publicar la información contenida en la base de datos de MusicBrainz como información estructurada en la Web empleando tecnologías de la Web Semántica. De esta manera, el objetivo primordial de LinkedBrainz es proveer un mapeo del esquema de la base de datos definida por la NGS y las relaciones ARs a la tecnología RDF. Para cumplir con este objetivo, LinkedBrainz mapea conceptos descritos en MusicBrainz con conceptos descritos por la ontología Music Ontology (Raimond et al. 2007), de la cual se habla en la siguiente sección.

5.2.2.1 Music Ontology

La ontología Music Ontology provee un marco de trabajo para publicar información estructurada relacionada con la música, la cual va desde datos editoriales hasta anotaciones temporales de señales de audio (Raimond and Sandler 2012). Esta ontología ha sido utilizada por proyectos tales como DBTune (Raimond, Sandler, and Mary 2008) y el sitio web de música de la BBC.

La especificación de la Music Ontology define 54 clases que permiten describir conceptos del dominio de la música tales como artistas, grupos musicales, compositores, géneros, instrumentos musicales, lanzamientos, canciones, entre otros. Además, provee 153 propiedades para describir dichos conceptos tales como codificación, discografía, duración, sitios de descarga, entre otros. Music Ontology a su vez está basada en las siguientes ontologías:

- **FOAF** (Brickley and Miller 2012). FOAF es un proyecto dedicado a vincular personas e información a través de la Web. Los términos principales de FOAF son agrupados en tres categorías amplias.

  - ***Core***. Las clases y propiedades correspondientes a este grupo permiten describir características de gente y grupos sociales que son independientes del tiempo y tecnología, de tal manera que puedan ser utilizadas para describir información básica referente a la gente en la actualidad, la historia, patrimonio cultural y contextos de bibliotecas digitales. Entre las clases más sobresalientes de este grupo se encuentran *Agent*, *Person*, *Project*, *Organization*, entre otros.
  - ***Social Web***. Además de los términos del grupo anterior, FOAF provee una serie de términos que permiten describir cuentas de internet, libretas de direcciones y otras actividades basadas en la Web. Algunos de estos términos son *Nick*, *homepage*, *workplaceHomepage*, entre otros.
  - ***Utilidades de Linked Data***. FOAF es un elemento importante para el crecimiento de Linked Data. Este grupo registra términos útiles para la comunidad Web, haciendo énfasis en la idea central de FOAF, que consiste en vincular redes de información con redes de personas. Ejemplos de estos términos son *geekcode*, *focus* y *LabelProperty*.

- **Event Ontology** (Raimond and Abdallah 2007). Esta ontología provee un vocabulario para describir eventos, definidos como la forma en que los agentes cognitivos clasifican las regiones tiempo/espacio. Esta ontología permite describir los eventos físicos existentes en el proceso de producción musical, que se producen en un determinado lugar y tiempo y que pueden implicar la participación de una serie de objetos físicos tanto animados como inanimados. Dichos eventos incluyen presentaciones, los cuales involucran la participación de músicos e instrumentos.
- **Timeline Ontology** (Raimond and Abdallah 2006). Esta ontología provee un vocabulario para describir intervalos e instantes de tiempo en múltiples líneas de tiempo (posiblemente relacionadas). Esta ontología es utilizada por la Music Ontology para resolver la necesidad de representar información temporal relacionada con la música. Por ejemplo, permite expresar la fecha de lanzamiento de una canción específica utilizando la línea de tiempo física, o información tal como *de 0 a 10 segundos en esa canción específica* utilizando la línea de tiempo de la señal de audio. Entre las clases más relevantes se encuentran *TimeLine*, *Instant* e *Interval*.
- **FRBR Ontology** (Boeuf 2001). (*Functional Requirements for Bibliographic Records*) Esta ontología provee un vocabulario para describir trabajos, expresiones, manifestaciones y sus relaciones, tal como se definen en los

requisitos funcionales para registros bibliográficos. Music Ontology utiliza esta ontología principalmente por sus conceptos *Work*, *Manifestation* e *Item*, los cuales permiten representar una creación artística, un registro musical, o algún vinilo particular, respectivamente.

5.3 Metodología de validación

La interfaz de lenguaje natural propuesta en este trabajo de tesis doctoral tiene como objetivo proveer la respuesta correcta a una pregunta expresada en lenguaje natural a partir de una base de conocimiento basada en ontologías. En este sentido, se ha llevado a cabo una evaluación para determinar la precisión de la interfaz presentada en esta tesis. Este proceso se compone de cuatro tareas fundamentales que se describen a continuación.

1. **Recolección del corpus de preguntas**. Esta tarea consistió en recopilar un conjunto de preguntas expresadas en lenguaje natural para cada uno de los contextos en los que sería evaluada la interfaz, es decir, DBpedia y MusicBrainz. Una gran parte de las preguntas recolectadas corresponden a los corpus provistos por la campaña de evaluación QALD (Lopez et al. 2013) correspondientes a su segunda y tercera edición. La razón de no considerar por completo estos corpus es que la cantidad de preguntas existentes para algunos de los tipos de preguntas soportados por la interfaz propuesta en este trabajo era muy baja y en algunos casos nulo, lo que impediría realizar un análisis detallado de las fortalezas o debilidades de la interfaz. A las preguntas obtenidas a través de QALD, se añadió un conjunto generado por un grupo de estudiantes de maestría de la Facultad de Informática de la Universidad de Murcia. Con el objetivo de que las preguntas provistas por estos usuarios estuvieran dentro de los dominios consideradas. Para ello, se les proporcionó una descripción de la información contenida en estas.
2. **Generación de consultas SPARQL**. Un grupo de expertos en lenguajes de consulta y bases de conocimiento basadas en ontologías generó la consulta SPARQL para recuperar la información relacionada con cada una de las preguntas generadas por los estudiantes involucrados en la tarea anterior. En términos de evaluación, la información recuperada de esta consulta se considera la respuesta correcta. En cuanto a las preguntas obtenidas de los retos QALD, cabe mencionar que estas ya proveen sus correspondientes consultas SPARQL.
3. **Ejecución de las preguntas en lenguaje natural contra la interfaz propuesta**. Básicamente, esta tarea se refiere al hecho de proveer a la interfaz

de lenguaje natural desarrollada cada una de las preguntas contenidas en los corpus y almacenar la información provista como respuesta.

4. **Evaluación de la interfaz**. Las respuestas provistas por la interfaz para cada una de las preguntas del corpus se compararon con la información recuperada a través de sus respectivas consultas SPARQL. Posteriormente, se evaluó la interfaz mediante las métricas de evaluación estándar precisión, exhaustividad y medida-F.

5.4 Recolección del corpus de preguntas

5.4.1 Proceso de obtención del corpus

El proceso de validación de la interfaz propuesta en esta tesis doctoral requirió de un corpus de preguntas expresadas en lenguaje natural concernientes a las bases de conocimiento de DBpedia y MusicBrainz. La generación de este corpus consistió en dos pasos: 1) búsqueda de corpus de preguntas disponibles en la Web, entre los que destacaron los provistos por la campaña QALD (Lopez et al. 2013) correspondientes a su segunda y tercera edición; y 2) generación de preguntas por parte de personas ajenas a este trabajo de investigación. A continuación, se describe a detalle el proceso de recolección de preguntas.

La campaña QALD provee corpus orientados a DBpedia y MusicBrainz. Sin embargo, la decisión de no considerar los corpus completos se basa en dos razones principales. En primer lugar, algunos de las preguntas contenidas en ellos hacen referencia a conceptos que no pertenecen a las ontologías del dominio. Esta característica fue incluida por la campaña con el objetivo de agregar el reto de identificar preguntas fuera del alcance de la ontología, un reto interesante pero que no entra en los objetivos específicos de la interfaz propuesta en este trabajo de tesis doctoral. En segundo lugar, la cantidad de preguntas contenidas en estos corpus para algunos de los tipos de pregunta soportados por nuestra propuesta era muy baja y en ocasiones nula. Esta situación impediría identificar de manera más específica las fortalezas y debilidades de nuestro enfoque para cada uno de los tipos de pregunta considerados.

Para obtener al menos 10 preguntas de cada uno de los tipos soportados por la interfaz, se le solicitó a un grupo de estudiantes de maestría de la Facultad de Informática de la Universidad de Murcia generar preguntas en lenguaje natural que pudieran ser respondidas a través del contenido disponible en DBPedia y MusicBrainz. Para cumplir con este objetivo, se les proporcionó una descripción de la información provista por estas.

El resultado final son dos corpus compuestos por 100 preguntas cada uno, donde para cada uno de ellos, se consideraron 50 preguntas de los corpus de QALD y el restante lo conforman las preguntas provistas por los estudiantes involucrados en el proceso.

5.4.2 Corpus de preguntas de DBpedia

En la Tabla 5-2 se muestra una descripción del corpus de preguntas para DBpedia, donde se logra apreciar la cantidad de preguntas para cada uno de los tipos considerados. Además, se proveen algunos ejemplos de las preguntas contenidas en este.

Tabla 5-2. Descripción del corpus de preguntas para el dominio de DBpedia.

Tipo	Cantidad	Ejemplos
What	17	What is the currency of the Czech Republic? What is the area code of Berlin?
Who	13	Who is the mayor of Berlin? Who wrote the book The pillars of the Earth?
How	14	How many people live in the capital of Australia? How tall is Michael Jordan?
Where	10	Where did Abraham Lincoln die?
When	10	When did Michael Jackson die? When was the Battle of Gettysburg?
Which	24	Which river does the Brooklyn Bridge cross? Which countries have places with more than two caves?
Give/List/Name	12	Give me all movies directed by Francis Ford Coppola. Give me all members of Prodigy.

Como se puede observar en la tabla anterior, la categoría con más preguntas es la de *Which* con un total de 24, mientras que las categorías con menor número de preguntas recolectadas son *Where* y *When*, con 10 cada una. En la Figura 5-2 se presenta de manera gráfica la cantidad de preguntas recolectadas para cada uno de los tipos considerados.

Figura 5-2. Corpus de preguntas para DBpedia.

5.4.3 Corpus de MusicBrainz

En la Tabla 5-3 se muestra un resumen del corpus obtenido para el dominio de MusicBrainz, así como algunos ejemplos de las preguntas contenidas en el.

Tabla 5-3. Descripción del corpus de preguntas para el dominio de MusicBrainz

Tipo	Cantidad	Ejemplos
What	16	What is the legal name of Loona? What is the longest song by Eminem?
Who	14	Who produced the album Infinite? Who was Whitney Houston's husband?
How	11	How old is Paul McCartney? How many tracks does Nirvana have?
Where	10	Where was born Frank Sinatra?
When	12	When did Ludwig van Beethoven die? When were the Dixie Chicks founded?
Which	26	Which singles did Slayer release? Which groups was David Bowie a member of?
Give/List/Name	11	Give me the titles of all singles by Phil Collins. Give me all Kraftwerk albums

En la tabla anterior se aprecia, que al igual que sucedió con el corpus de preguntas para el dominio de DBpedia, el grupo con mayor número de preguntas fue *Which*, con un total de 26. Este hecho nos indica una clara tendencia hacia el uso de este determinante (*Which*) para expresar necesidades de información. Por otro lado, el tipo de preguntas con un menor número corresponde a *Where*, con 10, seguida de los tipos *How* y *Give* con 11 para cada una. En la Figura 5-3 se aprecia de manera gráfica la cantidad de preguntas provistas para cada uno de los tipos de pregunta soportados por nuestro enfoque.

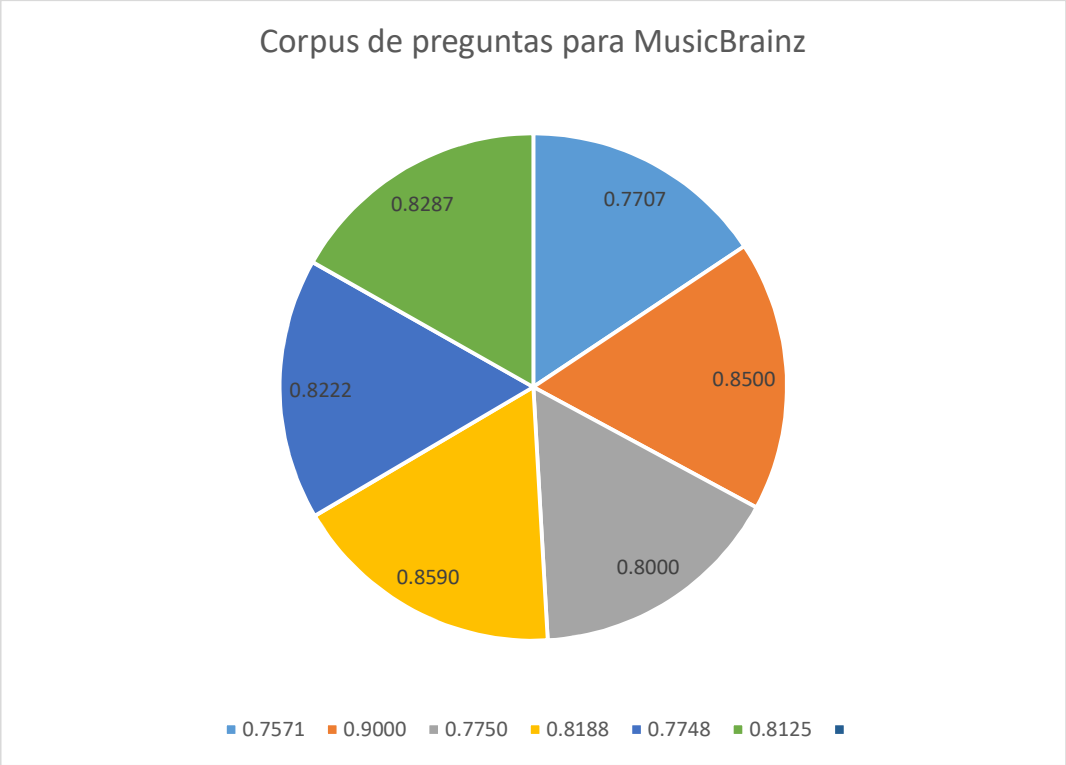

Figura 5-3. Corpus de preguntas para MusicBrainz.

5.5 Generación de consultas SPARQL

Con respecto a la consulta SPARQL de las preguntas, cabe mencionar que los corpus de QALD ya proveen dicha información. Para el caso de las preguntas solicitadas a los estudiantes, un grupo de expertos en tecnologías de la Web Semántica y lenguajes de consulta generó las correspondientes consultas. La consulta SPARQL es de vital importancia para el proceso de validación de la interfaz, ya que la información recuperada a través de ésta es considerada la respuesta correcta, por lo que el proceso de evaluación consistirá en comparar dicha información con los resultados provistos por la interfaz propuesta en esta tesis.

5.5.1 Representación de los corpus en formato XML

Los corpus generados se almacenaron en un archivo en formato XML que describe las características principales relacionadas con la pregunta, tales como su clase principal, su subclase, el tipo de respuesta esperada, las entidades de la base de conocimiento que deben ser identificadas en ella, y la consulta SPARQL que permite obtener la respuesta correcta. Un ejemplo de pregunta descrita en formato XML se muestra en la Figura 5-4.

```
<question id="74" questionclass="which" questionsubclass="which-
what" answertype="dboWork">
  <string>Which software has been developed by organizations founded
in California?</string>
  <KnowledgeBaseEntity>
    <KnowledgeBaseConcept>http://dbpedia.org/ontology/Software
</KnowledgeBaseConcept>
            <KnowledgeBaseConcept>
                  http://dbpedia.org/ontology/Organisation
            </KnowledgeBaseConcept>
            <KnowledgeBaseObjectProperty>
                  http://dbpedia.org/ontology/developer
            </KnowledgeBaseObjectProperty>
            <KnowledgeBaseObjectProperty>
                  http://dbpedia.org/ontology/foundationPlace
            </KnowledgeBaseObjectProperty>
            <KnowledgeBaseIndividual>
                  http://dbpedia.org/resource/California
            </KnowledgeBaseIndividual>
      </KnowledgeBaseEntity>
      <sparqlquery><![CDATA[
            SELECT ?software
            WHERE {
                  ?software <http://dbpedia.org/ontology/developer>
?organization.
                  ?software a <http://dbpedia.org/ontology/Software>.
                  ?organization a
<http://dbpedia.org/ontology/Organisation>
                  ?organisation
<http://dbpedia.org/ontology/foundationPlace>
<http://dbpedia.org/resource/California>
            } ]]>
      </sparqlquery>
</question>
```

Figura 5-4. Registro XML de una pregunta en el corpus de preguntas de DBpedia.

5.6 Medidas de evaluación

Existen diversos enfoques para llevar a cabo la evaluación de interfaces de lenguaje natural orientadas a fuentes de información estructurada tales como bases de datos relacionales o las bases de conocimiento basadas en ontologías. En este trabajo de tesis doctoral se empleó un conjunto de métricas de evaluación estándar, a saber, precisión, exhaustividad y su medida armónica conocida como medida-F, con el objetivo de medir la

efectividad de la interfaz propuesta para recuperar la información de la base de conocimiento que responda correctamente una pregunta expresada en lenguaje natural. Las métricas antes mencionadas fueron propuestas inicialmente por (Salton and McGill 1986) y desde entonces han sido ampliamente utilizadas en procesos de evaluación de sistemas de PLN y recuperación de información.

A pesar de que las métricas en cuestión estaban enfocadas a la evaluación del rendimiento de sistemas de búsqueda y recuperación de información, así como el reconocimiento de patrones, actualmente un gran número de áreas de investigación las han adaptado para medir el rendimiento de sistemas enfocados a contextos tales como la bioquímica (Alexopoulou et al. 2008), oftalmología (Milios et al. 2003), biomedicina (Krauthammer and Nenadic 2004), por mencionar solo algunos. A continuación, se provee una definición de las métricas de evaluación estándar a utilizar en este trabajo.

5.6.1 Definición de las métricas de evaluación estándar

5.6.1.1 Precisión

Existen diversas definiciones de la métrica de precisión, por ejemplo, de acuerdo con (Clarke and Willett 1997) la precisión se puede definir como: "*la fracción de una salida de búsqueda que es relevante para una consulta determinada*". Y, por tanto, su cálculo requiere el conocimiento de los éxitos relevantes y no relevantes en el conjunto de documentos evaluados. Por otro lado, los autores Salton y McGill (Salton and McGill 1986) la definen como: "*una medida de exactitud que determina la fracción de entidades relevantes de todas las entidades recuperadas en un sistema de extracción de información*". Tomando en cuenta las definiciones anteriores, la precisión se puede calcular mediante la fórmula mostrada a continuación:

$$precisión = \frac{|entidades\ extraídas| \cap |entidades\ relevantes|}{|entidades\ extraídas|}$$

5.6.1.2 Exhaustividad

Clarke y Willett (Clarke and Willett 1997) definen a la exhaustividad como: "*la capacidad de que un sistema de recuperación tiene para obtener todos o la mayoría de los documentos relevantes en una colección*". Por lo tanto, se requiere del conocimiento de los documentos no solo relevantes y recuperados, sino también, de aquellos que no fueron recuperados. Así, la exhaustividad puede ser calculada a través de la siguiente fórmula:

$$exhaustividad = \frac{|entidades\ extraídas| \cap |entidades\ relevantes|}{|entidades\ relevantes|}$$

5.6.1.3 Medida-F

Por último, la Medida-F, también conocida como *F-measure*, *balanced F-Score* o F_1 *measure*, es la media armónica de los valores de precisión y exhaustividad (Yang and Liu 1999). Concretamente, esta métrica se emplea para evaluar el rendimiento global de las dos métricas descritas anteriormente. Acorde a su definición, la medida-F se calcula a partir de la siguiente fórmula, y la cual provee un resultado entre 0 y 1.

$$medida - F = 2\,\frac{precisión\ x\ exhaustividad}{precisión + exhaustividad}$$

5.6.2 Adaptación de las métricas de evaluación

Como se mencionó anteriormente, el proceso de validación incluyó la recolección de un conjunto de preguntas enfocadas a cada uno de los dominios considerados. Además, para cada pregunta fue generada una consulta SPARQL. De esta manera, los resultados obtenidos tras la ejecución de las preguntas de lenguaje natural a través de la interfaz propuesta son comparados con la información recuperada mediante su respectiva consulta SPARQL. A partir de estos resultados, se evalúa la interfaz haciendo uso de las métricas citadas en apartados anteriores. Así, la precisión y exhaustividad de la interfaz se calcula a partir de las siguientes fórmulas:

$$precisión = \frac{recursos\ correctos\ obtenidos\ por\ la\ interfaz}{total\ de\ recursos\ recuperados\ por\ la\ interfaz}$$

$$exhaustividad = \frac{recursos\ correctos\ obtenidos\ por\ la\ interfaz}{recursos\ obtenidos\ por\ la\ consulta\ SPARQL}$$

Antes de entrar en detalles de la evaluación, es necesario aclarar algunos conceptos que son determinantes para entender este proceso. La respuesta provista por la interfaz puede constar de uno o más recursos, donde un recurso es la URI de un elemento de la base de conocimiento, o una cadena de texto, un número o una fecha correspondiente al valor de una de sus propiedades. Dicho esto, la precisión se obtiene dividiendo el número de recursos correctos obtenidos por la interfaz, es decir, aquellos que forman parte de la respuesta correcta, entre el total de recursos obtenidos por la interfaz. Por otro lado, la exhaustividad se obtiene dividendo el número de recursos correctos obtenidos por la interfaz entre los recursos obtenidos por la consulta SPARQL, los cuales representan la respuesta correcta. Finalmente, la medida-F es calculada mediante la fórmula presentada en la sección anterior.

5.7 Evaluación de la interfaz de lenguaje natural para bases de conocimiento basadas en ontologías

Una vez recolectado el corpus de preguntas, el siguiente paso consistió en medir la efectividad de la interfaz presentada en este trabajo para proveer la respuesta correcta a preguntas expresadas en lenguaje natural. Para llevar a cabo esta tarea, se ejecutaron cada una de las preguntas en lenguaje natural usando la interfaz propuesta. El siguiente paso consistió en ejecutar la consulta SPARQL asociada a cada pregunta con el objetivo de obtener lo que se considera como la respuesta correcta. Finalmente, los resultados obtenidos por ambas tareas fueron comparadas y se llevó a cabo una evaluación mediante las métricas de precisión, exhaustividad y medida-F. A continuación, se presentan los resultados obtenidos en ambos dominios.

5.7.1 Resultados obtenidos en el dominio de DBpedia

En la Figura 5-5. se muestran los resultados obtenidos en el dominio de DBpedia para cada uno de los tipos de pregunta soportados por la interfaz presentada en este trabajo, a saber, *What*, *Who*, *How* (para los contextos ¿Cuánto? ¿Qué tan…?), *Where*, *When*, *Which*, y *Give*. Este último tipo de pregunta incluye aquellas solicitudes de información que son expresadas mediante oraciones imperativas, tales como *Give me the name…* (Dame el nombre…), *List all members…* (Lista todos los integrantes…), o *Name the...* (Nombra las…). Este tipo de pregunta fue considerado tras analizar las formas de solicitar información en lenguaje natural, siendo este uno de los más utilizados.

En la Figura 5-5 se observa que no existe diferencia significativa entre los resultados obtenidos para cada uno de los tipos de pregunta soportados por la interfaz presentada en esta tesis. Esto se puede traducir como una efectividad constante por parte de la interfaz para proveer la respuesta correcta sin importar el tipo de pregunta provista por el usuario.

Los mejores valores para las métricas de exhaustividad y medida-F fueron obtenidos por el tipo de pregunta *Who*, con valores de 0.8889 y 0.8828 respectivamente. Esto se puede traducir en el hecho de que el sistema es capaz de proveer la información correcta a la mayoría de preguntas cuya respuesta esperada corresponde con instancias de clases tales como persona, organización, entre otros. Por otro lado, el tipo de pregunta *How* obtuvo los más bajos resultados para las tres métricas utilizadas, con valores de 0.7571 de precisión, 0.7707 de exhaustividad y 0.7630 de medida-F.

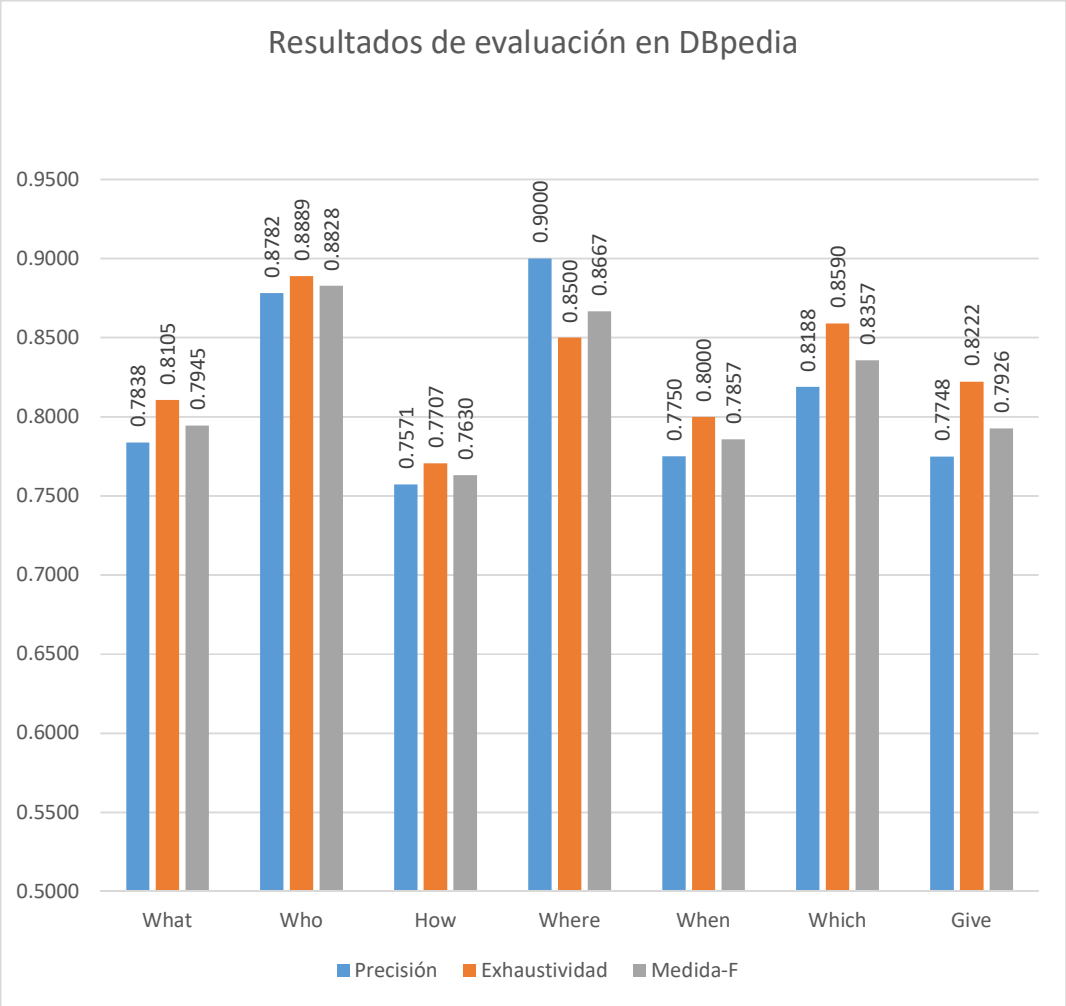

Figura 5-5. Resultados de evaluación en DBpedia.

En la Tabla 2-1 se presentan nuevamente los resultados obtenidos por la interfaz en el dominio de DBPedia. Sin embargo, esta vez se incluye la cantidad de preguntas por cada tipo de pregunta, se resaltan en negrita los mejores valores obtenidos, y se presenta el promedio general obtenido por la interfaz para las métricas de precisión, exhaustividad y medida-F, con valores de 0.8125, 0.8287 y 0.8173 respectivamente.

Tabla 5-4. Resultados de evaluación en DBpedia.

Tipo	Cantidad	Precisión	Exhaustividad	Medida-F
What	17	0.7838	0.8105	0.7945
Who	13	0.8782	**0.8889**	**0.8828**
How	14	0.7571	0.7707	0.7630
Where	10	**0.9000**	0.8500	0.8667
When	10	0.7750	0.8000	0.7857
Which	24	0.8188	0.8590	0.8357
Give	12	0.7748	0.8222	0.7926
Total	100			
Promedio		0.8125	0.8287	0.8173

Por otro lado, para fines de este proceso de evaluación, existen cuatro tipos de resultados que puede tener la interfaz en base a la respuesta provista, estos son; *correcta*, *incorrecta*, *fallida* y *parcial*. El tipo *fallida* se refiere a cuando la interfaz no pudo interpretar la pregunta. Por su parte, el término *parcial* se refiere al hecho que ocurre cuando, dada una pregunta cuya respuesta debe ser una lista de recursos, la respuesta completa provista por la interfaz corresponde a solo una parte de la respuesta correcta, o devuelve más de los esperados, incluyendo todos los que conforman la respuesta correcta. Dicho esto, en la Figura 5-6 se presentan la distribución de los tipos de resultado de acuerdo a las respuestas provistas por la interfaz en el contexto de DBpedia. Además, se aprecia que el número de preguntas que fueron respondidas correctamente es mucho mayor que los otros tres tipos juntos. Esto nos habla de la buena efectividad que obtuvo la interfaz dentro del dominio de DBpedia.

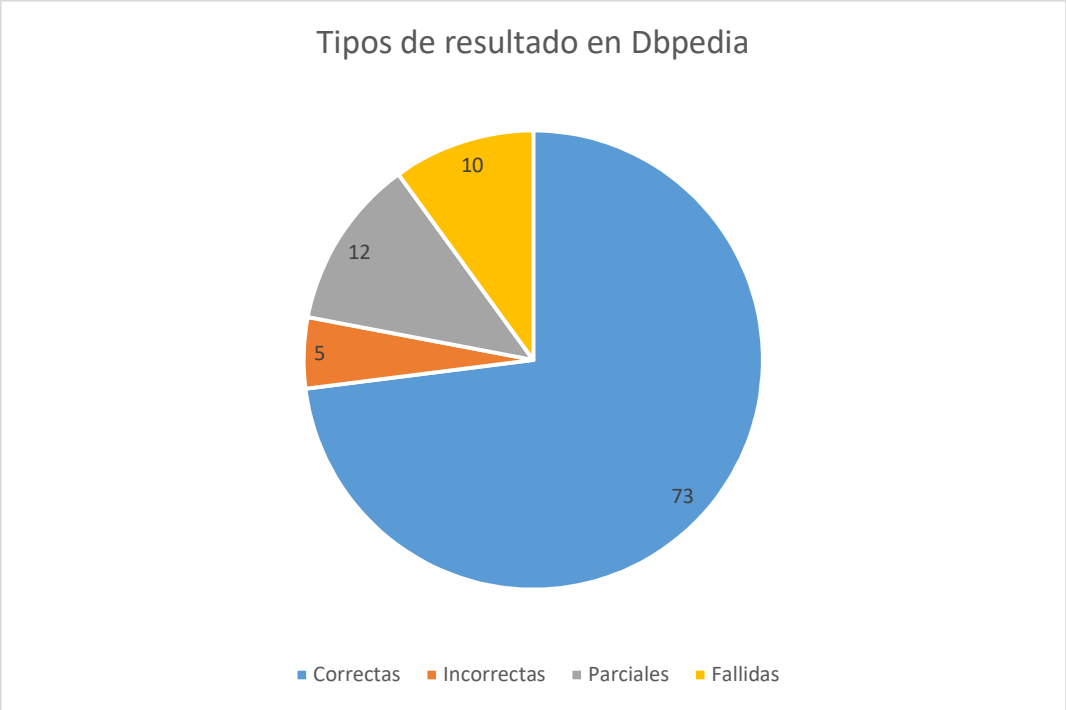

Figura 5-6. Tipos de resultado en DBpedia.

5.7.2 Resultados obtenidos en el dominio de MusicBrainz

En la Figura 5-7 se muestran los resultados obtenidos en el dominio de MusicBrainz para cada uno de los tipos de pregunta soportados por la interfaz. En dicha figura podemos observar que, al igual que en el dominio de DBpedia, los mejores resultados fueron obtenidos para el tipo de pregunta *Who* con valores de 0.8265, 0.8389 y 0.8319 para las métricas de precisión, exhaustividad y medida-F respectivamente. Por otro lado, el valor más bajo obtenido para la métrica de precisión corresponde al tipo de pregunta *How* con un valor de 0.7770. Con respecto a las métricas de exhaustividad y medida-F, los resultados más bajos fueron obtenidos por el tipo de pregunta *Give*, con valores de 0.7745 y 0.7721 respectivamente.

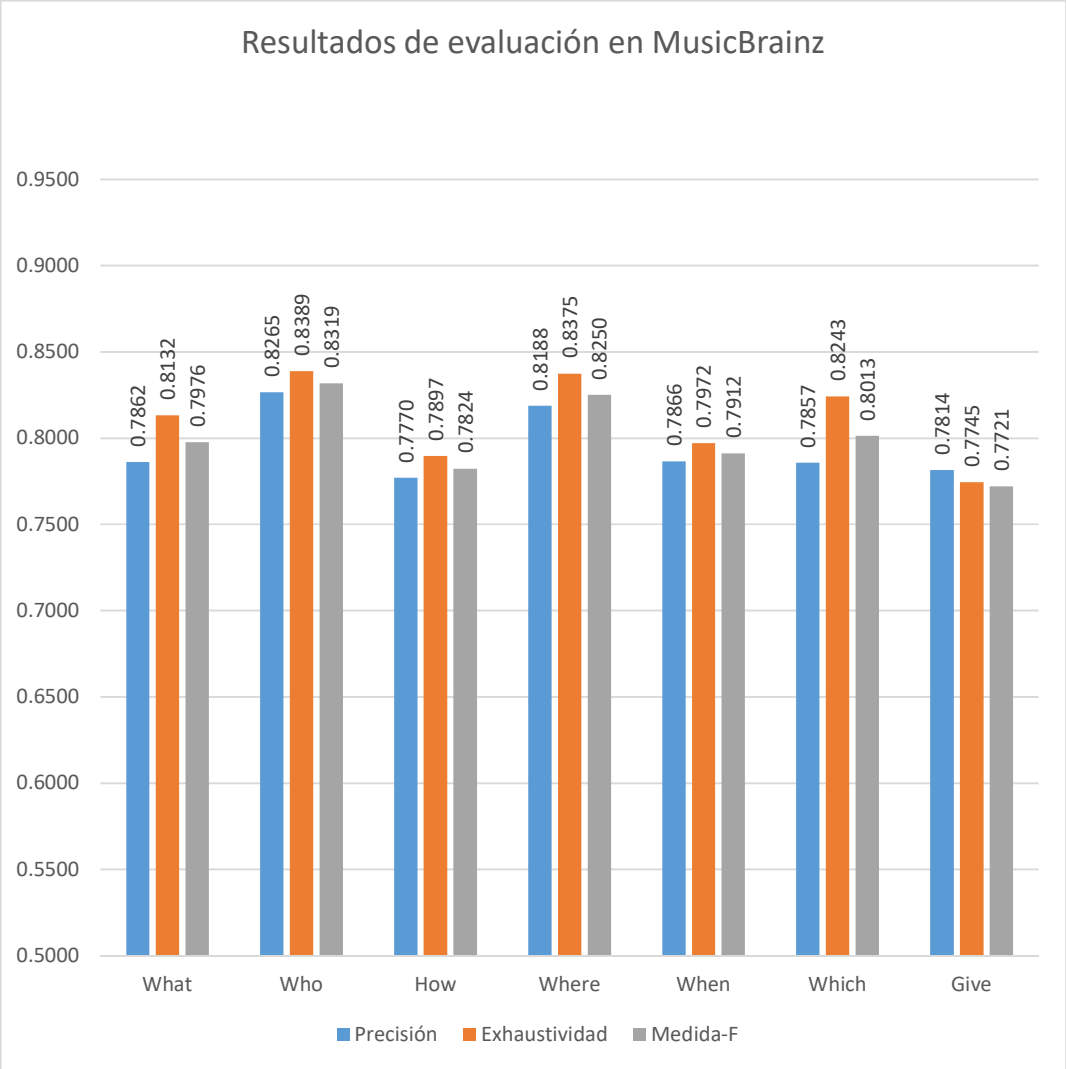

Figura 5-7. Resultados de evaluación en MusicBrainz.

Con el objetivo de proveer una mejor perspectiva de los resultados obtenidos con la base de conocimiento de MusicBrainz. La Tabla 5-5 presenta la cantidad de preguntas para grupo, se resaltan en negrita los mejores valores obtenidos, y en la parte inferior de la tabla se proporcionan los valores promedio para cada una de las métricas utilizadas, siendo estos valores 0.7946, 0.8108 y 0.8002 para la precisión, exhaustividad y medida-F respectivamente.

Tabla 5-5. Resultados obtenidos en MusicBrainz.

Tipo	Cantidad	Precisión	Exhaustividad	Medida-F
What	16	0.7862	0.8132	0.7976
Who	14	**0.8265**	**0.8389**	**0.8319**
How	11	0.7770	0.7897	0.7824
Where	10	0.8188	0.8375	0.8250
When	12	0.7866	0.7972	0.7912
Which	26	0.7857	0.8243	0.8013
Give/List/Name	11	0.7814	0.7745	0.7721
Total	100			
Promedio		0.7946	0.8108	0.8002

Finalmente, en la Figura 5-8 se presenta la distribución de los tipos de resultado de acuerdo a las respuestas provistas por la interfaz, que son correctas, incorrectas, fallidas (cuando no es capaz de interpretar la pregunta) y parciales (cuando devuelve solo una parte de los resultados esperados). Al igual que en el dominio de DBpedia, la interfaz pudo contestar correctamente a la gran mayoría de las preguntas. Solo que, en esta ocasión, el número de preguntas respondidas correctamente fue ligeramente menor, 73 para DBpedia y 71 para el presente dominio.

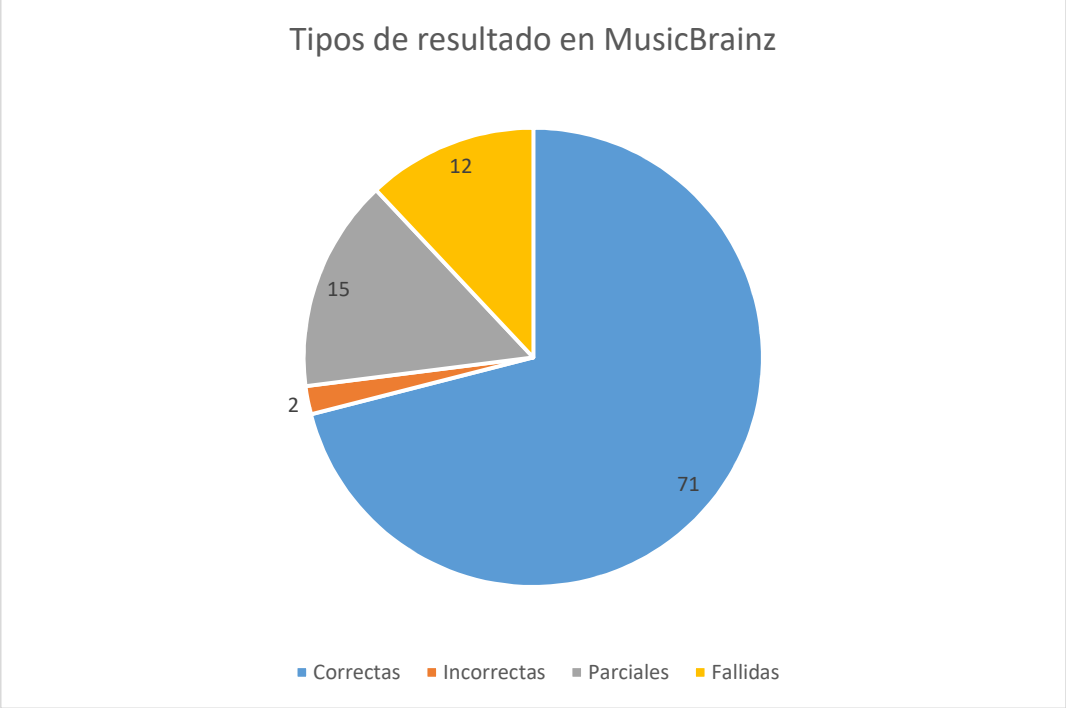

Figura 5-8. Tipos de resultado en MusicBrainz.

El hecho de que el número de preguntas contestadas correctamente no variará considerablemente al cambiar la base de conocimiento, se puede interpretar como un buen nivel de portabilidad de la interfaz propuesta en este trabajo.

5.7.3 Resultados generales

Con el objetivo de proveer una mejor perspectiva del funcionamiento general de la interfaz, en la Figura 5-9 se presentan los promedios de los resultados obtenidos en ambos dominios. Como se puede apreciar en dicha imagen, el tipo de pregunta con mejores resultados fue *Who*, con valores de 0.8639 y 0.8573 para la exhaustividad y medida-F respectivamente. Mientras tanto, el tipo de pregunta con resultados más bajos corresponde al tipo de pregunta *How* con una precisión de 0.7671, exhaustividad de 0.7802 y una medida-F de 0.7727.

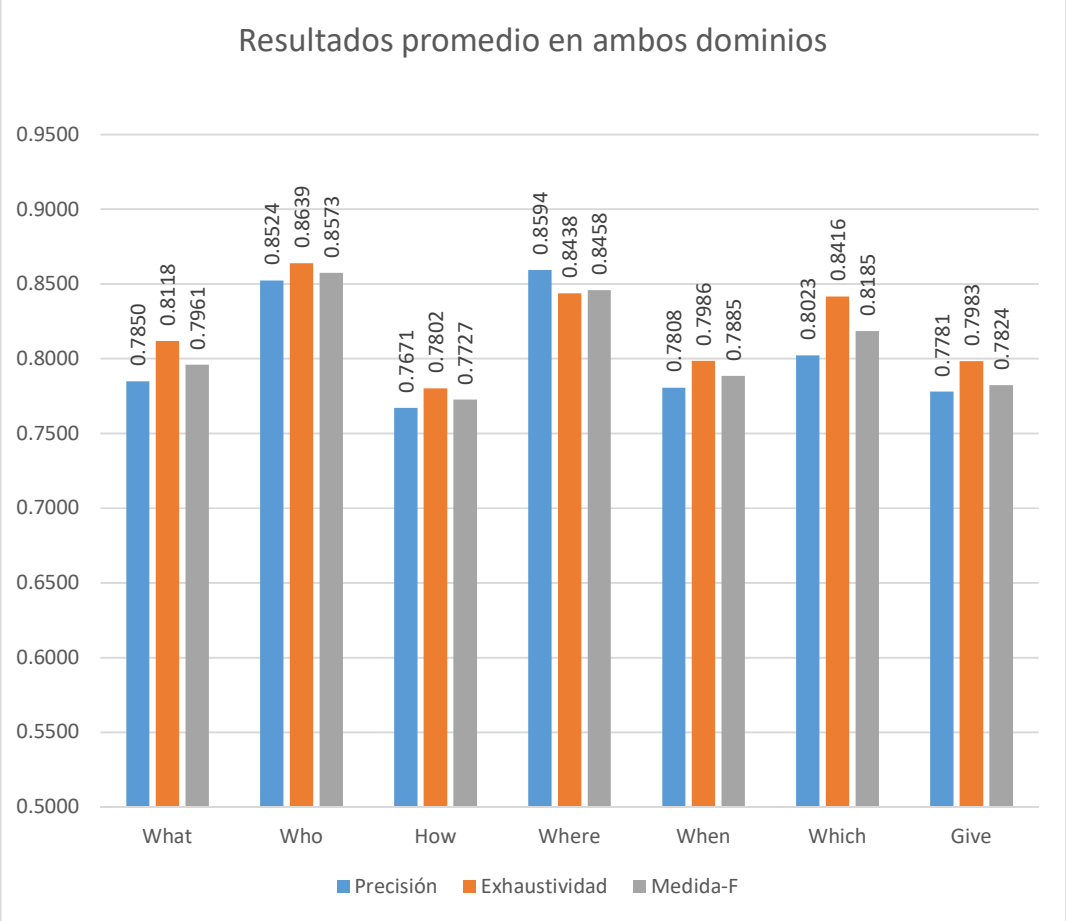

Figura 5-9. Resultados promedio en ambos dominios.

Además, en la Figura 5-10 se presenta la distribución de los tipos de resultado obtenidos en ambos conjuntos de experimentos. Es dicha figura podemos notar que el sistema pudo proveer la respuesta correcta a un alto porcentaje de preguntas, en

específico a un 72%. Otro punto a destacar es el 11% que corresponde al número de preguntas que no pudieron ser interpretadas por la interfaz. Por último, en color gris se representa el porcentaje de preguntas cuya respuesta provista por la interfaz coincidió de manera parcial con la obtenida a través de su respectiva consulta SPARQL, el cual es de 14%.

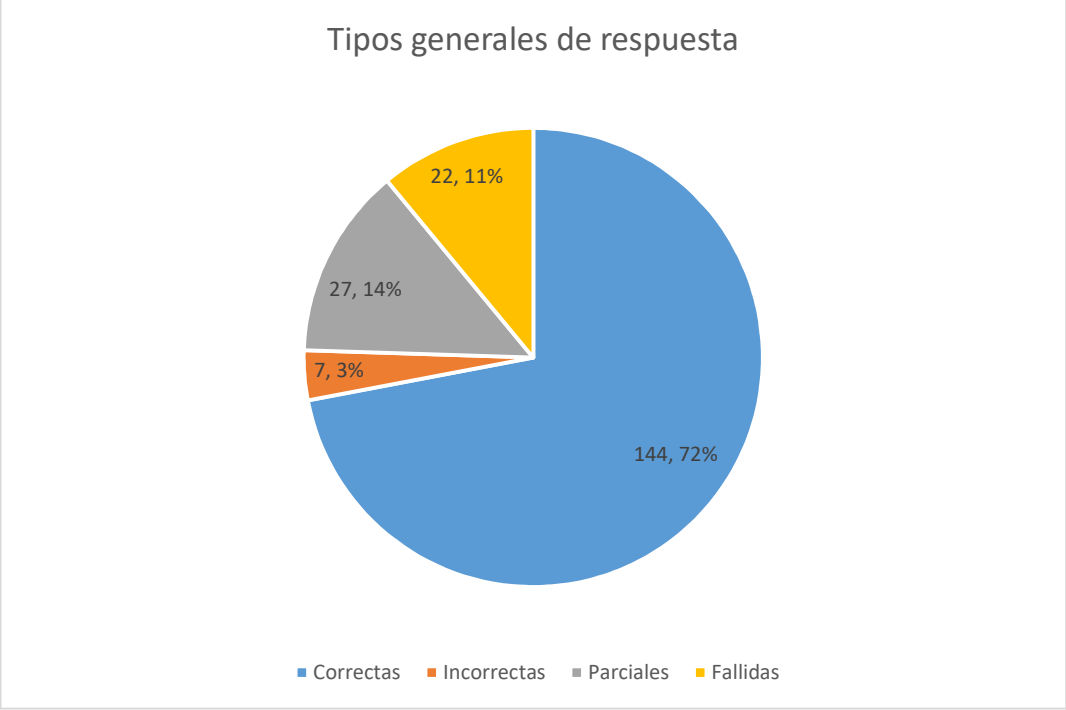

Figura 5-10. Tipos de resultados en ambos dominios.

En la siguiente sección se discuten a detalle los resultados de evaluación obtenidos en ambos dominios, haciendo hincapié en las fortalezas y debilidades de la interfaz propuesta en esta tesis doctoral.

5.7.4 Discusión de resultados

Una vez obtenidos los resultados de evaluación, la siguiente tarea consistió en analizar minuciosamente cada una de las preguntas utilizadas, así como sus respectivas respuestas provistas por la interfaz descrita en este trabajo. El objetivo fue determinar las razones por las cuales esta última no fue capaz de proveer la respuesta correcta, y de esta manera, ser capaces de identificar nuevas líneas de trabajo futuro a realizar que se centren en incrementar el grado de efectividad manteniendo la independencia en el dominio y el tipo de pregunta proporcionada. A continuación, se discuten nuestras observaciones.

5.7.4.1 Limitaciones

Desde un punto de vista estricto, la interfaz propuesta en este trabajo pudo responder correctamente solo al 72% de las preguntas que se le proporcionaron y el resto se puede considerar como una respuesta incorrecta. Sin embargo, para efectos de la evaluación y análisis de resultados, se establecieron cuatro tipos de resultados con el objetivo de interpretar de mejor manera el nivel de efectividad de la interfaz. Estos tipos de resultado son: correcto, incorrecto, fallidas y parcial. El primer tipo, al cual le corresponde el 72%, significa que la interfaz fue capaz de obtener las tripletas RDF que coinciden totalmente con las obtenidas por la consulta SPARQL. En este sentido, como se puede observar en la Figura 5-9, el tipo de pregunta que obtuvo mejores resultados fue *Who* que, como se estableció en la clasificación de preguntas utilizada por esta interfaz, el tipo de respuesta esperada por este tipo de preguntas corresponde a instancias de clases tales como persona u organización. Analizando el conjunto de preguntas notamos que su correspondiente consulta SPARQL contenían un menor número de tripletas en comparación con la de otros tipos de preguntas tales como las del tipo *Give* o *How*, los dos tipos de preguntas con resultados más bajos. Este hecho se debe, a que el número de elementos de la pregunta a relacionar con recursos de la base de conocimiento era mucho menor.

En cuanto a los tres tipos restantes pudimos identificar cuatro razones principales por las cuales la interfaz no pudo obtener la respuesta correcta, que son: 1) problemas de ambigüedad; 2) falló al relacionar palabras de la pregunta con términos del dominio; 3) nombres de individuos compuestos por múltiples palabras; y 4) no considerar términos que influyen en la cantidad de resultados a obtener. Estas razones son discutidas en los siguientes párrafos.

La ambigüedad fue la principal causa de que las respuestas provistas por la interfaz fueran categorizadas como fallidas o incorrectas. El problema más frecuente en este contexto fue que las preguntas contenían solo una parte del nombre del individuo sobre el cual buscaban información, el cual, como recordaremos, esta interfaz supone que es provisto a través de la propiedad *rdfs:label* del recurso. Dicho problema ocasiona que el término contenido en la pregunta fuese relacionado con más de un recurso de la base de conocimiento. A pesar de que nuestro enfoque intenta resolver este tipo de problemas a través de establecimiento del contexto de la pregunta, el cual consiste en seleccionar aquellos elementos que estén relacionados con las propiedades (*object property* o *datatype*) identificadas en la pregunta a través de su rango o dominio, muchos de los recursos obtenidos a través de elemento de la pregunta pertenecían a la misma clase, es

decir, tenían relación con la propiedad identificada. En este sentido, somos conscientes de que los mecanismos implementados por esta interfaz para resolver dicho problema carecen de la robustez necesaria para afrontar este problema. Desde esta perspectiva, estamos convencidos de que nuestro enfoque puede ser mejorada mediante la incorporación de mecanismos de retroalimentación que permitan al usuario seleccionar la opción deseada de un conjunto de opciones provistas por el sistema.

Con respecto al segundo problema mencionado, notamos que en ocasiones el sistema no fue capaz de relacionar elementos contenidos en la pregunta con recursos de la base de conocimiento a través de la similitud de caracteres o mediante el uso de sinónimos. Por ejemplo, en la pregunta: *When were The Beatles founded?* (¿Cuándo se fundaron Los Beatles?), el sistema fue capaz de identificar el individuo *The Beatles*. Sin embargo, no pudo encontrar una relación entre la palabra *founded* y la propiedad de tipo *datatype activity_start*. En este sentido, es importante mencionar que para llevar a cabo la tarea en cuestión nuestro enfoque depende en gran medida de la descripción del recurso entendible por el humano provista a través de la propiedad *rdfs:label*. A pesar de que esta interfaz implementa mecanismos para extraer tal descripción, ya sea a través de la propiedad antes mencionada o mediante el identificador del recurso (su URI), hemos percibido que no todas las entidades de la base de conocimiento están correctamente anotadas, es decir, ellas no reflejan un término que pueda ser utilizado por el humano para hacer referencia a este. Por ejemplo, los recursos contenidos en la base de conocimiento DBpedia cuentan con etiquetas completas. Sin embargo, al ser esta fuente de información un esfuerzo comunitario, no podemos garantizar que todos estos recursos estén correctamente anotados. El etiquetado correcto de las entidades de la base de conocimiento es una recomendación de métodos de evaluación de la calidad de uso de las ontologías, tales como ONTOMETRIC (Lozano-Tello and Gómez-Pérez 2004). Por otro lado, a pesar de que nuestro enfoque integra técnicas de PLN tales como lematización y la búsqueda de sinónimos con el objetivo de reducir el número de casos cuando algunas palabras no pueden ser mapeadas con elementos de la base de conocimiento, es necesario que nuestro enfoque integre mejores mecanismos que le permitan aprender las distintas formas que utilizan los usuarios de un determinado dominio para referirse a un recurso, sea este un individuo o un concepto.

En cuanto al reconocimiento de individuos, lo primero a aclarar es que el sistema debe reconocer dentro de la pregunta aquellos elementos que hagan referencia a un individuo de la base de conocimiento. Por ejemplo, a algún libro, o en el contexto de la música, a alguna canción o grupo musical. El problema de este tipo de individuos es que su nombre

está compuesto por múltiples palabras, lo que provoca que ni el módulo de reconocimiento de individuos ni el analizador de dependencias puedan llevar a cabo su tarea. Por ejemplo, consideremos la pregunta: *Who wrote the song Freeway of Love?* (¿Quién escribió la canción Freeway of Love?"), donde *Freeway of Love* debe ser identificada como un individuo. Cuando el usuario provee el nombre de la canción utilizando mayúsculas, tal como se aprecia en la pregunta anterior, el sistema determina qué conjunto de palabras puede hacer referencia al nombre de un individuo. Sin embargo, cuando el usuario emplea solo minúsculas, el sistema no puede reconocerlo como tal.

Finalmente, uno de los principales problemas que propiciaron que un alto porcentaje de respuestas provistas por la interfaz fuera categorizado como *parcial* fue el hecho de no considerar ciertos términos contenidos en la pregunta que influyen directamente en el total de elementos a recuperar de la base de conocimiento. Por ejemplo, en la pregunta *Give me all songs from Bruce Springsteen released between 1980 and 1990* (Dame todas las canciones de Bruce Springsteen lanzadas entre 1980 y 1990), el sistema no pudo interpretar el periodo especificado por el usuario, por lo que devolvió como resultado todas las canciones de Bruce Springsteen, y no solo las del rango especificado.

5.7.4.2 Portabilidad de la interfaz

Uno de los principales objetivos de llevar a cabo la evaluación en un segundo dominio es el de determinar el nivel de portabilidad de la interfaz propuesta, es decir, analizar en qué grado mejora o disminuye la efectividad de la interfaz para proveer la respuesta correcta cuando cambia la base de conocimiento. Como se puede observar en la Figura 5-5 y Figura 5-7 no existe diferencia significativa en los resultados obtenidos, lo que se pude interpretar como un buen grado de portabilidad por parte de la interfaz propuesta en este trabajo de tesis.

A pesar de lo antes dicho, es importante notar que el número de preguntas contestadas correctamente disminuyó de un dominio a otro. Sin embargo, tal diferencia la atribuimos al hecho de que las preguntas del corpus de MusicBrainz hacen referencia en mayor medida a individuos de la base de conocimiento a través de términos compuestos por múltiples palabras, tal vez por el hecho de tratarse del dominio de la música, donde un número significativo de las preguntas están dirigidas a obtener información de canciones o grupos musicales, cuyos nombres se componen de múltiples palabras. Además, las preguntas de MusicBrainz incluían un mayor número de modificadores que afectaban la cantidad de recursos a obtener como respuesta. A pesar de que la interfaz contempla patrones SPARQL para afrontar este tipo de preguntas, aún es necesario refinar los

existentes e incluir un mayor número de estos con el objetivo de incrementar la efectividad de la interfaz ante este tipo de preguntas.

5.8 Resumen

En este capítulo se presentó la evaluación llevada a cabo de una de las contribuciones de esta tesis, la interfaz de lenguaje natural. Este proceso se realizó mediante el análisis de la efectividad de recuperación de la respuesta correcta de bases de conocimiento basadas en ontologías.

El capítulo comienza por describir las bases de conocimiento utilizadas en este proceso de validación, que son DBpedia y MusicBrainz. La primera de ellas contiene información de un contexto demasiado general, pues representa un esfuerzo comunitario por proveer el contenido de Wikipedia a través de tecnologías de la Web Semántica. La segunda de ellas incluye información relacionada con el dominio de la música, tal como canciones, grupos musicales, entre otros.

Una vez descritas las bases de conocimiento, este capítulo presentó la metodología de validación a seguir, la cual se compone de cuatro tareas fundamentales: recolección del corpus de preguntas, generación de consultas SPARQL, ejecución de preguntas en lenguaje natural contra la interfaz propuesta, y finalmente la evaluación de la interfaz. Después, se analiza una descripción de los corpus de preguntas recolectados incluyendo detalles tales como la cantidad de preguntas para cada tipo de pregunta soportada por la interfaz. Posteriormente, se describen las métricas de validación empleadas en este proceso, que son: precisión, exhaustividad y medida-F.

En la parte final de este capítulo se presentan los resultados de evaluación para cada uno de las bases de conocimiento utilizadas. En esta sección se describen los resultados obtenidos para cada tipo de pregunta en ambos dominios, a través de los cuales se puede apreciar un nivel de efectividad elevado por parte de la interfaz sin importar el tipo de pregunta a resolver, ni la base de conocimiento utilizada. Finalmente, se discuten las razones principales que propiciaron que algunas de las preguntas provistas a la interfaz no fueran contestadas correctamente, destacando la ambigüedad, el mapeo de elementos de la pregunta con recursos de la base de conocimiento y el uso de términos compuestos por múltiples palabras.

Capítulo 6. Conclusiones y líneas futuras

6.1 Conclusiones

En los últimos años, las bases de conocimiento basadas en ontologías han sido adoptadas por individuos y organizaciones de múltiples dominios debido a que, entre otras cosas, permiten asignar a la información un significado bien definido que puede ser entendido tanto por humanos como por computadoras, quienes podrán automatizar, integrar y reutilizar información de múltiples fuentes con el fin de resolver necesidades específicas de información. Sin embargo, el acceso a estas bases de conocimiento sigue siendo un reto para una gran parte de los usuarios pues demanda el conocimiento de tecnologías de la Web Semántica, lenguajes formales de consulta, así como de la estructura de datos dicha fuente de información.

Para contribuir a la solución del reto antes planteado, el trabajo de investigación descrito en esta tesis presenta una interfaz de lenguaje natural enfocada a bases de conocimiento basadas en ontologías. El funcionamiento general de esta interfaz se basa en un modelo ontológico que permite representar a través de una estructura semántica las relaciones sintácticas existentes entre los elementos contenidos en la pregunta en lenguaje natural provista por el usuario. De igual manera, este modelo permite almacenar toda la información concerniente a los recursos de la base de conocimiento con los cuales guardan relación los términos de interés identificadas en la pregunta, permitiendo así, el establecimiento del contexto de la misma. A partir de la información representada en dicho modelo, el sistema es capaz de establecer tanto el tipo de pregunta proporcionada como el tipo de respuesta esperado, lo que incrementa la posibilidad de encontrar la respuesta al delimitar el espacio de búsqueda de entidades que pertenezcan a una clase específica. Además de lo antes dicho, el modelo ontológico de la pregunta facilita en gran medida la generación de consultas SPARQL ya que tal representación corresponde con el formato de tripletas RDF utilizado por lenguajes de consulta formal tal como SPARQL.

Tal como se describió en esta tesis, existen otros esfuerzos de investigación por resolver el reto en cuestión. Sin embargo, trabajos tales como SWIP (Pradel, Haemmerlé, and Hernandez 2012) o REHABROBO-CNL (Dogmus, Patoglu, and Erdem 2014) lo afrontan a través del uso de un lenguaje controlado, lo que limita la expresividad del usuario. En este

sentido, nuestra propuesta emplea el lenguaje natural con el objetivo dar al usuario la libertad de utilizar el poder de expresividad del lenguaje que ya conoce. Sin embargo, la manera en que los usuarios formulan sus preguntas en ocasiones no corresponde directamente con la forma en la cual la base de conocimiento es modelada por la ontología. Es decir, debido a la libertad del uso del lenguaje otorgado bajo este enfoque, el usuario puede utilizar diversas variantes léxicas y sintácticas para preguntar por la misma información. Esto denota el gran reto que representa el desarrollo de una interfaz de lenguaje natural, independientemente del contexto, pues es una tarea demasiado compleja donde se deben considerar demasiadas variantes. Es por eso, que a pesar de que los resultados de evaluación obtenidos por la interfaz son notorios, es imprescindible seguir trabajando en la mejora o inclusión de nuevos mecanismos que permitan afrontar con los problemas inherentes al uso del lenguaje natural, para así lograr reducir la brecha existente entre este tipo de información y usuarios no expertos en tecnologías de la Web semántica, sobre todo en la era actual donde la información juega un rol importante en gran parte de las actividades que desarrollamos cotidianamente.

Cuando se planteó este trabajo de investigación, uno de los principales objetivos que se establecieron fue el de lograr que la interfaz fuese independiente del dominio. Para cumplir con este objetivo, se diseñó una arquitectura compuesta por diversos módulos que llevan a cabo funciones muy específicas. Así, la integración del modelo ontológico de la pregunta propuesto permitió alcanzar en gran medida el objetivo planteado. Esta afirmación se sustenta en los resultados de evaluación obtenidos para ambos dominios. Sin embargo, una adecuación que se tuvo que realizar al cambiar de dominio fue provocada por el uso de la propiedad *rdfs:label*, es decir, a pesar de que la especificación RDF Schema recomienda el uso de dicha propiedad para proveer una versión entendible por el humano del nombre del recurso, la ontología Music Ontology utiliza la propiedad *dc:title* para definir conceptos y propiedades en el dominio de la música, por ejemplo, artistas, álbumes, canciones, entre otros. El módulo de preprocesamiento fue extendido para que integrase la propiedad *dc:title* en la generación del vocabulario del dominio. En este sentido, es importante alentar a la comunidad científica a anotar todos los recursos descritos en la ontología mediante términos que puedan ser utilizado por el humano para hacer referencia a ellos, con el objetivo de facilitar la interacción humano-computadora. El etiquetado correcto de las entidades de la base de conocimiento es una recomendación de métodos de evaluación de la calidad de uso de las ontologías, tales como ONTOMETRIC (Lozano-Tello and Gómez-Pérez 2004).

6.2 Aportaciones

De acuerdo a los discutido en la sección anterior, las principales aportaciones de esta tesis doctoral son las siguientes:

- **Modelo ontológico de la pregunta**. Una de las características sobresaliente de la interfaz presentada en esta tesis, es el establecimiento de un modelo ontológico que permite describir la estructura sintáctica de la pregunta, así como el contexto de esta en términos de la base de conocimiento del dominio y de las relaciones semánticas existentes entre ellos. La obtención de la estructura sintáctica de la pregunta se basa en la técnica de análisis de dependencias. Esta técnica obtiene relaciones binarias entre los elementos de la pregunta, las cuales, gracias al modelo de la pregunta pueden ser representados en forma de tripletas Sujeto-predicado-objeto, en los casos donde dichas relaciones comparten un mismo elemento. Esto facilita en gran medida la generación de consultas SPARQL pues tal estructura corresponde con el formato de tripletas RDF utilizado en este enfoque formal de consulta. Por otro lado, el contexto de la pregunta se basa en el mapeo de elementos de la pregunta y recursos de la base de conocimiento con los cuales guarde relación tanto a nivel de similitud de cadenas, como a través del dominio y rango de las relaciones identificadas.
- **Adaptación de una clasificación de preguntas y respuestas al contexto de bases de conocimiento basadas en ontologías**. En este trabajo se presentó la adaptación de la clasificación de preguntas propuesta por (D. Moldovan et al. 2000) al contexto de las bases de conocimiento basadas en ontologías. Esta adaptación consistió en sustituir los tipos de respuesta esperados por clases establecidas en ontologías y vocabularios que han sido ampliamente adoptados por individuos y organizaciones para representar su información. Entre dichos recursos se encuentra el lenguaje XML Schema (World Wide Web Consortium 2016b), así como las ontologías FOAF (Brickley and Miller 2012) y la de DBpedia. El establecer los tipos de respuesta esperada por el usuario acorde a los tipos de datos existentes en bases de conocimiento basadas en ontologías incrementa la posibilidad de obtener la respuesta correcta, ya que permite delimitar el espacio de búsqueda de tal forma que los recursos a obtener deberán ser solo aquellos que correspondan con el tipo de dato establecido, o sean subclase de este.
- **Conjunto de plantillas de tripletas RDF.** Una de las tareas fundamentales de toda interfaz de lenguaje natural enfocada a bases de conocimiento

estructuradas consiste en la generación de consultas en lenguaje formal que permitan obtener la información de la base de conocimiento que dé respuesta a la pregunta proporcionada por el usuario. En este sentido, la presente interfaz provee un conjunto de plantillas de tripletas RDF las cuales corresponden a las relaciones semánticas existentes entre los elementos de interés identificados en la pregunta que son almacenadas en el modelo ontológico de la pregunta propuesta en este trabajo. Este conjunto de plantillas, que han probado ser independientes del dominio, permiten la generación de consultas SPARQL formadas por múltiples tripletas.

- **Validación de la interfaz en diferentes dominios.** El proceso de validación de la interfaz se llevó a cabo en dos dominios diferenciados, a saber: DBpedia y MusicBrainz. Los experimentos realizados demandaron la búsqueda de corpus de preguntas en lenguaje natural para los dominios especificados, así como la participación de personas ajenas al trabajo de investigación quienes proporcionaron parte de las preguntas utilizadas. El objetivo primordial de los experimentos fue evaluar la efectividad de la interfaz para proveer la respuesta correcta a una pregunta a partir de la información contenida en la base de conocimiento del dominio. El segundo conjunto de experimentos, tuvo como objetivo verificar la portabilidad de la interfaz. Los resultados obtenidos no variaron significativamente con respecto a los del primer experimento, hecho que se puede interpretar como un buen nivel de portabilidad por parte de la interfaz propuesta.

6.3 Limitaciones y trabajo a futuro

A pesar de que los resultados de evaluación obtenidos por la interfaz propuesta en esta tesis doctoral lucen alentadores, somos conscientes que este enfoque tiene ciertas limitaciones que, sin embargo, pueden ser direccionadas a futuro. Algunas de las limitaciones ya fueron discutidas en la sección 5.7.4.1. Sin embargo, a continuación, se provee un resumen de ellas y se provee una más.

- **Tipos de pregunta soportadas.** La interfaz de lenguaje natural permite el uso de preguntas factuales, es decir, aquellas que esperan como respuesta un hecho concreto. Por ejemplo, el nombre de una persona o lugar; la altura de una persona, la fecha en que ocurrió un evento, entre otros. Además, permite el uso de oraciones imperativas para solicitar información. Por ejemplo, a través de los términos *Name*, *Give*, o *List*. Esta manera de definir las consultas de información

puede limitar bastante la expresividad del usuario. En este sentido, es importante considerar más tipos de preguntas como de opción múltiple, verdadero/falso, entre otras. Esto sin duda otorgará al usuario una mayor libertad del uso del lenguaje. Sin embargo, será una tarea ardua pues demandará el análisis de corpus con preguntas de este tipo, para identificar las relaciones de dependencia que ayudarían a obtener una representación semántica de la pregunta, la cual, como recordaremos, se almacena en el modelo de la pregunta propuesto en este trabajo.

- **Problemas de ambigüedad**. La libertad en el uso del lenguaje provista por la interfaz de esta tesis da paso a problemas tales como la ambigüedad, la cual se refiere al fenómeno que se presenta cuando una palabra, un sintagma, o una oración puede ser interpretada de más de una forma. A pesar de que la interfaz propuesta provee mecanismos para afrontar algunos casos de ambigüedad, somos conscientes de que estos presentan limitaciones que les impiden direccionar de mejor manera este problema. Con el objetivo de hacer frente a esta situación, se plantea la integración de mecanismos de retroalimentación que permitan al usuario desambiguar las preguntas, quizá, a través de la reformulación de la pregunta o a través de seleccionar alguna opción de un conjunto provisto por la interfaz.
- **Nombres de individuos compuestos por múltiples palabras**. La interfaz debe identificar dentro de la pregunta aquellos elementos que hagan referencia a un individuo de la base de conocimiento, como lo puede ser un libro o una canción. Sin embargo, en ocasiones el nombre del individuo está compuesto por múltiples palabras. Por ejemplo, la canción del grupo Metallica *For Whom the Bell Tolls*. Cuando este fenómeno ocurre, la interfaz puede reconocer este nombre, siempre y cuando combine el uso de mayúsculas y minúsculas. Cuando el nombre de la canción está escrito completamente en minúsculas, el sistema falla para reconocer este individuo. Este fenómeno representa un gran reto en el contexto de PLN en general, tal como se describe en (Sag et al. 2002), donde de igual manera se presentan algunas técnicas que podrían complementar a esta interfaz, tal como el uso de reglas, métodos estadísticos, entre otros.

Además del trabajo a futuro antes citado, en los siguientes puntos se presentan algunos temas que no han sido considerados como parte del desarrollo de la presente interfaz, pero que proporcionan nuevas líneas de investigación.

- **Conjuntos de datos distribuidos y enlazados.** Como se ha comentado a lo largo de esta tesis, existe un gran número de individuos y organizaciones que han adoptado ya el enfoque de la Web Semántica, y en específico, el de Linked Data, para publicar sus datos. Uno de los objetivos de Linked Data es enlazar distintos datos estructurados que se encuentran distribuidos en la Web. Debido a esto último, es importante considerar esa distribución al momento de buscar una respuesta, pues en ocasiones esta puede depender de más de una base de conocimientos. En esta línea de investigación, se propone llevar a cabo un estudio del arte de enfoques que permitan llevar a cabo consultas en fuentes de información descentralizadas como Linked Data. Algunos de los enfoques más sobresaliente son las consultas federadas a SPARQL endpoints, fragmentos de patrones de tripletas y flujos de Linked Data, los cuales son descritos en (Saleem et al. 2015). Tras el estudio podremos establecer un punto de partida para abordar el problema en cuestión e integrarlo en la interfaz de este trabajo.
- **Multilingüismo.** Actualmente, la mayoría de interfaces de lenguaje natural orientadas a bases de conocimiento basadas en ontologías no son capaces de responder a preguntas formuladas en múltiples lenguajes. El llevar a cabo esta tarea requiere que los recursos descritos en las ontologías (clases, propiedades e individuos) cuenten con una propiedad a través de la cual referenciarlos en cada uno de los lenguajes a considerar. Un ejemplo claro de esto es DBpedia, la cual provee la propiedad *rdfs:label* para idiomas tales como español, inglés, ruso, entre otros. Para aprovechar esta característica, en esta línea de investigación se propone la adaptación de la interfaz a otro lenguaje, concretamente al español. Esto demandará analizar herramientas que permitan llevar a cabo el análisis sintáctico de dependencias, y de esa manera reducir aún más la brecha existente entre usuarios y bases de conocimiento basadas en ontologías.

Capítulo 7. Contribuciones científicas

7.1 Publicaciones JCR

1. Paredes-Valverde Mario Andrés, Valencia-García Rafael, Rodríguez-García Miguel Ángel, Colomo-Palacios Ricardo, Alor-Hernández Giner. A semantic-based approach for querying linked data using natural language. Journal of Information Science, Volume 42, Issue 6, (2015) Impact Factor: 0.878.
2. Paredes-Valverde Mario Andrés, Rodríguez-García Miguel Ángel, Ruiz-Martínez Antonio, Valencia-García Rafael, Alor-Hernández Giner, ONLI: An ontology-based system for querying DBpedia using natural language paradigm. Expert Systems with Applications. Volume 42, Issue 12, 5163-5176 (2015) Impact Factor: 2.981.

7.2 Publicaciones en revistas

1. Vivancos-Vicente Pedro José, Castejón-Garrido Juan Salvador, Paredes-Valverde Mario Andrés, Salas-Zárate María del Pilar, Valencia-García Rafael, IXHEALTH: A multilingual platform for advanced speech recognition in healthcare. (2016) Communications in Computer and Information Science, Volume 658, pp. 26-38.
2. Noguera-Arnaldos José Ángel, Paredes-Valverde Mario Andrés, Valencia-García Rafael, Rodríguez-García Miguel Ángel, Sistema de diálogo basado en mensajería instantánea para el control de dispositivos en el internet de las cosas. Procesamiento del Lenguaje Natural Volume 55: 173-176 (2015).
3. Paredes-Valverde Mario Andrés, Noguera-Arnaldos José Ángel, Rodríguez-Enríquez Cristian Aarón, Valencia-García Rafael, Alor-Hernández Giner, A Natural Language Interface to Ontology-Based Knowledge Bases. In Distributed Computing and Artificial Intelligence, 12th International Conference, Volume 373, (pp. 3-10), Springer International Publishing.
4. Noguera-Arnaldos José Ángel, Rodríguez-García Miguel Ángel, Ochoa José Luis, Paredes-Valverde Mario Andrés, Alcaraz-Mármol Gema, Valencia-García Rafael, Ontology-Driven Instant Messaging-Based Dialogue System for Device Control. Lectures Notes in Computer Science (including subseries Lecture Notes in

Artificial Intelligence and Lecture Notes in Bioinformatics) 2015: Volume 9416, 299-308.

7.3 Capítulos en libro

1. Noguera-Arnaldos José Ángel, Paredes-Valverde Mario Andrés, Salas-Zárate María del Pilar, Rodríguez-García Miguel Ángel, Valencia-García Rafael, José Luis Ochoa, im4Things: An ontology-based Natural Language Interface for controlling devices in the Internet of Things, Current Trends on Knowledge-Based Systems, vol. 120, ISBN 978-3-319-51905-0.

7.4 Congresos internacionales

1. Conferencia: "A Natural Language Interface to Ontology-Based Knowledge Bases", In the 12th International Conference in Distributed Computing and Artificial Intelligence, Salamanca, España, junio 2015.

Capítulo 8. Summary

8.1 Introduction

The Semantic Web is an extension of current one, in which information is given well-defined meaning, better enabling computers and people to work in cooperation (Berners-Lee et al. 2001). Semantic Web enables people to create data stores on the Web, build vocabularies, and write rules for handling this data. Tim Berners-Lee proposed a Semantic Web architecture, known as Semantic Web Stack, which is divided into various layers of metadata, each one provides different degrees of expressivity. Ontologies are considered one of the pillars of the Semantic Web[5]. An ontology can be defined as a formal and explicit specification of a shared conceptualization (Studer, Benjamins, and Fensel 1998); in other words, it allows to formally and explicitly represent structures of knowledge through concepts, their properties, relations with other concepts, and the axioms related to them (Legaz García 2015). According to (Hadzic et al. 2009), two important functions of ontologies are that they enable agents to work cooperatively to communicate with each others and they make the available information more accessible to automated agents.

Given the Semantic Web advantages described above, a lot of individuals and organizations from different domains have adopted the ontology-based approach to publish their information. This has led to an exponential growth of information available on the Web and intranets represented by RDF (RDF Working Group 2016). Nowadays, the access to this kind of information is performed by using formal query languages such as SPARQL (Prud'Hommeaux, Seaborne, and others 2008). However, this approach is complicated for casual users (Kaufmann, Bernstein, and Fischer 2007) because of the necessity of learning formal query languages or the functioning of graphical interfaces, and even, of knowing the underlying knowledge base structure.

The need to make the ontology-based information accessible to all kind of users, whether casual or experts users, demands for new intuitive and easy to use information retrieval mechanisms. In this sense, the natural language paradigm is generally deemed to be very intuitive from a use point of view to address the gap between knowledge bases

[5] http://semanticweb.org/wiki/Ontology.html

and end users (Cimiano et al. 2008). A natural language interface (NLI) is a system that allows users to access information stored in some repository by formulating requests in natural language (Kaufmann and Bernstein 2010). NLIs allow users to use all communicative power of language that they already possess instead of being forced to use an unnatural and limited mode of communication. Furthermore, NLIs hide from users the formality of a knowledge base as well as the formal query language.

Hence, in this thesis we propose a natural language based solution for reducing the gap between ontology-based knowledge bases and users. This solution will be guided by natural language processing techniques and Semantic Web technologies. The defined solution integrates a question ontological model as the main vehicle for representing the syntactic structure of the question as well as for storing all information related to the its context. From the information contained in this model, the SPARQL queries are generated, which will allow to obtain answers from the domain knowledge base.

8.2 Aims of the thesis

This thesis aims to provide natural language based solutions for reducing the gap between ontology-based knowledge bases and users. More specifically, the objectives of this thesis can be summarized as follow:

- Design and implementation of a domain independent question ontological model for representing the syntactic structure and context of the natural language question.
- Design and implementation of a natural language interface to ontology-based knowledge bases.
- Design and implementation of a question analysis process guided by natural language processing techniques and Semantic Web technologies.
- Design and implementation of a SPARQL queries generation process from a semantic representation of the natural language question.
- Validation of the results on Linked Data based knowledge bases.

The methodology followed in this research is decomposed on next main tasks:

- **Analysis of the state of art**. It involves studying the technologies used in this research, namely, Semantic Web, Natural Language Processing, and Natural Language Interfaces.
- **Formalization of the proposal**. It involves the formalization of a natural language interface to ontology-based knowledge bases. This NLI is guided by a

domain independent question ontological model that allows representing the syntactic structure and context of the natural language question.

- **Implementation of the proposal**. It refers to the implementation of the proposal by means of current natural language tools.
- **Validation of the proposal**. It consists in validating the proposal on Linked Data based knowledge bases, namely, DBpedia and MusicBrainz.

8.3 State of art

In this section, we presented a description of all technologies used in this research, namely, Semantic Web, Natural Language Processing, and Natural Language Interfaces. Regarding Semantic Web, we described the ontologies, which are considered one of the pillars of the Semantic Web. Concerning natural language processing, we described the different levels on which it is decomposed. In this section, we emphasized the syntactic analysis process, a key part of the overall performance of our approach. Finally, with regards to the natural language interfaces, we described the generic architecture of a natural language interface, which represented the basis for the formalization of our work. Subsequently, we provided a study of the natural language interfaces oriented to relational databases, including their antecedents, advantages and disadvantages, as well as the main architectures used in the development of this type of applications. Finally, we presented a study of natural language interfaces for knowledge bases existing in the literature. Following, this study is briefly described.

Nowadays, there are prominent efforts to offer NLI for ontology-based knowledge bases. In this work, we presented an analysis of the most relevant works. For instance, there are works such as SWIP (Pradel, Haemmerlé, and Hernandez 2012) and REHABROBO-CNL (Dogmus, Patoglu, and Erdem 2014) that use an approach based on pivot language and controlled language respectively, instead of allowing users to use all communicative power of language they already possess. With regards to portability, there are works such as Aqualog (Lopez, Pasin, and Motta 2005) and ORAKEL (Cimiano et al. 2008) that require a configuration process aiming to they can be applied to other domains. On the other hand, works such as QACID (Ferrández et al. 2009) demand the collection of questions of the specific domain to generate question patterns compatible only with the underlying source of information. This process requires a considerable effort and the ability to ensure a meaningful relation between the questions collected and the domain. Concerning the NLP techniques used for analyzing the question, works such as PANTO (C. Wang et al. 2007) and FREyA (Damljanovic, Agatonovic, and Cunningham 2010) use

parsers conform to the phrase structure grammar, however, it can represent little semantic information (S. Li, He, and Wu 2014). In this sense, the dependency grammar is better suited than phrase structure grammar for languages with free or flexible word order (Kübler, McDonald, and Nivre 2009). Finally, about the approach used to represent the question in a formal way before the generation of the formal language query, works such as Aqualog (Lopez, Pasin, and Motta 2005), PANTO (C. Wang et al. 2007) and FREyA (Damljanovic, Agatonovic, and Cunningham 2010) use triple-based approaches which are guided by phrase structure grammar. On the other hand, works such as TR-Discover (Song et al. 2015), AutoSPARQL (Unger et al. 2012), MYAutoSPARQL (Sharef, Noah, and Murad 2015) and the presented in (Hamon, Grabar, and Mougin 2016) use patterns to map the question and obtain the corresponding formal queries.

Considering the analysis briefly described in the previous paragraph, in this work we propose natural language based solutions for reducing the gap between ontology-based knowledge bases and users. These solutions are guided by the dependency grammar and by an ontology that allows obtaining a semantic representation of the structure and context of the question. Furthermore, this model will guide the SPARQL-based query generation.

8.4 Results

The main results of this thesis are a domain independent question ontological model for representing the syntactic structure and context of the natural language question, a question analysis process guided by natural language processing techniques and Semantic Web technologies, and a natural language interface to ontology-based knowledge bases. In the following sections, these results are briefly described.

8.4.1 Architecture

The NLI architecture proposed in this work is based on the generic architecture for NLIs proposed in (R. W. Smith 2006), and the high-level components of question answering systems over Linked Data presented in (Unger, Freitas, and Cimiano 2014). Figure 8-1 depicts the architecture proposed, which relies on five main modules: (1) question ontology model, (2) knowledge base pre-processing, (3) question processing, (4) question classification, and (5) query construction and execution. In a nutshell, the NLI works as follow. Firstly, the NLI takes the domain ontology as input parameter and process it to generate a domain lexicon. Once this lexicon is generated, the NLI can accept question expressed in natural language. When the question is provided, it is processed by

means of natural language techniques such as tokenization, POS-tagging, lemmatization, NER, and dependency analysis. Also, some terms contained in the question are mapped to domain knowledge resources based on the string similarity between them. The information obtained by this process is stored in the question ontology model, which will allow obtaining a semantic representation of the lexical structure of the question as well to establish the context of it. Based on such information, the NLI determines the question type provided by the users and the answer type expected by them. Furthermore, thanks to the semantic relations described by the question model, the NLI generates a SPARQL-based query to obtain the information that could represent the correct answer. Finally, the information is provided to the user as the answer. Next subsections will describe the outstanding modules of the NLI proposed.

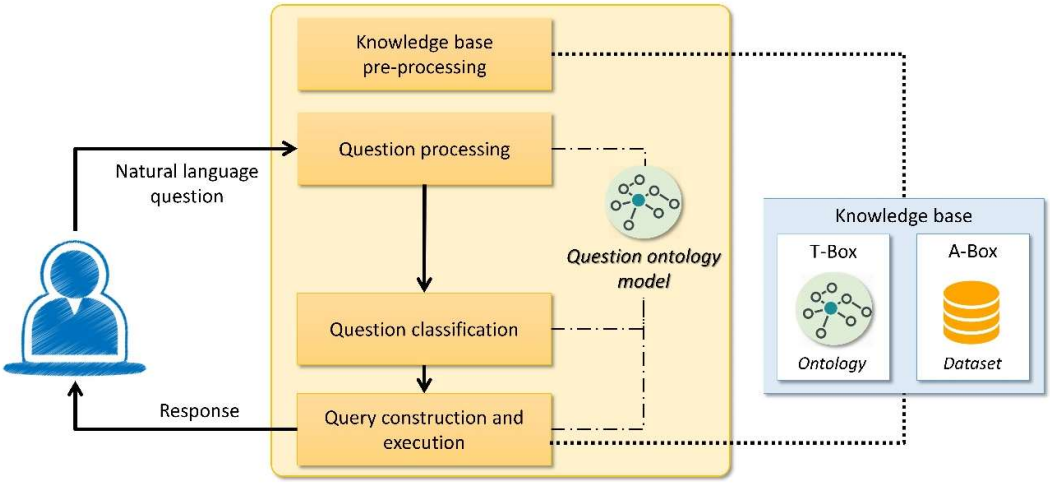

Figure 8-1. NLI's architecture.

8.4.2 Question ontology model

In this work, we propose a domain independent question ontological model for representing the syntactic structure and context of the natural language question. An excerpt of this model is presented in Figure 8-2.

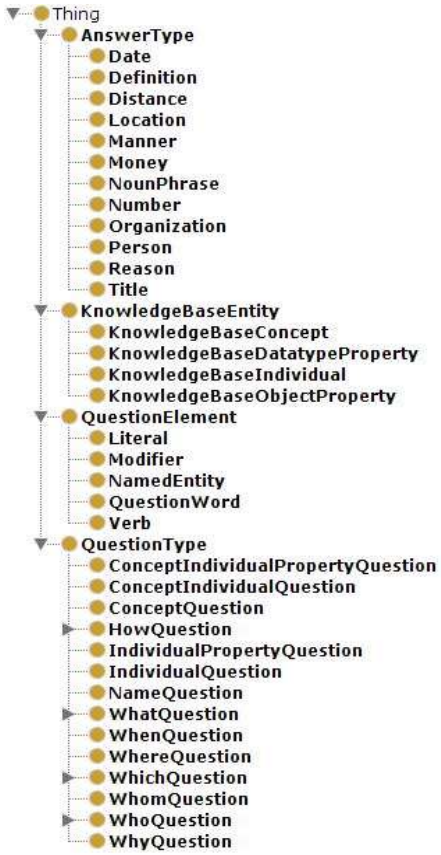

Figure 8-2. Question model.

On the one hand, the question model concepts that allows to describe the syntactic structure of the question are:

- ***question:QuestionElement***. It represents an element extracted from the question. It has three main properties: (1) *originalContent,* which represents the elements such as it appears in the question; (2) *lemma*, which represents the canonical form of the word; (3) POS, which represents the lexical category of the word; and (4) synonym, which represents words that can be used instead of the original word.
- ***question:Literal***. It represents fixed values.
- ***question:Modifier***. It is a word, phrase, or clause which functions as an adjective or an adverb to describe a word or make its meaning more specific.
- ***question:NamedEntity*.** It represents a word that potentially belongs to a predefined semantic category such as people's names, organizations and locations, among others.
- ***question:QuestionWord*.** It represents the interrogative particle detected in the question. Some examples of this class are: interrogative pronouns (what, which,

who and whom), and interrogative pro-adverb (how, when, whence, where, why), among others.

- ***question:QuestionType***. It represents the question type provided by the user.
- ***question:AnswerType***. It represents the answer type expected by the user.

On the other hand, the question model concepts that help to describe the question context are:

- ***question:KnowledgeBaseEntity***. It represents all knowledge base resources with which a word contained in the question has a relation. The main properties of this class are: (1) *URI*, which represents the uniform resource identifier of the resource; (2) *comment*, it is used to provide a human-readable description of a resource; and (3) *score*, which represents the string similarity level among the knowledge base resource and the question's word.
- ***question:KnowledgeBaseConcept***. It represents a concept described in the domain knowledge base.
- ***question:KnowledgeBaseIndividual***. It represents an individual of the domain knowledge base. It has a main property: *Type,* which represents the knowledge base class to which it belongs.
- ***question:KnowledgeBaseObjectProperty***. It represents an object property of the domain knowledge base.
- **question:KnowledgeBaseDatatypeProperty**. It represents a datatype property of the domain knowledge base.

8.4.3 Knowledge base pre-processing

Considering that the main challenge for an NLIKB is the question's interpretation in terms of relations and concepts defined in the knowledge base ontology (Damljanovic, Agatonovic, and Cunningham 2011), this module extracts all text descriptions of classes, object and datatype properties from the domain knowledge base to establish a domain lexicon. The text descriptions correspond to those provided through the *rdfs:label* which is an instance of *rdf:Property* that may be used to provide a human-readable version of a resource's name (Brickley and Guha 2016).

8.4.4 Question processing

This module processes the question aiming to obtain syntactic relationships between outstanding words in a question, as well as to map these words with knowledge base

resources. All information obtained by this process is stored in the question ontology model. This model performs next processes:

- **Tokenization**. It divides the question into a list of tokens. In this context, a token can be a keyword, an operator, or a punctuation mark.
- **POS tagging**. It assigns the lexical category to each word contained in the question.
- **Lemmatization**. It determines the canonical, dictionary or citation form of the word.
- **NER**. It aims to detect Named Entities (Grishman and Sundheim 1996), that is, entities belonging to a predefined semantic category such as people's names, organizations, and locations, among others.
- **Search for synonyms**. It extends the lexicon obtained from the question using synonyms of each extracted word. The synonyms used by this work are extracted from the lexical database WordNet (Miller 1995). This process increases the mapping possibility of a question's word to a knowledge base resource.
- **Knowledge base entities recognition**. It is responsible for identifying occurrences of knowledge base resources in the question, i.e., concepts, object and datatype properties, and individuals.
- **Dependence analysis**. It is based on the Stanford NLP (De Marneffe and Manning 2008) dependency parser, and it represents all grammatical relationships between occurrences of knowledge base resources found in the question. These relations are stored in the question model as triples subject-predicate-object.
- **Disambiguation**. Ambiguity is a type of uncertainty of meaning in which several interpretations are plausible. In this sense, this module performs a disambiguation process which is based on the lexical category of the word, as well as on the relationships that it has with object or datatype properties found in the question, whether through the range or domain of this property.

8.4.5 Question classification

This module determines the question type by means of a set of rules based on the dependency relationships between knowledge base resources found in the questions. Also, this classification is guided by the question classification proposed by Moldovan (D. Moldovan et al. 2000) which has been adapted to the context of knowledge bases. This

adaptation refers to the mapping of answer types to datatypes established by ontologies and vocabularies such as XML Schema (World Wide Web Consortium 2016b), FOAF (Brickley and Miller 2012), and the DBpedia's ontology. This question and answers classification is shown in Table 8-1.

Table 8-1. Question and answers classification.

Class	Subclass	Answer type
what	basic what	money – xsd:double | number – xsd:integer | definition – rdfs:comment, dbo:abstract | noun phrase – foaf:agent, dbo:agent
	what-who	person – foaf:Person, dbo:Person | organization – foaf:Organization, dbo:Organization
	what-when	date – xsd:date
	what-where	location – dbp:Place
who		person – foaf:Person, dbo:Person | organization – foaf:Organization, dbo:Organization
how	how-many	number – xsd:integer
	how-much	money – xsd:double | price - xsd:double
	how-far	distance - xsd:integer, xsd:double
	how-tall	number - xsd:integer, xsd:double
	how-large	number - xsd:integer, xsd:double, dbo:length
where		location – dbp:Place
when		date – xsd:date
which	which-who	person – foaf:Person, dbo:Person
	which-where	location – dbp:Place
	which-when	date – xsd:date
	which-what	noun phrase - foaf:agent, dbo:agent, dbo:Work | organization – foaf:Organization, dbo:Organization
Give/Name/List		noun phrase - foaf:agent, dbo:agent, dbo:Work | organization – foaf:Organization, dbo:Organization | location – dbp:Place

8.4.6 Query construction and execution

This module generates SPARQL-based queries that are executed against the domain knowledge base aiming to retrieve the information that could represent the correct answer. This process is based on a set of RDF Query language templates. These templates are associated with a specific triple pattern contained in the question ontology model, which provides a semantic representation of all grammatical relationships between

occurrences of knowledge base resources found in the question. Some of these RDF-based templates are shown in Table 8-2.

Table 8-2. RDF-based templates.

Question model relation	RDF-based template
KnowledgeBaseConcept **acl**[6] KnowledgeBaseObjectProperty	?concept a <CONCEPT>
KnowledgeBaseObjectProperty **dobj** KnowledgeBaseConcept	?var <OBJECTPROPERTY> ?concept. ?concept a <CONCEPT>
KnowledgeBaseObjectProperty **dobj** KnowledgeBaseIndividual	<INDIVIDUAL> <OBJECTPROPERTY> ?var
KnowledgeBaseObjectProperty **nmodAgent** KnowledgeBaseConcept	?var <OBJECTPROPERTY> ?concept. ?concept a <CONCEPT>
KnowledgeBaseObjectProperty **nmodAgent** KnowledgeBaseIndividual	?var < OBJECTPROPERTY> INDIVIDUAL
KnowledgeBaseObjectProperty **nmodBy** KnowledgeBaseIndividual	?var < OBJECTPROPERTY> INDIVIDUAL
KnowledgeBaseObjectProperty **nmodIn** KnowledgeBaseConcept	?var <OBJECTPROPERTY> ?concept. ?concept a <CONCEPT>
KnowledgeBaseObjectProperty **nmodIn** KnowledgeBaseIndividual	<INDIVIDUAL> <OBJECTPROPERTY> ?var
KnowledgeBaseConcept **nmodOf** KnowledgeBaseIndividual	<INDIVIDUAL> ?relation ?concept. ?relation rdfs:range <CONCEPT>
KnowledgeBaseConcept **nmodOf** KnowledgeBaseDatatypeProperty	?concept < DATATYPE > ?var
KnowledgeBaseDatatypeProperty **nmodOf** KnowledgeBaseIndividual	<INDIVIDUAL> < DATATYPE> ?var
KnowledgeBaseObjectProperty **nmodOf** KnowledgeBaseIndividual	<INDIVIDUAL> <OBJECTPROPERTY> ?var
KnowledgeBaseIndividual **nmodPoss** KnowledgeBaseObjectProperty	<INDIVIDUAL> <OBJECTPROPERTY> ?var
KnowledgeBaseObjectProperty **nmodPoss** KnowledgeBaseIndividual	<INDIVIDUAL> <OBJECTPROPERTY> ?var
KnowledgeBaseIndividual **nsubj** KnowledgeBaseConcept	<INDIVIDUAL> ?relation ?concept. ?concept a <CONCEPT>
KnowledgeBaseIndividual **nsubj** KnowledgeBaseObjectProperty	<INDIVIDUAL> <OBJECTPROPERTY> ?var
KnowledgeBaseObjectProperty **nsubjpass** KnowledgeBaseConcept	?concept < OBJECTPROPERTY> ?var. ?concept a <CONCEPT>

[6] This pattern is used in conjunction with the nmod:in and nmod:by relations.

The SPARQL query generation is guided partially by the relations described by the question ontology model that share a knowledge base resource. Finally, we execute the generated SPARQL queries in the knowledge base.

8.5 Evaluation

In this section, we evaluated the NLI to measure its effectiveness in providing information (domain knowledge base resources) that answers the user's question. Furthermore, this evaluation involved two different knowledge bases aiming to determine the portability of the NLI. The knowledge bases involved are:

- **DBpedia** (Lehmann et al. 2015). It represents a crowd-sourced community effort to structured information from Wikipedia and makes this information available on the Web. This knowledge base provides a description of 4.58 million of things such as places, people, and organizations, among others.
- **MusicBrainz** (Swartz 2002). It is already widely used as information source for music-related URIs in the Linked Data community.

The validation process performed in this work was decomposed by four tasks:

1. **Collection of a corpus of natural language questions**. This task aimed to obtaine a set of question expressed in natural language that must be able to be answered through the information contained in the knowledge bases involved. The corpus obtained consists of questions obtained from corpus provided by QALD (Lopez et al. 2013) as well as question provided by people outside the project. The resulting corpus has 200 questions, 100 for the DBpedia domain and the rest for the MusicBrainz domain.
2. **SPARQL queries generation**. A group of expert users built a SPARQL-based query for each question provided. It is an important task for this evaluation because the knowledge base resources recovered through this query are considered the correct answer.
3. **The natural language questions contained in the corpus were executed through the NLI proposed in this work**.
4. **Evaluation**. The information provided as answer by the NLI was compared with the information retrieved by means of its respective SPARQL-based query. In this evaluation, we used the primary metrics precision and recall, and their harmonic mean, known as the F-measure.

The evaluation results obtained by this process are described below.

8.5.1 Evaluation results obtained in the DBpedia's domain.

Figure 8-3 shows the evaluation results obtained in the DBpedia's domain for each of the question type supported by the NLI presented in this work. In such figure, it can be observed that there are no significant differences among the results obtained for each question type. This fact can be interpreted as a constant effectiveness of the interface to provide the correct answers regardless the question type provided. Furthermore, in this figure, we can observe that the highest values for the recall and F-measure metrics were obtained by the *Who* question type, with values of 0.8889 and 08828 respectively. This fact indicates that the system can provide the correct answer to most questions whose expected answer corresponds with instances of classes such as Person, Organization, etc. Meanwhile, the lowest values were obtained by the *How* question type, with a precision value of 0.7571, a recall value of 0.7707, and an F-measure value of 0.7630.

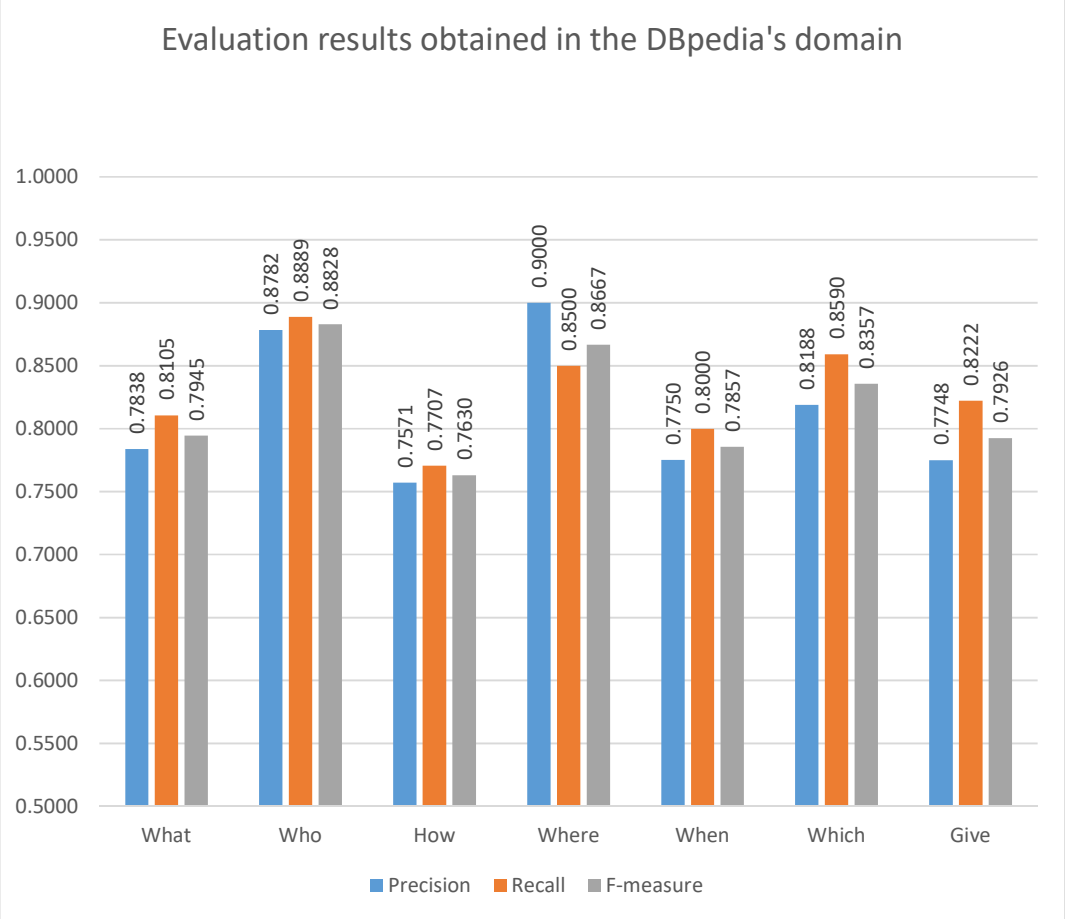

Figure 8-3. Evaluation results obtained in the DBpedia's domain.

8.5.2 Evaluation results obtained in the MusicBrainz's domain.

The evaluations results obtained in the MusicBrainz's domain are shown in Figure 8-4. Here, it can be observed that, such as occurs in the DBpedia's domain, there are no big differences between the results obtained for each question type. Again, the question type with the highest values is *Who* with values of 0.265, 0.8389, y 0.8319 for the precision, recall, and F-measure metrics respectively. The question type with the lowest precision value is *Who* with 0.7770. Meanwhile, the lowest values for the recall and F-measure metrics were obtained by the question type *Give* with the values of 0.7745 and 0.7721 respectively.

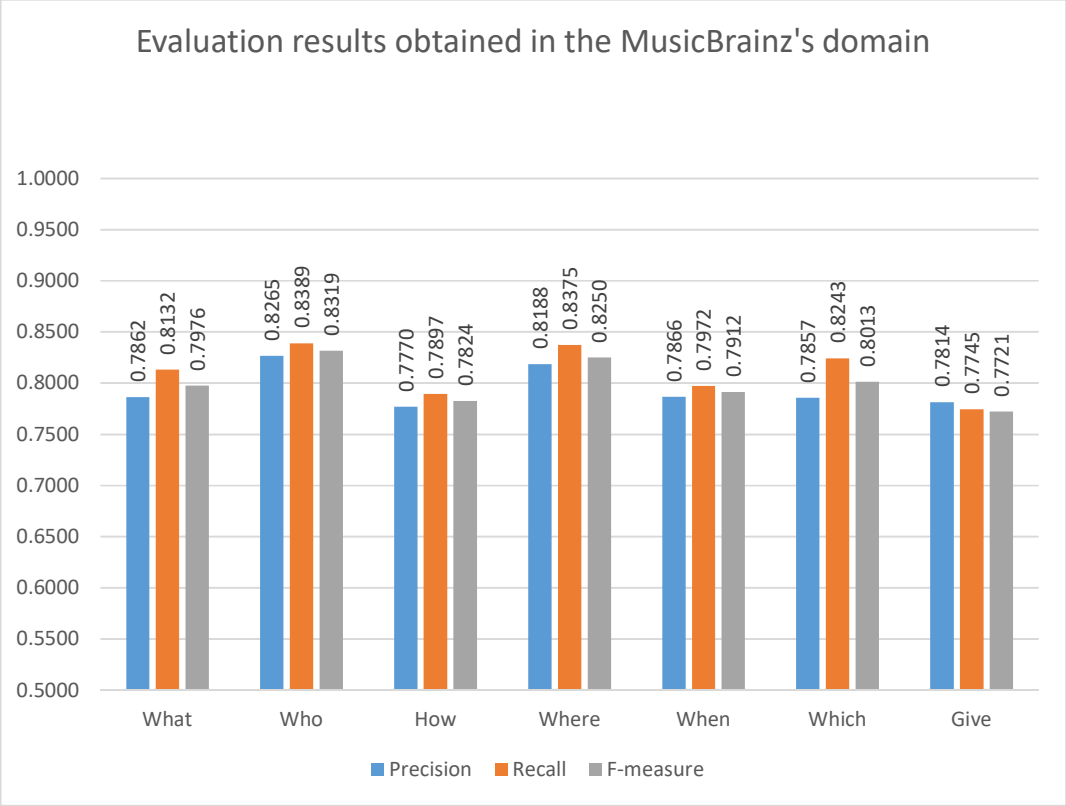

Figure 8-4. Evaluation results obtained in the MusicBrainz's domain.

As was previously mentioned, one of the main goals of this last validation scenario was to validate the portability of the NLI proposed. Considering the results presented in Figure 8-3 and Figure 8-4, we can conclude that this NLI can provide the correct answers to most questions regardless the domain knowledge base. Despite there is a bit difference among above-mentioned results, we have perceived that this occurred due to in the MusicBrainz domain, a higher number of question have references to knowledge base individuals whose names are multiword terms. The recognition of this kind of terms is a limitation of our NLI which, in addition to others, will be discussed in next section.

8.6 Conclusions and future work

The ontology-based knowledge bases have been adopted by individuals and organizations of different domains. However, current mechanisms for accessing this type of knowledge bases are intended to be used by users with knowledge and expertise on Semantic Web technologies. Due to this reason, there is a need for making this kind of information accessible to all kind of users by means of intuitive and easy to use mechanisms. Therefore, in this piece of research, we have presented our effort to provide natural language solutions for reducing the gap between non-expert users and ontology-based knowledge bases. These solutions have been successfully used in the DBpedia and MusicBrainz domains obtaining encouraging results. In summary, the main contributions of this research effort are the following:

- **Question ontology model**. One of the main contributions of this work is a domain independent question ontological model for representing the syntactic structure and context of the natural language question. This model allows obtaining semantic representations of the question that guide the SPARQL-based queries generation.
- **Question and answer classification adapted to the context of ontology-based knowledge bases**. This contribution makes references to the adaption of the question classification proposed by Moldovan (D. Moldovan et al. 2000). This adaptation consisted of mapping of answer types to datatypes established by ontologies and vocabularies such as XML Schema (World Wide Web Consortium 2016b), FOAF (Brickley and Miller 2012), and the DBpedia's ontology.
- **Set of RDF-based templates**. This contribution consists of a set of RDF-based templates that are the basis for the generation of a graph pattern that constitute the SPARQL-based query.
- **Validation of the NLI in multiple domains**. The NLI was evaluated on Linked Data based knowledge bases, namely, DBpedia and MusicBrainz. The results seem promising and they can be interpreted as a good portability level of the NLI formalized in this thesis.

Despite all the advantages and possibilities of the presented approach, we have detected several limitations that might be improved in future; these ones are summarized as follows:

- **Question types supporting**. The NLI presented in this work can only deal with factual questions and some imperative forms. We are aware that this fact restricts the expressivity of the user. Therefore, it is necessary to consider a wider set of question types, which will demand the generation of new RDF-based templates though which could be possible to generate the corresponding SPARQL-based queries.
- **Ambiguity problems**. Sometimes the NLI here presented provided wrong answers due to ambiguity problems. Despite our approach deals with some of these problems, we are aware that it is not enough. However, this issue can be addressed through the application of feedback mechanisms that enable users to solve possible ambiguities that cannot be solved by our approach.
- **Multiword expressions**. The NLI relates terms contained in the question with knowledge base resources, however, sometimes the system fails recognizing terms composed of multiple words, especially when these terms are in lowercase letters. Multiword expressions are a key problem for the development of natural language processing technology (Sag et al. 2002). Some techniques that can be adopted to address this limitation are presented in the aforementioned work.

Furthermore, some future research lines are:

- **Querying distributed linked data**. It is important to consider the distribution of information when we look for a response because sometimes it can depend on more than one knowledge base. In this sense, it is necessary to implement mechanisms that allow performing queries in decentralized information sources such as Linked Data. Some of these mechanisms are federated queries to SPARQL endpoints, fragments of triplet patterns, and Linked Data flows, which are described in (Saleem et al., 2015).
- **Multilingualism**. It is necessary to implement mechanisms that allow providing answers to question formulated in multiple languages. This task demands that knowledge bases resources have a description of them in each language to be considered. A clear example of a knowledge base that provides this information is DBpedia which provides a description of its resources in languages such as Spanish, English, Russian, among others. In this sense, we think that our approach could integrate mechanisms that allow it to address the above discussed.

Referencias

Aasman, Jans. 2006. “Allegro Graph: RDF Triple Database.” *Cidade: Oakland Franz Incorporated*.

Adida, Ben, and Mark Birbeck. 2008. “RDFa Primer: Bridging the Human and Data Webs.” *Retrieved June* 20: 2008.

Alexopoulou, Dimitra, Thomas Wächter, Laura Pickersgill, Cecilia Eyre, and Michael Schroeder. 2008. “Terminologies for Text-Mining; an Experiment in the Lipoprotein Metabolism Domain.” *BMC Bioinformatics* 9 (4): S2.

Alonso, Laura, Irene Castellón, Salvador Climent, Maria Fuentes, Lluís Padró, and H Rodrıguez. 2004. “Approaches to Text Summarization: Questions and Answers.” *Inteligencia Artificial* 8: 22.

Androutsopoulos, Ion, Graeme D Ritchie, and Peter Thanisch. 1995. “Natural Language Interfaces to Databases–an Introduction.” *Natural Language Engineering* 1 (01): 29–81.

Antoniou, Grigoris, and Frank Van Harmelen. 2004. *A Semantic Web Primer*. MIT Press.

Athenikos, Sofia J., and Hyoil Han. 2010. “Biomedical Question Answering: A Survey.” *Computer Methods and Programs in Biomedicine* 99 (1): 1–24. doi:10.1016/j.cmpb.2009.10.003.

Attardi, Giuseppe, Antonio Cisternino, Francesco Formica, Maria Simi, and Alessandro Tommasi. 2002. “PiQASso 2002.” In *TREC*.

Baader, Franz, Sebastian Brandt, and Carsten Lutz. 2005. “Pushing the EL Envelope.” In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, 364–369. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. http://dl.acm.org/citation.cfm?id=1642293.1642351.

Baader, Franz, Ian Horrocks, and Ulrike Sattler. 2008. “Description Logics.” *Foundations of Artificial Intelligence* 3: 135–179.

Baader, Franz, and Werner Nutt. 2003. “The Description Logic Handbook.” In , edited by Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and

Peter F. Patel-Schneider, 43–95. New York, NY, USA: Cambridge University Press. http://dl.acm.org/citation.cfm?id=885746.885749.

Beckett, Dave, and Brian McBride. 2004. “RDF/XML Syntax Specification (Revised).” *W3C Recommendation* 10.

Berners-Lee, Tim. 2011. “Linked Data-Design Issues (2006).” *URL Http://Www. w3. Org/DesignIssues/LinkedData. Html*.

Berners-Lee, Tim, James Hendler, Ora Lassila, and others. 2001. “The Semantic Web.” *Scientific American* 284 (5): 28–37.

Binot, JL, L Debille, D Sedlock, and B Vandecapelle. 1991. “Natural Language Interfaces: A New Philosophy.” *SunExpert Magazine* 2 (1): 67–73.

Bizer, Christian, Tom Heath, and Tim Berners-Lee. 2009. “Linked Data-the Story so Far.” *Semantic Services, Interoperability and Web Applications: Emerging Concepts*, 205–227.

Blackburn, Patrick, and Johan Bos. 2005. “Representation and Inference for Natural Language.” *A First Course in Computational Semantics. CSLI*.

Bledsoe, W. W., and I. Browning. 1959. “Pattern Recognition and Reading by Machine.” In *Papers Presented at the December 1-3, 1959, Eastern Joint IRE-AIEE-ACM Computer Conference*, 225–232. New York, NY, USA: ACM. doi:10.1145/1460299.1460326.

Boeuf, Patrick Le. 2001. “FRBR and Further.” *Cataloging & Classification Quarterly* 32 (4): 15–52.

Borgo, Stefano, Nicola Guarino, and Claudio Masolo. 1996. “Stratified Ontologies: The Case of Physical Objects.” In *In Proceedings of ECAI-96 Workshop on Ontological Engineering*, 5–15.

Borst, Pim, Hans Akkermans, and Jan Top. 1997. “Engineering Ontologies.” *International Journal of Human-Computer Studies* 46 (2): 365–406. doi:10.1006/ijhc.1996.0096.

Brickley, Dan, and R.V. Guha. 2016. “RDF Schema 1.1.” October 9. https://www.w3.org/TR/rdf-schema/.

Brickley, Dan, and Libby Miller. 2012. “FOAF Vocabulary Specification 0.98.” *Namespace Document* 9.

Brill, Eric, Jimmy J Lin, Michele Banko, Susan T Dumais, Andrew Y Ng, and others. 2001. “Data-Intensive Question Answering.” In *TREC*, 56:90.

Cahlink, George. 2000. “Data Mining Taps the Trends.” *Government Executive Magazine*.

Carbonell, Jaime G, Steve Klein, David Miller, Mike Steinbaum, Tomer Grassiany, and Jochen Frey. 2006. “Context-Based Machine Translation.” *Proceedings of the Association for Machine Translation of the Americas (AMTA-2006)*, 19–28.

Chen, Hsinchun, Roger H. L. Chiang, and Veda C. Storey. 2012. “Business Intelligence and Analytics: From Big Data to Big Impact.” *MIS Q.* 36 (4): 1165–1188.

Chomsky, Noam. 1956. “Three Models for the Description of Language.” *IRE Transactions on Information Theory* 2 (3): 113–124.

Chowdhury, Gobinda G. 2003. “Natural Language Processing.” *Annual Review of Information Science and Technology* 37 (1): 51–89. doi:10.1002/aris.1440370103.

Chu-Carroll, Jennifer, John Prager, Christopher Welty, Krzysztof Czuba, and David Ferrucci. 2006. “A Multi-Strategy and Multi-Source Approach to Question Answering.” DTIC Document.

Church, Kenneth Ward. 1980. “On Memory Limitations in Natural Language Processing.”

Cimiano, Philipp, Peter Haase, Jörg Heizmann, Matthias Mantel, and Rudi Studer. 2008. “Towards Portable Natural Language Interfaces to Knowledge bases–The Case of the ORAKEL System.” *Data & Knowledge Engineering* 65 (2): 325–354.

Clarke, Sarah J, and Peter Willett. 1997. “Estimating the Recall Performance of Web Search Engines.” In *Aslib Proceedings*, 49:184–189. MCB UP Ltd.

Codd, Edgar F. 1974. *Seven Steps to Rendezvous with the Casual User*. IBM Thomas J. Watson Research Division.

Cohen, Philip R. 1992. “The Role of Natural Language in a Multimodal Interface.” In *Proceedings of the 5th Annual ACM Symposium on User Interface Software and Technology*, 143–149. ACM.

Collins, Michael, and Nigel Duffy. 2001. “Convolution Kernels for Natural Language.” In *Advances in Neural Information Processing Systems*, 625–632.

Colmerauer, Alain. 1975. “Les Grammaires de Métamorphose GIA.” *Internal Publica*.

Colombo-Mendoza, Luis Omar, Rafael Valencia-García, Alejandro Rodríguez-González, Giner Alor-Hernández, and José Javier Samper-Zapater. 2015. “RecomMetz: A Context-Aware Knowledge-Based Mobile Recommender System for Movie Showtimes.” *Expert Systems with Applications* 42 (3): 1202–22. doi:10.1016/j.eswa.2014.09.016.

Cowie, Jim, and Wendy Lehnert. 1996. “Information Extraction.” *Commun. ACM* 39 (1): 80–91. doi:10.1145/234173.234209.

Croft, Bruce, and John Lafferty. 2013. *Language Modeling for Information Retrieval*. Vol. 13. Springer Science & Business Media.

Croset, Samuel, John P Overington, and Dietrich Rebholz-Schuhmann. 2013. “Brain, a Library for the OWL2 EL Profile.” In *OWLED*.

Cunningham, Hamish. 2005. “Information Extraction, Automatic.” *Encyclopedia of Language and Linguistics,* 665–677.

Damljanovic, Danica, Milan Agatonovic, and Hamish Cunningham. 2010. “Natural Language Interfaces to Ontologies: Combining Syntactic Analysis and Ontology-Based Lookup through the User Interaction.” In *Extended Semantic Web Conference*, 106–120. Springer.

Damljanovic, Danica, Milan Agatonovic, and Hamish Cunningham. 2011. “FREyA: An Interactive Way of Querying Linked Data Using Natural Language.” In *Extended Semantic Web Conference*, 125–138. Springer.

Damljanović, Danica, and Kalina Bontcheva. 2010. “Towards Enhanced Usability of Natural Language Interfaces to Knowledge Bases.” In *Web 2.0 & Semantic Web*, edited by Vladan Devedžić and Dragan Gaševic, 105–33. Springer US. http://link.springer.com/chapter/10.1007/978-1-4419-1219-0_5.

Davies, John, Dieter Fensel, and Frank Van Harmelen. 2003. *Towards the Semantic Web: Ontology-Driven Knowledge Management*. John Wiley & Sons.

De Marneffe, Marie-Catherine, and Christopher D Manning. 2008. "The Stanford Typed Dependencies Representation." In *Coling 2008: Proceedings of the Workshop on Cross-Framework and Cross-Domain Parser Evaluation*, 1–8. Association for Computational Linguistics.

Dijk, Teun Adrianus van, and José Antonio Mayoral. 1987. *Pragmática de la comunicación literaria*. Arco/Libros.

Doddington, George R, Alexis Mitchell, Mark A Przybocki, Lance A Ramshaw, Stephanie Strassel, and Ralph M Weischedel. 2004. "The Automatic Content Extraction (ACE) Program-Tasks, Data, and Evaluation." In *LREC*, 2:1.

Dogmus, Zeynep, Volkan Patoglu, and Esra Erdem. 2014. "Ontological Query Answering about Rehabilitation Robotics." *Standardized Knowledge Representation and Ontologies for Robotics and Automation*, September, 20.

Edelstein, Herbert A. 1998. *Introduction to Data Mining & Knowledge Discovery*. Edición: 2nd Spiral. Potomac, MD: Two Crows Corp.

Elbedweihy, Khadija, Stuart N Wrigley, and Fabio Ciravegna. 2012. "Evaluating Semantic Search Query Approaches with Expert and Casual Users." In *International Semantic Web Conference*, 274–286. Springer.

Erdmann, Maike, Kotaro Nakayama, Takahiro Hara, and Shojiro Nishio. 2008. "Extraction of Bilingual Terminology from a Multilingual Web-Based Encyclopedia." *Journal of Information Processing* 16: 68–79. doi:10.2197/ipsjjip.16.68.

Esuli, Andrea, and Fabrizio Sebastiani. 2006. "Sentiwordnet: A Publicly Available Lexical Resource for Opinion Mining." In *Proceedings of LREC*, 6:417–422. Citeseer.

Feldman, Susan. 1999. "NLP Meets the Jabberwocky: Natural Language Processing in Information Retrieval." *ONLINE-WESTON THEN WILTON-* 23: 62–73.

Ferrández, Óscar, Rubén Izquierdo, Sergio Ferrández, and José Luis Vicedo. 2009. "Addressing Ontology-Based Question Answering with Collections of User

Queries." *Information Processing & Management* 45 (2): 175–88. doi:10.1016/j.ipm.2008.09.001.

Ferrández, Óscar, Christian Spurk, Milen Kouylekov, Iustin Dornescu, Sergio Ferrández, Matteo Negri, Rubén Izquierdo, et al. 2011. "The QALL-ME Framework: A Specifiable-Domain Multilingual Question Answering Architecture." *Web Semantics: Science, Services and Agents on the World Wide Web* 9 (2): 137–145.

Finkel, Jenny Rose, Trond Grenager, and Christopher Manning. 2005. "Incorporating Non-Local Information into Information Extraction Systems by Gibbs Sampling." In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, 363–370. Association for Computational Linguistics.

Frank, Anette, Hans-Ulrich Krieger, Feiyu Xu, Hans Uszkoreit, Berthold Crysmann, Brigitte Jörg, and Ulrich Schäfer. 2007. "Question Answering from Structured Knowledge Sources." *Journal of Applied Logic* 5 (1): 20–48. doi:10.1016/j.jal.2005.12.006.

Fuhr, Norbert, Jaap Kamps, Mounia Lalmas, and Andrew Trotman, eds. 2008. *Focused Access to XML Documents*. Vol. 4862. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg. http://link.springer.com/10.1007/978-3-540-85902-4.

García Moreno, Carlos. 2015. "Desarrollo de un modelo para la gestión de la I+D+i soportado por tecnologías de la web semántica," December. https://digitum.um.es/xmlui/handle/10201/46942.

Giannakopoulos, George, Petra Mavridi, Georgios Paliouras, George Papadakis, and Konstantinos Tserpes. 2012. "Representation Models for Text Classification: A Comparative Analysis over Three Web Document Types." In *Proceedings of the 2nd International Conference on Web Intelligence, Mining and Semantics*, 13. ACM.

Girle, Roderic A, and Melvin Fitting. 1998. *First-Order Logic and Automated Theorem Proving*. JSTOR.

Green, Cordell. 1969. "Theorem Proving by Resolution as a Basis for Question-Answering Systems." *Machine Intelligence* 4: 183–205.

Green Jr, Bert F, Alice K Wolf, Carol Chomsky, and Kenneth Laughery. 1961. "Baseball: An Automatic Question-Answerer." In *Papers Presented at the May 9-11, 1961, Western Joint IRE-AIEE-ACM Computer Conference*, 219–224. ACM.

Grishman, Ralph, and Beth Sundheim. 1996. "Message Understanding Conference-6: A Brief History." In *COLING*, 96:466–471.

Grosz, Barbara J, and others. 1977. "The Representation and Use of Focus in a System for Understanding Dialogs." In *IJCAI*, 67:76.

Gruber, Thomas R. 1993. "A Translation Approach to Portable Ontology Specifications." *Knowledge Acquisition* 5 (2): 199–220. doi:10.1006/knac.1993.1008.

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

Guarino, Nicola, and L Boldrin. 1993. "Concepts and Relations." *Pre-Proceedings of the International Workshop on Formal Ontology*.

Guarino, Nicola, Daniel Oberle, and Steffen Staab. 2009. "What Is an Ontology?" In *Handbook on Ontologies*, edited by Steffen Staab and Rudi Studer, 1–17. Springer Berlin Heidelberg. http://link.springer.com/chapter/10.1007/978-3-540-92673-3_0.

Haarslev, Volker, Ralf Möller, and AY Turhan. 2001. "Racer User's Guide and Reference Manual." *Hamburg Universität*.

Habernal, Ivan, and Miloslav KonopíK. 2013. "SWSNL: Semantic Web Search Using Natural Language." *Expert Systems with Applications* 40 (9): 3649–3664.

Habert, Benoit, Gilles Adda, M Adda-Decker, P Boula de Marëuil, S Ferrari, O Ferret, G Illouz, and P Paroubek. 1998. "Towards Tokenization Evaluation." In *Proceedings of LREC*, 98:427–431.

Hadzic, Maja, Pornpit Wongthongtham, Tharam Dillon, and Elizabeth Chang. 2009. "Introduction to Ontology." In *Ontology-Based Multi-Agent Systems*, 37–60. Studies in Computational Intelligence 219. Springer Berlin Heidelberg. doi:10.1007/978-3-642-01904-3_3.

Hamon, Thierry, Natalia Grabar, and Fleur Mougin. 2016. "Querying Biomedical Linked Data with Natural Language Questions." *Semantic Web*, no. Preprint: 1–19.

Harabagiu, Sanda M, Dan I Moldovan, Marius Paşca, Rada Mihalcea, Mihai Surdeanu, Răzvan Bunescu, Corina R Gîrju, Vasile Rus, and Paul Morărescu. 2000. "Falcon: Boosting Knowledge for Answer Engines."

Harris, Larry R. 1984. "Experience with INTELLECT: Artificial Intelligence Technology Transfer." *AI Magazine* 5 (2): 43.

Heath, Tom, and Christian Bizer. 2011. *Linked Data:Evolving the Web into a Global Data Space*. Morgan & Claypool. http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6812934.

Heflin, Jeff, James Hendler, and Sean Luke. 1998. "Reading between the Lines: Using SHOE to Discover Implicit Knowledge from the Web." In *AAAI-98 Workshop on AI and Information Integration*. Vol. 297.

Heilman, Michael, and Noah A Smith. 2010. "Tree Edit Models for Recognizing Textual Entailments, Paraphrases, and Answers to Questions." In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 1011–1019. Association for Computational Linguistics.

Hendrix, Gary G. 1982. "Natural-Language Interface." *American Journal of Computational Linguistics* 8 (2): 57.

Hendrix, Gary G. 1986. "Q&a: Already a Success." In *International Conference on Computational Linguistics*, 164–166.

Hendrix, Gary G, Earl D Sacerdoti, Daniel Sagalowicz, and Jonathan Slocum. 1978. "Developing a Natural Language Interface to Complex Data." *ACM Transactions on Database Systems (TODS)* 3 (2): 105–147.

Hermjakob, Ulf, Abdessamad Echihabi, and Daniel Marcu. 2002. "Natural Language Based Reformulation Resource and Wide Exploitation for Question Answering." In *TREC*, 90:91.

Hernández, Myriam Beatriz, and José M Gómez. 2013. "Aplicaciones de Procesamiento de Lenguaje Natural." *Revista Politécnica* 32.

Hickl, Andrew, Kirk Roberts, Bryan Rink, Jeremy Bensley, Tobias Jungen, Ying Shi, and John Williams. 2007. “Question Answering with LCC’s CHAUCER-2 at TREC 2007.” In *TREC*, 2:2–1.

Hobbs, Jerry R. 1978. “Resolving Pronoun References.” *Lingua* 44 (4): 311–338.

Hobbs, Jerry R, Mark Stickel, Paul Martin, and Douglas Edwards. 1988. “Interpretation as Abduction.” In *Proceedings of the 26th Annual Meeting on Association for Computational Linguistics*, 95–103. Association for Computational Linguistics.

Horrocks, Ian. 1998. “The FaCT System.” In *Automated Reasoning with Analytic Tableaux and Related Methods*, edited by Harrie de Swart, 307–12. Springer Berlin Heidelberg. http://link.springer.com/chapter/10.1007/3-540-69778-0_30.

Horrocks, Ian. 2005. “OWL: A Description Logic Based Ontology Language.” In *Logic Programming*, edited by Maurizio Gabbrielli and Gopal Gupta, 1–4. Springer Berlin Heidelberg. http://link.springer.com/chapter/10.1007/11562931_1.

Hovy, Eduard H, Laurie Gerber, Ulf Hermjakob, Michael Junk, and Chin-Yew Lin. 2000. “Question Answering in Webclopedia.” In *TREC*, 52:53–56.

Hovy, Eduard, Ulf Hermjakob, and Deepak Ravichandran. 2002. “A Question/Answer Typology with Surface Text Patterns.” In *Proceedings of the Second International Conference on Human Language Technology Research*, 247–251. Morgan Kaufmann Publishers Inc.

Hripcsak, George, and Adam S. Rothschild. 2005. “Agreement, the F-Measure, and Reliability in Information Retrieval.” *Journal of the American Medical Informatics Association* 12 (3): 296–98. doi:10.1197/jamia.M1733.

Hsieh, Haowei, and Frank M. Shipman. 2002. “Manipulating Structured Information in a Visual Workspace.” In *Proceedings of the 15th Annual ACM Symposium on User Interface Software and Technology*, 217–226. New York, NY, USA: ACM. doi:10.1145/571985.572018.

Indurkhya, Nitin, and Fred J. Damerau. 2010. *Handbook of Natural Language Processing*. 2nd ed. Chapman & Hall/CRC.

Jacobson, Kurt, Simon Dixon, and Mark Sandler. 2010. “LinkedBrainz: Providing the MusicBrainz Next Generation Schema as Linked Data.” In *Late-Breaking Demo*

Session at the 11th International Society for Music Information Retrieval Conference.

Johnson, Tim. 1984. "Natural Language Computing: The Commercial Applications." *The Knowledge Engineering Review* 1 (3): 11–23. doi:10.1017/S0269888900000588.

Jones, K Sparck, and others. 1999. "Automatic Summarizing: Factors and Directions." *Advances in Automatic Text Summarization*, 1–12.

Joshi, Aravind K, and Phil Hopely. 1996. "A Parser from Antiquity." *Natural Language Engineering* 2 (04): 291–294.

Jubilson, E. Ajith, P. Dhanavanthini, P. Victer Paul, V. Pravinpathi, M. RamCoumare, and S. Paranidharan. 2016. "Intelligent Telecommunication System Using Semantic-Based Information Retrieval." In *Proceedings of the Second International Conference on Computer and Communication Technologies*, edited by Suresh Chandra Satapathy, K. Srujan Raju, Jyotsna Kumar Mandal, and Vikrant Bhateja, 137–43. Springer India. http://link.springer.com/chapter/10.1007/978-81-322-2526-3_15.

Jurafsky, Dan, and James H Martin. 2014. "Semantic Analysis." In *Speech and Language Processing*, 545–87. Pearson.

Kando, Noriko. 2005. "Overview of the Fifth NTCIR Workshop." In *NTCIR*.

Kaplan, Ronald M, and Martin Kay. 1981. "Phonological Rules and Finite-State Transducers." In *Linguistic Society of America Meeting Handbook, Fifty-Sixth Annual Meeting*, 27–30.

Kaufmann, Esther, and Abraham Bernstein. 2007. "How Useful Are Natural Language Interfaces to the Semantic Web for Casual End-Users?" In *Proceedings of the 6th International The Semantic Web and 2Nd Asian Conference on Asian Semantic Web Conference*, 281–294. Berlin, Heidelberg: Springer-Verlag. http://dl.acm.org/citation.cfm?id=1785162.1785184.

Kaufmann, Esther, and Abraham Bernstein. 2010. "Evaluating the Usability of Natural Language Query Languages and Interfaces to Semantic Web Knowledge Bases." *Web Semantics: Science, Services and Agents on the World Wide Web* 8 (4): 377–393.

Kaufmann, Esther, Abraham Bernstein, and Lorenz Fischer. 2007. “NLP-Reduce: A Naive but Domainindependent Natural Language Interface for Querying Ontologies.” In *4th European Semantic Web Conference ESWC*, 1–2.

Kaufmann, Esther, Abraham Bernstein, and Renato Zumstein. 2006. “Querix: A Natural Language Interface to Query Ontologies Based on Clarification Dialogs.” In *5th International Semantic Web Conference (ISWC 2006)*, 980–981. Springer.

Kazakov, Yevgeny, Markus Krötzsch, and Frantisek Simancik. 2012. “ELK Reasoner: Architecture and Evaluation.” In *ORE*.

Khamis, SK, F Mosteller, and DL Wallace. 1966. *Inference and Disputed Authorship: The Federalist*. JSTOR.

Khayrallah, Huda, Sean Trott, and Jerome Feldman. 2015. “Natural Language For Human Robot Interaction.” In *Proceedings of the Workshop on Human-Robot Teaming at the 10th ACM/IEEE International Conference on Human-Robot Interaction, Portland, Oregon*.

Kiryakov, Atanas, Borislav Popov, Ivan Terziev, Dimitar Manov, and Damyan Ognyanoff. 2004. “Semantic Annotation, Indexing, and Retrieval.” *Web Semantics: Science, Services and Agents on the World Wide Web* 2 (1): 49–79. doi:10.1016/j.websem.2004.07.005.

Klyne, Graham, Jeremy Carrol, and Brian McBride. 2016. “RDF 1.1 Concepts and Abstract Syntax.” October 11. https://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/.

Kolomiyets, Oleksandr, and Marie-Francine Moens. 2011. “A Survey on Question Answering Technology from an Information Retrieval Perspective.” *Information Sciences* 181 (24): 5412–34. doi:10.1016/j.ins.2011.07.047.

Kontokostas, Dimitris, Charalampos Bratsas, Sören Auer, Sebastian Hellmann, Ioannis Antoniou, and George Metakides. 2012. “Internationalization of Linked Data: The Case of the Greek DBpedia Edition.” *Web Semantics: Science, Services and Agents on the World Wide Web* 15 (September): 51–61. doi:10.1016/j.websem.2012.01.001.

Krauthammer, Michael, and Goran Nenadic. 2004. “Term Identification in the Biomedical Literature.” *Journal of Biomedical Informatics* 37 (6): 512–526.

Kübler, Sandra, Ryan McDonald, and Joakim Nivre. 2009. “Dependency Parsing.” *Synthesis Lectures on Human Language Technologies* 1 (1): 1–127.

Kucera, Henry, W. Nelson Francis, John B. Carroll, and W. F. Twaddell. 1967. *Computational Analysis of Present Day American English*. 1St Edition edition. Providence: Brown University Press.

Kullback, Solomon, and Richard A Leibler. 1951. “On Information and Sufficiency.” *The Annals of Mathematical Statistics* 22 (1): 79–86.

Kumar, Anand, and Barry Smith. 2005. “Oncology Ontology in the NCI Thesaurus.” In *Artificial Intelligence in Medicine*, edited by Silvia Miksch, Jim Hunter, and Elpida T. Keravnou, 213–20. Springer Berlin Heidelberg. http://link.springer.com/chapter/10.1007/11527770_30.

Kupiec, Julian. 1993. “MURAX: A Robust Linguistic Approach for Question Answering Using an on-Line Encyclopedia.” In *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 181–190. ACM.

Lee, Gary Geunbae, Jungyun Seo, Seungwoo Lee, Hanmin Jung, Bong-Hyun Cho, Changki Lee, Byung-Kwan Kwak, et al. 2001. “SiteQ: Engineering High Performance QA System Using Lexico-Semantic Pattern Matching and Shallow NLP.” In *TREC*.

Legaz García, Legaz. 2015. “Integración de información biomédica basada en tecnologías semánticas avanzadas,” September. https://digitum.um.es/xmlui/handle/10201/45979.

Lehmann, Jens, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, et al. 2015. “DBpedia–a Large-Scale, Multilingual Knowledge Base Extracted from Wikipedia.” *Semantic Web* 6 (2): 167–195.

Lehnert, Wendy G. 1977. “A Conceptual Theory of Question Answering.” In *Proceedings of the 5th International Joint Conference on Artificial Intelligence-Volume 1*, 158–164. Morgan Kaufmann Publishers Inc.

Levenshtein, Vladimir I. 1966. “Binary Codes Capable of Correcting Deletions, Insertions, and Reversals.” In *Soviet Physics Doklady*, 10:707–710.

Li, Shusen, Zhiyang He, and Ji Wu. 2014. “An Ontology Semantic Tree Based Natural Language Interface.” In *Chinese Spoken Language Processing (ISCSLP), 2014 9th International Symposium on*, 226–230. IEEE.

Li, Xin, and Dan Roth. 2002. “Learning Question Classifiers.” In *Proceedings of the 19th International Conference on Computational Linguistics-Volume 1*, 1–7. Association for Computational Linguistics.

Li, Xin, and Dan Roth. 2006. “Learning Question Classifiers: The Role of Semantic Information.” *Natural Language Engineering* 12 (03): 229–249.

Liddy, Elizabeth. 2001. “Natural Language Processing.” *Center for Natural Language Processing*, January. http://surface.syr.edu/cnlp/11.

Lin, Dekang. 2003. “Dependency-Based Evaluation of MINIPAR.” In *Treebanks*, 317–329. Springer.

Lin, Jimmy, and Chris Dyer. 2010. “Data-Intensive Text Processing with MapReduce.” *Synthesis Lectures on Human Language Technologies* 3 (1): 1–177.

Linckels, Serge, and Christoph Meinel. 2007. “Semantic Interpretation of Natural Language User Input to Improve Search in Multimedia Knowledge Base (Semantische Interpretation Einer Benutzer-Eingabe in Natürlicher Sprache Für Eine Verbesserte Suche in Einer Multimedialen Wissensdatenbank).” *It-Information Technology* 49 (1): 40–48.

Litkowski, Kenneth C. 2002. “Question Answering Using XML-Tagged Documents.” In *TREC*.

Liu, Bing. 2012. “Sentiment Analysis and Opinion Mining.” *Synthesis Lectures on Human Language Technologies* 5 (1): 1–167.

Lloret, Elena. 2008. “Text Summarization: An Overview.” *Paper Supported by the Spanish Government under the Project TEXT-MESS (TIN2006-15265-C06-01)*.

López, L. Alfonso Ureña. 2001. *Resolución de la ambigüedad léxica en tareas de clasificación automática de documentos*. Editorial Club Universitario.

Lopez, Vanessa, Michele Pasin, and Enrico Motta. 2005. “Aqualog: An Ontology-Portable Question Answering System for the Semantic Web.” In *European Semantic Web Conference*, 546–562. Springer.

Lopez, Vanessa, Christina Unger, Philipp Cimiano, and Enrico Motta. 2013. "Evaluating Question Answering over Linked Data." *Web Semantics: Science, Services and Agents on the World Wide Web* 21: 3–13.

Lopez, Vanessa, Victoria Uren, Marta Sabou, and Enrico Motta. 2011. "Is Question Answering Fit for the Semantic Web?: A Survey." *Semant. Web* 2 (2): 125–155. doi:10.3233/SW-2011-0041.

Lozano-Tello, Adolfo, and Asunción Gómez-Pérez. 2004. "Ontometric: A Method to Choose the Appropriate Ontology." *Journal of Database Management* 2 (15): 1–18.

Luke, Sean, Lee Spector, David Rager, and James Hendler. 1997. "Ontology-Based Web Agents." In *Proceedings of the First International Conference on Autonomous Agents*, 59–66. New York, NY, USA: ACM. doi:10.1145/267658.267668.

Mahesh, Kavi, Sergei Nirenburg, and others. 1995. "A Situated Ontology for Practical NLP." In *Proceedings of the IJCAI-95 Workshop on Basic Ontological Issues in Knowledge Sharing*, 19:21. Citeseer.

Makulowich, John. 1999. "Government Data Mining Systems Defy Definition." *Washington Technology* 22: 393–3.

Malik, S., A. Goel, and S. Maniktala. 2010. "A Comparative Study of Various Variants of SPARQL in Semantic Web." In *2010 International Conference on Computer Information Systems and Industrial Management Applications (CISIM)*, 471–74. doi:10.1109/CISIM.2010.5643493.

Manaris, Bill. 1998. "Natural Language Processing: A Human-Computer Interaction Perspective." *Advances in Computers* 47: 1–66.

Mani, Inderjeet. 2001. *Automatic Summarization*. John Benjamins Publishing.

Mani, Inderjeet, and Mark T Maybury. 1999. *Advances in Automatic Text Summarization*. Vol. 293. MIT Press.

Manning, Christopher D., Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press.

Manning, Christopher D, and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*. Vol. 999. MIT Press.

Marcus, Mitchell P., Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. "Building a Large Annotated Corpus of English: The Penn Treebank." *Comput. Linguist.* 19 (2): 313–330.

Martínez Barco, Patricio, José Luis Vicedo González, Estela Saquete Boró, David Tomás Díaz, and others. 2007. "Sistemas de Pregunta-Respuesta."

Masinter, Larry, Tim Berners-Lee, and Roy T Fielding. 2005. "Uniform Resource Identifier (URI): Generic Syntax."

McCallum, Andrew, Karl Schultz, and Sameer Singh. 2009. "Factorie: Probabilistic Programming via Imperatively Defined Factor Graphs." In *Advances in Neural Information Processing Systems*, 1249–1257.

McDonald, David D. 2010. "Natural Language Generation." *Handbook of Natural Language Processing* 2: 121–144.

McGuinness, Deborah L., and Frank Van Harmelen. 2016. "OWL Web Ontology Language Overview." October 10. https://www.w3.org/TR/owl-features/.

Mendes, Pablo N, Max Jakob, and Christian Bizer. 2012. "DBpedia: A Multilingual Cross-Domain Knowledge Base." In *LREC*, 1813–1817. Citeseer.

Milios, E, Y Zhang, B He, and L Dong. 2003. "Automatic Term Extraction and Document Similarity in Special Text Corpora." In *Proceedings of the Sixth Conference of the Pacific Association for Computational Linguistics*, 275–284. Citeseer.

Miller, George A. 1995. "WordNet: A Lexical Database for English." *Commun. ACM* 38 (11): 39–41. doi:10.1145/219717.219748.

Minock, Michael, Peter Olofsson, and Alexander Näslund. 2008. "Towards Building Robust Natural Language Interfaces to Databases." In *Proceedings of the 13th International Conference on Natural Language and Information Systems: Applications of Natural Language to Information Systems*, 187–198. NLDB '08. Berlin, Heidelberg: Springer-Verlag. doi:10.1007/978-3-540-69858-6_19.

Moens, Marie-Francine. 2006. *Information Extraction: Algorithms and Prospects in a Retrieval Context*. Vol. 21. Springer Science & Business Media.

Moldovan, Dan, Christine Clark, Sanda Harabagiu, and Steve Maiorano. 2003. "Cogex: A Logic Prover for Question Answering." In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, 87–93. Association for Computational Linguistics.

Moldovan, Dan, Sanda Harabagiu, Marius Pasca, Rada Mihalcea, Roxana Girju, Richard Goodrum, and Vasile Rus. 2000. "The Structure and Performance of an Open-Domain Question Answering System." In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, 563–570. Association for Computational Linguistics.

Moldovan, Dan I, Sanda M Harabagiu, Marius Paşca, Rada Mihalcea, Richard A Goodrum, Corina R Gîrju, and Vasile Rus. 1999. "Lasso: A Tool for Surfing the Answer Net."

Moldovan, Dan, Marius Paşca, Sanda Harabagiu, and Mihai Surdeanu. 2003. "Performance Issues and Error Analysis in an Open-Domain Question Answering System." *ACM Transactions on Information Systems (TOIS)* 21 (2): 133–154.

Mollá, Diego. 2006. "Learning of Graph-Based Question Answering Rules." In *Proceedings of the First Workshop on Graph Based Methods for Natural Language Processing*, 37–44. Association for Computational Linguistics.

Moreda, P. 2009. "Los Roles Semánticos En La Tecnologıa Del Lenguaje Humano: Anotación Y Aplicación." *María Teresa Vicente-Díez, Paloma Martínez, Ángel Martínez-González* 42: 125–126.

Morton, Thomas S. 1999. "Using Coreference for Question Answering." In *Proceedings of the Workshop on Coreference and Its Applications*, 85–89. Association for Computational Linguistics.

Motik, Boris, Bernardo Cuenca Grau, Ian Horrocks, Zhe Wu, Achille Fokoue, and Carsten Lutz. 2016. "OWL 2 Web Ontology Language Profiles (Second Edition)." October 12. https://www.w3.org/TR/owl2-profiles/.

Motik, Boris, and R Studer. 2005. “KAON2–A Scalable Reasoning Tool for the Semantic Web.” In *Proceedings of the 2nd European Semantic Web Conference (ESWC’05), Heraklion, Greece*. Vol. 17.

Mustaffa, Supiah, Ros’aleza Zarina Ishak, and Dickson Lukose. 2012. “Ontology Model for Herbal Medicine Knowledge Repository.” In *Knowledge Technology*, edited by Dickson Lukose, Abdul Rahim Ahmad, and Azizah Suliman, 293–302. Springer Berlin Heidelberg. http://link.springer.com/chapter/10.1007/978-3-642-32826-8_30.

Neches, Robert, Richard E. Fikes, Tim Finin, Thomas Gruber, Ramesh Patil, Ted Senator, and William R. Swartout. 1991. “Enabling Technology for Knowledge Sharing.” *AI Magazine* 12 (3): 36. doi:10.21918/aimag.v12i3.902.

Nenkova, Ani, Sameer Maskey, and Yang Liu. 2011. “Automatic Summarization.” In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts of ACL 2011*, 3. Association for Computational Linguistics.

Nirenburg, Sergei. 1989. “Knowledge-Based Machine Translation.” *Machine Translation* 4 (1): 5–24.

Nivre, Joakim, Johan Hall, and Jens Nilsson. 2006. “Maltparser: A Data-Driven Parser-Generator for Dependency Parsing.” In *Proceedings of LREC*, 6:2216–2219.

Noguera-Arnaldos, Jose Ángel, Mario Andrés Paredes-Valverde, Rafael Valencia-García, and Miguel Ángel Rodríguez-García. 2015. “Sistema de diálogo basado en mensajería instantánea para el control de dispositivos en el internet de las cosas.*.” *Procesamiento del Lenguaje Natural* 55 (0): 173–76.

Oard, Douglas W, Jianqiang Wang, Dekang Lin, and Ian Soboroff. 2000. “Trec-8 Experiments at Maryland: Clir, Qa and Routing.” DTIC Document.

O’Grady, William, Michael Dobrovolsky, and Mark Aronoff. 1993. *Contemporary Linguistics: An Introduction*. 2nd edition. New York, NY: St Martins Pr.

Ott, Nikolaus. 1992. “Aspects of the Automatic Generation of SQL Statements in a Natural Language Query Interface.” *Information Systems* 17 (2): 147–159.

Ovchinnikova, Ekaterina. 2012. *Integration of World Knowledge for Natural Language Understanding*. Vol. 3. Atlantis Thinking Machines. Paris: Atlantis Press. http://www.springerlink.com/index/10.2991/978-94-91216-53-4.

OWL Working Group. 2016. "OWL 2 Web Ontology Language Document Overview (Second Edition)." October 11. https://www.w3.org/TR/owl-overview/.

Padró, Lluís, and Evgeny Stanilovsky. 2012. "Freeling 3.0: Towards Wider Multilinguality." In *LREC2012*.

Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan. 2002. "Thumbs up?: Sentiment Classification Using Machine Learning Techniques." In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing-Volume 10*, 79–86. Association for Computational Linguistics.

Parr, Terence. 2013. *The Definitive ANTLR 4 Reference*. Pragmatic Bookshelf.

Pearl, Judea. 2014. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann.

Perrault, C Raymond, and Barbara J Grosz. 1988. "Natural Language Interfaces." In *Exploring Artificial Intelligence, Survey Talks from the National Conferences on Artificial Intelligence*.

Popescu, Ana-Maria, Oren Etzioni, and Henry Kautz. 2003. "Towards a Theory of Natural Language Interfaces to Databases." In *Proceedings of the 8th International Conference on Intelligent User Interfaces*, 149–157. New York, NY, USA: ACM. doi:10.1145/604045.604070.

Pradel, Camille, Ollivier Haemmerlé, and Nathalie Hernandez. 2012. "A Semantic Web Interface Using Patterns: The SWIP System." In *Graph Structures for Knowledge Representation and Reasoning*, 172–187. Springer.

Prud'Hommeaux, Eric, Andy Seaborne, and others. 2008. "SPARQL Query Language for RDF." *W3C Recommendation* 15.

Radev, Dragomir, Weiguo Fan, Hong Qi, Harris Wu, and Amardeep Grewal. 2005. "Probabilistic Question Answering on the Web." *Journal of the American Society for Information Science and Technology* 56 (6): 571–583.

Raimond, Yves, and Samer Abdallah. 2006. “The Timeline Ontology.” *OWL-DL Ontology*.

Raimond, Yves. 2007. “The Event Ontology.” Technical report, 2007..

Raimond, Yves, Samer A Abdallah, Mark B Sandler, and Frederick Giasson. 2007. “The Music Ontology.” In *ISMIR*, 417–422. Citeseer.

Raimond, Yves, and Mark Sandler. 2012. “Evaluation of the Music Ontology Framework.” In *Extended Semantic Web Conference*, 255–269. Springer.

Raimond, Yves, Mark B Sandler, and Q Mary. 2008. “A Web of Musical Information.” In *ISMIR*, 263–268.

Ravichandran, Deepak, and Eduard Hovy. 2002. “Learning Surface Text Patterns for a Question Answering System.” In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, 41–47. Association for Computational Linguistics.

RDF Working Group. 2016. “RDF - Semantic Web Standards.” October 5. https://www.w3.org/RDF/.

Resnik, Philip. 1989. “Access to Multiple Underlying Systems in JANUS.” DTIC Document.

Resnik, Philip, and David Yarowsky. 1999. “Distinguishing Systems and Distinguishing Senses: New Evaluation Methods for Word Sense Disambiguation.” *Nat. Lang. Eng.* 5 (2): 113–133. doi:10.1017/S1351324999002211.

Roberts, Ian, and Robert Gaizauskas. 2004. “Evaluating Passage Retrieval Approaches for Question Answering.” In *European Conference on Information Retrieval*, 72–84. Springer.

Rodríguez García, Rodríguez. 2014. “Extracción semántica de información basada en evolución de ontologías,” November. https://digitum.um.es/xmlui/handle/10201/41246.

Rodríguez-García, Miguel Ángel, Rafael Valencia-García, Francisco García-Sánchez, and J. Javier Samper-Zapater. 2014. “Ontology-Based Annotation and Retrieval of Services in the Cloud.” *Knowledge-Based Systems* 56: 15–25. doi:10.1016/j.knosys.2013.10.006.

Ruiz-Martínez, Juana María, Rafael Valencia-García, Rodrigo Martínez-Béjar, and Achim Hoffmann. 2012. "BioOntoVerb: A Top Level Ontology Based Framework to Populate Biomedical Ontologies from Texts." *Knowledge-Based Systems* 36: 68–80.

Sag, Ivan A., Timothy Baldwin, Francis Bond, Ann Copestake, and Dan Flickinger. 2002. "Multiword Expressions: A Pain in the Neck for NLP." In *Computational Linguistics and Intelligent Text Processing*, edited by Alexander Gelbukh, 1–15. Lecture Notes in Computer Science. Springer Berlin Heidelberg. doi:10.1007/3-540-45715-1_1.

Salas-Zárate, María del Pilar, Rafael Valencia-García, Antonio Ruiz-Martínez, and Ricardo Colomo-Palacios. 2016. "Feature-Based Opinion Mining in Financial News: An Ontology-Driven Approach." *Journal of Information Science*, May, 0165551516645528. doi:10.1177/0165551516645528.

Saleem, Muhammad, Muhammad Intizar Ali, Ruben Verborgh, and Axel-Cyrille Ngonga Ngomo. 2015. "Federated Query Processing over Linked Data." *Tutorial at ISWC*.

Salton, Gerard, and Michael J. McGill. 1986. *Introduction to Modern Information Retrieval*. New York, NY, USA: McGraw-Hill, Inc.

Saquete, Estela, J Luis Vicedo, Patricio Martínez-Barco, Rafael Munoz, and Hector Llorens. 2009. "Enhancing QA Systems with Complex Temporal Question Processing Capabilities." *Journal of Artificial Intelligence Research* 35: 775–811.

Schmachtenberg, Max, Christian Bizer, and Heiko Paulheim. 2014. "State of the LOD Cloud 2014." *University of Mannheim, Data and Web Science Group [En Ligne]* 30.

Schulz, Stefan, Elena Beisswanger, Udo Hahn, Joachim Wermter, Anand Kumar, and Holger Stenzhorn. 2006. "From GENIA to BIOTOPTowards a Top-Level Ontology for Biology." In *Proceedings of the 2006 Conference on Formal Ontology in Information Systems: Proceedings of the Fourth International Conference (FOIS 2006)*, 103–114. Amsterdam, The Netherlands, The Netherlands: IOS Press. http://dl.acm.org/citation.cfm?id=1566079.1566094.

Seifert, Jeffrey W. 2004. “Data Mining: An Overview.” *National Security Issues*, 201–217.

Sertkaya, Baris. 2013. “The ELepHant Reasoner System Description.” In *ORE*, 87–93.

Shannon, Claude Elwood. 2001. “A Mathematical Theory of Communication.” *ACM SIGMOBILE Mobile Computing and Communications Review* 5 (1): 3–55.

Shapiro, S. 1995. “Propositional, First-Order And Higher-Order Logics: Basic Definitions, Rules of Inference, Examples.” *Natural Language Processing and Knowledge Representation: Language for Knowledge and Knowledge for Language. AAAI Press/The MIT Press, Menlo Park, CA*.

Sharef, Nurfadhlina Mohd, Shahrul Azman Noah, and Masrah Azrifah Azmi Murad. 2015. “Linguistic Rule-Based Translation of Natural Language Question into Sparql Query for Effective Semantic Question Answering.” *Journal of Theoretical and Applied Information Technology* 80 (3): 557.

Shen, Dan, and Mirella Lapata. 2007. “Using Semantic Roles to Improve Question Answering.” In *EMNLP-CoNLL*, 12–21.

Shi, Lei, and Rada Mihalcea. 2005. “Putting Pieces Together: Combining FrameNet, VerbNet and WordNet for Robust Semantic Parsing.” In *International Conference on Intelligent Text Processing and Computational Linguistics*, 100–111. Springer.

Sidner, Candace. 1986. “Focusing in the Comprehension of Definite Anaphora.” In *Readings in Natural Language Processing*, 363–394. Morgan Kaufmann Publishers Inc.

Sirin, Evren, Bijan Parsia, Bernardo Cuenca Grau, Aditya Kalyanpur, and Yarden Katz. 2007. “Pellet: A Practical Owl-Dl Reasoner.” *Web Semantics: Science, Services and Agents on the World Wide Web* 5 (2): 51–53.

Smith, Barry. 2004. “Beyond Concepts: Ontology as Reality Representation.” In *Formal Ontology in Information Systems (FOIS)*.

Smith, R.W. 2006. “Natural Language Interfaces.” In *Encyclopedia of Language & Linguistics*, 496–503. Elsevier. http://linkinghub.elsevier.com/retrieve/pii/B0080448542009755.

Song, Dezhao, Frank Schilder, Charese Smiley, Chris Brew, Tom Zielund, Hiroko Bretz, Robert Martin, et al. 2015. “TR Discover: A Natural Language Interface for Querying and Analyzing Interlinked Datasets.” In *International Semantic Web Conference*, 21–37. Springer.

Sosa, Eduardo. 1997. “Procesamiento Del Lenguaje Natural: Revisión Del Estado Actual, Bases Teóricas Y Aplicaciones (Parte I).” *Revista Internacional Científica Y Profesional*.

Soubbotin, Martin M, and Sergei M Soubbotin. 2002. “Use of Patterns for Detection of Likely Answer Strings: A Systematic Approach.” In *TREC*, 52:90.

Sowa, John F. 1995. “Top-Level Ontological Categories.” *International Journal of Human-Computer Studies* 43 (5): 669–85. doi:10.1006/ijhc.1995.1068.

Sowa, John F. 2000. *Knowledge Representation: Logical, Philosophical and Computational Foundations*. Pacific Grove, CA, USA: Brooks/Cole Publishing Co.

Srihari, Rohini, and Wei Li. 2000. “A Question Answering System Supported by Information Extraction.” In *Proceedings of the Sixth Conference on Applied Natural Language Processing*, 166–172. Association for Computational Linguistics.

Stenmark, Maj, and Pierre Nugues. 2013. “Natural Language Programming of Industrial Robots.” In *Robotics (ISR), 2013 44th International Symposium on*, 1–5. IEEE.

Steve, Geri, Aldo Gangemi, and Domenico M. Pisanelli. 1997. “Integrating Medical Terminologies with ONIONS Methodology.” In *Information Modelling and Knowledge Bases VIII (IOS*. Press.

Studer, Rudi, V. Richard Benjamins, and Dieter Fensel. 1998. “Knowledge Engineering: Principles and Methods.” *Data & Knowledge Engineering* 25 (1): 161–97. doi:10.1016/S0169-023X(97)00056-6.

Sussna, Michael. 1993. “Word Sense Disambiguation for Free-Text Indexing Using a Massive Semantic Network.” In *Proceedings of the Second International Conference on Information and Knowledge Management*, 67–74. New York, NY, USA: ACM. doi:10.1145/170088.170106.

Swartz, A. 2002. “MusicBrainz: A Semantic Web Service.” *IEEE Intelligent Systems* 17 (1): 76–77. doi:10.1109/5254.988466.

Tellex, Stefanie, Boris Katz, Jimmy Lin, Aaron Fernandes, and Gregory Marton. 2003. “Quantitative Evaluation of Passage Retrieval Algorithms for Question Answering.” In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, 41–47. ACM.

Tennant, Harry R, Kenneth M Ross, Richard M Saenz, Craig W Thompson, and James R Miller. 1983. “Menu-Based Natural Language Understanding.” In *Proceedings of the 21st Annual Meeting on Association for Computational Linguistics*, 151–158. Association for Computational Linguistics.

Thompson, Bozena H, and Frederick B Thompson. 1983. “Introducing Ask, a Simple Knowledgeable System.” In *Proceedings of the First Conference on Applied Natural Language Processing*, 17–24. Association for Computational Linguistics.

Tjong Kim Sang, Erik F, and Fien De Meulder. 2003. “Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition.” In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003-Volume 4*, 142–147. Association for Computational Linguistics.

Tripathi, Sneha, and Juran Krishna Sarkhel. 2010. “Approaches to Machine Translation.” *ALIS Vol.57(4) [December 2010]*, December. http://nopr.niscair.res.in/handle/123456789/11057.

Tsatsaronis, George, Michael Schroeder, Georgios Paliouras, Yannis Almirantis, Ion Androutsopoulos, Eric Gaussier, Patrick Gallinari, et al. 2012. “BioASQ: A Challenge on Large-Scale Biomedical Semantic Indexing and Question Answering.” In *AAAI Fall Symposium: Information Retrieval and Knowledge Discovery in Biomedical Text*. Citeseer.

Unger, Christina, Lorenz Bühmann, Jens Lehmann, Axel-Cyrille Ngonga Ngomo, Daniel Gerber, and Philipp Cimiano. 2012. “Template-Based Question Answering over RDF Data.” In *Proceedings of the 21st International Conference on World Wide Web*, 639–648. ACM.

Unger, Christina, André Freitas, and Philipp Cimiano. 2014. “An Introduction to Question Answering over Linked Data.” In *Reasoning Web International Summer School*, 100–140. Springer.

Unicode Staff, CORPORATE. 1991. *The Unicode Standard: Worldwide Character Encoding*. Addison-Wesley Longman Publishing Co., Inc.

Vallin, Alessandro, Bernardo Magnini, Danilo Giampiccolo, Lili Aunimo, Christelle Ayache, Petya Osenova, Anselmo Peñas, et al. 2005. “Overview of the CLEF 2005 Multilingual Question Answering Track.” In *Workshop of the Cross-Language Evaluation Forum for European Languages*, 307–331. Springer.

Vicente Moreno, Esther Marta, Cristina Barros, Fernando Peregrino Torregrosa, Francisco Agulló Antolín, and Elena Lloret. 2015. *La Generación de Lenguaje Natural: Análisis Del Estado Actual*.

Vivancos-Vicente, Pedro José, Juan Salvador Castejón-Garrido, Mario Andrés Paredes-Valverde, María del Pilar Salas-Zárate, and Rafael Valencia-García. 2016. “IXHEALTH: A Multilingual Platform for Advanced Speech Recognition in Healthcare.” In *Technologies and Innovation*, edited by Rafael Valencia-García, Katty Lagos-Ortiz, Gema Alcaraz-Mármol, Javier del Cioppo, and Nestor Vera-Lucio, 26–38. Springer International Publishing. http://link.springer.com/chapter/10.1007/978-3-319-48024-4_3.

Voorhees, Ellen M. 2001. “The TREC Question Answering Track.” *Natural Language Engineering* 7 (04): 361–378.

Wang, Chong, Miao Xiong, Qi Zhou, and Yong Yu. 2007. “Panto: A Portable Natural Language Interface to Ontologies.” In *European Semantic Web Conference*, 473–487. Springer.

Wang, Gang, Jianshan Sun, Jian Ma, Kaiquan Xu, and Jibao Gu. 2014. “Sentiment Classification: The Contribution of Ensemble Learning.” *Decision Support Systems* 57: 77–93.

Warren, David HD, and Fernando CN Pereira. 1982. “An Efficient Easily Adaptable System for Interpreting Natural Language Queries.” *Computational Linguistics* 8 (3–4): 110–122.

Winograd, Terry. 1972. “Understanding Natural Language.” *Cognitive Psychology* 3 (1): 1–191.

Woods, William A. 1973. “Progress in Natural Language Understanding: An Application to Lunar Geology.” In *Proceedings of the June 4-8, 1973, National Computer Conference and Exposition*, 441–450. ACM.

Woods, William A. 1979. *Semantics for a Question-Answering System*. Vol. 27. Garland Pub.

Woods, William A, Ronald M Kaplan, and Bonnie Nash-Webber. 1972. *The Lunar Sciences Natural Language Information System: Final Report*. Bolt, Beranek and Newman, Incorporated.

World Wide Web Consortium. 2016a. “Extensible Markup Language (XML).” October 5. https://www.w3.org/XML/.

World Wide Web Consortium. 2016b. “W3C XML Schema.” October 5. https://www.w3.org/XML/Schema.

Yang, Yiming, and Xin Liu. 1999. “A Re-Examination of Text Categorization Methods.” In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 42–49. ACM.

Yusof, Nor Nadiah, Azlinah Mohamed, and Shuzlina Abdul-Rahman. 2015. “Reviewing Classification Approaches in Sentiment Analysis.” In *Soft Computing in Data Science*, edited by Michael W. Berry, Azlinah Hj Mohamed, and Bee Wah Yap, 43–53. Springer Singapore. http://link.springer.com/chapter/10.1007/978-981-287-936-3_5.

Zhang, Dell, and Wee Sun Lee. 2003. “Question Classification Using Support Vector Machines.” In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, 26–32. ACM.

Zheng, Zhiping. 2002. “AnswerBus Question Answering System.” In *Proceedings of the Second International Conference on Human Language Technology Research*, 399–404. Morgan Kaufmann Publishers Inc.

Zhou, GuoDong, and Jian Su. 2002. “Named Entity Recognition Using an HMM-Based Chunk Tagger.” In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, 473–480. Stroudsburg, PA, USA: Association for Computational Linguistics. doi:10.3115/1073083.1073163.

Zloof, Moshé M. 1975. “Query-by-Example: The Invocation and Definition of Tables and Forms.” In *Proceedings of the 1st International Conference on Very Large Data Bases*, 1–24. New York, NY, USA: ACM. doi:10.1145/1282480.1282482.