UNIVERSIDAD DE MURCIA

FACULTAD DE INFORMÁTICA

Extracción Semántica de Información basada en Evolución de Ontologías.

D. Miguel Ángel Rodríguez García

2014

UNIVERSIDAD DE MURCIA

Facultad de Informática

TESIS DOCTORAL

EXTRACCIÓN SEMÁNTICA DE INFORMACIÓN BASADA EN EVOLUCIÓN DE ONTOLOGÍAS.

Miguel Ángel Rodríguez García

SEPTIEMBRE 2014

Directores:

Dr. Rafael Valencia García
Dr. Francisco García Sánchez

Agradecimientos

A mi hermana por mostrarme el camino.
A mi madre y mi madrina por su apoyo infinito.
A mi padre por sus sabios consejos.
A toda mi familia y amigos.
A Rafa por darme la oportunidad.
A mis compañeros de investigación con los
que he trabajado todos estos años.

ÍNDICES

ÍNDICE DE CONTENIDO

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

LISTA DE ACRÓNIMOS

Término	Significado
AC	Aprendizaje Computacional
ANNIE	A Neraly-New Information Extraction System
AIE	Adaptive Information Extraction
API	Application Programming Interface
BNF	Backus-Naur Form
CERN	European Organization for Nuclear Research
COHSE	Conceptual Open Hypermedia Service
CREAM	CREAtion of Metadata
CSS	Cascading Style Sheets
DAML	DARPA Agent Markup Language
DARPA	Defense Advanced Research Project Agency
DL	Description logic
DTD	Document Type Definition
FVP	Facet Value Pair
GATE	General Architecture for Text Engineering
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
IA	Inteligencia Artificial
II	Information Integration
IDF	Inverse Document Frequency
IE	Information Extraction
IEEE	Institute of Electrical and Electronics Engineers
IR	Information Retrieval
KB	Knowledge base
KIMO	KIM Ontology
KMi	Knowledge Media Institute
KNN	K nearest neighbors
MUC	Message Understanding Conference
NER	Named-entity recognition
NLP	Natural language processing
OIL	Ontology Inference Layer
OWL	Web Ontology Language
PDF	Portable Document Format
PLN	Procesamiento del Lenguaje Natural
POS-Taggers	Part-of-speech Taggers
QL	Query Language
RDF	Resource Description Framework
RDFS	Resource Description Framework Schema
RNE	Reconocimiento de Nombres de Entidades

S-CREAM	Semi-automatic CREAtion of Metadata
SemTag	Semantic Tag
SGML	Standard Generalized Markup Language
SHOE	Simple HTML Ontology Extensions
TF	Term Frequency
TOR	Termino-ontologías
TXL	Turing eXtender Language
URI	Uniform Resource Identifier
URN	Uniform Resource Name
W3C	World Wide Web Consortium
WWW	World Wide Web
XML	eXtensible Markup Language

RESUMEN

La ingente cantidad de información que se produce constantemente en Internet y su heterogeneidad dieron lugar a la concepción de una extensión de la Web actual para alcanzar una Web más inteligente, la Web Semántica, con más significado y orientada a que los usuarios encuentren las respuestas a sus necesidades de una forma más rápida y sencilla. Este incremento de significado se basa en la utilización de metadatos semánticos y ontológicos para describir de manera más específica el contenido de los recursos de la Web, independientemente de las fuentes de conocimiento. Así, la utilización de anotaciones semánticas permite explicitar el significado contenido en los recursos de manera que cualquier sistema de computación sea capaz de interpretarlo.

En la actualidad, a pesar de la relevancia de las anotaciones dentro de la Web Semántica, éste es aún un campo sin estandarizar. Varios enfoques se han desarrollado a lo largo de los últimos años pero, debido a las carencias que algunos de estos enfoques presentan, aún no existe ninguna metodología estándar. La motivación que ha servido de guía en esta tesis doctoral ha sido, por tanto, proponer, en el ámbito de la Web Semántica, una nueva metodología de anotación semántica basada en ontologías que cubra todo el ciclo de vida de las anotaciones así como las posibles actualizaciones de los recursos. Esta metodología está constituida por una serie de fases entre la que se pueden destacar las siguientes: representación y anotación semántica, extracción de términos, indexación semántica y evolución de ontologías.

La primera etapa tiene por objetivo la anotación semántica de los recursos y la extracción de términos. La anotación semántica se lleva a cabo utilizando una ontología con un modelo predefinido del dominio. Para la extracción de términos se emplea una metodología basada en patrones morfosintácticos construidos a partir del análisis de los recursos anotados. Esta extracción terminológica dará como resultado una lista de términos que será utilizada en la fase de evolución de ontologías.

La siguiente etapa se centra en la indexación semántica. En esta etapa las anotaciones semánticas definidas en la fase anterior son enriquecidas utilizando la

taxonomía del modelo ontológico formal de representación del conocimiento empleado en la etapa de anotación semántica. Por último, la última fase está constituida por el proceso de evolución de ontologías. Este proceso precisa acceder a un repositorio de información externo y utiliza la lista de términos generada durante la primera fase para, aplicando una metodología basada en un algoritmo de búsqueda en anchura, enriquecer la ontología transformando cada uno de los términos de la lista de términos extraídos en conceptos del dominio ontológico subyacente.

Con objeto de evaluar el rendimiento de la metodología desarrollada, se diseñó una estrategia de evaluación del sistema de anotación semántico basado en las métricas de "precisión", "exhaustividad" y "medida-F", métricas todas ellas muy utilizadas científicamente en la validación de este tipo de sistemas. La validación consistió en la selección de dos dominios de aplicación y en la utilización de estas métricas sobre algunos de los módulos que componían el sistema propuesto. Los índices obtenidos tras los experimentos en los diferentes módulos fueron muy prometedores y llevaron a conclusiones muy favorables acerca del rendimiento de la metodología y su aplicabilidad en diversos contextos.

El desarrollo de esta metodología de anotación dio lugar al desarrollo de varias aplicaciones que aprovechan su potencial. Entre las aplicaciones más destacables es posible resaltar la utilización de la metodología de anotación semántica para el cálculo de la similitud entre entidades. La aplicación de esta metodología en el cálculo de la similitud supuso la definición de un método capaz de realizar comparaciones entre cualquier par de entidades almacenadas en un sistema de información. Este método lleva a cabo la comparación entre entidades al nivel de granularidad más fino, esto es, los atributos que caracterizan cada una de las entidades comparadas. Por lo tanto, los atributos de las entidades constituyen la principal fuente de información para realizar tales comparativas. Así, la única restricción existente en este método de comparación es que los tipos de datos de los atributos a comparar entre sí sean del mismo tipo. Con todo, la comparación se realiza a través de una serie de algoritmos matemáticos que permiten cuantificar la similitud para cada par de atributos. Al igual que ocurría en el caso de la metodología de anotación semántica, la metodología de cálculo de similitud semántica también se encuentra dividida en una serie de fases, a saber,

representación y anotación semántica, indexación semántica, cálculo de similitud y motor de inferencia semántica.

La primera y segunda fases, ya comentadas anteriormente por formar parte de la metodología de anotación semántica, tienen por objetivo la anotación semántica y la representación del conocimiento, y la indexación semántica, respectivamente.

La siguiente fase se centra en el cálculo de similitud. La función principal de esta etapa es la construcción de las matrices de similitud. Estas matrices se comportan como contenedores numéricos que representan, cuantitativamente, la similitud entre grupos de pares de entidades. Las matrices de similitud conforman una fuente de información fundamental para el motor de inferencia semántica, que constituye la última fase de la metodología propuesta. El motor de inferencia semántica es, básicamente, el responsable de proporcionar una interfaz altamente funcional que permite al usuario explotar toda la información generada por los diferentes módulos que componen la plataforma de comparación semántica. Entre otras funciones, el motor de inferencia semántica permitirá la recuperación de entidades similares a una entidad proporcionada. También incluye un buscador semántico que facilita las tareas de búsqueda a partir de descripciones en lenguaje natural.

La evaluación de esta aplicación de la metodología de anotación semántica para el cálculo de la similitud entre entidades se llevó a cabo a través de la aplicación de métricas que proporcionan índices cuantitativos relacionados con la “precisión”, la “exhaustividad” y la “medida-F”. La evaluación se inició seleccionado un dominio particular de aplicación para, posteriormente, aplicar las métricas mencionadas sobre uno de los sistemas que componen la metodología, a saber, el motor de inferencia semántica. Esto permitió recoger los índices necesarios para realizar el estudio de evaluación. Los resultados obtenidos fueron muy favorables, proporcionando conclusiones muy prometedoras acerca de la aplicación de la metodología de anotación semántica en el cálculo de la similitud entre entidades.

INTRODUCCIÓN

La Web Semántica es una extensión de la WWW más inteligente, significativa y eficiente donde el significado de la información que contienen los recursos en la Web puede ser representado formalmente y de forma explícita para conseguir que los contenidos sean comprensibles por los sistemas de computación (Malik et al., 2010). En este sentido, las ontologías proporcionan esta forma de representación formal y explicita constituyendo uno de los pilares fundamentales de la Web Semántica (Davis, 2013).

La Inteligencia Artificial ha encontrado en las ontologías el modelo de representación del conocimiento ideal para describir de manera formal el contenido de los recursos en la Web haciendo explicito el significado subyacente del contenido. La anotación semántica cumple una labor muy relevante dentro de la Web Semántica debido a que facilitan una representación formal de los contenidos de los recursos en la Web más interpretable por las aplicaciones. Esta representación formal permite a las aplicaciones obtener una visión más precisa del significado del contenido y proporcionar respuestas más rápidas y sencillas a los usuarios. Además, las anotaciones semánticas facilitan un acceso más inteligente a los recursos en la Web y permiten explotar nuevos enfoques de inferencia de conocimiento y recuperación de información (Aguado de Cea et al., 2003).

Las razones expuestas en los párrafos anteriores han sido las principales motivaciones para la realización de la investigación que se describe en esta tesis doctoral. La solución que se propone en este trabajo de investigación se sustenta en el desarrollo de una nueva metodología de anotación y recuperación semántica basada en modelos ontológicos formales. Esta innovadora metodología cubre todo el ciclo de vida de las anotaciones semánticas teniendo en cuenta las posibles actualizaciones de información y facilitando la adaptación continua del dominio de aplicación a través de la evolución de la ontología subyacente.

Para lograr este objetivo, se ha seguido la siguiente metodología:

- Análisis del estado del arte en Web Semántica, Ontologías, Evolución de Ontologías, Procesamiento de Lenguaje Natural, Extracción y Recuperación de Información y Anotación Semántica. Este estudio del estado del arte implicó el estudio y análisis de los antecedentes de las tecnologías a incorporar en el proyecto.
- Análisis detallado de las metodologías actuales en la evolución de las ontologías y en la extracción y recuperación de información.
- Definición y formalización de un entorno para la anotación y recuperación semántica de información basado en ontologías. Esta tarea se dividió en distintas partes: (i) definición y formalización de una metodología de anotación semántica; (ii) definición y formalización de una metodología de extracción terminológica; (iii) definición y formalización de una metodología de evolución de ontologías; y (iv) definición y formalización de una metodología de búsqueda y recuperación semántica de información.
- Diseño de una aplicación software para la anotación y recuperación semántica de información de manera automática. Se ha desarrollado una interfaz Web que integra las funciones de anotación y búsqueda semántica basada en ontologías.
- Validación de la metodología de anotación y recuperación semántica desarrollada en dos dominios bien diferenciados, a saber, el de la computación en la nube y el de la gestión de la "investigación, desarrollo e innovación" (I+D+i). La validación fue aplicada sobre algunos de los diversos módulos que componen la metodología. En concreto, se llevó a cabo (i) la validación del motor de búsqueda semántico; (ii) la validación del módulo de extracción de términos y, por último, (iii) la validación del módulo de evolución de ontologías. Para cada uno de estos procesos de validación se aplicaron las métricas de "precisión", "exhaustividad" y "medida-F", ampliamente utilizadas en el análisis de los sistemas de procesamiento de lenguaje natural y los sistemas de recuperación y extracción de información. En cada validación se redefinieron estas métricas para adaptarlas al dominio de evaluación. A partir de los resultados de aplicar estos procesos de validación, se realizó todo el estudio de evaluación para cada uno de los módulos analizados.
- Definición y formalización de un entorno para el cálculo de la similitud semántica entre recursos. La concepción de este entorno se ha dividido en

varias partes: (i) reutilización de la metodología de anotación y recuperación semántica de información diseñada anteriormente; (ii) diseño y formalización de una metodología de comparación de entidades; (iii) diseño y formalización de un motor de inferencia que ofrece un conjunto de servicios relacionados con la búsqueda y recomendación de entidades basada en la similitud semántica.

- Diseño de una aplicación software para el cálculo de la similitud semántica entre recursos. Se ha desarrollado una interfaz Web que facilita la utilización de la aplicación a través de los navegadores Web. Este sistema proporciona un servicio de recomendación basado en la similitud de los recursos insertados en el sistema.
- Validación de la metodología de cálculo de similitud semántica desarrollada en el dominio de la gestión de la I+D+i. La validación fue aplicada sobre el motor de inferencia semántico y consistió, al igual que la anterior validación, en la utilización de las métricas de "precisión", "exhaustividad" y "medida-F". Posteriormente, a partir de los valores obtenidos para estas métricas se realizó el estudio de evaluación sobre el motor de inferencia semántico.

Los objetivos que se establecieron en el desarrollo de la metodología descrita se han realizado con éxito, y los resultados que se han conseguido se presentan en esta memoria con la siguiente organización:

En el Capítulo I se realiza un detallado estudio del estado del arte de las tecnologías relacionadas con esta investigación que se encuentran integradas en las metodologías desarrolladas. El estudio comienza con el análisis de la Web Semántica, tecnología que se utiliza para contextualizar los demás paradigmas tecnológicos utilizados. A continuación, se proporciona una extensa definición de Ontología, que cubre desde su primera utilización hasta la actualidad, pasando por descripciones más intrínsecas acerca de los lenguajes que se utilizan para describirlas o el conjunto de propiedades que pueden ser definidas o incluso las diferentes clasificaciones de ontologías que existen en función sus características. El siguiente apartado está relacionado con el campo de la evolución de ontologías y proporciona una clasificación de las diferentes metodologías más extendidas que se utilizan en la actualidad para enriquecer el contenido de las ontologías. Seguidamente, se describe la disciplina del Procesamiento de Lenguaje Natural,

analizando en profundidad cómo opera sobre cada uno de los diferentes niveles del lenguaje. En el siguiente apartado, se presenta la clasificación y descripción de algunas de las técnicas fundamentales utilizadas en los procesos de extracción y recuperación de información relacionadas con el procesamiento de lenguaje natural. Por último, se utiliza el análisis de la anotación y recuperación semántica de información para introducir un estudio sobre diferentes herramientas. Este análisis concluye con una comparativa entre todas estas herramientas y la que se propone en esta tesis doctoral a través una serie de parámetros preestablecidos.

En el Capítulo II se describe la metodología de anotación y recuperación semántica de información propuesta en este trabajo de investigación. Esta descripción se divide en varias secciones. En la primera sección se describe la metodología diseñada para representar semánticamente la información y cómo se transforma esta información en metadatos para que el sistema de anotación semántica los utilice durante el proceso de anotación. En la siguiente sección se detalla la metodología de indexación semántica utilizada para enriquecer las anotaciones definidas por la metodología anterior. A continuación, se explica la metodología de extracción de términos, que tiene la función de extraer términos de la información anotada para construir una lista que se proporcionará a la metodología de evolución de ontologías, descrita en el siguiente apartado. Por último, se muestra la metodología de recuperación de información integrada como servicio en un motor búsqueda semántica.

El siguiente capítulo, Capítulo III, se centra en la validación de esta metodología de anotación y recuperación semántica de información. En este capítulo se presenta un conjunto de evaluaciones que han sido aplicadas para la validación en dos dominios diferentes, a saber, el de la computación en la nube y el de la I+D+i. En este capítulo se describen características de los escenarios de evaluación que se han utilizado, las metodologías de evaluación desarrolladas, y los resultados obtenidos.

La definición y el desarrollo del sistema de anotación semántica no se centraron únicamente en un intento de mejorar las metodologías existentes en este mismo dominio sino que se plantearon también para concebir una herramienta que fuera utilizada en otros entornos. Esta perspectiva motivó la aparición de numerosas aplicaciones entre las que se destaca la descrita en el Capítulo IV, referida a la

aplicación de cálculo de similitud semántica entre entidades. En este Capítulo IV se describe la metodología de cálculo de similitud semántica a partir de sus módulos constituyentes: (i) un repositorio de ontologías, que representa la base de conocimiento donde se almacenan todos los modelos formales, así como las anotaciones semánticas creadas; (ii) el sistema de representación y anotación semántica descrito en el Capítulo II; (iii) el módulo de indexación semántica (también descrito en el Capítulo II); (iv) el módulo de cálculo de similitud que permite cuantificar la similitud semántica entre dos tipos de entidades cualesquiera; y, por último, (v) el motor de inferencia semántica que presenta una interfaz Web con la que se permite a los usuarios interactuar con el sistema para mostrar los resultados de búsquedas semánticas realizadas.

Al igual que en el caso de la metodología de anotación y recuperación semántica de información descrita en el Capítulo II, la aplicación de la metodología para el cálculo de la similitud semántica también se evaluó utilizando como dominio la gestión de la I+D+i en proyectos relacionados con las Tecnologías de la Información y las Comunicaciones (TIC). El dominio seleccionado para evaluar esta aplicación facilitó la reutilización de muchos de los recursos que fueron desarrollados para la evaluación de la metodología de anotación semántica. Además, esta evaluación requirió de la colaboración de ciertos departamentos de una organización de desarrollo de software externa como, por ejemplo, los departamentos de recursos humanos, así como la colaboración de expertos que validaran los resultados obtenidos por la aplicación desarrollada. Todo este proceso de validación se encuentra descrito minuciosamente en el Capítulo V.

Por último, el Capítulo VI muestra las conclusiones obtenidas durante todo este trabajo de investigación, así como un análisis de las posibles líneas futuras de investigación que podrían continuarse partiendo de esta tesis doctoral.

Capítulo I. ESTADO DEL ARTE

I.1. INTRODUCCIÓN

En este capítulo se describe el estado actual de las diferentes tecnologías que componen el núcleo central de esta tesis doctoral.

En el primer apartado, se define el marco tecnológico a partir del que nacen todas las tecnologías que han sido utilizadas en esta tesis. Este marco está compuesto por las tecnologías de la Web Semántica y, más concretamente, las ontologías. Este apartado comienza con la definición del concepto de Web Semántica y una contextualización que permite conocer los orígenes de la Web actual. Posteriormente, se analizan las principales características que aporta esta nueva versión de la Web con respecto a las anteriores. El análisis de la Web Semántica concluye con la descripción en capas de su arquitectura. En el caso de las ontologías, se utiliza una metodología de análisis similar comenzando con una extensa definición del término "ontología". Esta definición cubre la mayoría de acepciones del término ontología, desde la primera vez que se utilizó este término hasta las definiciones más actuales proporcionadas por investigadores de hoy día. Después de este análisis, se describen las diferentes clasificaciones propuestas por varios autores y los distintos tipos de ontologías existentes a partir de estas clasificaciones. A continuación, se examina la composición de una ontología, es decir, qué elementos la componen y qué función desempeñan. Por último, se realiza un estudio de los diferentes y más actuales lenguajes de representación de ontologías.

La segunda sección contiene un detallado análisis del estado del arte del campo de la evolución de ontologías. Este estudio comienza con la definición del concepto de evolución de ontologías. Posteriormente, se clasifican los distintos paradigmas existentes que actualmente se utilizan para evolucionar una ontología y se enumeran los distintos problemas que conlleva el proceso de evolución. Para concluir este apartado, se analiza el aprendizaje de ontologías como metodología de evolución de ontologías empleada en esta tesis. El estudio de esta metodología

se inicia con una definición que explica en qué consiste esta metodología. Después, se descompone la metodología en un modelo de capas, que refleja las distintas tareas que constituyen el proceso de construcción de ontologías desde texto en lenguaje natural. Por último, se describen las tareas de refinamiento e instanciación automática de ontologías, que se corresponden con las últimas fases del aprendizaje de ontologías y que están estrechamente relacionadas con el proceso de evolución.

En el tercer apartado, se presenta la disciplina del procesamiento del lenguaje natural, una de las tecnologías clave en la elaboración de esta tesis doctoral. En primer lugar, se proporciona la definición de este campo de la inteligencia artificial para, después, realizar un breve análisis de los antecedentes, donde se describe el progreso de esta área de investigación desde sus inicios hasta el presente. Por último, se enumeran los diferentes niveles del lenguaje que la investigación en procesamiento del lenguaje natural abarca. Para ello, se proporciona para cada nivel, una definición del nivel lingüístico al que pertenece y, además, se identifican qué técnicas de procesamiento han sido desarrolladas para procesar la información en ese nivel.

En la cuarta sección, se muestra un minucioso estudio sobre los procesos de extracción y recuperación de información, que constituye la base de muchas de las tecnologías descritas en esta tesis (p.ej., evolución de ontologías o anotación semántica). Este apartado comienza comentando algunas de las definiciones más relevantes encontradas en la literatura sobre la extracción y recuperación de información. Después, se separan ambos estudios en dos apartados para analizar de manera individual la extracción y la recuperación de información. El análisis de la extracción de información comienza describiendo los diferentes tipos de información y enumerando las diferentes técnicas, modelos y estrategias de extracción de información que se pueden emplear para cada uno de los tipos de información existentes. Finalmente, se proporciona una descripción detallada de las tareas más relevantes en las que se divide el proceso de extracción de información, las técnicas de extracción de información que utilizan y sus metodologías. Seguidamente, el apartado se centra en analizar el proceso de recuperación de información. Este análisis comienza mostrando una clasificación de técnicas de recuperación de información que se fundamenta en la exactitud del

tipo de modelo de solicitud de información. A partir de esta distinción, se establecen dos grandes clasificaciones de técnicas de recuperación, los basados en modelos de solicitud de coincidencia exacta y los basados en coincidencia parcial. Ambos grupos constituyen el primer nivel de la clasificación y, a partir de ellos, se organizan todas las demás técnicas de recuperación de información. Para concluir este apartado, se describen dos de las técnicas más relevantes de recuperación de información en cada modelo de solicitud clasificado.

En el quinto apartado, se aborda el campo de la anotación semántica. En primer lugar, se proporciona una definición generalista del concepto "anotar" para, seguidamente, definir este término desde el punto de vista semántico. A continuación, se describen los distintos tipos de anotación semántica existentes y se analizan las ventajas y desventajas de utilizar una estrategia de anotación frente a las demás. Por último, se realiza un breve estudio del estado del arte sobre diferentes herramientas de anotación semántica que se han propuesto hasta el momento y se comparan éstas con la herramienta presentada en esta tesis doctoral.

En los dos últimos apartados de este capítulo, se expone brevemente cuál es el problema que se ha tratado de resolver en esta investigación y se presenta un completo resumen del capítulo, en el que se analizan de manera concisa y clara las tecnologías que se han descrito a lo largo del mismo.

I.2. WEB SEMÁNTICA Y ONTOLOGÍAS

I.2.1. WEB SEMÁNTICA

I.2.1.1. Definición

El término "Web Semántica" fue presentado por Tim Berners-Lee, considerado el padre de la Web actual, James Hendler y Ora Lassila en su conocido artículo "The Semantic Web" (Berners-Lee et al., 2001). En este artículo los autores remarcan la carencia de significado de la Web y proponen el cambio de representación de contenido para que incluya "una semántica bien definida", que no solo sea entendible para los seres humanos sino también para las máquinas, de forma que

sean capaces de realizar determinadas tareas de forma autónoma. El siguiente extracto del artículo define brevemente el principal objetivo que se persigue con la inclusión de semántica en la Web:

"*La Web Semántica aportará estructura al contenido significativo de las páginas Web, creando un entorno donde los agentes software itinerantes de página en página, podrán llevar a cabo tareas sofisticadas para los usuarios*"

Según Tim Berners-Lee y sus colegas (2001), "*la Web Semántica es una red de datos que puede ser procesada directa o indirectamente por máquinas. Es una Web extendida que permitirá a humanos y máquinas trabajar en cooperación mutua*". En esta definición, Tim Berners-Lee establece diversas características importantes que tendrá la Web Semántica y asegura su interoperabilidad con las demás versiones de la Web. De hecho, no define la Web Semántica como una versión independiente, sino una versión extendida que dota de nuevas propiedades a la Web existente. Además, esta nueva versión está centrada en disponer de mayor significado que no sólo sea entendible por los humanos, sino también por las máquinas. Es decir, el objetivo de la Web Semántica es la incorporación de información estructurada, que facilite las tareas de procesamiento de la misma por parte de programas de ordenador, de tal forma que sean capaces de entender esta información para mejorar la experiencia del humano.

Por otro lado, el W3C (*World Wide Web Consortium)* define la Web Semántica como "*una Web extendida, dotada de mayor significado, en la que cualquier usuario en Internet podrá encontrar respuestas a sus preguntas de forma más rápida y sencilla gracias a una información mejor definida*". Esta definición vuelve a establecer la Web Semántica como una extensión sobre la Web actual, que tiene como objetivo dotar de inteligencia a la Web actual a partir de información mejor definida. Para ello, es necesario contar con mecanismos que permitan la adición de descripciones explicitas, que ayuden a las máquinas a entender la información contenida en los sitios y proporcionar a los usuarios respuestas más rápidas y de mejor calidad para los usuarios.

La Web Semántica surge, por lo tanto, como respuesta a las limitaciones que presenta la Web actual en lo referente al procesado automático de la información, la interoperabilidad con los sistemas de información, e incluso una indexación que permita manejar, de manera más eficiente, la ingente cantidad de información para

buscar y encontrar de manera rápida, fácil y precisa la información que se desea. Actualmente, la Web es un inmenso grafo formado por nodos del mismo tipo que representarían los contenidos interconectados por hiperenlaces. La mayoría de contenidos en la Web se encuentran construidos mediante lenguajes de etiquetado orientados a la presentación de datos. Estos lenguajes ofrecen escasa información acerca de los documentos y de su contenido, y poseen un número finito de etiquetas para describir los contenidos por lo que su nivel de expresividad es bastante limitado. Además, a esta expresividad restringida hay que añadirle que la mayor parte de la información existente en la Web ha sido creada para los humanos dificultando a las máquinas su interpretación. Sin embargo, la Web Semántica ha sido diseñada para mejorar su expresividad mediante la incorporación de lenguajes estructurados como XML (del inglés, "*eXtensible Markup Language*") y RDF (del inglés, "*Resource Description Framework*") que dotan a cada contenido de una estructura lógica y un significado, permitiendo mejorar el procesamiento de la información y la interoperabilidad entre los sistemas de información.

I.2.1.2. Antecedentes

La '*World Wide Web*' (WWW) tuvo su origen en el Consejo Europeo para la Energía Nuclear (CERN)[1] donde Tim Berners-Lee, considerando los problemas de intercambio de información entre los investigadores, escribió una propuesta donde definía una gran base de datos de hipertexto con enlaces tipados. El objetivo de esta propuesta era facilitar la forma de compartir y actualizar información. Inicialmente esta propuesta no tuvo mucho éxito hasta que, más tarde y con la ayuda de Robert Cailliau, se revisara y fuera aceptada por los supervisores del CERN. Esta nueva versión proponía la unión entre dos tecnologías de esta época, Internet y el hipertexto (esto es, HTTP (del inglés, "*Hypertext Transfer Protocol*") y HTML (del inglés, "*HyperText Markup Language*")). En la literatura esta propuesta ha sido considerada como el germen de la WWW y describía un sistema global para la transferencia de hipertextos basado en identificadores únicos, que

[1] http://home.web.cern.ch

pretendía resolver el problema de la pérdida de información y permitir su intercambio entre los científicos de dicho centro de investigación. Estos identificadores se conocen con el nombre de hiperenlaces. Los hiperenlaces son cadenas de caracteres que identifican de manera unívoca a un recurso dentro de una red mediante el uso de una dirección URI (del inglés, "*Uniform Resource Identifier*"). Estas direcciones representaban vías de acceso a otros recursos como otros documentos, páginas, o incluso a otros servicios. El funcionamiento del sistema requirió de la implementación del primer servidor Web y del primer navegador que mostrara la información, además de las primeras páginas Web (que aún se encuentran disponibles en la página del centro de investigación[2][3]). En primera instancia, el sistema sólo se utilizó en el centro de investigación y, poco a poco, fue extendiéndose a otros centros, pasando de una decena de servidores Web a varias decenas y varios miles de usuarios. Fue entonces, cuando en 1994 los directivos del CERN decidieron liberar la WWW a dominio público, asegurando siempre su mantenimiento como un estándar abierto. Coincidiendo con este hecho surgió el W3C, que es actualmente el ente responsable de desarrollar los estándares que aseguran el crecimiento de la Web.

I.2.1.3. Fundamentos de la Web Semántica

La Web Semántica propone la modificación de la forma en que se presentan los contenidos de la Web de manera que no sólo contenga información para formatear el contenido, sino que también incluya información que describa el contenido para facilitar su procesamiento por parte de las máquinas.

La Figura I.1 muestra la evolución de la forma de representar los contenidos en la Web tradicional con respecto a la Web Semántica. Ambas imágenes reflejan la misma estructuración de contenidos pero unidos con diferentes tipos de enlaces. Mientras que la Web tradicional (parte izquierda de la figura) se asemeja a un grafo constituido por nodos del mismo tipo y enlaces unidireccionales igualmente indiferenciados, la Web Semántica (parte derecha de la figura) cambia estos

[2] http://line-mode.cern.ch/www/hypertext/WWW/TheProject.html

[3] http://info.cern.ch/hypertext/WWW/TheProject.html

enlaces unidireccionales y simples por nodos que tienen un tipo y enlaces que representan relaciones explícitamente diferenciables que pueden ser interpretables tanto por humanos como por agentes software. Es decir, en la Web actual los recursos son almacenados como conjuntos de palabras que no tienen ningún tipo de relación. Sin embargo, en la Web Semántica los recursos no sólo son almacenados como conjuntos de palabras sino que, además, incluyen un significado y una estructura.

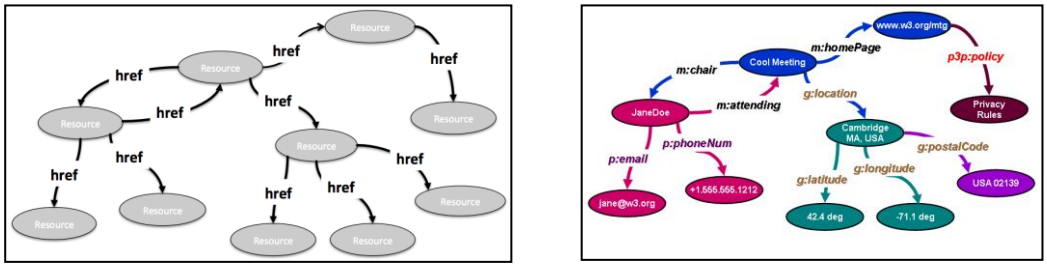


Figura I.1 Evolución de la Web a la Web Semántica (W3C Oficina España, 2007)

Esta nueva versión requiere la adición de elementos que den soporte a la nueva forma de estructurar los contenidos y aportar significado. Ante este problema de representación del conocimiento, las ontologías proporcionan un vocabulario que permite describir las relaciones entre diferentes términos de manera flexible y sin ambigüedades, facilitando su interpretación por las máquinas y los humanos (Horrocks et al., 2003).

I.2.1.4. Arquitectura de la Web Semántica

La arquitectura estándar de la Web Semántica sigue el modelo de capas definido por Tim Berners-Lee en 2000[4] (véase Figura I.2). El objetivo de esta arquitectura es proporcionar a los usuarios y agentes software capacidades de procesamiento, razonamiento y deducción sobre los contenidos de la Web.

[4] http://www.w3.org/2000/Talks/1206-xml2k-tbl/slide10-0.html

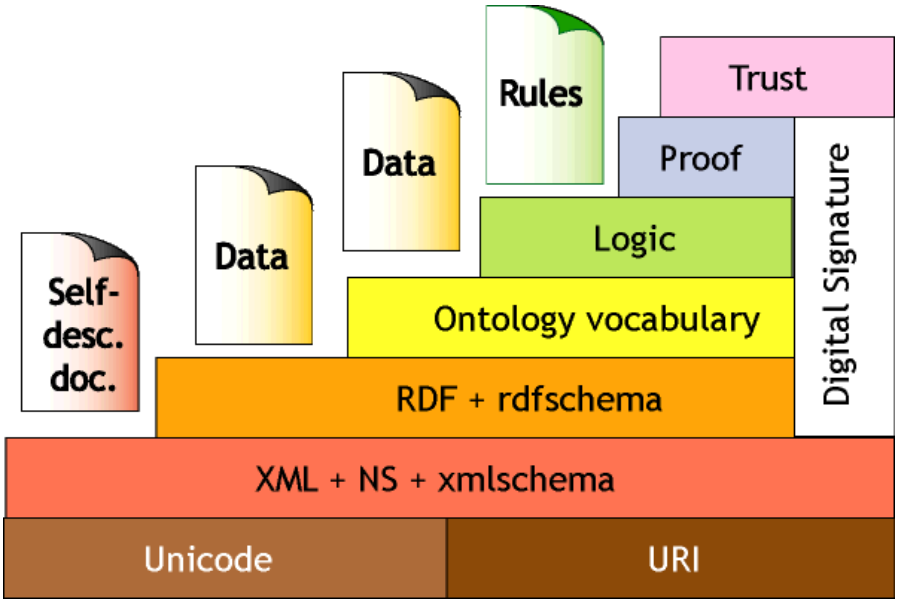


Figura I.2 Arquitectura en capas de la Web Semántica

Seguidamente se analizará individualmente la función que desempeña cada capa dentro de la arquitectura de la Web Semántica:

- **Unicode**. La capa Unicode permite que en la Web Semántica toda la información pueda expresarse en cualquier idioma. Esta capa se encarga de proporcionar una codificación de caracteres estándar que permita utilizar distintos símbolos internacionales y alfabetos.
- **URI/IRI**. La capa URI proporciona un identificador uniforme único que está compuesto por una URL (del inglés, "*Uniform Resource Locator*"), que describe la localización del recurso, y una URN (del inglés, "*Uniform Resource Name*"), que describe el espacio de nombres del recurso.
- **XML + XML Query + Namespaces + XML Schema**. La capa XML , '*XML Query*', junto con el espacio de nombres (del inglés, '*NameSpace*') y el esquema XML (del inglés, '*XML schema*'), sirven de base ofreciendo un formato común de documento que hace posible que los agentes puedan entenderse entre ellos independientemente de las fuentes de información que se utilicen para crear los documentos. Cada una de las tecnologías que son agrupadas en esta capa tiene una función bien definida. En primer lugar, el lenguaje de marcas XML proporciona un formato común para el intercambio de documentos. '*XML Query*', también conocido como XQuery, define un lenguaje de consultas estándar basado en la notación XML para definir consultas y manejar los

resultados. El espacio de nombres sirve para cualificar elementos y atributos de nombres, y asociarlos con los espacios de nombres identificados a través de referencias URI. Finalmente, los esquemas XML proporcionan plantillas que facilitan la elaboración de documentos estándar.

- **Modelo RDF y Sintaxis**. La capa RDF y '*RDF Schema*' se basan en la capa anterior para definir el lenguaje universal con el que se expresa el conocimiento en la Web Semántica. El lenguaje universal se basa en un formato 3-upla o tripleta de RDF. Por su parte, '*RDF Schema*' proporciona un vocabulario para describir las propiedades y las clases de los recursos RDF, con una semántica claramente definida con la que se pueden establecer jerarquías de generalización entre propiedades y clases.
- **Ontología**. La capa de ontologías mejora la expresividad de la capa anterior, incluyendo más vocabulario que facilita la descripción precisa de conceptos, relaciones y propiedades con los que conceptualizar un dominio concreto.
- **Lógica**. La capa lógica está compuesta por un conjunto de reglas de inferencia que permiten a los agentes software procesar y relacionar información. Estas reglas permiten procesar de manera automática la información a nivel semántico.
- **Prueba**. La capa "Prueba" tiene como objetivo el intercambio de hechos y reglas estándar para facilitar la interoperabilidad entre recursos de la Web Semántica.
- **Web Semántica Confiable**. Esta capa tiene la función de evaluar las pruebas ofrecidas por la capa anterior para comprobar de forma exhaustiva si las fuentes de información son confiables.
- **Firma y cifrado (Firma digital)**. El objetivo de esta capa es definir el ámbito de confianza para las capas "Prueba" y "Web Semántica Confiable" de manera que permita a los ordenadores y agentes software verificar la seguridad de la información adjuntada y que ésta se envió por una fuente confiable.

La tecnología más importante en el contexto de la Web Semántica son las ontologías. De hecho, constituyen el medio principal para lograr el objetivo de la Web Semántica al facilitar la definición formal de las entidades y conceptos presentes en los diferentes dominios, la jerarquía que les sustenta y las diferentes relaciones que los unen entre sí. De esta manera se garantiza una representación

del conocimiento consensuada y reutilizable que puede ser compartida y utilizada automáticamente por cualquier sistema informático. En la siguiente sección se describe en profundidad el concepto de ontología.

I.2.2. ONTOLOGÍAS

I.2.2.1. Definición

El término ontología (del griego οντος, 'del ente', y λόγος, 'ciencia, estudio, teoría') hace referencia a una rama de la metafísica que estudia la naturaleza de la existencia. Desde la aparición de este término, según Lawson (2004), en 1606 en el libro "*Ogdoas scholastica*" (Lorhard, 1606) del filósofo y pedagogo alemán Jacob Lorhard (1561-1609), el término ontología ha tenido diferentes definiciones de entre las que destacamos las presentadas a continuación.

La primera definición del término ontología fue dada por el profesor en filosofía, metafísica y ética Rudolph Gockel (1547-1628), quien en su obra "Lexicon philosophicum, quo tantam clave philosophiae fores aperiuntur" de 1613 afirma "*la ontología como una filosofía del ente*" (Gockel, 1613).

Gottfried Wilhelm Leibniz (1646-1716) en su obra "Introductio ad Encyclopaediam Arcanam" publicada en 1683-1685 define ontología como la "*ciencia de lo que es y de la nada, del ente y del no ente, de las cosas y de sus modos, de la sustancia y del accidente*" (Leibniz, 1683).

Jean LeClerc (1657-1736), en su obra publicada en 1692 "Ontologia sive de ente in genere", define el término ontología como "*la rama de la filosofía que estudia el ser, o sea, la realidad*" (Clerc, 1692).

Christian Wolff (1679-1754), discípulo de Gottfried Leibniz, populariza el término definiendo ontología como "*ciencia del ente en general, en cuanto que ente*" en su obra publicada 1736 titulada "Philosophia prima sive Ontologia" (Wolff, 1736).

Hasta ahora, todas las definiciones propuestas proporcionan una noción de ontología centrada en la rama de la filosófica metafísica. La vinculación de este término con la Inteligencia Artificial se produjo cuando el filósofo Willard Van Orman Quine (1908-2000), a través de su interpretación en 1961 "*todo lo que*

puede ser cuantificado existe", relacionó el término de ontología a través de la analogía con la Inteligencia Artificial "*todo lo que existe es exactamente aquello que puede ser representado computacionalmente*" (Quine, 1961).

Pasarían varios años hasta que Robert Neches definiera por primera vez en su publicación "Enabling technology for knowledge sharing" la palabra ontología en el dominio de la Inteligencia Artificial, proporcionando la siguiente definición (Neches et al., 1991):

"*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*"

Aunque ésta fue la primera definición de ontología dentro del dominio de la Inteligencia Artificial, la más extendida y popular ha sido la proporcionada por Tom Gruber (1993):

"*Una ontología es una especificación explicita de una conceptualización*"

El autor define ontología como una conceptualización que está compuesta por objetos, conceptos y otras entidades que existen dentro de una determinada área, y las relaciones que pueden ser definidas entre ellos. Una conceptualización se refiere a un modelo abstracto, que puede ser definido como una interpretación estructurada de una parte del mundo del que se identifican los conceptos más relevantes. Por explícita se entiende a la necesidad de especificar de manera consciente los distintos conceptos que constituyen la ontología.

Aunque ha sido la más extendida, la definición de ontología propuesta por Tom Gruber ha recibido varias críticas por ser considerada demasiado genérica. Entre las voces discordantes más destacadas se encuentra la de Nicola Guarino, quién afirmó lo siguiente tras examinar diversas posibles interpretaciones de ontología (Guarino, 1995):

"*Un punto de inicio en este esfuerzo clarificador será el cuidadoso análisis de la interpretación dada por Tom Gruber. El problema principal de dicha interpretación es que se basa en la noción de conceptualización. Una conceptualización es un conjunto de relaciones extensionales que describen un estado particular, mientras que la noción que tenemos en mente es intencional, esto es, algo como una rejilla conceptual al que le imponemos varios posibles estados*"

Nicola Guarino propuso entonces una definición alternativa del concepto ontología (Guarino, 1995):

"*En el sentido filosófico, podemos referirnos a una ontología como un sistema particular de categorías que representa una cierta visión del mundo. Como tal, este sistema no depende de un lenguaje particular: la ontología de Aristóteles es siempre la misma, independientemente del lenguaje usado para describirla. Por otro lado, en su uso más típico en IA, una ontología es un artefacto ingenieril constituido por un vocabulario específico para describir una cierta realidad, más un conjunto de supuestos explícitos concernientes al significado pretendido de las palabras del vocabulario. Este conjunto de supuestos tiene generalmente la forma de teorías lógicas de primer orden, donde las palabras del vocabulario aparecen como predicados unarios o binarios, respectivamente llamados conceptos y relaciones. En el caso más simple, una ontología describe una jerarquía de conceptos relacionados por relaciones de subsunción: en los casos más sofisticados, se añaden axiomas para expresar otras relaciones entre conceptos y restringir la posible interpretación*"

Willem Nico Borst refinó la definición propuesta por Tom Gruber incluyendo el término "compartida" (Borst, 1997):

"*Una ontología es una especificación formal de una conceptualización compartida*"

En este contexto, formal se refiere a la necesidad de disponer de ontologías comprensibles por las máquinas. Además, enfatiza la necesidad de consenso en la conceptualización, refiriéndose al tipo de conocimiento contenido en las ontologías, esto es, conocimiento consensuado y no privado de ahí el término "compartida".

Posteriormente, Rudi Studer y sus colegas, basándose en las definiciones de Tom Gruber y Willem Nico Borst, formularon la siguiente definición, que ha sido la adoptada para esta investigación (Studer et al., 1998):

"*Una ontología es una especificación formal y explícita de una conceptualización compartida*"

Esta definición fue extendida en el mismo trabajo, explicando lo siguiente:

"*Conceptualización se refiere a un modelo abstracto de algún fenómeno en el mundo a través de la identificación de los conceptos relevantes de dicho fenómeno. Explicita significa que el tipo de conceptos y restricciones usados se definen explícitamente. Formal representa el hecho de que la ontología debería ser*

entendible por una computadora. Compartida refleja la noción de que una ontología captura conocimiento consensual, esto es, que no es de un individuo, si no que es aceptado por un grupo"

Actualmente, podemos destacar que sigue sin existir una definición consensuada del término ontología. Autores como Tom Gruber, William Nico Borst o Rudi Studer, que han proporcionado las definiciones más reconocidas y aceptadas, siguen replanteándose las mismas. Este es el caso de Tom Gruber quien respondió en octubre de 2004, a través de un boletín, si modificaría la definición que proporcionó de ontología y que ha sido tantas veces referenciada manifestando lo siguiente (Gruber, 2004):

"*Bien, los componentes más importantes de esa definición de ontología son que la ontología es un artefacto de representación (una especificación), distinta del mundo que modela, y que es un artefacto diseñado, construido para un propósito. Creo que la mayoría de científicos en computación obtienen la diferencia entre una especificación del mundo, incluso para mundos sintéticos. Retrospectivamente, no cambiaría la definición pero intentaría enfatizar que nosotros diseñamos ontologías. La consecuencia de esta vista es que podemos aplicar una disciplina de ingeniería en su diseño y evaluación. Si las ontologías son cosas derivadas de una ingeniería, entonces no tenemos que preocuparnos tanto sobre si son correctas y favorecer el negocio de construirlas para hacer algo útil. Podemos diseñarlas para conocer objetivos funcionales y restricciones. Podemos construir herramientas que nos ayuden a gestionarlas y validarlas. Y podemos tener múltiples ontologías que se coordinen o compitan basadas en un criterio objetivo más que en una marca de fábrica o una autoridad*"

I.2.2.2. Tipos de Ontologías

En la literatura se pueden encontrar varias clasificaciones de ontologías basadas en diferentes criterios. En este apartado se analizan las clasificaciones más extendidas.

I.2.2.2.1. Clasificación por el conocimiento que contienen

A partir del contenido de la ontología, Riichiro Mizoguchi y sus colegas establecen las siguientes categorías ontológicas (Mizoguchi et al., 1995):

- Ontologías del dominio: expresan conceptualizaciones especificas a un dominio en particular, es decir, describen los conceptos y sus relaciones respecto a un dominio especifico.
- Ontologías de tarea: describen el vocabulario específico de los conceptos de manera que se pueda utilizar el conocimiento del dominio para realizar tareas específicas.
- Ontologías generales: proporcionan descripciones generales acerca de objetos, eventos, relaciones temporales, relaciones causales, modelos de comportamiento y funcionalidades.

Gertjan van Heijst y sus colegas proponen una clasificación alternativa a la presentada por Mizoguchi y sus colegas basada en el volumen, tipo de estructura y en la conceptualización específica del conocimiento (van Heijst et al., 1997):

- Ontologías terminológicas o lingüísticas: especifican los términos que son usados para representar conocimiento en un dominio determinado. Se suelen utilizar para unificar el vocabulario de un dominio concreto.
- Ontologías de información: especifican la estructura de los registros de almacenamiento de una base de datos. Estas ontologías proporcionan un marco estándar para el almacenamiento de la información.
- Ontologías para modelar conocimiento: especifican conceptualizaciones de conocimiento. Estas ontologías tienen una estructura interna mucho más rica que las anteriores, que las hace interesantes para los desarrolladores de sistemas basados en conocimiento. Estas ontologías se ajustan al uso particular del conocimiento que describen.

I.2.2.2.2. Clasificación por motivación

Mediante esta clasificación se categorizan las ontologías a partir del motivo por el que éstas se elaboran. Steve y sus colegas, de acuerdo a su motivación, establecieron la siguiente clasificación (Steve et al., 1997):

- Ontologías para la representación del conocimiento, también conocidas como representacionales: permiten especificar las conceptualizaciones que subyacen en los formalismos de representación de conocimiento (Davis et al., 1993), por lo que también se denominan meta-ontologías (del inglés, '*meta-level ontologies*' o '*top-level ontologies*').
- Ontologías genéricas: definen conceptos considerados genéricos y fundacionales del conocimiento como las estructuras parte/todo, la cuantificación, los procesos o los tipos de objetos. También se conocen por el nombre de ontologías abstractas o superteorías, debido a que permiten definir conceptos abstractos y pueden ser reutilizables en diferentes dominios.
- Ontologías del dominio: se utilizan para representar el conocimiento especializado pertinente de un dominio o subdominio.

Nicola Guarino presenta una clasificación similar a la de Steve y sus colegas añadiendo aquellas ontologías desarrolladas para actividades o tareas específicas. Este tipo de ontologías reciben el nombre de '*task ontology*'. Las categorías de ontologías que constituyen esta clasificación son las siguientes (Guarino, 1998):

- Ontologías genéricas o de alto nivel: describen conceptos generales que normalmente son independientes del dominio o problema particular. Se consideran, generalmente, de utilidad en todos los dominios o aplicaciones.
- Ontologías de dominio: definen un vocabulario para describir conceptos y relaciones de un dominio específico. Estos conceptos son normalmente especializaciones de términos introducidos en la ontología de alto nivel.
- Ontologías de tareas: describen el vocabulario relacionado con una actividad, tarea genérica o artefacto especializando los términos introducidos en la ontología de alto nivel.
- Ontologías de aplicación: describen conceptos dependientes de un dominio y de una tarea en particular, los cuales son frecuentemente especializaciones de ontologías relacionadas. Estos conceptos normalmente suelen corresponder a funciones realizadas por las entidades de dominio mientras desempeñan una determinada actividad.
- Ontologías de grano grueso y de grano fino: con esta clasificación Guarino distingue las ontologías según la exactitud con la que reflejan la conceptualización. Para Guarino, según este criterio, existen dos tipos de

ontologías: (1) las ontologías de grano grueso, y (2) las de grano fino. Las ontologías de grano grueso, que desarrollan una axiomatización más rica, pueden consistir en un conjunto mínimo de axiomas escritos en un lenguaje de expresividad mínima, para apoyar un conjunto de servicios específicos a una amplia comunidad de usuarios que ya han acordado una conceptualización subyacente. Este tipo de ontologías también son llamadas ontologías "off-line", porque son sólo accesibles temporalmente por los usuarios para fines de referencia. Por otro lado, las ontologías de grano fino se desarrollan adoptando un dominio más rico y un conjunto más amplio de axiomas y relaciones conceptuales relevantes que permiten definir con bastante precisión los conceptos a los que se refiere. También pueden ser llamadas ontologías "on-line", debido a que su función es proporcionar soporte al núcleo del sistema.

Roberto Poli propone una clasificación alternativa que distingue los siguientes tipos de ontologías (Poli, 2001):

- Ontologías generales: aquellas que se encuentran centradas en la arquitectónica de la teoría y describen las categorías superiores o fundamentales y oposiciones, además de sus conexiones de dependencia.
- Ontologías categóricas: estas ontologías son más sensibles a los detalles de las categorías individuales. Estudian las diversas formas en las que una categoría da cuenta de los diversos estratos ontológicos, determinando la posible presencia de una teoría general que subsume sus concreciones.
- Ontologías del dominio: se refieren a la estructuración detallada de un contexto de análisis con respecto a los subdominios que lo componen.
- Ontologías genéricas: estas ontologías se encuentran ligadas a corpus lingüísticos y léxicos conceptuales que permiten la clasificación de los términos en varios niveles. Esto significa que cada término debería ser accesible, por defecto, únicamente en su sentido genérico, mientras que su significado especializado queda para cuando se activa una ontología del dominio específica.
- Ontología regional: estas ontologías se centran en analizar las categorías y sus conexiones de interdependencia para cada nivel ontológico (estrato o capa).
- Ontología aplicada: estas ontologías son la aplicación concreta de la infraestructura ontológica a un objeto específico.

I.2.2.2.3. Clasificación por el grado de formalidad de la ontología

Según Roberto Poli, existe otro criterio que puede ser utilizado para clasificar ontologías basado en el grado de formalidad de la ontología (Poli, 2003):

- Ontologías descriptivas: Poli define estas ontologías como un recolector de elementos del mundo, ya sea de un dominio específico de análisis o en general. El autor define el mundo como resultado de un complejo entramado de conexiones de dependencia y formas de independencia entre los elementos que lo componen. Es decir, en el mundo existen cosas materiales, plantas y animales, así como los productos de los talentos y actividades de animales y humanos, pensamientos, sensaciones y decisiones, así como el completo espectro de actividades mentales y reglas que se organizan en relación a la dependencia o independencia entre los mismos elementos. En este ámbito, las ontologías descriptivas están relacionadas con esta recopilación de información.
- Ontologías formales: según Poli, este tipo de ontologías tienen la función de destilar, filtrar y organizar los resultados de una ontología descriptiva, ya sea en su entorno local o global. Según esta interpretación, la ontología formal es formal en el sentido descrito por Husserl en el libro "Logical Investigations" (Husserl, 1970). El filósofo alemán fue el primero en introducir la ontología formal en la filosofía. Husserl define las ontologías formales basándose en la mereología, la teoría de la dependencia y la topología. Para él, una ontología formal se basa en interconexiones de las cosas, con objetos y propiedades, parte y todo que representan categorías puras que caracterizan aspectos y tipos de realidad que todavía no tienen nada que ver con el uso de ningún formalismo específico.

I.2.2.3. Elementos de una ontología

Las ontologías proporcionan un vocabulario común de un área y definen, a diferentes niveles de formalismo, el significado de los términos y relaciones entre ellos. El conocimiento en ontologías se formaliza principalmente usando cinco

tipos de componentes (Gruber, 1993): clases, atributos, relaciones, axiomas e instancias.

- **Clases**. Se suele usar tanto el término "clases" como "conceptos". Un concepto representa cualquier entidad que se puede describir, tiene asociado un identificador único, puede poseer diferentes atributos y establecer relaciones con otros conceptos. Las clases en la ontología se suelen organizar en taxonomías. Algunas veces, la noción de ontología se diluye en el sentido que las taxonomías se consideran ontologías completas (Studer et al., 1998).
- **Atributos**. Los atributos representan la estructura interna de los conceptos. Atendiendo a su origen, los atributos se clasifican en específicos y heredados. Los específicos son los propios del concepto al que pertenecen, mientras que los heredados vienen dados por las relaciones taxonómicas en las que el concepto desempeña el rol de hijo y, por tanto, hereda los atributos del padre. Los atributos se caracterizan por el rango en el cual pueden tomar valor.
- **Relaciones**. Las relaciones representan un tipo de interacción entre los conceptos del dominio. Se definen formalmente como cualquier subconjunto del producto cartesiano de n conjuntos, esto es: "R: C1 x C2 x … x Cn". Las relaciones normalmente suelen ser binarias entre dos conceptos. No todas las relaciones tienen el mismo significado. Existen relaciones binarias de especialización como "*is-a*" o de composición como "*part-whole*", que se pueden modelar con distintas propiedades como la simetría, reflexividad, transitividad, asimetría, etc.
- **Axiomas**. Los axiomas son expresiones que son siempre ciertas, es decir, modelan las "verdades" que siempre se cumplen en el modelo y en el dominio. Pueden ser incluidas en una ontología con muchos propósitos, tales como definir el significado de los componentes ontológicos, definir restricciones complejas sobre los valores de los atributos, argumentos de relaciones, etc. Estos axiomas verifican la corrección de la información especificada en la ontología y pueden ayudar en la generación de nuevo conocimiento.
- **Instancias**. Las instancias son las ocurrencias en el mundo real de los conceptos. En una instancia todos los atributos del concepto tienen asignado un valor concreto.

I.2.2.4. Lenguajes para la representación de ontologías

Los lenguajes ontológicos surgieron como medio para proporcionar capacidad de representación del conocimiento. El primer lenguaje para la definición de ontologías para la Web fue SHOE (Luke et al., 1997). Posteriormente surgieron otros lenguajes con funciones similares, entre los que se pueden destacar, como más representativos, RDF, DAML+OIL y, por último, OWL (del inglés, "*Web Ontology Language*").

A continuación se proporciona una breve descripción de estos lenguajes para la representación de ontologías.

I.2.2.4.1. Simple HTML Ontology Extensions (SHOE)

SHOE (Luke et al., 1997) fue el primer lenguaje de etiquetado utilizado para definir ontologías en la Web. Las ontologías y las etiquetas son incrustadas en archivos HTML. Este lenguaje permitía definir clases, relaciones entre clases, así como reglas de inferencia expresadas en forma de cláusulas de Horn (Heflin et al., 1998). Una de las principales limitaciones de este lenguaje es que no dispone de ningún mecanismo para expresar negaciones o disyunciones. Aunque el proyecto fue abandonado a medida que se desarrollaron nuevos lenguajes de representación de ontologías como OIL y DAML (McGuinness, 2000), este lenguaje disponía de diferentes herramientas, API y editores de anotaciones que, entre otras funciones, permitían la serialización de este lenguaje en XML.

I.2.2.4.2. Resource Description Framework (RDF)

El lenguaje RDF es un modelo de datos que utiliza tripletas para representar recursos y las relaciones que pueden ser establecidas entre ellos. Una tripleta está compuesta por dos nodos (sujeto y objeto) unidos por un arco (predicado) (Klyne & Carroll, 2004):

- Sujeto: identifica el recurso (persona, lugar o cosa) que la sentencia describe. Un recurso RDF puede ser cualquier cosa en un modelo de datos (documento, usuario, producto, etc.). Cada recurso está identificado de forma única a través de una URI.

- Predicado: representa una propiedad del sujeto. Al igual que ocurre con los sujetos, cada propiedad se identifica con una URI única.
- Objeto: especifica el valor del predicado para un sujeto. En RDF un objeto puede ser otro recurso o un literal. En el caso de que sea un recurso, el objeto definirá otra URI que identifique al recurso. Por su parte, los literales son una cadena simple de caracteres u otro tipo de datos primitivo definido por XML. En términos de RDF, un literal puede contener marcado XML pero no es interpretado por RDF. El modelo RDF distingue a los literales de los recursos restringiendo a los literales ser sujeto o predicado en una declaración.

La Figura I.3 muestra un ejemplo del modelo de datos en RDF para almacenar información mediante la utilización de las tripletas.

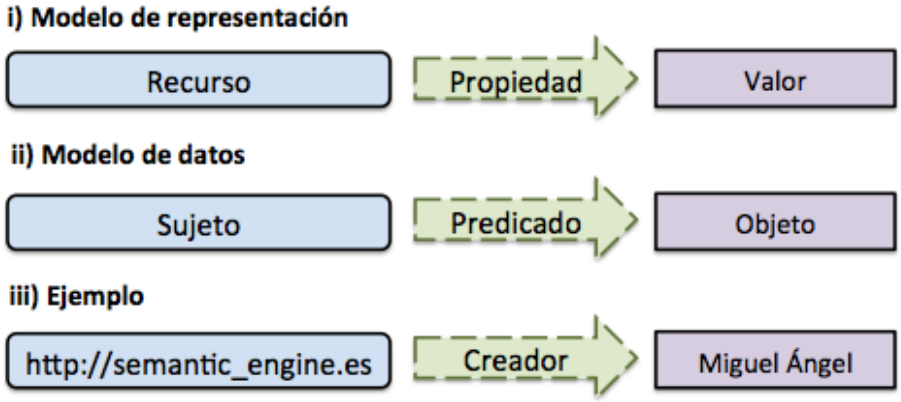


Figura I.3 Ejemplo de grafo RDF

RDF provee un mecanismo capaz de describir recursos e identificarlos unívocamente. Para identificar globalmente los recursos y las propiedades se utilizan la URI. Una URI es una cadena de caracteres que identifica unívocamente a un recurso, y está compuesta por un espacio de nombres y un identificador que debe ser único dentro de este espacio de nombres. Dentro de las tripletas, los sujetos deben ser nodos identificados por una URI. Sin embargo, los objetos de las tripletas pueden ser tanto otros recursos como valores literales de propiedades. La Figura I.4 muestra un ejemplo de cómo RDF es capaz de describir recursos e identificarlos unívocamente utilizando los identificadores URI.

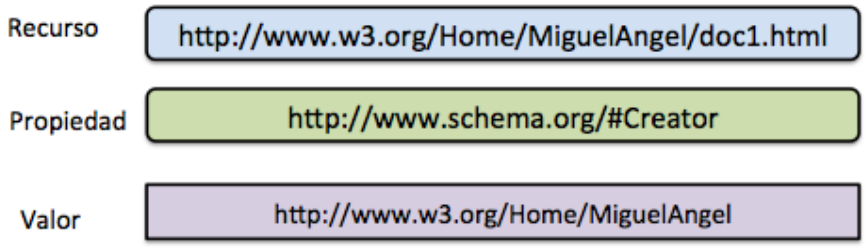


Figura I.4 Ejemplo grafo RDF con URI

La Figura I.4 representa un modelo de datos RDF para acceder a la propiedad de un recurso que, en este ejemplo, modela el acceso a la propiedad *Creator* del recurso http://www.w3.org/Home/MiguelAngel/*doc1.html*.

I.2.2.4.3. Resource Description Framework Schema (RDFS)

RDFS o esquema RDF proporciona un vocabulario de modelado de datos para RDF, constituyendo una extensión del vocabulario básico de RDF[5]. Este vocabulario permite expresar clases y sus relaciones jerárquicas, al tiempo que se definen propiedades que se asocian con clases (Cyganiak et al., 2004).

RDFS junto con RDF constituye el *lenguaje universal* que se utiliza para expresar el conocimiento en la Web Semántica. La incorporación de este vocabulario a RDF proporciona un lenguaje de mayor capacidad para representar relaciones semánticas más complejas. A través de esta extensión, RDF puede definir clases, relaciones de pertenencia entre clases, dominios y rangos para las propiedades, pudiendo restringir en qué clases está cada propiedad y que valores pueden ser asignados. Este sistema de clases y propiedades que se definen en RDFS es similar al sistema de tipos de los lenguajes de orientación a objetos. La diferencia radica en la metodología de definición que ambas utilizan. Mientras que en los lenguajes orientados a objetos se definen las clases en relación a las propiedades que se desea que posean las instancias, en RDFS es al contrario, las propiedades se definen basándose en las clases de recursos a los que éstas se pueden aplicar. Por lo tanto, si definimos los roles 'dominio' y 'rango' como clases, se podría definir la propiedad "*ma:escritor*" con un dominio "*ma:Libro*" y un rango "*ma:Persona*" y,

[5] http://www.w3.org/TR/rdf-schema/

además, se podrían definir tantas propiedades que tengan como dominio "*ma:Libro*" y rango "*ma:Persona*" como sean necesarias sin la necesidad de modificar la descripción original de las clases.

Esta descripción nos lleva a reconocer que RDFS es un lenguaje ontológico simplista, que ofrece ciertas primitivas para el modelado de clases, relaciones de subclases, relaciones de subpropiedades y restricciones de dominio y rango, con un significado fijo (Antoniou & Van Harmelen, 2004). El problema de RDFS es que es un lenguaje demasiado primitivo, que carece de muchas características de modelado interesantes que otros lenguajes más actuales aportan. Entre las limitaciones existentes, se destacan las siguientes (Antoniou & Van Harmelen, 2004):

- Básicamente permite la organización de vocabularios en jerarquías.
- El ámbito local de las propiedades que restringe la aplicación para que sólo se aplique a algunas clases.
- No permite expresar clases disjuntas.
- No permite expresar combinación booleana de clases.
- No permite establecer restricciones de cardinalidad.
- No permite expresar características de las propiedades tales como transitividad, simetría, unicidad, propiedad inversa, etc.

A pesar de estas limitaciones, la unión de RDF y RDFS permite realizar tareas de razonamiento automáticas para inferir nuevas relaciones sobre una base de conocimiento dada.

I.2.2.4.4. Web Ontology Language (OWL)

OWL es un lenguaje ontológico para la Web basado en RDF que facilita su publicación (McGuinness & Van Harmelen, 2004). Se trata de una recomendación de la W3C. El primer borrador de la especificación del lenguaje, OWL 1.0, apareció en julio de 2002 y fue presentado de manera formal por la W3C en febrero de 2004. En 2009 apareció la última versión hasta el momento, la 2.0. El lenguaje OWL proporciona un mecanismo para interpretar el contenido de la Web que supera los propuestos previamente, XML, RDF, y esquema RDF (RDFS), proporcionando vocabulario adicional junto con una semántica formal.

El primer elemento que se declara en un documento RDF/OWL es "*owl:ontology*". Este elemento se utiliza en el lenguaje OWL para definir el concepto de ontología. La declaración de una ontología puede tener asociadas diferentes propiedades o relaciones como, por ejemplo, la propiedad "*owl:imports*" que permite importar definiciones de ontologías externas. Además, por defecto OWL define dos clases que estarán presentes en todas las ontologías de manera implícita: (i) "*owl:Thing*", que representa la clase raíz de la ontología, es decir, la clase de todos los individuos; todas las clases que sean definidas en la ontología serán sub-clases de ésta; y (ii) "*owl:Nothing*", que representa la clase vacía y todas las clases en la ontología tienen como subclase a esta clase.

Las diferentes versiones de OWL han venido acompañadas de diferentes sub-lenguajes expresivos con los que construir las ontologías. La primera versión de OWL definió los lenguajes OWL-Lite, OWL DL y OWL FULL con una expresividad creciente. Cada sub-lenguaje presenta características diferentes que se analizan a continuación:

- OWL Lite: es el sub-lenguaje más simple. Añade una serie de restricciones en el uso de los constructores de OWL tales como restricciones de rango local, restricciones existenciales, restricciones de cardinalidad simple y varios tipos de propiedades (inversa, transitiva y simétrica). Básicamente, permite modelar jerarquías de clasificación y restricciones sencillas.
- OWL-DL: proporciona la máxima expresividad posible garantizando que se mantiene la integridad de cómputo y la decidibilidad en un tiempo finito para tareas de inferencia. Este sub-lenguaje incluye todas las construcciones de OWL pero con ciertas restricciones como la condición de separación en el tipo de recurso. OWL DL se denomina así por su correspondencia con la lógica descriptiva.
- OWL-FULL: con este sub-lenguaje se pueden utilizar todos los constructores y primitivas definidos en OWL y, además, aporta la libertad sintáctica ofrecida por RDF. Esto implica que no se garantiza su decidibilidad pero, en cambio, como familia de "*Description Logics*", posee un gran poder expresivo al permitir tratar clases como instancias o definir propiedades sobre tipos de datos (string, float, etc.).

La versión 2 de OWL tiene una estructura muy similar a la versión 1 pero, sin embargo, la sintaxis de ambas versiones son diferentes. La sintaxis funcional en OWL 2 se diferencia en la forma de la sintaxis abstracta de OWL 1, pero el rol que desempeña sobre la estructura OWL es exactamente la misma, de ahí que la compatibilidad de las versiones sea prácticamente completa. La sintaxis de OWL 2 es mucho más cercana a la representación en grafos RDF. Además, OWL 2.0 añade nuevas funcionalidades con respecto a OWL 1.0 que mejoran la expresividad del lenguaje. Entre las nuevas características que aporta este lenguaje se pueden destacar las siguientes: (i) definición de claves en las clases, (ii) cadenas de propiedades, (iii) tipos de datos y rangos de datos más complejos, (iv) restricciones de cardinalidad cualificadas, (v) propiedades asimétricas, reflexivas y disjuntas, y (vi) mejora de las características de las anotaciones. OWL 2 no mantiene la misma estructura de sub-lenguajes que OWL 1 y ésta cambia por completo (véase Figura I.5). En esta versión se definen tres perfiles, siendo cada uno de ellos más restrictivo que OWL DL. Los diferentes perfiles existentes en OWL 2 son:

- OWL 2 EL: se define como un subconjunto de OWL 2 que está diseñado para aplicaciones que trabajen con ontologías que contienen un gran número de propiedades y/o clases. El acrónimo "EL" en el nombre del perfil refleja la incorporación a la familia de lógicas descriptivas EL, que permite realizar tareas básicas de razonamiento en tiempos polinómicos, aunque su expresividad sea limitada restringiendo el uso de cuantificadores universales.
- OWL 2 QL: está destinado para aplicaciones que requieran grandes volúmenes de instancias de datos y donde la consulta de información es la tarea más importante de razonamiento. En OWL 2 QL se permite la unión de consultas utilizando los sistemas de base de datos relacionales. Como en el perfil OWL 2 EL, incorpora algoritmos de tiempo polinomial para comprobar la consistencia de la ontología y las expresiones lógicas. El acrónimo QL relaciona el perfil con las consultas de información y con el hecho de que permite la utilización de un lenguaje de consultas estándar relacional.
- OWL 2 RL: está destinado para aplicaciones que requieren un razonamiento escalable sin sacrificar excesivamente el poder de la expresividad. Este perfil ha sido diseñado para permitir que las aplicaciones OWL 2 puedan intercambiar la

expresividad del lenguaje por la eficiencia. También ha sido diseñado para aplicaciones RDFS que necesitan expresividad añadida. Los sistemas de razonamiento OWL 2 RL pueden utilizar motores de razonamiento basados en reglas. Problemas como la consistencia de la ontología, satisfacibilidad de las expresiones, subsunción de las expresiones, chequeo de instancias y unión de consultas pueden ser solucionados en tiempo polinómico respecto al tamaño de la ontología. El acrónimo RL refleja el hecho de que el razonamiento en este perfil puede ser implementado mediante un lenguaje de reglas estándar.

La Figura I.5 representa gráficamente la estructura de sub-lenguajes de OWL 2.0. A diferencia de OWL 1.0, la organización de los sub-lenguajes no es estrictamente jerárquica, sino que en esta versión los diferentes sub-lenguajes comparten características entre ellos. La lógica descriptiva se encuentra presente en cada uno de ellos y la perspectiva del diseño cambia radicalmente con respecto a la versión 1.0. En este caso, cada uno de los sub-lenguajes ha sido diseñado para ofrecer importantes ventajas en escenarios concretos de aplicación, en lugar de estar jerárquicamente diseñados como ocurre en el caso de la versión OWL 1.0.

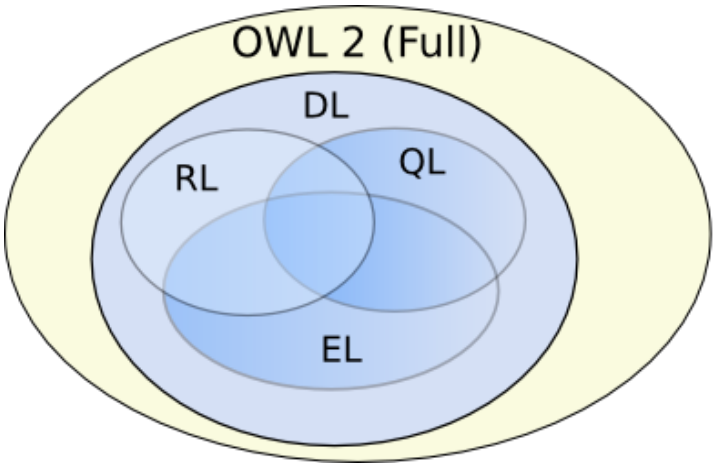


Figura I.5 Perfiles de OWL 2.0 (Diagrama de Venn)

Una de las nuevas funcionalidades innovadoras que se han incorporado en la versión 2 es la función de "punning", que permite la definición de recursos como instancias y clases al mismo tiempo. La función "punning" permite definir una clase y una instancia asociada a la misma clase y que, además, compartan la misma URI, a diferencia de OWL versión 1, donde a cada recurso se le asigna una URI diferente.

Por lo tanto, OWL 2 permite agrupar los recursos a través de una URI que representa un recurso que, en función del contexto, puede ser tratado como una clase o como una instancia. A nivel de clase, estos recursos pueden tener a su vez sub-clases o propiedades declaradas, y a nivel de instancia pueden contener otras instancias del modelo ontológico. El objetivo de esta nueva funcionalidad es facilitar la reutilización a nivel de instancia de la semántica de una conceptualización de un dominio a nivel de clase.

En OWL 2 se definen diferentes sintaxis que añaden diferentes funcionalidades al lenguaje. La característica que todas las sintaxis comparten es que deben estar obligatoriamente basadas en RDF/XML. De hecho, según el estándar, todas las herramientas que implementen OWL 2 deben de poder leer y escribir usando esta sintaxis. La sintaxis RDF/XML (W3C, 2014) permite representar la información de las ontologías para que se puedan almacenar en documentos OWL y, además, tiene por objeto servir como medio de intercambio de documentos OWL 2. Por su parte, el objetivo de la sintaxis OWL/XML (W3C, 2012b) es cumplir el estándar XML para que puedan reutilizarse herramientas XML para procesar este tipo de ontologías. En OWL 2 también se ha definido una sintaxis funcional (W3C, 2012a) orientada a facilitar la lectura de la estructura/forma de las ontologías. Finalmente, la sintaxis Turtle (Beckett & Berners-Lee, 2008) facilita la lectura y escritura de tripletas RDF.

I.3. EVOLUCIÓN DE ONTOLOGÍAS

I.3.1. DEFINICIÓN

Según Stojanovic (2004), la evolución de ontologías se define como un proceso de adaptación puntual de una ontología a los cambios surgidos y la propagación constante de estos cambios a los demás objetos dependientes. Un cambio en una parte de una ontología puede causar inconsistencias en otras partes de la ontología, así como en los artefactos dependientes de ella. La gran variedad de causas y consecuencias de los cambios dentro de las ontologías hacen de este proceso una tarea muy compleja (Stojanovic et al., 2002).

Es importante determinar cuáles son las causas de estos cambios, es decir, por qué las ontologías cambian, qué cambios deben realizarse y sobre qué aspectos de la ontología. Para ello, Flouris y sus colegas (2007), basándose en el trabajo de Klein y Fensel (2001), manifiestan que la evolución de una ontología puede ser causada por cualquier cambio producido en el dominio, en la conceptualización o en la especificación. Sin embargo, no todos los cambios en las ontologías están orientados a evolucionarla. En el siguiente apartado se presenta una breve clasificación de los diferentes tipos de cambio que pueden producirse en una ontología.

I.3.2. CLASIFICACIÓN DE CAMBIOS EN LA ONTOLOGÍA

El término de "cambio en las ontologías" establece el problema de decidir qué tipo de modificaciones deben de realizarse sobre una ontología en respuesta de una cierta necesidad, así como la aplicación de estas modificaciones y la gestión de sus efectos en función de datos, servicios, aplicaciones, agentes y otros elementos (Flouris et al., 2007).

Una ontología puede requerir una modificación por algunos de los siguientes motivos: el dominio de interés ha cambiado (Stojanovic et al., 2003); un cambio en la perspectiva de cómo se ve el dominio modelado (Noy & Klein, 2004); la existencia de defectos de diseño en la conceptualización del dominio (Plessers & Troyer, 2005); actualizaciones para añadir nuevas funcionalidades que provienen de un cambio de las necesidades por parte de los usuarios de las aplicaciones (Haase & Stojanovic, 2005); descubrimiento de incoherencias o inconsistencias en el modelado (Flouris & Plexousakis, 2006); y conocimiento de nueva información desconocida anteriormente que hace diferentes las características del dominio (Heflin et al., 1999).

Flouris y sus colegas (2007) definen los diferentes tipos de cambios que cubren de manera individual cada uno de los sucesos comentados anteriormente. De entre todos los tipos de cambios analizados en este trabajo, nos centraremos tanto en la evolución de ontologías, como en el versionado, que es considerado como una fuerte variante de la evolución de ontologías.

I.3.2.1. Evolución de ontologías vs. versionado de ontologías

La evolución de ontologías resulta ser un importante problema porque la eficiencia de la aplicación depende, en gran medida, de la calidad de la conceptualización del dominio de la ontología subyacente (Stojanovic et al., 2003) y, a su vez, esta calidad se ve afectada directamente, por la capacidad del algoritmo de evolución para adaptar apropiadamente la ontología con los cambios del dominio y los cambios en la conceptualización del mismo. La Figura I.6 representa gráficamente el proceso de cómo se lleva a cabo la evolución de ontologías.

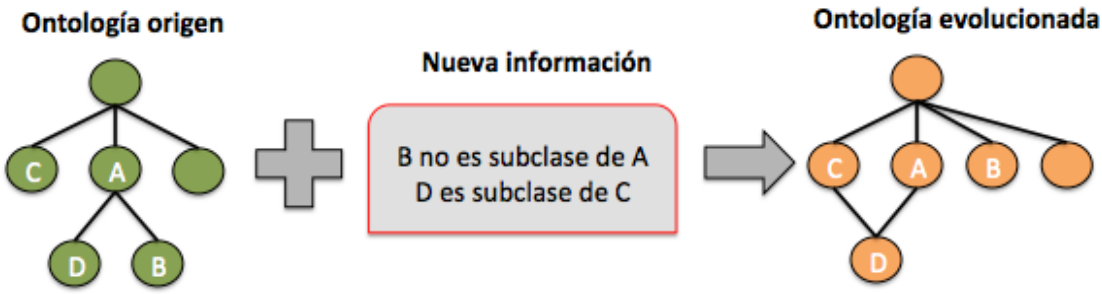


Figura I.6 Evolución de ontologías

En general, para este proceso se requiere la ontología origen y una lista de cambios que describan las operaciones que se van a realizar. La Figura I.6 muestra gráficamente este proceso con una ontología origen que representa la ontología sobre la que se van a realizar los cambios, una lista de operaciones a realizar sobre la ontología y la ontología final evolucionada. Esta tarea se compone de un proceso cíclico de seis fases (Stojanovic et al., 2002): fase de captura de cambios, fase de representación de cambios, fase de cambios semánticos, fase de implementación, fase de propagación y, por último, la fase de validación. El objetivo general de todo el proceso de evolución es aplicar los cambios necesarios y mantener la consistencia en la ontología evolucionada.

Como ya se ha mencionado anteriormente, existen varios trabajos que consideran el versionado de las ontologías como una variante de la evolución de ontologías. Noy y Klein (Noy & Klein, 2004) fusionan los conceptos de evolución y versionado de ontologías como la capacidad de gestionar cambios en la ontología y sus efectos mediante la creación y el mantenimiento de las diferentes variantes de

la ontología. Además, en este trabajo se realiza un estudio que analiza las causas más importantes de por qué es necesaria la evolución en la ontología y también recalcan las diferencias existentes entre una ontología y un esquema de base de datos. Más concretamente, se analiza la existencia de dos tipos de evolución de ontologías, por un lado la evolución trazable y, por otro lado, la evolución no trazable. La diferencia entre ellas radica en que la evolución trazable trata la evolución como una serie de operaciones sobre la ontología, de tal forma que se conocen los diferentes efectos sufridos por la ontología originaria para llegar a la ontología evolucionada. Sin embargo, en la evolución no trazable al terminar el proceso de evolución se dispone de dos versiones diferentes de la ontología, la ontología original y la evolucionada, sin conocimiento alguno de qué pasos fueron realizados durante el proceso de evolución.

El versionado de ontologías se considera un tipo de cambio muy útil dentro de la Web Semántica dada su naturaleza distribuida y descentralizada (Heflin & Pan, 2004). El versionado de ontologías cumple la difícil función de proporcionar un acceso transparente e individual a diferentes versiones de una misma ontología por parte de los elementos dependientes. La labor que desempeña este tipo de cambio es intentar minimizar cualquier efecto adverso que un cambio produce sobre una ontología o cualquier entidad dependiente de la misma como agentes, aplicaciones u otros elementos.

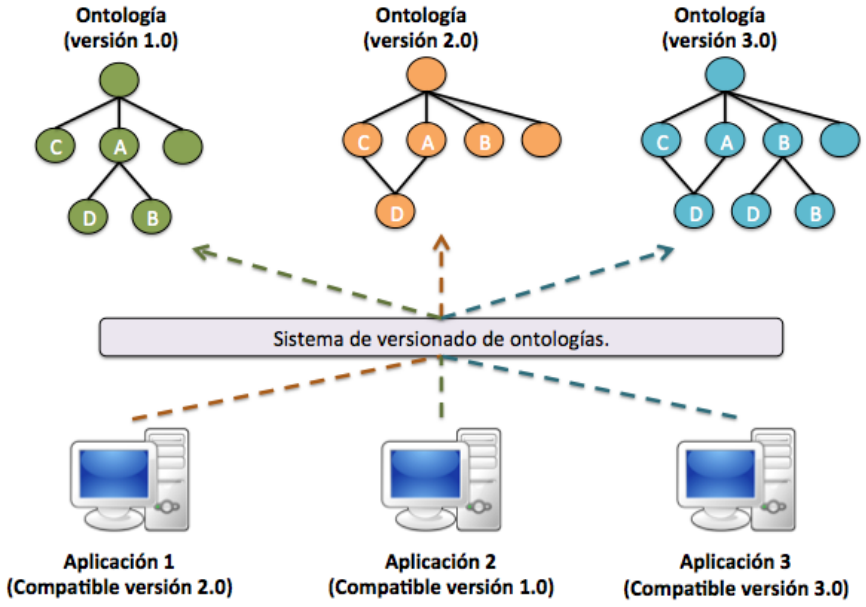


Figura I.7 Versionado de ontologías

El versionado de ontologías comienza cuando se produce un cambio en una ontología y los algoritmos de control de versiones entran en juego. La función que desempañan los sistemas de control de versiones de ontologías no solo se restringe al almacenamiento de las diferentes versiones de una ontología, sino que también deben de disponer de mecanismos de identificación que permitan diferenciar varias versiones de una misma ontología, así como el tipo de relación que existe entre cada versión de la ontología y las compatibilidades ente ellas. Además, los sistemas de versionado de ontologías (véase Figura I.7) proporcionan un acceso transparente a las diferentes versiones de una ontología, relacionando automáticamente distintas versiones con fuentes de datos, aplicaciones u otros elementos dependientes (Klein & Fensel, 2001).

I.3.3. ANÁLISIS DE PROBLEMAS EN LA EVOLUCIÓN DE ONTOLOGÍAS

Stojanovic (2004) establece que la evolución de ontologías tiene dos problemas principales que tienen que ser analizados y tratados. El primer problema es entender cómo una ontología puede evolucionar a través de la aplicación de un conjunto de cambios. El otro gran problema a tener en cuenta es el mantenimiento de la consistencia, es decir, que los nuevos cambios incorporados en la ontología mantengan y no violen ninguna de las restricciones que habían sido establecidas en la ontología original.

La solución para el primer problema se basa en el trabajo de Banerjee y sus colegas (1987), donde proponen un esquema de evolución de ontologías basado en el paradigma de orientación a objetos. Esta solución propone una taxonomía de cambios para aspectos generales de un modelo ontológico como incorporación de conceptos en la jerarquía, la definición de cardinalidades o restricciones de dominio o de rango. En concreto, la utilización de cada uno de estos cambios se basa en definir una ontología como una composición de entidades que evoluciona a través de la eliminación o incorporación de nuevas entidades. Sin embargo, esta taxonomía no cubre todas las posibles operaciones que se pueden realizar sobre los modelos ontológicos.

Por otro lado, el otro objetivo que persigue la evolución de ontologías, es asegurar que todos los cambios aplicados sobre una ontología, mantienen el

conjunto de restricciones semánticas definidas en ésta. La aplicación de cualquier cambio en una ontología no siempre la deja en estado consistente, por lo que es necesario la aplicación de un proceso de verificación que compruebe la exactitud de la ontología con respecto a su consistencia. Existen dos enfoques que facilitan la labor de verificación de la consistencia (Hürsch et al., 1995), la verificación a priori y a posteriori. La verificación a posteriori ejecuta, en primer lugar, la operación de cambio y, después, se valida la ontología actualizada para comprobar si satisface las restricciones de consistencia. Por otro lado, la verificación a priori establece que se mantendrá la consistencia en una ontología si ésta es consistente antes de una actualización y las condiciones previas se cumplen. Este enfoque define una serie de precondiciones para cada cambio, que deben ser satisfechas para que el cambio sea aplicado.

I.3.4. EVOLUCIÓN DE ONTOLOGÍAS BASADO EN APRENDIZAJE DE ONTOLOGÍAS

I.3.4.1. Definición de aprendizaje de ontologías

Uno de las técnicas más utilizadas en la evolución de ontologías es el aprendizaje de ontologías (del inglés, '*Ontology Learning*'). El aprendizaje de ontologías es un área de investigación que tiene sus orígenes en las áreas de Procesamiento del lenguaje natural (PLN), Inteligencia Artificial (IA) y Aprendizaje Computacional (AC). Maedche y Staab (2001) establecen el objetivo del aprendizaje de ontologías como una herramienta que permite y facilita la tarea de construir ontologías al experto. Posteriormente, en otro trabajo, Maedche y Staab (2004) definen el objetivo del aprendizaje de ontologías como "*la integración de una multitud de disciplinas con el fin de facilitar la construcción de ontologías, como la ingeniería ontológica y el aprendizaje computacional*".

Por otro lado, Yang y Callan (2008) definen la construcción automática de ontologías, también conocido como aprendizaje de ontologías, como una tarea importante en Inteligencia Artificial, Web Semántica y Gestión del Conocimiento. Se trata del proceso de construir una ontología, un modelo de datos que representa un conjunto de conceptos dentro de un dominio y las relaciones entre los conceptos.

Gulla y sus colegas (2007) definen aprendizaje de ontologías como "*el proceso de construir ontologías de manera automática o semiautomática sobre la base de las descripciones textuales de un dominio*".

Si se analizan detalladamente cada una de las definiciones aportadas, todas coinciden en que el aprendizaje de ontologías trata de facilitar tarea de construcción de ontologías, ya que los enfoques tradicionales de ingeniería de ontologías son tediosos, requieren mucho trabajo y representan un cuello de botella para la creación de la Web Semántica (Gulla et al., 2007). Otra característica que se contempla en todas las definiciones aportadas es que el aprendizaje de ontologías es un proceso. Éste está compuesto por una serie de sub-tareas o subprocesos que serán analizados en el siguiente apartado.

I.3.4.2. Modelo de capas del aprendizaje de ontologías

Aunque existen varias metodologías desarrolladas en esta área de investigación, la comunidad del aprendizaje de ontologías no ha definido ninguna solución estándar. Sin embargo, varios trabajos en este campo (Wong et al., 2012) presentan el aprendizaje de ontologías como un proceso compuesto por varias sub-tareas específicas (véase Figura I.8).

Figura I.8 Representación de las sub-tareas del aprendizaje de ontologías

La Figura I.8 recoge las diferentes sub-tareas que componen el proceso de aprendizaje de ontologías (Buitelaar & Magnini, 2005). Cada nivel hace referencia al tipo lingüístico extraído y la sub-tarea que tiene asociada ese nivel. Así, el nivel base tiene como sub-tarea la extracción terminológica de un dominio. Esta tarea tiene el objetivo de obtener términos y sinónimos al final del proceso de extracción. Los siguientes niveles se encargan de la extracción de conceptos, relaciones taxonómicas y no taxonómicas y, por último, la extracción de reglas o axiomas. Por encima del nivel de descubrimiento existen otros niveles superiores en el proceso que se encargan de poblar la ontología, extender la jerarquía de conceptos y detectar eventos y marcos.

Utilizando este desglose de sub-tareas en el aprendizaje de ontologías como base, en los siguientes apartados se explican sólo aquellas sub-tareas que se encuentran realmente relacionadas con la aproximación propuesta en esta tesis.

I.3.4.2.1. Extracción terminológica

La extracción terminológica permite la identificación y extracción de candidatos a términos a partir del análisis de grandes cantidades de información. Vivaldi y sus colegas (2001) proporcionan la siguiente definición: "*la extracción de términos es la tarea de detectar de forma automática, a partir de un corpus textual, unidades léxicas que designen conceptos en dominios temáticamente restringidos*". Esta definición establece la extracción de terminologías como una tarea automática que facilita la obtención de unidades léxicas en un dominio concreto.

Las unidades léxicas representan los términos que se obtienen al finalizar la tarea de extracción. Según Dubuc (1997) un término es "*el elemento constitutivo de cualquier nomenclatura terminológica que esté relacionada con una lengua de especialidad, es la denominación de un objeto, propio de una determinada área de especialidad*". La definición aportada por Dubuc establece que la extracción de un término de su contexto o de su área de especialidad le hace perder automáticamente la categoría de término salvo que esté definido en la nueva especialidad. Siguiendo en la misma línea, Barrón y sus colegas proporciona otra reflexión que nos permite ahondar más en esta definición (Barrón et al., 2006):

"*Un término es un sintagma asociado a un concepto, dentro de un área de especialidad, en donde un sintagma es un conjunto de una o más palabras cohesionadas que tienen un concepto asociado*"

Según Barrón los términos son sintagmas que no están limitados a una sola palabra, sino que pueden estar constituidos por varias palabras y formar un sintagma multi-palabra, también conocido con el nombre de "sintagma compuesto". Un ejemplo de sintagma compuesto sería "anotador semántico", donde el conjunto de palabras hace referencia a un solo término.

La extracción terminológica del dominio es un paso crucial que constituye un prerrequisito en el aprendizaje de ontologías (Cimiano et al., 2006). En general, un término es una representación superficial de un concepto en un dominio especifico (Pazienza et al., 2005). Los términos son realizaciones lingüísticas de los conceptos específicos de dominio y, por lo tanto, la base para otras tareas más complejas (Buitelaar & Magnini, 2005). La principal característica de los términos es que son "mono referenciales", es decir, un término referencia a un concepto especifico en un campo particular (Ananiadou, 1994). En el aprendizaje de ontologías, los términos que se extraen de un corpus se suelen representar como conceptos candidatos a utilizarse para la evolución de ontologías.

Actualmente, existen varios enfoques y estrategias que utilizan técnicas supervisadas y no supervisadas para extraer y reconocer términos. Dentro de las diferentes aproximaciones existentes, se pueden destacar tres enfoques principalmente (Pazienza et al., 2005). Por un lado, se han definido medidas estadísticas para definir el grado de terminologización (del inglés, '*termhood*'), que es una característica de los términos utilizada en el área de investigación terminológica que hace referencia al grado de unidad lingüística que está relacionado con conceptos del dominio especifico (Kageura & Umino, 1996). Otro de los enfoques se basa en tratar de identificar y reconocer términos mediante la búsqueda de propiedades lingüísticas puras, utilizando técnicas de filtrado lingüístico con el objetivo de identificar patrones sintácticos de términos (Gianluca et al., 1997). Por último, las aproximaciones híbridas utilizan medidas estadísticas y técnicas de filtrado lingüístico para el reconocimiento de términos (Pazienza et al., 2005).

En concreto, la técnica de extracción de términos que se utiliza en este trabajo de investigación se basa en el enfoque de filtrado lingüístico. El funcionamiento de esta técnica se basa en tres importantes pasos (Krauthammer & Nenadic, 2004):

- **Reconocimiento de términos**. El objetivo principal de esta tarea es saber diferenciar entre posibles términos dentro de un corpus. En (Kageura & Umino, 1996) se analizan los principios y métodos del reconocimiento automático de términos.
- **Clasificación de términos**. La tarea de clasificación de términos tiene el objetivo de clasificar el término reconocido en una categoría semántica. En el ámbito de las ontologías, las categorías semánticas suelen ser conceptos del dominio.
- **Mapeo de términos**. El objetivo de esta última tarea es relacionar una ocurrencia de un término con una fuente de datos referente como vocabularios, léxicos, tesauros o bases de datos. El mapeo de términos se enfrenta a dos problemas principales, el problema de la ambigüedad léxica y la amplia variabilidad de las representaciones léxicas. El problema de la ambigüedad léxica se soluciona en la mayoría de los casos durante la tarea de clasificación, donde se utilizan procesos de desambiguación para eliminar la ambigüedad de un término en el caso de que disponga de distintos significados. El problema de la variabilidad léxica incluye cualquier variación simple relacionada con diferencias ortográficas en el término, como puede ser la detección de cambios de género y número, y variaciones más complejas relacionadas con la sinonimia, que facilita la identificación de grupos de términos que representan el mismo concepto.

Como se puede ver, la extracción de términos requiere de unos niveles avanzados de procesamiento lingüístico para tratar de identificar sentencias nominales complejas, analizar su estructura semántica interna y obtener los términos relacionados.

En el siguiente nivel, por encima de los términos, se encuentra la sinonimia. Según Petasis y sus colegas (2011) "*todos los términos que son sinónimos hacen referencia al mismo objeto real o evento y, de esta forma, todos materializan un concepto o relación*". La identificación de sinónimos ayuda a evitar conceptos

redundantes dado que dos o más términos pueden representar el mismo concepto (Drumond & Girardi, 2008).

En la literatura existen diferentes enfoques para solucionar el problema de extracción o detección de sinónimos. Entre ellos se pueden resaltar las soluciones basadas en sistemas de bases de datos léxicas y las soluciones basadas en metodologías estadísticas. Las bases de datos léxicas son construidas a mano y aseguran cierto nivel de calidad. Algunos ejemplos son Wordnet (Miller, 1995), BRICO (Haase, 2000) y EuroWordnet (Vossen, 1998). Por otro lado, existen diferentes tipos de aproximaciones estadísticas, como las aproximaciones basadas en coocurrencia (Manning & Schütze, 1999) y otras aproximaciones que se basan en medidas de similitud semántica y que utilizan técnicas de aprendizaje no supervisadas. También existen algunas soluciones híbridas que mezclan técnicas estadísticas y bases de datos léxicas (Turney, 2001). Según Cimiano y sus colegas (2006) el enfoque más común para solucionar la detección de sinónimos en el aprendizaje de ontologías es el enfoque estadístico, utilizando técnicas de clustering para hacer grupos de palabras similares o utilizar medidas de asociación para detectar pares de términos estadísticamente correlacionados.

I.3.4.2.2. Descubrimiento de conceptos

Los conceptos se representan por un conjunto de términos relacionados con un significado común (Reiterer et al., 2010). Un concepto puede describirse a través de un "triángulo de significado" (Ogden et al., 1946). Un "triángulo de significado" representa que un concepto, pensamiento o referencia describe o simboliza un objeto que, a su vez, puede representar un elemento o referente específico. Tomemos, por ejemplo, la palabra "gato". Esta palabra se encuentra definida por la combinación de cuatro letras que, cada vez que escuchamos, leemos o escribimos, hace referencia a un animal felino, doméstico, mamífero y carnívoro. Además, cada vez que vemos un animal de estas características podemos diferenciarlo de otro tipo de animales como, por ejemplo, un perro.

La Figura I.9 muestra las relaciones básicas de un "triángulo de significado" (Ogden et al., 1946). En la parte inferior izquierda de la figura se muestra un icono que se asemeja a un gato llamado "Yojo". A la derecha aparece un símbolo impreso

que representa su nombre. La nube en la parte superior proporciona una excitación neuronal inducida por los rayos de luz que rebotan sobre "Yojo" y sus alrededores. Esta excitación, llamada concepto, es el mediador que relaciona a un símbolo con su objeto (Sowa, 2000).

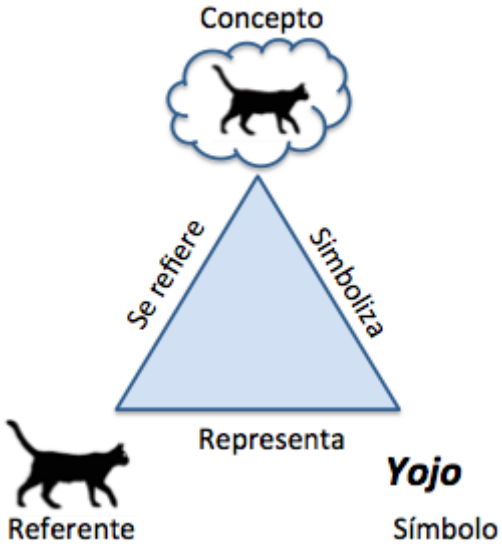


Figura I.9 Representación de un "triángulo de significado"

Una definición de "concepto" bastante extendida en el área de investigación del aprendizaje de ontologías es la propuesta por Buitelaar y sus colegas (2005), en la que se define un concepto a través de la intensión del concepto, su extensión y su realización léxica. La intensión puede ser una definición formal identificando sus propiedades y las relaciones que tiene con otros conceptos, o una definición informal del concepto tal como una descripción textual dentro de un diccionario. La extensión representa un conjunto de instancias del mismo concepto. La realización léxica está constituida por un conjunto de términos que representan este concepto en uno o varios idiomas.

La mayoría de aproximaciones tratan de identificar la existencia de sinónimos entre todos los términos candidatos. Si se encuentran sinónimos entre los términos, entonces éstos se agrupan representando un concepto. En caso contrario, se crea un concepto por cada término diferente extraído en la etapa anterior.

I.3.4.2.3. Derivación jerárquica de conceptos

El siguiente nivel está relacionado con la construcción de jerarquías de conceptos o taxonomías. Una taxonomía puede definirse como una colección controlada de vocabulario de términos organizada en una estructura jerárquica. Cada término está relacionado con uno o más términos en la taxonomía con relaciones del tipo padre-hijo (Ryu & Choi, 2006). Las taxonomías son artefactos útiles para organizar muchos aspectos del conocimiento. Como componentes de las ontologías, las taxonomías proporcionan un modelo de organización para un dominio o un modelo adecuado para tareas especificas (Burgun & Bodenreider, 2001).

Una de las partes más importantes de una ontología es su taxonomía o su jerarquía de conceptos. Las relaciones taxonómicas, conocidas como relaciones "IS-A", permiten una vista de árbol de la ontología y determinan la herencia entre conceptos (Petasis et al., 2011).

Buitelaar y sus colegas (2005) distinguen tres tipos diferentes de paradigmas para crear taxonomías desde texto. El primero de estos paradigmas utiliza patrones léxico-sintácticos, tales como los patrones Hearst (Hearst, 1992), para detectar relaciones de hiponimia ("IS-A"). En concreto, este enfoque también se utiliza en estrategias que analizan la estructura interna de los sintagmas nominales para derivar las relaciones taxonómicas entre clases expresadas como núcleo sintáctico del sintagma nominal y sus subclases, que pueden ser derivadas de la combinación del núcleo sintáctico y sus modificadores (Buitelaar et al., 2004).

El segundo paradigma se basa en la hipótesis de distribución de Harris (1954), según la cual, la similitud semántica entre unidades léxicas puede detectarse a través de la búsqueda de coincidencias en el contexto lingüístico. Este paradigma utiliza algoritmos de agrupamiento jerárquico para construir automáticamente jerarquías desde texto (Cimiano et al., 2005).

El último paradigma procede del área de investigación de recuperación de información y utiliza un tipo de coocurrencia conocido como "subsunción" para organizar jerárquicamente la información extraída de documentos. Un ejemplo de la utilización de este paradigma se puede encontrar en el trabajo (Sanderson & Croft, 1999).

Como ejemplo del uso de los distintos paradigmas, el trabajo presentado en (Maedche & Staab, 2004) describe un framework de aprendizaje de ontologías que incluye tres tipos de diferentes estrategias para extraer relaciones taxonómicas de texto: extracción estadística basada en técnicas de clustering, extracción estadística basada en técnicas de clasificación y extracción basada en patrones léxico-sintácticos. La primera estrategia se basa en técnicas de agrupación, por lo que estaría basado en el segundo paradigma explicado anteriormente. La segunda estrategia se basa en técnicas de clasificación, por lo que estaría relacionado con el tercer paradigma. Y, por último, la tercera estrategia estaría basada en la utilización de patrones léxicos-sintácticos, lo que se relaciona con el primer paradigma.

Hasta aquí, todos los niveles del proceso de aprendizaje de ontologías que se encuentran incluidos en la aproximación de evolución de ontologías que se describe en esta tesis. Los demás niveles, relaciones y reglas no serán analizados. A continuación se analiza la fase de refinamiento de ontologías que tiene por objetivo la actualización y el enriquecimiento de ontologías.

I.3.4.3. Refinamiento de ontologías

Entre las diferentes fases que componen el proceso de evolución de ontologías, es posible destacar la fase de refinamiento, que cumple un rol similar a la fase de extracción de información, donde las principales partes de la ontología utilizada son modeladas a través de la información extraída de los documentos Web. La fase de refinamiento trata de afinar el objetivo de la ontología y proporciona un soporte a su naturaleza evolutiva (Maedche & Staab, 2001). Las fases de extracción y de refinamiento cumplen el mismo rol, en ambas pueden utilizarse los mismos algoritmos, con la diferencia de que el proceso de refinamiento debe considerar los elementos y las conexiones existentes en la ontología, mientras que la fase de extracción no tiene por qué considerar estos aspectos.

En su tesis doctoral, Stojanovic (2004) establece que dentro de la evolución de ontologías se pueden dar dos tipos de cambios: “top-down” y “bottom-up”. Los cambios “top-down” están caracterizados por cambios explícitos dirigidos normalmente por ingenieros ontológicos. Aquí se encuentran los cambios que se

producirían en una ontología para adaptarla a nuevos requisitos, modificaciones en el dominio de la aplicación, la incorporación de nuevas funcionalidades, nuevas necesidades por parte de los usuarios, etc. Por otro lado, los cambios "bottom-up" son cambios que se originan en los repositorios de información que afectan implícitamente al dominio de la ontología. Ambos tipos de cambios se corresponden con métodos de adquisición de conocimiento. El cambio "top-down" se parece al método de aprendizaje deductivo, debido a que utiliza técnicas bien definidas de obtención de conocimiento directo de los expertos humanos y de los usuarios finales. Por otro lado, los cambios "bottom-up" se parecen a los métodos de aprendizaje inductivo, que emplean técnicas propias de máquinas de aprendizaje que utilizan conjuntos de ejemplos para inferir patrones y obtener nuevo conocimiento. La Figura I.10 representa gráficamente los dos tipos de cambios explicados anteriormente dentro de un sistema basado en ontologías.

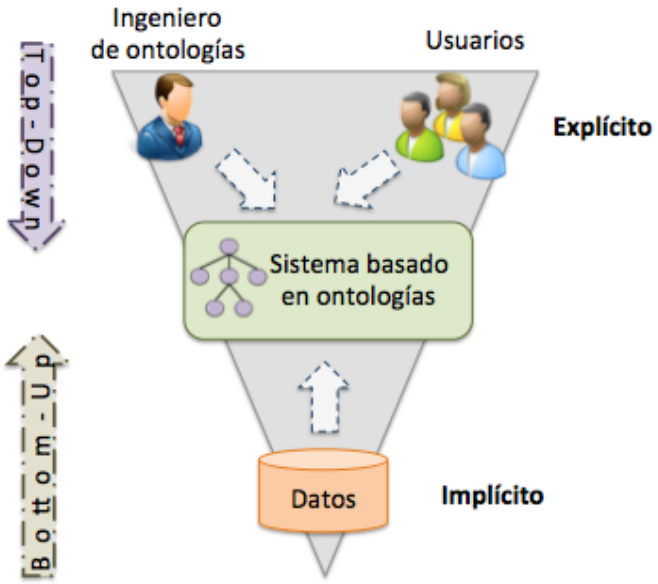


Figura I.10 Tipos de cambio en la evolución de ontologías

I.3.4.4. Instanciación automática de ontologías (Ontology Population)

I.3.4.4.1. Definición

La instanciación automática de ontologías (del inglés, '*Ontology Population*') permite evolucionar las ontologías mediante la adquisición de nuevas

descripciones semánticas de datos extraídas desde fuentes de información heterogéneas (Castano et al., 2008). Más concretamente, la instanciación automática de ontologías es el proceso de insertar y relacionar instancias en una ontología definida sin cambiar la estructura de la ontología, es decir, la jerarquía de conceptos y las relaciones no taxonómicas no se modifican (Petasis et al., 2011). Como se ha comentado anteriormente, la instancia de un concepto es una realización del concepto en el dominio. El proceso de instanciación de ontologías no cambia la estructura de la ontología.

La instanciación automática de ontologías requiere de una ontología inicial que será instanciada y un motor de extracción de instancias que se responsabiliza de localizar instancias de conceptos y relaciones en corpus multimedia para poblar la ontología (Ruiz-Martínez et al., 2012).

I.3.4.4.2. Clasificación de técnicas en Ontology Population

Según Tanev y Magnini (2008) existen dos formas de clasificar los métodos de instanciación automática de ontologías. Estas clasificaciones se basan en los datos de entrenamiento utilizados o en las distintas técnicas de extracción empleadas.

En el primer tipo de clasificación, existen principalmente dos enfoques diferentes, a saber, aplicar técnicas no supervisadas o aplicar técnicas supervisadas. Mientras que las técnicas no supervisadas tienen un bajo rendimiento, las técnicas supervisadas alcanzan una precisión alta debido a la utilización de datos de entrenamiento anotados. El inconveniente principal de estos métodos es que requieren de la construcción manual de la base de entrenamiento, por lo que limita su utilización en aplicaciones de gran escala.

El segundo tipo de clasificación distingue entre dos enfoques principales. El primer enfoque está basado bien en la utilización de patrones (Hearst, 1998) o bien en el análisis de la estructura de términos (Velardi et al., 2005). Mientras tanto, el segundo enfoque está más centrado en el análisis de las características textuales (Cimiano & Völker, 2005).

Las técnicas basadas en patrones han sido diseñadas para buscar en frases que explícitamente muestren la existencia de una relación “IS-A” entre dos palabras como, por ejemplo, “la hormiga es un insecto” o “las hormigas y otros insectos”. El

problema de estas técnicas es que sólo funcionan para este tipo de sentencias, que no son muy utilizadas en los corpus. Por otro lado, una relación "IS-A" puede representar una relación entre subclases o bien que una instancia pertenece a una clase dada. Por estas razones, es necesario incorporar nuevas técnicas que amplíen y refinen los resultados obtenidos por las técnicas basadas en patrones (Velardi et al., 2005);(Schlobach et al., 2004).

Por otro lado, existen otras técnicas que utilizan un corpus para extraer características a partir del contexto en el que una clase semántica tiende a aparecer. Las características pueden ser superficiales (Fleischman & Hovy, 2002) o sintácticas (Almuhareb & Poesio, 2004);(Lin, 1998). Algunos ejemplos de características superficiales son: frecuencia de palabra, tópicos exclusivos o características como hiperonimia o sinonimia definidas en Wordnet (Fellbaum, 2005). Las características sintácticas se refieren, fundamentalmente, a las morfológicas y gramaticales. Cimiano y Völker (2005) realizan una evaluación comparativa donde las características sintácticas muestran un mejor rendimiento que las superficiales. También existen técnicas híbridas que mezclan todos los enfoques analizados anteriormente (Cimiano et al., 2003).

I.4. PROCESAMIENTO DEL LENGUAJE NATURAL (PLN)

I.4.1. DEFINICIÓN

El lenguaje natural puede definirse como el medio oral o escrito que los humanos utilizan para propósitos generales de comunicación. El término "procesamiento del lenguaje natural" abarca un amplio conjunto de técnicas para la generación automática, manipulación y análisis de textos en lenguaje natural. Más concretamente, el procesamiento del lenguaje natural es un área de investigación y aplicación que explora cómo los ordenadores pueden ser usados para entender el significado del lenguaje que usamos los humanos para comunicarnos.

Liddy (2001) propone la siguiente definición: "*El procesamiento del lenguaje natural es 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".

Esta definición destaca la delicada labor que desempeñan estos sistemas, que necesitan de diferentes técnicas para llevar a cabo su misión. Además, resalta la capacidad de representación de estos sistemas en diferentes niveles de análisis lingüísticos que serán analizados en los siguientes apartados.

Chowdhury (2003) define el procesamiento del lenguaje natural 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*".

Ambas definiciones establecen el objetivo del procesamiento del lenguaje natural, que se centra en el desarrollo de técnicas y construcción de herramientas que permitan a los sistemas informáticos entender y manipular los lenguajes naturales de tal manera que se facilite la comunicación entre seres humanos y sistemas informáticos.

I.4.2. ANTECEDENTES

La investigación en el campo del procesamiento del lenguaje natural tiene su comienzo hace varias décadas, a finales de 1940, donde la traducción automática fue la primera aplicación informática que relacionaba el mundo de los ordenadores con el lenguaje natural. Durante esta década y hasta finales de los años 50 se realizó un trabajo intenso sobre dos paradigmas fundamentales: los autómatas y los modelos probabilísticos. Este trabajo dio lugar a numerosos avances tecnológicos, comenzando por la primera utilización de los ordenadores para manipular los lenguajes naturales para intentar automatizar la traducción entre inglés y ruso (Locke & Booth, 1956). También en esta época se desarrolló el primer autómata según el modelo de computación algorítmica de Turing, que fue utilizado por McCulloch y Pitts (1943) como base para desarrollar, en 1943, un simplificado modelo de neuronas como tipo de elemento computacional descrito utilizando la lógica proposicional. Este modelo neuronal dio lugar al trabajo de autómatas finitos de Kleene (1951) y, posteriormente, a su trabajo en expresiones regulares

(Shannon et al., 1956). En 1948, Shannon definió la teoría de autómatas (Shannon, 1948) y aplicó la teoría de la probabilidad de procesos de Markov para definir sistemas discretos, parecidos a los autómatas finitos, que procesaran el lenguaje humano. Chomsky (1956) consideró las máquinas de estados finitos para caracterizar las gramáticas y definió un lenguaje de estados como lenguaje generado por una gramática de estados finitos. Estos modelos iniciales llevaron a la teoría del lenguaje formal a incorporar el álgebra y la teoría de conjuntos para definir lenguajes formales como secuencias de símbolos. En este trabajo, Chomsky incluye la renombrada jerarquía de gramáticas formales que generan lenguajes formales (Chomsky, 1956). Esta jerarquía se compone de cuatro niveles: (i) gramáticas de tipo 0, donde se incluyen todas las gramáticas formales; (ii) gramáticas de tipo 1, que recoge las gramáticas sensibles al contexto que generan los lenguajes sensibles al contexto; (iii) gramáticas de tipo 2, donde se sitúan las gramáticas libres de contexto que generan lenguajes independientes del contexto; y, por último, (iv) gramáticas de tipo 3, donde se sitúan las gramáticas regulares que generan los lenguajes regulares. Otro elemento fundamental desarrollado durante esta época fueron los algoritmos de probabilidad y de procesamiento del lenguaje, también aportación de Shannon (1948).

A finales de la década de 1950 y principios de 1960, el procesamiento del lenguaje natural se dividió en dos paradigmas: simbólico y estocástico. El paradigma simbólico se originó a partir de dos líneas de investigación. La primera línea de investigación se basó en las obras de Chomsky y sus colegas sobre la teoría del lenguaje formal y la gramática generativa (Chomsky, 1956);(Moore, 1956). La segunda línea de investigación fue una nueva área de investigación conocida con el nombre de "inteligencia artificial", nombre ideado por un grupo de investigadores compuesto por John McCarthy, Marvin Minsky, Claude Shannon, y Nathaniel Rochester durante el verano de 1955 (McCarthy et al., 1955). Aunque al principio, el número de investigadores en este campo era pequeño, el trabajo realizado se enfocó en el estudio de algoritmos estadísticos y estocásticos que incluían modelos probabilísticos y redes neuronales. El principal objetivo de este nuevo campo de investigación fue el trabajo en razonamiento y lógica tipificado en los trabajos de Newell y Simon llamados "Logic Theorist" (Newell & Simon, 1956) y "The General Problem Solver" (Simon et al., 1959). En este punto histórico, se destaca el

desarrollo de sistemas capaces de entender el lenguaje natural, sistemas simples que trabajaban en dominios concretos que utilizaban combinaciones de patrones y búsquedas basadas en palabras clave con simples heurísticas y técnicas de pregunta-respuesta.

Por otro lado, el paradigma estocástico fue seguido por departamentos de estadística e ingeniería eléctrica. A finales de los años 50, el método bayesiano comenzaba a aplicarse en los problemas de reconocimiento óptico de caracteres. Un ejemplo sería el trabajo presentado por Bledsoe y Browning, en 1959 (Bledsoe & Browning, 1959), donde se describe un sistema bayesiano de reconocimiento de texto. Otro ejemplo fue el trabajo de Mosteller y Wallace, en 1964 (Mosteller & Wallace, 1964), donde se aplicaron métodos bayesianos para el problema de la atribución de autoría en artículos.

En el siguiente periodo (1970-1983) los investigadores dividieron aún más el área de investigación abarcando nuevas áreas con más tecnología y conocimiento disponible. El paradigma estocástico desempeñó una labor muy importante en el desarrollo de algoritmos de reconocimiento de voz, particularmente con el uso del modelo oculto de Markov y los teoremas de codificación de canal ruidoso y decodificación desarrollados, independientemente, por Jelinek, Bahl, Mercer, y colegas del Centro de Investigación IBM Thomas J. de Watson, y Baker de la Universidad Carnegie Mellon (Bahl & Mercer, 1976);(Baker, 1979);(Jelinek et al., 1975). El paradigma lógico desarrollado por Colmerauer y sus colegas en Q-systems y gramáticas metamorfosis (del inglés, '*metamorphosis grammars*') (Colmerauer, 1975) fue el primer formalismo gramatical basado en las cláusulas de Horn. En el campo de la comprensión del lenguaje natural, aunque estuvo inactivo durante esta época, apareció el sistema de Terry Winograd, "SHRDLU", que simulaba un robot integrado en un mundo de bloques que aceptaba comandos de texto en lenguaje natural (Winograd, 1972). Por último, los paradigmas basados en lógicas y en la comprensión del lenguaje natural, se unificaron en sistemas que utilizaban la lógica de predicados como representación semántica. Un ejemplo de estos entornos es el sistema LUNAR (Woods, 1973).

El siguiente periodo (1983-1993) se centró en recuperar varias tendencias de los años 50 y 60 que recibieron numerosas críticas. La primera tendencia fue el modelo de estados finitos, que comenzó a recibir atención después de introducir

este modelo en el estudio de la fonología de estados finitos (Kaplan & Kay, 1981) y la morfología de los modelos de estados finitos de la sintaxis de Church (1980). La segunda tendencia fue lo que se conoce como el "retorno del empirismo", donde comenzaron a surgir métodos y enfoques probabilísticos de reconocimiento de voz y de procesamiento del lenguaje.

En los últimos años el área de investigación de procesamiento del lenguaje natural ha cambiado bastante. En primer lugar, los modelos probabilísticos y los modelos conducidos por datos se encuentran bastante estandarizados en todo el procesamiento del lenguaje natural. Los algoritmos de análisis, etiquetado gramatical, resolución de referencia y discurso han comenzado a incorporar modelos probabilísticos y a emplear metodologías de evaluación en el reconocimiento de voz y recuperación de la información. Estos años también se caracterizan por el incremento del nivel de procesamiento y la rapidez de los ordenadores, lo que ha permitido la explotación de diferentes áreas del procesamiento del lenguaje natural como el reconocimiento de la voz y la detección de errores ortográficos y gramaticales. Por último, el auge de la Web ha resaltado la necesidad de innovación en los procesos de recuperación y extracción de información de los sistemas de almacenamiento.

I.4.3. NIVELES DEL PROCESAMIENTO DEL LENGUAJE NATURAL

Según Liddy y Feldman (1999) es importante ser capaz de distinguir entre los siete niveles interdependientes que los humanos utilizan para extraer el significado del lenguaje natural. La Figura I.11 muestra un cuadro resumen que relaciona los diferentes niveles del lenguaje con cada una de las unidades lingüísticas que se procesan en esos niveles y, además, se incluyen las herramientas utilizadas en cada nivel.

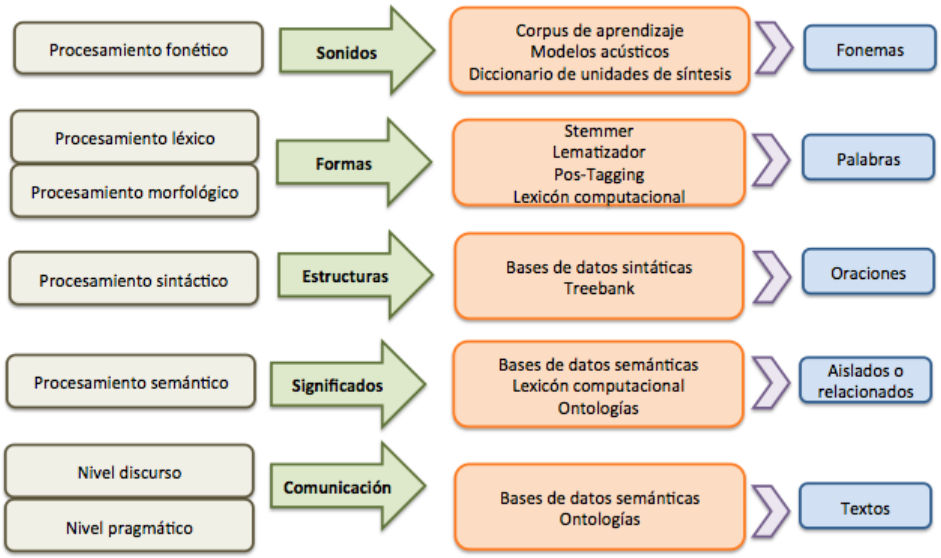


Figura I.11 Cuadro resumen de los niveles del lenguaje humano

Las fases o niveles de procesamiento del lenguaje no son niveles aislados sino que se encuentran interrelacionados. En concreto, conforme se avanza en el procesamiento existen niveles que, para desarrollar su proceso de análisis, requieren conocimiento de niveles anteriores, o incluso posteriores. Por ejemplo, para los procesos de desambiguación de las categorías morfológicas es necesario recurrir a la sintaxis para determinar qué función desempeña un término en una oración. En los siguientes apartados se analiza cada nivel del lenguaje, explicando las técnicas más importantes que se utilizan para llevar a cabo el análisis correspondiente al nivel.

I.4.3.1. Nivel fonológico

La fonética se encarga de describir las dimensiones físico-acústicas, articulatorias y auditivas de los sonidos en el lenguaje. A este nivel se encuentran tres tipos de reglas que se utilizan para realizar el análisis fonológico: (i) reglas fonéticas, para analizar los sonidos producidos por los humanos cuando dicen una palabra; (ii) reglas fonológicas, para detectar las variaciones de pronunciación producidas cuando las palabras se hablan unidas; y (iii) reglas prosódicas, utilizadas para analizar la fluctuación del acento y la entonación de los humanos cuando enuncian una sentencia. Normalmente, el funcionamiento de un sistema de procesamiento fonológico acepta como entrada la voz humana, y analiza y codifica

las ondas producidas en una señal digitalizada para interpretar varias reglas o para establecer comparaciones con otro modelo de lenguaje.

I.4.3.2. Nivel morfológico

La morfología estudia la estructura de la forma de las palabras a través de los morfemas, flexivos y derivativos, las unidades más pequeñas de significado que componen las palabras. Para ello, los sistemas de procesamiento morfológicos transforman cada secuencia de caracteres en una secuencia de morfemas, mediante la utilización de diferentes técnicas como stemmers, lematizadores, o POS-Taggers. Con estas técnicas es posible determinar aspectos como tiempo, género, número, grado, etc. y buscar sufijos, prefijos, sinónimos, generalizaciones, o especializaciones. También permiten clasificar las unidades lingüísticas en categorías gramaticales (p. ej., sustantivo, verbo, adjetivo, adverbio, etc.). Por ejemplo, si se analiza la palabra "*desarrollo*" morfológicamente, se obtendrían tres morfemas separados: "*des*" como prefijo, "*arroll*" como raíz y "*o*" como sufijo. Los seres humanos son capaces de descomponer una palabra desconocida en sus morfemas constituyentes para entender el significado. Del mismo modo, los sistemas de procesamiento morfológico pueden reconocer el significado de cada morfema para obtener el significado total de la palabra.

I.4.3.3. Nivel léxico

Este nivel se centra en la interpretación del significado de las palabras de manera individual. Los sistemas de procesamiento del lenguaje natural requieren de un léxico que les permita conocer las categorías léxicas existentes en el lenguaje. En este nivel existen diferentes técnicas que permiten el análisis léxico de la palabra. La técnica más extendida se basa en la asignación de etiquetas individuales a cada palabra del texto. Estas etiquetas determinan la categoría léxica de la palabra y, en el caso de que una palabra pueda desempeñar diferentes funciones dentro de la sentencia, será etiquetada con la categoría léxica más probable en función del contexto. El proceso de análisis léxico finaliza cuando todas las palabras del texto han sido etiquetadas.

I.4.3.4. Nivel sintáctico

La sintaxis se centra en estudiar las relaciones que se establecen entre las palabras dentro de una oración y las reglas que rigen estas oraciones con el fin de descubrir la estructura gramatical de la frase y analizar cómo las palabras se mezclan para componer oraciones gramaticalmente correctas. Este proceso de análisis requiere contar con los patrones sintácticos más frecuentes del lenguaje que se esté analizando y un analizador sintáctico que se encargue de obtener la estructura gramatical de la secuencia de unidades léxicas para representarla en forma de árbol o red. Por lo tanto, 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.

I.4.3.5. Nivel semántico

La semántica estudia el significado del lenguaje. El objetivo del procesamiento semántico es determinar los posibles significados de una frase, centrándose en las relaciones de significados a nivel de palabra. Este nivel de procesamiento puede requerir de técnicas de desambiguación semántica para aquellas palabras que tengan varios significados. La desambiguación semántica permite seleccionar y representar un solo significado de palabras polisémicas. Si tomamos la palabra "carpeta" como sustantivo, esta palabra puede tener varios significados, pudiendo referirse a una carpeta para almacenar documentos o una carpeta que se utiliza en los sistemas informáticos para almacenar información. Actualmente, existen varios métodos que pueden ser implementados en los sistemas de procesamiento semántico para eliminar las ambigüedades. Algunos de estos métodos precisan de información relacionada con la frecuencia en que aparece cada tópico de interés en el corpus y otros utilizan el conocimiento pragmático del dominio del documento.

I.4.3.6. Nivel del discurso o contextual

En los niveles sintáctico y semántico, los sistemas de procesamiento del lenguaje natural trabajan con unidades de longitud de sentencia, mientras que en el nivel del discurso trabajan con unidades de texto más largas que una sentencia. Es decir,

en este nivel el sistema no interpreta los textos como conjuntos independientes de sentencias concatenadas, sino que se centra en analizar las propiedades del texto en su conjunto que transmiten significado al conectar entre las oraciones que lo componen. El objetivo de este nivel de procesamiento es utilizar la estructura semántica del nivel semántico para desarrollar una interpretación final de la oración en función de circunstancias del contexto. Para lograr este objetivo, existen varios tipos de procesamiento del discurso que pueden darse en este nivel, siendo los más conocidos la resolución anafórica y el reconocimiento de la estructura del texto/discurso. La resolución anafórica se basa en reemplazar palabras vacías semánticamente, como pronombres, por la entidad a la que referencian. El reconocimiento de la estructura del texto/discurso determina las funciones de las sentencias en el texto y añade esta representación significativa al texto.

I.4.3.7. Nivel pragmático

La pragmática estudia las estrategias comunicativas enmarcándolas en un contexto socio-cultural. Es uno de los niveles de análisis más complejos donde se analiza el texto más allá de los límites de la frase. El análisis que se realiza en este nivel se encuentra relacionado con los factores extralingüísticos que condicionan el uso del lenguaje en situaciones comunicativas concretas. Precisamente todos aquellos factores extralingüísticos que no pueden ser analizados por los niveles anteriores, como la intención comunicativa, el contexto verbal, y la situación o conocimiento del mundo, resultan de gran importancia en este nivel. En este análisis se utiliza el contexto, por encima de los contenidos, para comprender el significado del texto. Para lograr 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.

I.5. EXTRACCIÓN Y RECUPERACIÓN DE INFORMACIÓN

I.5.1. DEFINICIONES

Actualmente, una creciente cantidad de información en lenguaje natural se encuentra disponible en formato electrónico online. La necesidad de procesar de

forma inteligente esta información hace de la extracción de información (IE, del inglés, "*Information Extraction*") una tarea fundamental ya que permite localizar partes específicas de datos de un documento en lenguaje natural (Califf & Mooney, 1999).

Jim Cowie y Yorick Wilks (1996) definen la extracción de información como "*cualquier proceso que selectivamente organiza y combina los datos que se encuentran de manera implícita o explícita en uno o más textos*". Estos autores también establecen la diferencia entre los sistemas de recuperación de información (IR, del inglés, "*Information Retrieval*"), que seleccionan un subconjunto relevante de documentos de un conjunto más amplio, y los sistemas de extracción de información (IE), que extraen información del texto de los documentos.

Por otro lado, Cunningham (2005) define la extracción de información como "*una tecnología basada en el análisis del lenguaje natural para extraer trozos de información. El proceso toma como entrada textos y produce un formato fijo de datos inequívocos como salida*". Además, el autor en este trabajo hace una diferenciación entre IR e IE definiendo el sistema IR como aquel que encuentra textos relevantes y los presenta al usuario, mientras que los sistemas IE se encargan de analizar los textos presentando únicamente la información específica que interesa al usuario.

El objetivo de los sistemas de extracción de información es automatizar el proceso de búsqueda de todo tipo de información en una serie de documentos. Para ello, el sistema es capaz de extraer fragmentos de texto con significado relevante, ignorando los fragmentos irrelevantes que se emplean para estructurarlos, de tal forma que el ordenador sea capaz de entenderla y almacenarla en un sistema de almacenamiento, como una base de datos, para su futura explotación (Cowie & Lehnert, 1996).

En el caso de los sistemas de recuperación de información, el objetivo se centra en encontrar el material, generalmente documentos, de naturaleza no estructurada, generalmente texto, que satisface una necesidad de información desde dentro de grandes colecciones habitualmente almacenadas en ordenadores (Manning et al., 2008).

En los siguientes apartados se describirán los diferentes tipos de información y las técnicas que se emplean comúnmente por parte de los sistemas de extracción y recuperación de información.

I.5.2. TIPOS DE INFORMACIÓN

I.5.2.1. Introducción

En sentido general, la información es un conjunto organizado de datos procesados (Alazraqui et al., 2006). A partir de cómo estén organizados estos datos se puede distinguir diferentes tipos de información. Por ejemplo: si estos datos no presentan ningún orden ni estructura, la información que conforman los datos se cataloga como (i) información no estructurada. Si por el contrario, todos los datos presentan un orden y una estructura, la información que constituyen se clasifica como (ii) información estructurada. Por último, si dentro de este conjunto de datos, sólo algunos de estos datos presentan una estructura u organización, entonces en este caso se clasifica como (iii) información semiestructurada. En lo siguientes apartados se realizará una descripción detallada de cada uno de los tipos de información enumerados.

I.5.2.2. Información semiestructurada

La información estructurada hace referencia a documentos cuya estructura es declarada explícitamente de alguna manera como, por ejemplo, asociando etiquetas a los elementos de estructura o utilizando cualquier tipo de sintaxis como ocurre con los lenguajes de programación. Por otro lado, la información semiestructurada es un tipo de información que no se ajusta a una estructura formal de modelo de datos asociada a una base de datos relacional u otra forma de organización de información. Este tipo de información se caracteriza por definir un conjunto de etiquetas o marcadores que permiten separar elementos semánticos y forzar jerarquías de registros y campos con datos. Este tipo de información está continuamente creciendo en Internet y en las intranets de las organizaciones, donde los documentos de texto y las bases de datos no son los únicos almacenes de

información existentes. Cada vez más información está siendo almacenada como información semiestructurada como, por ejemplo, correos, música, películas, productos, blogs, etc. La mayor parte de la información semiestructurada se compone de información estructurada que, generalmente, se encuentra en formato HTML, XML, CSS, etc.

Varias definiciones se han propuesto en la literatura sobre información semiestructurada. Quizás la más relevante es la que aporta Peter Buneman (1997): "*En los datos semiestructurados, la información que se asocia normalmente a un esquema está contenida dentro de los datos, y a veces se conoce con el nombre de auto-descripción (self-describing). En algunas formas de datos semiestructurados no existe tal esquema y en otros existe pero sólo impone las restricciones de organización de los datos*". El autor explica que la condición de esquema no se cumple restrictivamente para todos los tipos de información semiestructurada, es decir, que no todos los datos semiestructurados necesariamente requieren de este esquema de auto-descripción. Además, resalta que algunos datos semiestructurados utilizan este esquema para definir la posición que los datos ocupan.

Otra definición a destacar es la propuesta por Nevill-Manning y sus colegas (1996), a saber: "*Caracterizamos un texto semiestructurado como datos que son destinados a un procesamiento automático, pero que además son legibles por los humanos*".

La información semiestructurada se utiliza para diversas tareas, entre las que se pueden resaltar las siguientes (Buneman, 1997): realizar operaciones de base de datos sobre los datos, intercambiar datos entre diferentes bases de datos o navegar por datos estructurados. Para la información semiestructurada existen diversos modelos de representación (Zhang, 2013). El modelo más básico de representación muestra la información semiestructurada en forma de árbol, donde cada nodo se corresponde con un elemento de la estructura del documento a representar. Esta representación básica permite ver cada documento mediante un conjunto de pares aspecto-valor, donde cada aspecto se corresponde con un elemento en el nivel más bajo de la estructura del documento y el valor se corresponde al valor particular del elemento correspondiente. Siguiendo este modo de representación de información semiestructurada, un documento de investigación podría ser

representado de la siguiente forma: "*Titulo: Sistema de anotación semántica*", "*Fecha: 10 de junio de 2014*", "*Sección: Estado del arte*", "*Descriptor: Anotación semántica, ontologías y procesamiento del lenguaje natural*", etc.

Además de este modo básico de representación de información semiestructurada, existen otros que se analizarán a continuación en diferentes apartados.

I.5.2.2.1. Faceta textual

En este tipo de representación el valor, por lo general, es un fragmento de texto que puede incluir el título, encabezado, y un resumen de un documento de investigación. La representación de faceta textual se lleva a cabo mediante la utilización de modelos "bolsa de palabras" (*bag-of-words*) o N-gramas (subsecuencia de elementos de una secuencia dada).

I.5.2.2.2. Faceta nominal

En este tipo de representación de la información el valor es un término predefinido del vocabulario del documento. Ejemplos de este tipo de representación pueden ser la localización, los editores, personas implicadas en la noticia, objeto de investigación, género, etc. En este tipo de representación cada faceta-valor (FVP) se trata como una característica a la que se le asigna un valor '1' o '0' que denota si esta FVP aparece en el documento representado o no. Es decir, si se dispone del conjunto FVP "*Autor: Miguel Ángel Rodríguez-García*", "*Título: Ontología*", aplicando el método de faceta nominal para la representación de la información relativa a un documento dado, se le asignará un 1 o un 0 en función de si el autor es "Miguel Ángel", o el título es "Ontología".

I.5.2.2.3. Faceta numérica

En este tipo de representación de información semiestructurada el valor que se utiliza es un número que representa conceptos relacionados con el precio del producto, o el número de veces comentado (en el caso de ser, por ejemplo, un video, noticia o una película). Cada número representa una faceta distinta y es

tratada como una característica individual. Los valores numéricos pueden ser transformaciones/normalizaciones de los valores obtenidos durante el proceso de representación.

I.5.2.3. Información no estructurada expresada en lenguaje natural

La información no estructurada es un tipo de información que no tiene un modelo de datos predefinido (Chen et al., 2012), es decir, los datos no presentan ningún tipo de estructura organizativa reconocible. Normalmente, la información no estructurada representa el contenido generado y orientado para el ser humano que no encaja perfectamente en los sistemas de almacenamiento relacional.

La gente utiliza datos no estructurados todos los días (Weglarz, 2004). Aunque ellos no son conscientes, los utilizan para crear, almacenar, y recuperar documentos, e-mails, hojas de cálculo y otros tipos de documentos.

I.5.3. TAREAS DE LA EXTRACCIÓN DE INFORMACIÓN

I.5.3.1. Reconocimiento de Nombres de Entidades

El término "nombre de entidad" es ampliamente utilizado en los sistemas de extracción de conocimiento y fue acuñado en la sexta edición de las conferencias MUC (*Message Understanding Conference*) por Grishman y Sundheim (1996). Las MUC eran congresos financiados por DARPA (*Defense Advanced Research Project Agency*) y que estaban diseñados para promover y evaluar la investigación en el área de extracción de información. Hubo hasta siete ediciones de este congreso, desde la inicial, MUC-1 en 1987, hasta la última edición, MUC-7 en 1997. Concretamente, la MUC-6 aportó varias innovaciones importantes con respecto a las anteriores ediciones. En esta edición fueron reconocidos los tipos de entidades más estudiadas, como los nombres propios, lugares y organizaciones.

El Reconocimiento de Nombres de Entidades (RNE o NER) se puede definir como la tarea que se encarga de clasificar cada palabra de un documento en un conjunto de categorías predefinidas (Zhou & Su, 2002). Esta definición remarca la labor de identificación y clasificación que desarrolla la tarea de reconocimiento de nombres

de entidades dentro de los sistemas de extracción de información. La tarea de reconocimiento de nombres de entidades es bastante complicada de llevar a cabo, si se consideran los problemas asociados a la ambigüedad del lenguaje natural y la existencia de fenómenos lingüísticos como la metonimia, polisemia o elipsis.

En la literatura existen diferentes clasificaciones sobre los enfoques que se han desarrollado para resolver el problema de reconocer nombres de entidades. En el trabajo realizado por Nadeu y Skine (2007) se presenta un estudio de casi dos décadas de investigación en este campo, donde se describe la evolución de esta tarea desde los inicios en 1991, cuando Lisa F. Rau publica en "*The Seventh IEEE Conference on Artificial Intelligence Applications*" un artículo en el que se detalla un sistema capaz de extraer y reconocer nombres de compañías (Rau, 1991), hasta HAREM, sistema de reconocimiento de nombres de entidades para el lenguaje portugués (Santos et al., 2006). En el estudio de Nadeu y Skine también se analizan las tendencias más representativas que han seguido las técnicas de reconocimiento de nombres de entidades. Además, también describen los diferentes métodos de aprendizaje, supervisado, semisupervisado y no supervisado, utilizados por esta técnica y que se describen brevemente a continuación.

El aprendizaje supervisado es el método actual dominante para solucionar el problema de reconocimiento de entidades. Este tipo de aprendizaje incluye diferentes técnicas entre las que destacamos los modelos ocultos de Markov (Bikel et al., 1997), los árboles de decisión (Sekine, 1998), los modelos de máxima entropía (Borthwick et al., 1998), las máquinas de vectores de soporte (Asahara & Matsumoto, 2003) y los campos aleatorios condicionales (McCallum & Li, 2003). De forma genérica, todas estas técnicas consisten en un sistema que requiere como entrada un conjunto de textos anotados para, entonces, memorizar esta lista de entidades y crear reglas de desambiguación basadas en características discriminatorias.

Otro método de aprendizaje utilizado y que es relativamente reciente es el método de aprendizaje semisupervisado, también conocido como método débilmente supervisado (Nadeau & Sekine, 2007). La principal técnica de este método se llama "bootstrapping" y necesita un cierto grado de supervisión. Para comenzar el proceso de aprendizaje, este método requiere de una pequeña cantidad de ejemplos etiquetados y una cantidad considerable de muestras sin

etiquetar para evaluar el método propuesto. Durante la década de los noventa se presentaron diferentes propuestas que utilizan este método de aprendizaje. Por ejemplo, la propuesta de Brin (1999) utilizó rasgos léxicos implementados a través de expresiones regulares para generar listas de títulos de libros relacionados con sus autores. La propuesta Collins y Singer (1999) analizaba un corpus completo en búsqueda de patrones que identificaran candidatos de nombres de entidades. La propuesta de Riloff y Jones (1999) introduce un "bootstrapping" mutuo que consiste en el crecimiento de un conjunto de entidades y un conjunto de contextos a la vez. La propuesta de Cucchirelli y Velardi (2001) utilizaba relaciones sintácticas para descubrir evidencias contextuales más precisas alrededor de las entidades. Por último, la aproximación de Pasça (2007) también empleaba técnicas basadas en "bootstrapping" mutuo, pero esta vez incorporando los conceptos de distribución semántica propuestos por Lin (1998).

Finalmente, el método de aprendizaje no supervisado se caracteriza por utilizar técnicas de clustering que facilitan la recolección de nombres de entidades en clústeres agrupados por similitud de contexto. Este método también incluye otras técnicas que utilizan recursos léxicos como Wordnet (Fellbaum, 2005), patrones léxicos y estadísticas calculadas sobre grandes corpus no anotados. El objetivo principal de estas técnicas es construir representaciones de los datos para descubrir patrones o tendencias en éstos que faciliten la identificación y clasificación de entidades. Entre las propuestas relacionadas con el aprendizaje no supervisado podemos destacar el trabajo de Alfonseca y Manandhar (2002) en el que se estudia el problema de etiquetar una palabra de entrada con un tipo apropiado de nombre de entidad. Evans (2003) utilizó el método de identificación de hiperónimos desarrollado por Hearst (1992) para identificar hiperónimos potenciales en las palabras que aparecen en un documento. Shinyama y Sekine (2004) definieron una técnica que permite la identificación de nombres de entidades de forma no supervisada y que puede ser utilizada con otras técnicas de reconocimiento. Por último, Etzioni y sus colegas (2005) desarrollaron un algoritmo que proporciona una función que permite evaluar si una entidad con nombre puede clasificarse en un tipo determinado.

El análisis de resultados de los enfoques supervisados frente a los no supervisados establece que los sistemas basados en algoritmos de aprendizaje

supervisado obtienen mejores resultados que los sistemas basados algoritmos de aprendizaje no supervisado (Tjong Kim Sang, 2002). Sin embargo, dada la abundante disponibilidad de textos no etiquetados se han retomado los métodos semisupervisados (Solorio, 2005).

En el trabajo presentado en (Álvarez Silva et al., 2009) se establece que no solo existe este enfoque basado en técnicas de aprendizaje sino que existe otro que se caracteriza por la utilización de la ingeniería del conocimiento. Estos sistemas están orientados a trabajar con un idioma y dominio concreto. Las técnicas más utilizadas por este enfoque se basan en expresiones regulares y reglas heurísticas que permiten definir la sintaxis de las expresiones que constituyen los nombres de las entidades. Estas heurísticas se definen para establecer un orden de evaluación de las expresiones regulares de forma que se evite cualquier contradicción existente. Otra técnica que se utiliza en este enfoque es el conjunto de reglas especificadas como programas en lenguajes de programación lógicos o funcionales, donde es necesario establecer mecanismos de prioridad al igual que en la técnica de expresiones regulares. Por último, conviene resaltar la técnica que utiliza colecciones de nombres conocidos, también llamados gazetteers, que se procesan manualmente o de forma semiautomática.

I.5.3.2. Extracción de términos

Una tarea muy extendida en la extracción de información es la extracción de términos, que ya se ha explicado anteriormente en el apartado I.3.4.2.1 en el contexto del modelo de capas del aprendizaje de ontologías.

I.5.4. TÉCNICAS DE RECUPERACIÓN DE INFORMACIÓN

I.5.4.1. Clasificación de modelos de recuperación

Según Belkin y Croft (1987) los modelos de recuperación de información se pueden clasificar en dos categorías: los modelos de coincidencia exacta (del inglés, '*exact-match*') y los modelos de coincidencia parcial (del inglés, '*partial-match*'). Las técnicas de coincidencia exacta se caracterizan porque el modelo de solicitud

utilizado para recuperar información tiene que contener exactamente la misma información que la existente en el texto de los documentos. En cambio, las técnicas de coincidencia parcial utilizan representaciones de documentos basadas en conjuntos de características o términos índice representativos, por lo que no requieren de esa correspondencia exacta entre la información solicitada y la existente en los documentos. Esta representación de documentos favorece la utilización de consultas, no sólo basadas exclusivamente en contenidos específicos de documentos, sino que además facilita la utilización de consultas en la que los términos utilizados pueden ser derivados de una consulta expresada en lenguaje natural, o incluso, puede utilizar un vocabulario de indexación para consultas más específicas.

La Figura I.12 muestra la jerarquía de modelos de recuperación de información propuesta por Belkin y Croft (1987). Ambas categorías comentadas constituyen los niveles jerárquicos más altos de clasificación de modelos de recuperación de información. Por lo tanto, todos los demás modelos de recuperación se clasifican basándose en estas dos categorías.

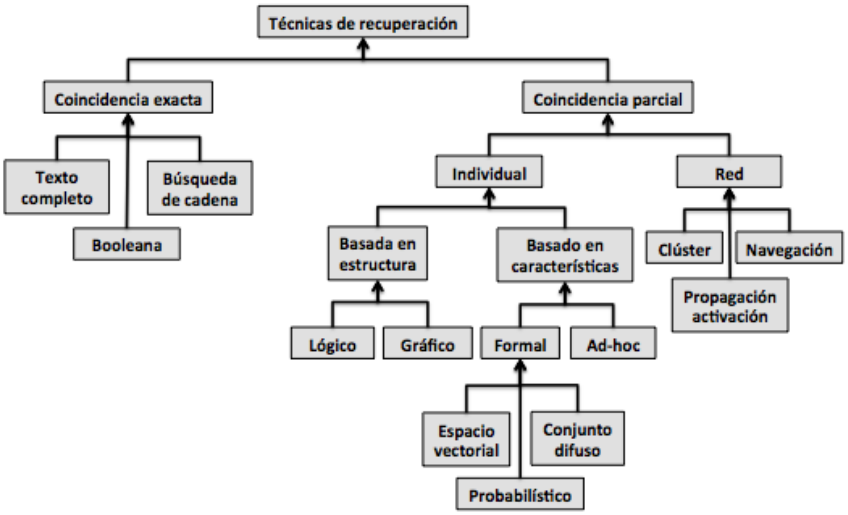


Figura I.12 Clasificación modelos de recuperación de información (Belkin & Croft, 1987)

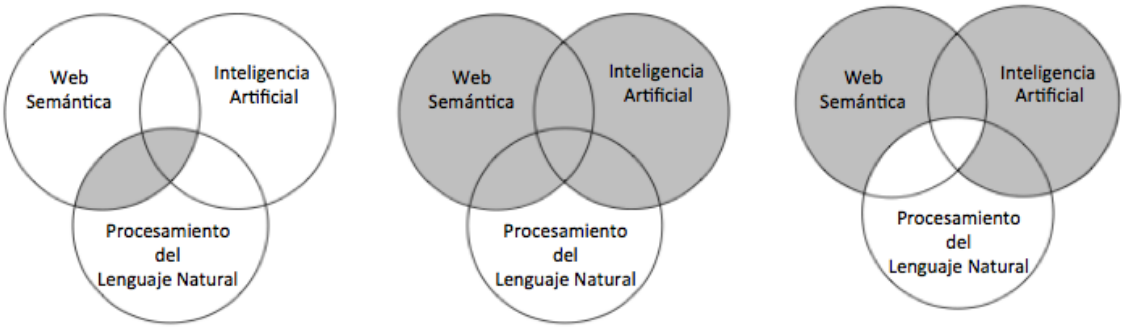
En los siguientes apartados se describen los modelos de recuperación de información más relevantes de esta clasificación.

I.5.4.2. Modelos de coincidencia exacta (Exact-Match)

I.5.4.2.1. Modelo booleano

El modelo booleano es el primer modelo de recuperación de información y, probablemente, el más criticado. Este modelo utiliza los términos de las consultas como definiciones inequívocas en agrupaciones de documentos, es decir, la consulta con el término "Web Semántica", utilizando este modelo, resultaría en el conjunto de documentos que han sido indexados exclusivamente con el término "Web Semántica". Este modelo permite la utilización de los operadores de la lógica matemática de George Boole (1854) para consultar varios términos y recuperar los conjuntos de documentos correspondientes. La aplicación de los operadores lógicos permite combinar distintos conjuntos de documentos para generar uno nuevo que cumpla las restricciones establecidas en la consulta. George Boole definió tres operadores básicos, a saber, '*AND*', '*OR*' y '*NOT*', esto es, el producto lógico, la suma lógica y la diferencia lógica, respectivamente.

La Figura I.13 muestra gráficamente el funcionamiento de estos operadores lógicos sobre tres conjuntos de documentos que contienen los términos "Web Semántica", "Inteligencia Artificial" y "Procesamiento del Lenguaje Natural". En la Figura I.13 las agrupaciones coloreadas representan gráficamente la recuperación de los documentos en función del operador lógico aplicado. La representación gráfica de la derecha, parte c), representa una combinación más compleja de operadores con el fin de denotar la riqueza lógica que permite este modelo de recuperación de información.



a) Web Semántica AND Procesamiento del Lenguaje Natural

b) Web Semántica OR Inteligencia Artificial

c) (Web Semántica OR Inteligencia Artificial) AND NOT (Web Semántica AND Procesamiento del Lenguaje Natural)

Figura I.13 Diagrama de Venn de los conjuntos resultantes de la aplicación de operadores lógicos booleanos

Entre las ventajas que proporciona este modelo de recuperación de información es posible destacar su fácil implementación, su alta eficiencia computacional y su alta expresividad mitigando cualquier problema relacionado con la ambigüedad propia del lenguaje natural. Por otro lado, su principal desventaja es que no proporciona una clasificación de los documentos recuperados.

Este modelo ha servido como base para el desarrollo de otros modelos como, por ejemplo, el modelo de regiones (Jaakkola & Kilpeläinen, 1999), el modelo booleano inteligente (Fox & Koll, 1988), o el modelo booleano extendido (Salton et al., 1983), entre otros.

I.5.4.3. Modelos de coincidencia parcial (Partial-Match)

I.5.4.3.1. Modelo espacio vectorial

Salton y sus colegas (1983) definieron un modelo de recuperación de información basado en el criterio de similitud propuesto en el trabajo de Luhn (1957). En este modelo, denominado modelo de espacio vectorial, los documentos y las consultas se representan por medio de vectores en un espacio multidimensional euclídeo, donde cada término se asigna a una dimensión aislada. La creación de este vector implica una exploración léxica que identifique los términos más representativos para los documentos. Cada uno de los términos

extraídos puede ser ponderado mediante distribuciones estadísticas que se aplican sobre los términos y los documentos (Salton & McGill, 1983) y que representan su importancia.

Es precisamente la utilización de estas distribuciones estadísticas en el modelo una de sus principales desventajas. En particular, existe generalmente ambigüedad al definir la medida numérica que expresa la relevancia de un término para un documento en una colección. El modelo espacio vectorial no define cómo deben ser asignados los componentes de los vectores. A este problema de asignación se le conoce con el nombre de "ponderación del término" (Hiemstra, 2009). Los primeros experimentos de Salton con el modelo advirtieron que el problema de la ponderación de términos no era algo trivial en absoluto (Salton & Yang, 1973). A partir de ahí surgieron varias propuestas como, por ejemplo, TF-IDF (Salton & McGill, 1983), Okapi (Robertson et al., 1999), y la ponderación basada en la normalización pivotada (Singhal et al., 1999). Cada una de estas propuestas determinan la relevancia de un término en un documento o colección mediante la definición de funciones que se basan en tres factores: a) frecuencia del término (TF), que refleja cómo de relevante es un término en un documento a través del número de ocurrencias de un término en un documento; b) frecuencia inversa del documento (IDF), que refleja el número de documentos que contienen ese término y que representa la capacidad discriminatoria de un término en un documento con respecto a una colección de documentos; y c) longitud del documento, que no representa un algoritmo de frecuencia como los anteriores analizados, sino que refleja simplemente el número de palabras que tiene un documento. Generalmente, la longitud del documento se utiliza para normalizar el cálculo de las ponderaciones, ya que dentro de un repositorio de documentos no todos tienen la misma longitud. En concreto, este factor se utiliza debido a que dentro de una colección de documentos cuya longitud no esté normalizada, los documentos de mayor longitud tienen más probabilidades de puntuar más alto ya que contienen más palabras y repeticiones de palabras, a diferencia de los documentos más cortos que no tienen tantas probabilidades. De ahí, que se utilice este factor para normalizar el método de cálculo de ponderación.

I.5.4.3.2. Modelo probabilístico 2-Poisson

Bookstein y Sawnson (1974) estudiaron el problema de desarrollar un conjunto de normas estadísticas para identificar los términos índice de un documento. En este trabajo los autores modelan el número de ocurrencias de términos en los documentos a través de dos distribuciones de Poisson (véase ecuación (I.1)), donde 'X' representa una variable aleatoria para el número de ocurrencias.

$$P(X = tf) = \lambda \frac{e^{-\mu_1}(\mu_1)^{tf}}{tf!} + (1 - \lambda) \frac{e^{-\mu_2}(\mu_2)^{tf}}{tf!} \qquad \textbf{(I.1)}$$

Este modelo asume que los documentos se crearon a partir de una secuencia aleatoria de términos y que todos los documentos tienen una longitud constante. Esta colección aleatoria de términos se divide en dos subconjuntos, de tal manera que los documentos que se encuentren en el mismo grupo trataran de un tema relacionado con un término en mayor medida que los documentos que se encuentren en el otro subconjunto. Este suceso se encuentra modelado en la fórmula por 'λ', que define la proporción de documentos que pertenecen al subconjunto uno, y por las distribuciones de Poisson 'μ 1' y 'μ 2' (donde $\mu 1 \geq \mu 2$), que permite estimar a partir de la media del número de apariciones del término en los respectivos subconjuntos. Para cada término, el modelo necesita de estos tres parámetros para determinar a qué subconjunto pertenece. Para la estimación de estos parámetros se pueden utilizar diversos algoritmos como, por ejemplo, el algoritmo de maximización de la expectativa (Dempster et al., 1977) o el método de momentos concebido por Harter (1975), entre otros.

Según esta definición, si un documento se selecciona aleatoriamente de un subconjunto, la probabilidad de pertinencia de este documento se asume que es igual o mayor que la probabilidad de relevancia del documento a partir de dos subconjuntos. Esto se debe a que la probabilidad de relevancia se correlaciona con el grado en que un sujeto se refiere a un término tratado, y a que $\mu 1 \geq \mu 2$. De esta forma, los términos útiles permitirán hacer una buena distinción entre los documentos relevantes y no relevantes debido a que los dos subgrupos se

encuentran bien distinguidos debido a que ' μ 1' y ' μ 2' tendrán un valor "Poisson" muy diferente.

La principal ventaja del modelo de 2-Poisson es que no requiere la implementación de ningún algoritmo de ponderación adicional. Sin embargo, el mayor problema que presenta este modelo es la estimación de los tres parámetros para cada término ya que éstos no pueden ser estimados directamente de los datos analizados durante la ejecución del modelo. Además, a pesar de la complejidad del modelo probabilístico, podría darse la situación de que no se ajustara a los datos reales si las frecuencias de términos difieren mucho en cada documento. Por ello, algunos estudios examinan el uso de más de dos funciones de Poisson, pero esto hace que el problema de estimación sea aún más intratable (Margulis, 1993).

I.6. ANOTACIÓN SEMÁNTICA

I.6.1. DEFINICIÓN

Leech (1997) define anotación como "*la práctica de añadir información lingüística interpretativa a un corpus*". Más concretamente, anotación o etiquetado es un proceso que permite asociar conceptos, relaciones, comentarios o descripciones a un documento o a un fragmento de texto. Siguiendo con la anotación como un proceso de asociación, Nagarajan (2006) define la anotación semántica como "*el proceso de asociar metadatos con recursos (audio, video, información estructurada, información no estructurada, páginas Web)*". Según el autor, la diferencia entre anotación y anotación semántica radica en el tipo de metadato que se utilice para anotar el recurso digital.

Para el objeto de esta tesis, la anotación semántica es un proceso que identifica formalmente conceptos y relaciones entre conceptos en documentos, y está destinado principalmente para el uso por los ordenadores (Uren et al., 2006). Este proceso consiste en insertar en un documento etiquetas que representen elementos ontológicos (esto es, conceptos, relaciones, atributos e instancias). Las etiquetas se asocian a fragmentos de texto, permitiendo así la inserción de metadatos que pueden ser procesados no solo por humanos sino también por

ordenadores (Kiyavitskaya et al., 2005). Hasta la fecha se han propuesto numerosos sistemas de anotación semántica, pero no existe ningún estándar unificado (Uren et al., 2006).

Para aclarar el concepto de anotación semántica, tengamos en cuenta el siguiente ejemplo. Partimos de un texto cualquiera donde aparece escrita la palabra "Paris" y una ontología que contiene una instancia "Paris" que pertenece al concepto abstracto de "Ciudad". Si se crea una anotación asociando la palabra "Paris" del documento con la instancia "Paris" de la ontología y, además, con el concepto abstracto de "Ciudad", entonces se eliminaría cualquier ambigüedad en lo relativo a lo que se refiere la palabra "Paris" en dicho texto.

Según (Oren et al., 2006) este proceso puede realizarse de manera manual, automática o semiautomática.

- **Anotación manual**. Los sistemas de anotación manual son también conocidos como herramientas de autor. El usuario es el encargado de anotar el texto en lenguaje natural de manera manual seleccionando qué conceptos son los que aparecen en el texto. La principal desventaja que presenta este método es que es necesario que el proceso lo realice una persona experta en el dominio y, al ser un proceso manual, es muy probable que se cometan errores.

- **Anotación automática**. Periódicamente el sistema analiza los textos expresados en lenguaje natural y, mediante el uso de un conjunto de reglas, los anotará con respecto a las ontologías. La ventaja principal de este método es que el usuario se ve liberado del proceso de anotación. El problema, sin embargo, es que para ello se tendrán que implementar un conjunto de reglas y es probable que se produzcan errores en la anotación debido a la ambigüedad del lenguaje.

- **Anotación semiautomática**. Por último, está la aproximación híbrida entre las dos técnicas anteriores. Esta vez el usuario realiza un conjunto reducido de anotaciones y, posteriormente, el sistema anotará una parte más compleja que el usuario no tiene que conocer. Con esta aproximación se consigue aunar las ventajas de los procedimientos anteriores, pero todavía se da la circunstancia de que es necesaria la verificación de las anotaciones por parte de un usuario.

Una vez definido el concepto de anotación semántica, a continuación se van a describir las aproximaciones existentes analizando las diferencias, ventajas e inconvenientes de cada una de las técnicas de anotación.

I.6.2. ANÁLISIS DE TÉCNICAS DE ANOTACIÓN SEMÁNTICA

I.6.2.1. Armadillo

Armadillo (Ciravegna et al., 2004) es un sistema para producir anotaciones automáticas sobre un dominio específico en grandes repositorios de información de manera no supervisada. La fuente natural de información del sistema es la Web. Armadillo propone una metodología para extraer información sobre un dominio específico con una intervención mínima por parte del usuario. El funcionamiento del sistema se basa en crear anotaciones extrayendo información de diferentes fuentes e integrándolas en un repositorio. El repositorio se utiliza como contenedor de información para almacenar toda la información extraída, de manera que éste puede ser usado para anotar las páginas donde se encuentre esta información. La Figura I.14 muestra de manera resumida la implementación en pseudocódigo del algoritmo que emplea este sistema.

Entrada:
- Una ontología.
- Un léxico inicial.
- Un repositorio de documentos.

Salida:
- Conjunto de tripletas que representan la información extraída y que será utilizada para anotar documentos.

do{
- Detección de información utilizada en el léxico.
- Búsqueda para confirmar la información identificada.
- Ampliación del léxico mediante la extracción adaptativa de información y búsqueda de la confirmación de la nueva información extraída.

} while (Mientras exista información para analizar).
- Integrar información desde diferentes documentos.
- Almacenar información en el repositorio.

Figura I.14 Principal algoritmo de Armadillo (Ciravegna et al., 2004)

El sistema Armadillo recibe como entradas una ontología, un lexicón inicial (esto es, un conjunto abstracto no ordenado de entradas léxicas que se definen de acuerdo a sus rasgos fónicos y gramaticales) y un repositorio de documentos. En la primera iteración del bucle que muestra la Figura I.14, el sistema se encarga de identificar las posibles anotaciones para un documento seleccionado utilizando el lexicón que el sistema recibió como entrada. Posteriormente, cada anotación detectada será validada utilizando estrategias de desambiguación o de múltiple evidencia. Por otra parte, las anotaciones identificadas que no se encuentran en el lexicón deben ser validadas e incorporadas al lexicón relacionado con el contexto, lo que se corresponde con el proceso de aprendizaje. Como resultado del proceso, todas las anotaciones son validadas, las nuevas son integradas en el lexicón y almacenadas en una base de datos.

El sistema Armadillo presenta una arquitectura bastante simple que se basa en el conjunto de metodologías que se detallan a continuación:

- Extracción adaptativa de información de textos (AIE, del inglés, "*Adaptative Information Extraction*"): metodología que se utiliza para detectar información y aprender nuevas instancias. Mientras que la mayoría de aproximaciones, como COHSE (Bechhofer et al., 2001) y SemTag (Dill et al., 2003), utilizan lexicones estáticos contenidos en una ontología para los procesos de anotación, Armadillo continuamente amplía el lexicón inicial reconociendo nuevos términos en el repositorio de manera automática.
- Integración de la Información (II, del inglés, "*Information Integration*"): se utiliza para descubrir un conjunto inicial de información que se utilizará como semilla de aprendizaje para el AIE y como metodología de verificación para validar la nueva información adquirida. Armadillo aprovecha la redundancia de la información existente en la Web para, o bien adquirir un lexicón inicial, o intentar extraer más información de la proporcionada por el lexicón aportado por el usuario como entrada del sistema. El proceso de extracción de información se lleva a cabo mediante agentes que utilizan sistemas adaptadores de inducción. Estos sistemas emplean metodologías capaces de extraer información de diferentes fuentes y formatos para integrarla. A partir de estos agentes es posible extraer información con diferentes grados de

fiabilidad y estos grados de fiabilidad son utilizados durante el proceso de validación de la información.

- Servicios Web: la arquitectura de Armadillo se compone de servicios donde cada servicio trabaja de manera independiente y está asociado a partes de la ontología, que pueden ser conceptos o relaciones. Cada servicio puede hacer uso de otros servicios, incluyendo servicios externos, para llevar a cabo distintas sub-tareas. Por ejemplo, si el sistema Armadillo necesitara reconocer los nombres de los investigadores en la página de una Universidad, podría hacer uso de servicios de reconocimiento de nombres de entidades (NER, del inglés, "*Named-Entity Recognition"*) como subservicio para obtener los nombres de los investigadores y luego validarlos mediante estrategias internas de validación.
- Repositorio RDF: los datos que son extraídos de la Web, junto con el enlace de donde se extrajeron, se almacenan en forma de tripletas RDF (Sujeto-Verbo-Objeto).

I.6.2.2. CREAM (CREAtion of Metadata)

CREAM (Handschuh & Staab, 2003) es un framework de anotación semántica orientado a permitir la creación fácil y sencilla de metadatos semánticos a través del análisis de recursos en HTML. CREAM integra un sistema de gestión de documentos que facilita el manejo de cambios en los documentos, datos y ontologías. Además, el framework soporta múltiples ontologías, por lo que un documento en la Web podría estar relacionado con múltiples ontologías.

La arquitectura que proporciona CREAM se caracteriza por ser flexible y abierta. Estas características propician la fácil incorporación de otras herramientas como OntoMat. OntoMat[6] es una herramienta software flexible basada en plugins que incorpora a CREAM diversas funciones que hace que pueda trabajar con diferentes formatos de documentos. El núcleo OntoMat, una vez descargado, consta de una ontología básica de orientación, navegador de información, un editor y visor de documentos, y una estructura interna donde almacenar la ontología y los

[6] http://projects.semwebcentral.org/projects/ontomat/

metadatos. Las capacidades semánticas se encuentran aisladas y tienen que ser accedidas a través del plugin de conexión al servidor de inferencia donde se almacenan las anotaciones.

A continuación se describen brevemente los diferentes módulos que componen la plataforma CREAM:

- *Editor de documentos* (del inglés, '*Document Editor*'): está constituido por dos componentes gráficos que facilitan las tareas de visor de documentos y de generación de contenido. El visor soporta diferentes formatos, tales como HTML, PDF, XML, etc., y su función es permitir al usuario visualizar el contenido de los documentos, además de facilitar la alineación entre partes de la ontología y fragmentos de texto mediante la creación de nuevos metadatos. La función del generador de contenido es la de almacenar nuevas instancias en el "servidor de inferencia de anotaciones". Estas nuevas instancias se generan en el proceso de anotación semántica cuando el generador de contenido, durante el análisis de un documento, detecta textos para anotar según la meta-ontología.
- *Guía ontológica y navegador* (del inglés, '*Ontology guidance & fact browser*'): la ontología que utiliza el framework debe ser consistente, de ahí que se utilice una guía que permita comprobar la consistencia de la misma. Además, ambos componentes son importantes para guiar a los creadores de metadatos durante el proceso de anotación.
- *Rastreador* (del inglés, '*Crawler*'): durante el proceso de creación de metadatos dentro de la Web Semántica, las herramientas de anotación deben ser conscientes de la existencia de entidades que ya han sido anotadas para evitar la creación de metadatos redundantes. La labor del rastreador es proporcionar esta información sobre las propiedades de entidades anotadas, de forma que se evite la co-existencia de metadatos redundantes en el sistema.
- *Servidor de inferencia de anotaciones* (del inglés, *Annotation inference server*): su función es proporcionar información acerca de las instancias almacenadas para (i) evitar anotaciones redundantes, (ii) permitir la creación de referencias apropiadas, y (iii) ayudar en el proceso de chequeo de consistencia. Además, proporciona toda la información necesaria al módulo "*guía ontológica y navegador*" permitiendo consultar clases, instancias y propiedades existentes

en la ontología. El servidor de inferencia de anotaciones soporta múltiples ontologías que se distinguen a través de diferentes espacios de nombres.

- *Gestor de documentos* (del inglés, '*Document management*'): su función es mantener la consistencia de las anotaciones en los documentos que ya fueron anotados. Cualquier cambio en un documento o una página Web anotada puede provocar que se generen nuevas anotaciones, que se mantengan las anotaciones antiguas o que muchas anotaciones queden invalidadas. Este módulo se encarga de decidir qué anotaciones debe incluir, invalidar o no tener en cuenta sobre la base de las anotaciones previamente establecidas a partir de los cambios realizados en el recurso. De esta forma se consigue mantener la consistencia de las anotaciones con respecto a los documentos anotados.
- *Re-reconocimiento de metadatos y extracción de información* (del inglés, '*Metadata re-recognition and information extraction*'): su objetivo es ofrecer un mecanismo semiautomático para la creación de metadatos utilizando técnicas de extracción de información para sugerir anotaciones a los creadores de metadatos. El funcionamiento de este módulo se basa en detectar nuevos metadatos comparándolos con los ya existentes. Después utiliza sistemas de extracción de conocimiento como Amilcare (Ciravegna et al., 2002b) como herramienta de extracción de información.
- *Meta-ontología* (del inglés, '*Meta-ontology*'): en CREAM se permite separar el diseño de la ontología de su utilización. La meta-ontología es una ontología que define cómo deben estar definidas las ontologías para que puedan ser utilizadas por el sistema de anotación en la plataforma. Es decir, esta ontología describe las clases, atributos y relaciones de tal manera que el anotador establezca la conexión que sea necesaria con la ontología del dominio para realizar la anotación a través de la meta-ontología. Este diseño, en varios niveles de ontologías, facilita el soporte para el uso de múltiples ontologías.
- *Almacenamiento* (del inglés, '*Storage*'): la plataforma CREAM proporciona dos mecanismos de almacenamiento, dentro del documento ubicado en el módulo de gestión de documentos, o en el servidor de inferencia de anotaciones.
- *Replicación* (del inglés, '*Replication*'): la plataforma CREAM proporciona un mecanismo sencillo de replicación de información que facilita el almacenamiento dentro del documento o en el servidor de inferencia de

anotaciones. El procedimiento de replicación es sencillo y se basa en mover las anotaciones desde el módulo de gestión de documentos al servidor de inferencia de anotaciones, donde los procesos de inferencia facilitan el descarte en caso de inconsistencias formales.

I.6.2.3. S-CREAM (Semi-automatic CREAtion of Metadata)

S-CREAM (Handschuh et al., 2002) es un framework que integra la herramienta OntoMat para proporcionar una metodología de anotación semiautomática que permite la creación de metadatos a través de la anotación de páginas Web. Esta metodología se basa en el componente de extracción de información de Amilcare, que provee un conjunto de reglas de extracción de conocimiento a OntoMat para extraer información procedente de las páginas Web. Estas reglas de extracción son generadas previamente a partir de un ciclo de aprendizaje basado en páginas previamente anotadas.

El objetivo principal del framework S-CREAM es facilitar la generación fácil de representaciones de páginas Web en grupos de anotaciones semánticas. Una anotación semántica, en el contexto de S-CREAM, se refiere a un conjunto de instancias asociadas a un documento HTML. Estas instancias pueden ser (i) instancias de clases DAML+OIL, (ii) propiedades instanciadas desde la instancia de una clase a la instancia de un tipo de dato, y (iii) propiedades instanciadas desde la instancia de una clase a la instancia de otra clase. Para S-CREAM estas anotaciones representan "metadatos relacionales". "Metadatos" porque son anotaciones que se utilizan para enriquecer los documentos HTML y "relacionales" porque, además, están relacionadas con instancias de la ontología del dominio.

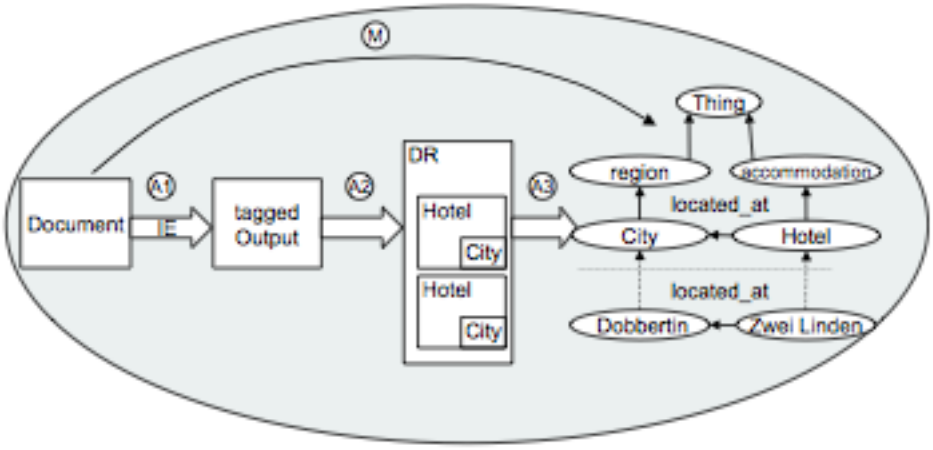


Figura I.15. Funcionamiento de S-CREAM (Handschuh et al., 2002)

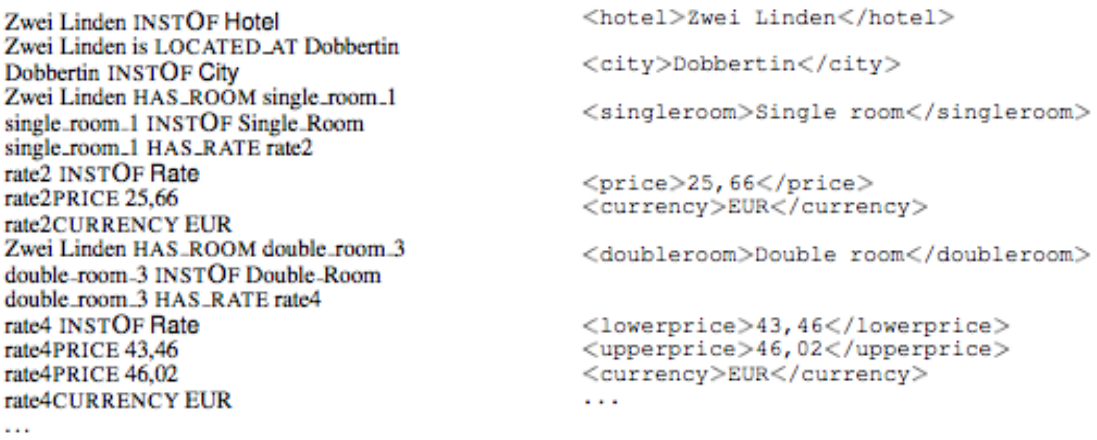
La Figura I.15 representa gráficamente el funcionamiento del framework S-CREAM. Éste se puede resumir en tres procesos: proceso de anotación manual, proceso de extracción de información y proceso de relacionar la información extraída con el dominio ontológico. El primer proceso se indica con el círculo 'M' y permite la adaptación de la herramienta a cualquier dominio específico. El proceso se encarga de la anotación manual y la creación de metadatos para transformar un documento en un conjunto de metadatos relacionales definidos a partir de un modelo ontológico subyacente. Para llevar a cabo la anotación manual se utiliza OntoMat, que proporciona una interfaz que permite realizar este proceso de manera sencilla sin la necesidad de tener grandes conocimientos de sistemas de procesamiento del lenguaje natural. Como salida, OntoMat proporciona las anotaciones manuales como etiquetas XML. Estas etiquetas serán utilizadas en la fase de entrenamiento por el componente de aprendizaje de Amilcare para inducir reglas que sean capaces de reproducir anotaciones.

Amilcare proporciona dos modos de funcionamiento: el modo de entrenamiento, utilizado para adaptar la herramienta a un nuevo dominio de aplicación, y el modo de extracción, utilizado para anotar textos. En ambos modos, lo primero que realiza Amilcare es un preprocesamiento del texto utilizando "Annie"[7] (en inglés, *A Neraly-New Information Extraction System*), un sistema de extracción de información incluido en la plataforma GATE (Cunningham et al., 2002). GATE (en

[7] http://gate.ac.uk/ie/annie.html

inglés, *General Architecture for Text Engineering*) es una suite de herramientas Java desarrollada originariamente por la Universidad de Sheffield a comienzos de 1995 que incluye todo tipo de tareas de procesamiento del lenguaje natural y extracción de información en diferentes idiomas.

El segundo proceso que lleva a cabo S-CREAM viene representado en la Figura I.15 con la etiqueta '`A1`' y consiste en la extracción de información. En este proceso la responsabilidad de obtener la información recae sobre herramienta Amilcare que desempeña la función de analizar un documento y producir un documento etiquetado en XML o una lista de fragmentos etiquetados en XML (véase Figura I.16 (b)).



(a) OntoMat (b) Amilcare

Figura I.16 Comparación de salidas OntoMat vs Amilcare (Handschuh et al., 2002)

El tercer y último proceso engloba lo representado en la Figura I.15 con las etiquetas '`A1`', '`A2`' y '`A3`'. Su función es la de vincular las etiquetas obtenidas por el sistema de extracción de información con el modelo ontológico destino a través de una representación del discurso explicita. Para ello, S-CREAM utiliza una versión simplificada del modelo de representación del discurso desarrollado por Grosz y Sidner (1986). La idea principal de este modelo es definir una serie de restricciones fijas y reglas "suaves" que guían todo el proceso de referencia. Las restricciones fijas proporcionan los objetos para resolver el problema de la anáfora y el establecimiento de nuevas relaciones lógicas inter-oracionales, mientras que las reglas suaves dan un orden de preferencias a estos posibles antecedentes. La versión simplificada de S-CREAM formula una serie de reglas básicas que son

utilizadas por las tareas de anotación. Básicamente, estas reglas se centran en el análisis de términos adyacentes al fragmento de texto que está siendo anotado.

I.6.2.4. KIM

KIM (Popov et al., 2003) es una plataforma que proporciona una infraestructura de gestión del conocimiento y de la información. También proporciona servicios para la anotación semántica automática, indexación y recuperación de documentos. KIM presenta una infraestructura madura, escalable y personalizable para la extracción de información. Esta infraestructura incluye, además, funciones para la gestión de anotaciones semánticas y documentos a través de GATE. Con el fin de proporcionar un nivel básico de rendimiento, la plataforma cuenta con una ontología de nivel superior y una base de conocimiento que cubre dominios de carácter general. Las ontologías y las bases de conocimiento involucradas en la plataforma siguen los estándares de la Web Semántica, incluyendo repositorios RDF, ontologías y razonamiento.

La ontología de nivel superior, denominada "KIMO", contiene definiciones de clases de entidades, atributos y relaciones, así como una rama de tipos de recursos léxicos y nombres de entidades como, por ejemplo, títulos, nombres de personas, etc. Las descripciones semánticas de las entidades y las relaciones entre ellas se mantienen en una base de conocimiento (KB) codificada en la ontología KIM, que reside en el mismo repositorio semántico. Así, KIM ofrece, para cada referencia de la entidad en el texto, un enlace (URI) para la clase más específica en la ontología y un enlace a la instancia específica en la KB. De esta forma, cada entidad nombrada extraída se encuentra ligada a su información de tipo específica.

Las bases de conocimiento han sido instanciadas previamente con nombres de entidades de importancia general como lugares, personas, y otras relaciones. Estas entidades nombradas extraídas podrían utilizarse, además, para la indexación semántica y para la recuperación de contenido relacionado con instancias de la entidad y el tipo, así como el nombre y los atributos, y las relaciones entre estas entidades.

La ontología de nivel superior es simple pero suficiente y adecuada para el uso general en el dominio de las anotaciones semánticas. KIM mantiene las

descripciones semánticas de las entidades en el "KIM KB", que se enriquece con las entidades y las relaciones extraídas. Las descripciones de estas entidades se almacenan en formato RDF en el repositorio de ontologías. Cada entidad tiene información sobre su tipo específico, alias, los atributos y sus relaciones.

La API "KIM Server" incorpora funciones que facilitan el acceso remoto y la integración de información. También dispone de la infraestructura para la anotación semántica, indexación y recuperación, así como la gestión de documentos y la base de conocimiento. La API de anotación semántica permite la anotación de documentos con respecto a la ontología KIM y la base de conocimiento. Además, este módulo proporciona infraestructura para administrar el contenido y las anotaciones. La API de indexación está basada en una versión modificada del motor Lucene IR[8], que permite la indexación y recuperación de recursos respecto a las entidades nombradas.

La API de consulta puede equipararse a un buscador semántico que permite la especificación de búsquedas tradicionales por palabras clave y otros métodos de búsqueda basados en ontologías. Además, proporciona la infraestructura para la construcción de composiciones de búsquedas, que permiten combinar diferentes tipos de búsqueda cómo, por ejemplo, búsquedas de entidades con búsquedas por palabras clave y búsquedas de patrones. Existe, del mismo modo, una API de acceso al repositorio semántico que permite consultar sobre la base de conocimiento subyacente utilizando el lenguaje de consultas SPARQL.

I.6.2.5. CERNO

Cerno (Kiyavitskaya et al., 2009) es un framework para anotación semántica semiautomática de documentos de acuerdo con un modelo semántico de dominio especifico. El framework está basado en técnicas y herramientas ligeras para el análisis y marcado del código fuente. El framework consiste en (i) un proceso sistemático de definición de palabras clave y reglas basadas en gramática para identificar instancias de conceptos en el texto, y (ii) una arquitectura basada en un sistema software de recuperación que permite la aplicación de reglas de marcado y la extracción de las instancias identificadas en un conjunto de documentos.

[8] http://www.jakarta.apache.org/lucene.

La arquitectura de Cerno se basa, en parte, en el sistema LS/2000 (Dean et al., 2001). Este sistema implementa un método automático para la recuperación de diseños de software de aplicación. De ahí, que los documentos con los que se ha validado este sistema eran programas informáticos escritos en lenguajes de programación formal. El funcionamiento del sistema estaba basado en una serie de fases: (i) análisis del documento, (ii) reconocimiento de hechos básicos, (iii) interpretación con respecto a un modelo semántico de dominio, y (iv) mapeo de la información identificada en una base de datos externa. En el framework Cerno estas tareas o fases han sido re-implementadas para extender las funciones de anotación a documentos de texto arbitrario. Así, la arquitectura Cerno incluye tres nuevos bloques que se analizan brevemente a continuación:

- Analizador sintáctico: tiene la misión de construir un árbol sintáctico a partir de fragmentos del texto de entrada. Para extraer el contenido del texto, Cerno requiere la construcción de una gramática independiente definida con el lenguaje de programación TXL (del inglés, "*Turing eXtender Language"*) (Cordy et al., 2000) y en notación BNF (del inglés, "*Backus-Naur Form"*) (Grune & Jacobs, 1990). La función del lenguaje de programación TXL es analizar el texto de entrada y transformarlo en un texto de salida aplicando una serie de reglas definidas en notación BNF. El proceso de transformación, llevado a cabo por el motor TXL, da como resultado un árbol sintáctico formado por fragmentos extraídos del texto de entrada.
- Marcado: su función esencial es reconocer instancias de conceptos, es decir, anotar las unidades de texto que contienen información relevante de acuerdo con un esquema de anotación previamente definido. Este esquema contiene un vocabulario relacionado con un dominio concreto. Entonces, durante el proceso de marcado, si uno de los indicadores de la lista definidos en el esquema se encuentra presente en un fragmento de texto, el fragmento se etiqueta con el nombre asociado al indicador.
- Mapeo: la función de este bloque es el almacenamiento de las anotaciones en un repositorio externo. Para realizar el almacenamiento, en primer lugar debe definirse una plantilla que especifique el esquema del almacén de información. En este esquema se define la lista de campos existente en el repositorio a través de un fichero DTD (del inglés, "*Document Type Definition*"), que representa una

plantilla que describe la estructura de destino donde van a ser almacenadas las anotaciones. Para enlazar los fragmentos de texto anotados con sus respectivos campos, se utiliza un esquema gramatical independiente del dominio que genera un documento XML que contiene toda la información ordenada en las etiquetas que fueron definidas en el fichero DTD. En el caso de que existan fragmentos de texto con varias anotaciones, se realiza la copia del fragmento tantas veces como anotaciones hayan sido asignadas.

I.6.2.6. EvOnto

EvOnto (Tissaoui et al., 2011) es tanto una herramienta como un método que soporta una gestión coherente y conjunta del cambio para término-ontologías y anotaciones semánticas. Las término-ontologías son ontologías que definen un componente léxico donde los términos aparecen como entidades en la ontología que se suman a las clases conceptuales. La utilización de este tipo de ontologías permite a EvOnto proporcionar sistema de anotación óptimo basado en ontologías simples que no pretenden describir un dominio completo, pero que son suficientes para proporcionar anotaciones óptimas en una colección de documentos.

EvOnto ha sido desarrollada basándose en tres aspectos fundamentales que caracterizan su funcionamiento. En primer lugar, presta especial atención a los términos que contribuyen a definir las anotaciones. En segundo lugar, define criterios de calidad para evaluar el resultado de la anotación automática de textos y detectar carencias en la ontología. Y por último, asume que la calidad de la ontología es evaluada a través de su utilización para la anotación.

EvOnto extiende la clasificación de cambios propuesta por (Stojanovic, 2004), donde se identifican todos los posibles cambios que pueden realizarse sobre una ontología para adaptarlo al uso de termino-ontologías (TOR, del inglés, "*Termino-Ontologies*"). De manera que el proceso de evolución de ontologías se simplifique para producir cambios sobre la taxonomía de la ontología y para detectar las consecuencias lógicas de estos cambios. Esta simplificación propicia que la evolución de la TOR se expresa a través de cambios producidos en los términos o en las relaciones término-concepto motivados por las necesidades de anotación. En este proceso de evolución, el objetivo de la herramienta es reducir el impacto

negativo que conlleva cada modificación en la ontología o en las anotaciones. Por impacto negativo se entiende a la incorporación de nuevas anotaciones inservibles o la producción de anotaciones de menor calidad.

El método EvOnto, por su parte, define un proceso cíclico que se lleva a cabo después de finalizar la anotación de nuevos documentos. Este proceso permite comprobar la calidad de la ontología e identificar necesidades de evolución. Si se identifica cualquier necesidad de evolución, EvOnto propondrá un conjunto de operaciones de cambio en la TOR, de forma que para cada operación propuesta la herramienta permita mantener la consistencia dentro de la TOR con las anotaciones. El impacto de las nuevas anotaciones es controlado a través del proceso que se describe a continuación:

- En primer lugar, se añaden documentos nuevos a la colección o, si han sido ya anotados, se eliminan de la misma. La retirada de documentos implica que muchos conceptos o términos en la ontología queden no referenciados.
- En segundo lugar, después de agregar los nuevos documentos a la colección, un ingeniero de conocimiento tiene que poner en marcha el módulo de anotación semántica. La novedad de esta herramienta es que permite la definición de criterios de calidad para evaluar las nuevas anotaciones. Cada criterio concreta un conjunto de conceptos y/o relaciones que esperan ser encontrados en cada anotación de texto. De esta forma, una anotación no será válida a menos que se utilice en la misma una instancia de estos conceptos/relaciones o subconceptos. La verificación de si las anotaciones cumplen o no los criterios establecidos proporciona un resultado cuantitativo que refleja si el documento queda bien descrito por sus anotaciones.
- En tercer lugar, el ingeniero del conocimiento puede explorar la lista de documentos, empezando por aquellos con puntuaciones más bajas. Esta exploración se trata de un proceso manual guiado de manera eficiente, que tiene por objetivo identificar aquellos términos y partes del texto que no han sido anotadas durante el proceso de anotación. Esta identificación facilita la detección de carencias, que pueden conllevar a cambios en la ontología, como por ejemplo incorporación de nuevos términos a los conceptos ya existentes o la incorporación de nuevas subclases previstas en los criterios de anotación.

El método EvOnto también contempla la posibilidad de que el proceso de evolución de la ontología esté guiado por el ingeniero de conocimiento. En este caso, EvOnto ayuda al ingeniero a ser consciente del impacto de cualquier cambio que se produzca en la ontología, tanto en los términos de referencia como en las anotaciones, y le permite realizar las adaptaciones que puedan ser necesarias. Este proceso de evolución se realiza en cuatro etapas, a saber: (i) selección de la entidad que se desea modificar y selección del tipo de cambio que va a ser aplicado, (ii) toma de decisión sobre las consecuencias o la estrategia de evolución a seguir, (iii) adaptación al impacto, y (iv) propagación del cambio en las anotaciones de tal manera que reduzcan los efectos secundarios y se eviten las anotaciones globales. Para expresar esa necesidad de evolución, el método EvOnto define los tipos de modificaciones que pueden darse en una TOR extendiendo la tipología de cambios propuesta por (Stojanovic, 2004). Concretamente, los cambios pueden producirse sobre términos o en las relaciones término-concepto si el dominio del vocabulario evoluciona. Cada cambio lleva consigo definida una estrategia de evolución en la que se describen las consecuencias correspondientes de este cambio. Estas consecuencias son las que se muestran al ingeniero del conocimiento antes de que se lleven a cabo los cambios.

La evolución de la TOR provoca la evolución de las anotaciones. Este proceso se lleva a cabo en dos etapas: (i) detección de anotaciones inconsistentes, esto es, anotaciones que hacen referencia a conceptos, términos o relaciones que han sido modificadas, y (ii) modificación de las anotaciones inconsistentes en coherencia con el nuevo contenido de la ontología. Concretamente, el primer paso se centra en navegar por el grafo de anotaciones para buscar elementos modificados en la ontología, mientras que el segundo se encarga de reparar las anotaciones una vez detectada la inconsistencia.

Por último, se describe algunos aspectos de la interfaz altamente funcional de la herramienta EvOnto. A continuación se listan algunas de las funciones más significativas que esta herramienta proporciona son:

- Permite capturar criterios de calidad de la anotación.
- Proporciona al usuario la posibilidad de seleccionar conceptos en el proceso de anotación haciendo posible que cada documento será anotado por una o varias instancias de conceptos o subclases.

- Permite al usuario la posibilidad de seleccionar relaciones de esos conceptos.
- Al final del proceso de anotación, la interfaz muestra la lista de anotaciones que se adecuan a los criterios de calidad definidos previamente. Las anotaciones que no cumplan los criterios establecidos pueden ser analizadas por el ingeniero de conocimiento para extraer nuevos conceptos y términos que incorporar a la ontología.

I.6.2.7. GoNTogle

GoNTogle (Bikakis et al., 2010) es un framework basado en tecnologías de la Web Semántica y de recuperación de información (IR) que facilita la anotación y recuperación de documentos. La herramienta soporta la anotación basada en ontologías para diferentes formatos de documento (p.ej., doc, pdf, txt, rtf, odt, sxw, etc.) en un entorno completamente colaborativo. GoNTogle implementa mecanismos de anotación manual y automática. La anotación automática se basa en un método de aprendizaje que explota el historial de anotación del usuario cuando se enfrenta a nuevos documentos. La anotación se basa en tecnologías estándar de la Web Semántica como OWL y RDFS. Cada anotación se almacena como una instancia de la ontología junto con la información del documento anotado. Esta información del documento se define a través de propiedades de la ontología que permiten la conexión bidireccional entre los documentos y las ontologías.

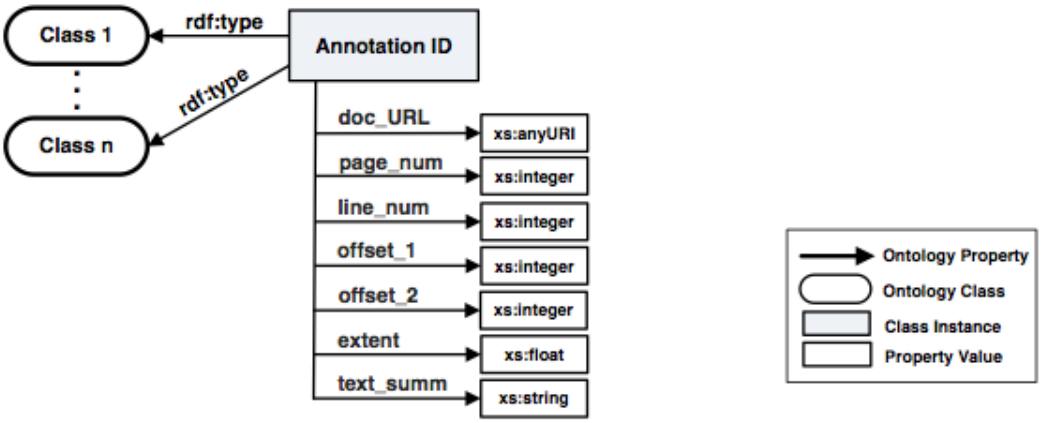


Figura I.17 Modelo de anotación basado en ontología (Bikakis et al., 2010)

En la Figura I.17 se presenta el modelo de anotación basado en ontologías que utiliza GoNTogle. Como se puede apreciar en la imagen, las anotaciones se representan como instancias que pueden pertenecer a una o más clases de la ontología. También se muestra en la imagen la utilización de propiedades de la ontología para lograr la comentada conexión bidireccional entre documentos y clases. Cada instancia define un conjunto de propiedades esenciales, entre las que se pueden destacar las siguientes: (i) "`doc_url`", que representa la URL del documento anotado, (ii) "`page_num`" y "`line_num`", propiedades que se corresponden con el número de página y línea donde comienza la anotación, y (iii) "`offset_1`" y "`offset_2`", que representan la posición de comienzo y final de la anotación en el documento.

Las anotaciones se almacenan separadas del documento original en un servidor centralizado para proporcionar el entorno colaborativo. Un método de aprendizaje explota la información textual y los historiales de anotación de los usuarios, almacenados en este servidor centralizado, para dar soporte a un mecanismo de anotación automática que sugiere anotaciones para los nuevos documentos entrantes.

GoNTogle también proporciona servicios de búsqueda más avanzados que los procesos de búsqueda tradicionales basados en palabras clave. La herramienta propone una combinación flexible de metodologías de búsqueda basada en palabras clave y búsqueda semántica junto con operaciones avanzadas de búsqueda basada en ontologías.

La arquitectura de GoNTogle se compone de cuatro subsistemas básicos, a saber, el anotador semántico (del inglés, '*Semantic Annotation*'), el servidor de ontologías (del inglés, '*Ontology Server*'), el indexador (del inglés, '*Indexing*') y el motor de búsqueda (del inglés, '*Search*'). Seguidamente, se describe brevemente la función que desempeña cada componente dentro de la plataforma:

- Anotador semántico: se encarga de anotar semánticamente los documentos utilizando una o varias ontologías para todo o parte del documento. Cada anotación realizada se almacena en el servidor de ontologías como una instancia que contiene información acerca de la parte del documento que ha

sido anotado. Este componente está constituido por tres módulos, el visor de documentos, el visor de ontologías y el editor de anotaciones.

El método de aprendizaje utilizado para anotar documentos automáticamente se basa en el método de clasificación supervisada "vecinos más cercanos con distancia ponderada" kNN (del inglés, "*Weighted K Nearest Neighbors*") (Mitchell, 1997), que permite explotar el historial de anotaciones y la información textual para sugerir automáticamente anotaciones en nuevos documentos. Este método de aprendizaje consta de una fase de entrenamiento que se alimenta de las anotaciones manuales realizadas por los usuarios. Cuando un documento se anota manualmente, el método de aprendizaje extrae el texto anotado y lo indexa utilizando un índice invertido que almacena información acerca de las clases en la ontología utilizadas para la anotación. Para anotar automáticamente los documentos, el usuario debe seleccionar primero un documento. Entonces, teniendo en cuenta el conjunto de datos de entrenamiento, el método de aprendizaje sugiere una lista de conceptos de la ontología (clases) para anotar el documento.

- Buscador: permite buscar documentos mediante una combinación flexible de distintas metodologías de búsqueda. Concretamente, los tres tipos de búsqueda que proporciona GoNTogle son:
    - Búsqueda basada en palabra clave: se ajusta al modelo de búsqueda tradicional, donde el usuario proporciona un conjunto de palabras clave que el sistema utiliza para recuperar documentos basándose en métricas de similitud textual. La métrica que GoNTogle ha adoptado es la que proporciona el motor Lucene IR.
    - Búsqueda basada en semántica: modelo de búsqueda que permite al usuario navegar por la taxonomía de la ontología y enfocar su búsqueda en una o más clases definidas.
    - Búsqueda hibrida: permite al usuario utilizar tanto palabras clave como las clases de la ontología. Además, en este modelo de búsqueda se pueden definir operaciones de conjuntos tales como la intersección o la unión.

- Servidor de ontologías: almacena todas las anotaciones semánticas de los documentos en forma de instancias en formato OWL. Este componente está formado por dos módulos, el administrador de ontologías y la base de conocimiento ontológico.
- Indexador: indexa los documentos mediante la construcción de un índice invertido.

I.6.2.8. MnM

MnM (Vargas-Vera et al., 2002) es una herramienta de anotación que implementa estrategias automáticas y semiautomáticas de marcado de contenido Web con contenidos semánticos. MnM integra un navegador Web con un editor de ontologías y proporciona varias APIs de código abierto para relacionar herramientas de extracción de información con repositorios de ontologías. Sin embargo, en un principio, el trabajo se centró en la creación de un modelo de proceso genérico para el desarrollo de contenidos Web semánticos enriquecidos. Actualmente, la herramienta MnM se compone de repositorios ontológicos y herramientas de extracción de información (IE).

El objetivo de esta herramienta es proporcionar utilidades para la creación de contenidos Web semánticos directamente, ya que la tarea de anotación semántica no es una tarea trivial. El funcionamiento de la herramienta se basa en cinco actividades que se explicarán a continuación:

- Examinar el repositorio de conocimiento: permite al usuario navegar entre la librería de modelos de conocimientos ubicada en el repositorio de ontologías. El usuario, a través de esta utilidad, puede tener una visión resumida de las ontologías en la herramienta para seleccionar cuál de ellas representa el dominio para la anotación. Esta selección determina qué ontología se utilizará para iniciar el proceso de marcado. La selección de una ontología permitirá al usuario navegar por el modelo y visualizar las clases. Dentro de una ontología pueden seleccionarse cada uno de los elementos existentes como punto de partida para la elección de un mecanismo de extracción de información. Es decir, la clase de la ontología seleccionada se empleará para crear una plantilla

que será utilizada por la herramienta de extracción para analizar el cuerpo de los documentos y crear instancias durante el proceso.

- Etiquetado semántico: define el proceso de anotar los documentos que contengan texto plano o hipertexto con un conjunto de etiquetas de la ontología. La ontología que utiliza esta herramienta es la ontología KMi (del inglés, "*Knowledge Media Institute*")[9], ontología que modela personas, proyectos, publicaciones y eventos sobre una organización. MnM se centra en detectar eventos y proporcionar funciones para navegar por la jerarquía de eventos que define la ontología KMi. En esta jerarquía, cada evento se representa a través de una clase y el componente de anotación extrae el conjunto de posibles etiquetas definidas en cada clase. Después de seleccionar una clase, es necesario proporcionar un corpus de entrenamiento etiquetado manualmente. Para realizar esta actividad, el usuario analiza los documentos apropiados a través del navegador de la herramienta y anota segmentos de texto utilizando etiquetas basadas en propiedades de clases definidas en la ontología. Entonces, la herramienta inserta las etiquetas en formato SGML/XML.
- Proceso de aprendizaje: incluye funciones de navegación Web, visor de ontologías y desarrollo de sistemas de extracción de información mediante la integración de herramientas de extracción a través de plugins. En este caso, se ha utilizado Amilcare (Ciravegna, 2001). Al inicio de la fase de aprendizaje, Amilcare preprocesa los textos usando la herramienta de extracción de información ANNIE de GATE (Cunningham et al., 2002), que se encarga de segmentar el texto en palabras para identificar tokens. ANNIE utiliza un motor de reconocimiento de entidades para identificar nombres de organizaciones, fechas, etc. Después, el sistema de extracción de información basado en Lazy-NLP (Ciravegna, 2001) aprovecha la información proporcionada por el componente ANNIE para generar reglas genéricas de inducción que permitan la extracción de la información.
- Test: MnM proporciona dos mecanismos para seleccionar un corpus de prueba y distinguirlo de un corpus de entrenamiento. Por un lado, el usuario puede

[9] http://kmi.open.ac.uk

seleccionar manualmente los corpus de entrenamiento en forma de archivos locales o en la Web. Por otro lado, también ofrece la posibilidad de seleccionar un solo corpus local o en la Web y dejar al sistema que cree corpus de test y entrenamiento de manera aleatoria.

- Proceso de extracción de información: Amilcare, que es el sistema de extracción de información empleado por MnM, presenta dos modos de operación, a saber, modo entrenamiento y modo extracción. Al finalizar la fase de entrenamiento, Amilcare cuenta con una biblioteca de reglas inducidas que se pueden utilizar para extraer información a partir de texto. Este conjunto de reglas, junto con unas etiquetas definidas por el usuario y con la herramienta ANNIE de la arquitectura GATE, se utilizará durante el modo de extracción para procesar colecciones de texto. Como resultado se obtiene el texto original con las anotaciones añadidas en XML o SGML/XML, en función de la versión de la arquitectura GATE utilizada. Una vez que la información ha sido extraída y verificada por el usuario, entonces se envía al repositorio semántico para poblar la ontología seleccionada.

I.6.3. COMPARACIÓN DE LAS HERRAMIENTAS DE ANOTACIÓN SEMÁNTICA

Actualmente, existen varios sistemas de anotación basados en ontologías, pero no existe ninguna metodología estándar con la que analizar y comparar este tipo de sistemas (Uren et al., 2006). En esta tesis doctoral se emplean los parámetros sugeridos en trabajos previos (Rodríguez-García et al., 2014a); (Rodríguez-García et al., 2014b) para realizar una comparativa entre las plataformas de anotación semántica más importantes en la actualidad. Estos parámetros son los siguientes:

- Soporte de ontologías y uso de estándares del W3C, tales como OWL y RDF, para su representación.
- Soporte para anotar formatos heterogéneos de los documentos. Las anotaciones pueden utilizarse en lenguajes de marcado como HTML o XML, pero también en formatos como PDF, doc, etc.
- Capacidad de almacenar las anotaciones generadas en una base de datos independiente de los documentos anotados.

- Uso de interfaz que facilite la anotación de los documentos. Es importante establecer los niveles de acceso a las anotaciones para asegurar su privacidad en algunos casos.
- Capacidad de mantener la consistencia de las relaciones entre las anotaciones y los textos anotados, de manera que si cambia el texto también cambien las anotaciones realizadas, y si cambia la ontología, que también se modifiquen las anotaciones en los documentos.
- Existencia de procesos automáticos basados en tecnologías de procesamiento del lenguaje natural para llevar a cabo la anotación.
- Capacidad de evolucionar una ontología proporcionando una metodología que gestione todo el proceso de mantenimiento de anotaciones en los recursos anotados.

En la Tabla I.1 se muestra un resumen de la comparación realizada a partir de las características comentadas anteriormente. En los siguientes apartados se analizarán por separado cada una de estas características para establecer comparativas entre las distintas herramientas que han sido analizadas (Rodríguez-García et al., 2014a).

Tabla I.1 Tabla comparativa de herramientas de anotación semántica

Sistema de anotación	Formato estándar	Soporte ontología	Formatos documentos	Evolución de documentos	Almacenamiento de anotación	Automatización técnica	Evolución ontología
Armadillo (Chapman et al., 2005)	RDF	Múltiples ontologías	Web y documentos de texto	Si	Modelo semántico	Automática	No
CERNO (Kiyavitskaya et al., 2009)	RDF OWL	No múltiples ontologías	Documentos de texto	No	Modelo semántico	Semiautomática	No
CREAM (Handschuh & Staab, 2003)	RDF	Múltiples ontologías	Web y documentos de texto	Si	Modelo semántico	Automática	Si
S-CREAM (Handschuh et al., 2002)	RDF	No múltiples ontologías	Web	No	Modelo semántico	Semiautomática	Si
KIM (Popov et al., 2003)	RDF extensible a OWL	Múltiples ontologías	Documentos	Si	Modelo semántico	Automática	Si
EVONTO (Tissaoui et al., 2011)	RDF OWL	No múltiples ontologías	Documentos de texto	Si	Modelo semántico	Automática	Si
GoNTogle (Bikakis et al., 2010)	RDF OWL	Múltiples ontologías	Documentos de texto	No	Modelo semántico	Manual and automática	No
MnM (Vargas-Vera et al., 2002)	RDF	No múltiples ontologías	Documentos	No	Modelo semántico	Semiautomática	No
Nuestra aproximación	OWL	Múltiples ontologías	Documentos de texto	Si	Modelo semántico	Manual and automática	Si

I.6.3.1. Formato estándar

El *World Wide Web Consortium* (W3C) ofrece diferentes técnicas para describir y definir formatos para crear ontologías. Algunas de estas técnicas son RDF, *RDFS*, SKOS (del inglés, "*Simple Knowledge Organization System*"), OWL y RIF (del inglés, "*Rule Interchange Format"*).

Actualmente, las técnicas más extendidas utilizadas para construir ontologías son RDF y OWL. Muchas de las herramientas que se han analizado utilizan el formato RDF para definir anotaciones. Estas herramientas son Armadillo (Chapman et al., 2005), MnM (Vargas-Vera et al., 2002), S-CREAM (Handschuh et al., 2002) y CREAM (Handschuh & Staab, 2003). Sin embargo, la tendencia en las herramientas de los últimos años ha sido utilizar OWL como formato para las anotaciones. Herramientas como CERNO (Kiyavitskaya et al., 2009), EVONTO (Tissaoui et al., 2011), GoNTogle (Bikakis et al., 2010) o KIM (Popov et al., 2003) utilizan este formato. No obstante, estos no son los únicos formatos empleados en las herramientas de anotación. El sistema presentado en (Zeni et al., 2007) utiliza un esquema alternativo de anotación basado en ficheros generados por el lenguaje de programación TXL (Cordy et al., 1988).

En esta tesis doctoral, el formato utilizado es la segunda versión de OWL, OWL2 (Grau et al., 2008). La selección de este formato se debió fundamentalmente a que es un modelo formal que soporta un conjunto importante de servicios de inferencia automática DL, entre los que podemos destacar:

- Chequeo de consistencia para asegurar que la ontología está bien construida y no sufre de problemas de inconsistencia.
- Concepto de "satisfacibilidad", que permite comprobar la posibilidad de que una clase tenga instancias. Por lo tanto, si una clase es "insatisfacible", quiere decir que, definir una instancia de esa clase hará que la ontología sea inconsistente.
- Servicio de clasificación para calcular las relaciones de subclase entre cada clase identificada y crear la jerarquía de clases completa. La jerarquía de clases puede ser utilizada para responder consultas tales como obtener todas las subclases o sólo las subclases directas a una clase especificada.

- Chequeo para encontrar las clases más específicas a las que los individuos pertenecen. Además, esta comprobación permite computar el cálculo de tipos directos.

Todos estos servicios de inferencia son proporcionados por diferentes razonadores de lógica descriptiva (DL) como, por ejemplo, HermiT, Pellet2, Fact++ o Racer (Sirin & Parsia, 2004).

I.6.3.2. Soporte de ontologías

La característica de dar soporte a múltiples ontologías permite a los sistemas de anotación semántica cubrir diferentes dominios. Existen dos estrategias que facilitan el soporte de múltiples ontologías por parte de los sistemas de anotación, a saber, mezclar todas las ontologías que se van a utilizar o definir explícitamente en las anotaciones la ontología concreta a la que se referencia. La unión de múltiples dominios puede dar lugar a una ontología muy grande que afecte negativamente al rendimiento del sistema debido a las restricciones computacionales que establece la utilización de este tipo de ontologías en sistemas de anotación semántica. Por ello, en este caso, es más apropiado la utilización de diversas ontologías de tamaño medio que una ontología en donde se fusionen todos los dominios. Además, en esta misma línea se han desarrollado diversas técnicas que permiten dividir grandes ontologías en varios módulos para hacerlas más manejables por los ordenadores (Grau et al., 2009).

Una de las principales ventajas de los sistemas de anotación semántica que soportan múltiples ontologías es que pueden cubrir diferentes dominios. Por ejemplo, KIM (Popov et al., 2003), CREAM (Handschuh & Staab, 2003) o Armadillo (Chapman et al., 2005) implementan esta funcionalidad, mientras que CERNO (Kiyavitskaya et al., 2009), S-CREAM (Handschuh et al., 2002), GoNTogle (Bikakis et al., 2010), MnM (Vargas-Vera et al., 2002) o EVONTO (Tissaoui et al., 2011) no incluyen esta característica.

El sistema de anotación que se describe en esta tesis doctoral proporciona soporte para múltiples ontologías. En concreto, implementa la segunda estrategia mencionada anteriormente para habilitar esta funcionalidad. Es decir, todas las anotaciones que se generan durante el proceso de anotación semántica tienen

definido explícitamente el modelo ontológico que ha sido utilizado para producir el metadato.

I.6.3.3. Soporte de tipos de formatos de documentos

La mayoría de herramientas de anotación semántica sólo proporcionan soporte para formatos basados en la Web como HTML o XML. Entre los sistemas analizados, S-CREAM (Handschuh et al., 2002) y CREAM (Handschuh & Staab, 2003) ofrecen funciones de anotación semántica únicamente para este tipo de documentos. Sin embargo, otras aproximaciones como, por ejemplo, KIM (Popov et al., 2003), MnM (Vargas-Vera et al., 2002) y Armadillo (Chapman et al., 2005) proporcionan soporte para analizar y anotar formatos de documentos alternativos. Finalmente, existen herramientas como, por ejemplo, CERNO (Kiyavitskaya et al., 2009), GoNTogle (Bikakis et al., 2010) y EVONTO (Tissaoui et al., 2011), que han sido específicamente diseñadas para soportar exclusivamente documentos con formato de texto.

El trabajo que proponemos en esta tesis ofrece soporte para diferentes tipos de formatos de documento, incluyendo los formatos de texto más importantes y formatos Web nativos.

I.6.3.4. Consistencia

Uno de los grandes problemas que tienen los sistemas de anotación semántica está relacionado con el mantenimiento de la consistencia entre los recursos anotados y los repositorios de ontologías donde las anotaciones son almacenadas. En este entorno, la consistencia está relacionada con mantener las relaciones de las expresiones lingüísticas del texto con los conceptos seleccionados de las ontologías. Esta consistencia se vería comprometida con, por ejemplo, la modificación del contenido de un documento previamente anotado. Cualquier tipo de cambio, ya sea modificar, añadir o eliminar información, debe ser reflejado en el repositorio de anotaciones semánticas. Si, por el contrario, el sistema no es capaz de detectar tales modificaciones en los documentos, las anotaciones de este documento en el repositorio serán inconsistentes. Además, si el sistema de

anotación soporta múltiples ontologías, entonces éste tiene que asegurarse de que se mantiene la coherencia y la consistencia de las anotaciones entre las ontologías utilizadas y los documentos cuando se produzcan modificaciones.

La mayoría de herramientas analizadas proporcionan soporte para mantener la coherencia y la consistencia en el caso de que se produzca cualquier tipo de modificación en los documentos. Por ejemplo, Armadillo (Chapman et al., 2005), CREAM (Handschuh & Staab, 2003), KIM (Popov et al., 2003) y EVONTO (Tissaoui et al., 2011) tienen implementadas funciones para actualizar las anotaciones en el caso de que se produzcan cambios en uno o varios documentos. Sin embargo, existen otras herramientas como S-CREAM (Handschuh et al., 2002), GoNTogle (Bikakis et al., 2010), MnM (Vargas-Vera et al., 2002) o CERNO (Kiyavitskaya et al., 2009) que no contemplan en su software tales funciones.

En esta tesis doctoral, el sistema de anotación planteado proporciona diversos mecanismos y metodologías para solucionar los problemas de inconsistencia e incoherencia que puedan surgir. Básicamente, los mecanismos dispuestos se basan en reinserción del recurso en el sistema para provocar su re-anotación.

I.6.3.5. Almacenamiento de las anotaciones

Actualmente, los sistemas de anotación semántica más innovadores utilizan dos enfoques para almacenar las anotaciones semánticas (Devedzic & Gaseviceds, 2009): el enfoque basado en un modelo semántico y el enfoque centrado en documentos. El primer enfoque está orientado a la Web Semántica y establece que las anotaciones deben ser almacenadas separadas del documento original. En cambio, el segundo enfoque presenta una visión centrada en el documento en la que se establece que las anotaciones deben ser almacenadas como parte integrante del documento.

En general, la gran mayoría de los sistemas actuales hacen uso de modelos de la Web Semántica, que permiten almacenar las anotaciones de los recursos en contenedores de información tales como bases de datos, aislando los documentos de sus anotaciones. De hecho, todas las herramientas que han sido analizadas implementan este tipo de almacenamiento.

La propuesta que se presenta en esta tesis doctoral también almacenará las anotaciones, siguiendo el enfoque semántico, en repositorios y no directamente en los documentos. Este enfoque de almacenamiento de anotaciones en un repositorio semántico centralizado facilita un entorno colaborativo que permite, incluso, que otros sistemas puedan hacer uso de estas anotaciones.

I.6.3.6. Automatización del sistema

Según el tipo de metodología de anotación implementada, se pueden distinguir tres tipos de sistemas de anotación semántica: manual, automático y semiautomático. Actualmente, los más extendidos son los semiautomáticos y los enfoques totalmente automáticos debido a que la anotación manual es una tarea que consume demasiado tiempo (Ciravegna et al., 2002a). De acuerdo con el tipo de anotación semántica soportada, entre las herramientas que se han analizado se distinguen CERNO (Kiyavitskaya et al., 2009), GoNTogle (Bikakis et al., 2010) y S-CREAM (Handschuh et al., 2002), que son herramientas que se basan en ofrecer una estrategia de anotación semiautomática, frente a Armadillo (Chapman et al., 2005), CREAM (Handschuh & Staab, 2003), MnM (Vargas-Vera et al., 2002), KIM (Popov et al., 2003), EVONTO (Tissaoui et al., 2011) y la herramienta que se presenta en esta tesis doctoral, que disponen de una estrategia de anotación totalmente automática.

La necesidad de integrar tecnologías de extracción de conocimiento totalmente automatizadas en los enfoques de anotación semánticas es particularmente acuciante cuando el proceso de anotación se va a realizar sobre grandes colecciones de documentos.

I.6.3.7. Evolución de ontologías

Como se mencionó anteriormente, la evolución de ontologías se puede definir como la adaptación oportuna de una ontología y la propagación constante de estos cambios a los artefactos dependientes (Stojanovic et al., 2002). Concretamente, se refiere al proceso de cambio que sufren las ontologías con el paso del tiempo como, por ejemplo, añadir nuevas clases o instancias, modificar clases e instancias o

eliminar conocimiento. Los sistemas de anotación semántica que integran procesos de evolución de ontologías deben garantizar la coherencia de las anotaciones con respecto a las ontologías que se están modificando.

Soportar la evolución ontológica implica que el sistema de anotación debe mantener la consistencia y la coherencia entre anotaciones y ontologías. EVONTO (Tissaoui et al., 2011), KIM (Popov et al., 2003), S-CREAM (Handschuh et al., 2002) y CREAM (Handschuh & Staab, 2003) implementan un enfoque de evolución de ontologías que atiende esta necesidad. Por el contrario, CERNO (Kiyavitskaya et al., 2009), GoNTogle (Bikakis et al., 2010), MnM (Vargas-Vera et al., 2002) y Armadillo (Chapman et al., 2005) no cubren esta función.

Tal y como se muestra en el Capítulo II, donde se describe la funcionalidad del sistema de anotación, el sistema que se presenta en esta tesis doctoral es totalmente compatible con la evolución de ontologías a partir de Wikipedia[10].

I.7. PROBLEMA A RESOLVER EN ESTA TESIS DOCTORAL

Las ontologías son tecnologías que permiten representar el conocimiento de manera estructurada y formal para que la información sea entendible no solo por el ser humano sino también por los sistemas de computación de manera automática. Además, constituyen los vocabularios de anotaciones semánticas que proporcionan los metadatos con los que enriquecer los recursos digitales. Una de las ventajas que proporciona la utilización de estos metadatos, en los motores de búsqueda es la mejora en la calidad de los resultados que se obtienen en tareas de recuperación de información gracias a la explotación de la semántica introducida. La búsqueda asistida semánticamente ayuda a la búsqueda tradicional, mejorándola, sin llegar a sustituirla. La búsqueda semántica acepta nuevos criterios de búsqueda como, por ejemplo, la búsqueda de palabras clave, por propiedades de los metadatos y consultas en lenguaje natural. Un buscador semántico identifica el contexto o la intención del usuario, de lo que quiere buscar, y realiza un acceso inteligente al conocimiento, a través de las anotaciones, aplicando algoritmos y criterios semánticos para recuperar la información

[10] https://www.wikipedia.org

requerida, aumentando la precisión de los resultados con respecto a los buscadores tradicionales.

Analizando las metodologías de anotación que se basan en modelos semánticos, se puede observar que la mayoría de las propuestas presentan limitaciones que impiden su implantación como solución plausible. Uno de los principales problemas que afecta a los sistemas de anotación basados en el modelo semántico es la evolución de la ontología subyacente. La mayoría de herramientas no disponen de un único dominio ontológico si no que utilizan múltiples ontologías para subsanar esta limitación. Además, la mayoría de metodologías incluyen modelos básicos de similitud que proporcionan funciones básicas de extracción y recuperación de información. Sin embargo, en esta investigación se ha considerado que, para que una metodología de anotación semántica posea una aplicabilidad real, debe, en primer lugar, ser totalmente automática, si bien esto no significa que no se deba tener en cuenta la anotación manual. Además, el modelo semántico subyacente debe atender a su naturaleza evolutiva y ser capaz de enriquecerse y evolucionar para subsanar los cambios que todos los entornos de aplicación sufren. Otra característica de la solución perseguida en este trabajo es que debe incluir modelos de similitud complejos que exploten los metadatos semánticos incorporados en los documentos para ofrecer búsquedas más eficientes.

A partir de las carencias detectadas en las metodologías de anotación semántica actuales, el trabajo de esta tesis se planteó para dar solución a los problemas indicados anteriormente. Para el desarrollo de esta solución se plantearon los siguientes objetivos:

- **Definición de una metodología de anotación semántica.**
    - La metodología incluirá las siguientes características:
        - Extracción de términos basada en algoritmos estadísticos y lingüísticos.
        - Evolución de las ontologías del dominio.
        - Anotación semántica automática capaz de anotar cualquier tipo de información textual ya sea estructurada, semiestructurada o no estructurada.

        - Indexación semántica que permita representar los conceptos que están relacionados con un contenido textual dado.
        - Búsqueda semántica basada en modelos estadísticos de búsqueda y recuperación de información.

- **Validación de la metodología desarrollada.**
    - La metodología desarrollada se validará siguiendo los pasos que se relatan a continuación: (i) definición de los dominios de validación, computación en la nube y Tecnologías de la Información y Comunicación (TIC); (ii) validación de la metodología de extracción de términos en cada dominio definido; (iii) validación de la metodología automática de evolución de ontologías en el dominio de computación en la nube y TIC; (iv) validación del motor de búsqueda semántica en ambos dominios.
- **Definición de una metodología de similitud basada en la anotación semántica.**
    - La metodología incluirá las siguientes características:
        - Construcción de un repositorio semántico que contenga una ontología que modele el dominio de una organización de Investigación, Desarrollo e innovación (I+D+i) y ontologías que proporcionen descripciones semánticas sobre proyectos y recursos humanos.
        - Extracción de información de un repositorio semántico.
        - Definición de diferentes modelos de cálculo de similitud para cada tipo de dato que va a ser almacenado en el repositorio semántico.
        - Motor de inferencia basado en similitud semántica.
- **Validación de la metodología desarrollada.**
    - La metodología desarrollada se validará siguiendo los pasos que se relatan a continuación: (i) definición del dominio de validación; (ii) validación de la metodología de búsqueda y recuperación de información en una organización I+D+i.

I.8. RESUMEN

En la primera sección, se incluye un estudio en profundidad acerca de la Web Semántica y las ontologías. En el caso de la Web Semántica, el estudio comienza definiéndola y proporcionado un breve estudio histórico que recoge los antecedentes de la Web actual. En este estudio se analiza la World Wide Web desde sus orígenes hasta los inicios de la Web Semántica. Después, se remarcan los fundamentos que estimularon la necesidad de diseñar una nueva Web. Además, en este apartado presenta una comparación entre la Web normal con la Web Semántica para detallar su evolución. Para terminar se incluye un estudio donde se analiza y detallan las capas de la arquitectura de la Web Semántica.

El estudio de las ontologías comienza con una extensa definición donde se muestran las diferentes acepciones que han sido proporcionadas por diversos autores con el paso del tiempo. Posteriormente, el apartado I.2.2.2 se centra en analizar varios criterios establecidos en la literatura que se utilizan para clasificar las ontologías. Cada clasificación analizada presenta un desglose de los tipos de ontologías que contiene y una breve definición de la misma. Después, en el apartado I.2.2.3 se analizan dos aspectos muy relevantes dentro del mundo de las ontologías, por un lado los elementos que componen una ontología y por otro lado los lenguajes de representación de ontologías más actuales.

En la segunda sección, se realiza un estudio del arte de los diferentes enfoques existentes para la evolucionar ontologías. Además, el apartado I.3.3, incluye un análisis acerca de los problemas que supone la difícil tarea de llevar a cabo el evolucionar de ontologías. Por último, entre los diferentes paradigmas definidos para evolucionar la ontología, se describe el aprendizaje de ontologías como técnica de construcción ontologías basadas en técnicas de extracción de información. En este apartado se hace alusión a esta tecnología debido a la gran influencia que tiene en la propuesta que se presenta.

En la tercera sección, se describe el procesamiento del lenguaje natural que junto con la Web Semántica y las ontologías establecen las bases para las demás tecnologías que se utilizan en esta tesis. El estudio de esta tecnología comienza con el análisis de varias definiciones propuestas por diferentes autores. Posteriormente, se describe la evolución que ha sufrido esta área de investigación

desde sus orígenes a finales de 1940 hasta los últimos años. Para concluir esta sección, se presenta una análisis del procesamiento del lenguaje natural en los niveles lingüísticos existentes. Este análisis pretende remarcar los diferentes niveles lingüísticos que cubre el procesamiento del lenguaje natural resaltando que técnicas son utilizadas en cada uno de los niveles.

En la cuarta sección, se analizan las técnicas de extracción y de recuperación de información, muy relacionadas con la evolución de ontologías y más concretamente con el aprendizaje de ontologías (del inglés, "*Ontology Learning*"). El estudio se inicia con el análisis de varias definiciones propuestas por autores que aportan diferentes enfoques sobre la definición de la extracción y de la recuperación de información. Concretamente, debido a la relación existente entre el tipo de información y los sistemas de extracción, esta sección recoge un breve estudio de los diferentes tipos de información actuales. Además la explicación detalla algunas de los modelos y metodologías utilizadas para extraer la información en cada tipo. Para concluir esta estudio de técnicas de extracción, se describen las tareas más sobresalientes que son utilizadas por los sistemas de extracción de información. Después, se presenta una clasificación de las técnicas de recuperación de información. Esta clasificación utiliza como criterio la exactitud del modelo de coincidencia utilizado por la técnica de recuperación de información. Así, se destacan dos modelos: modelo de coincidencia exacta y modelo de coincidencia parcial. A partir de estos dos modelos se construye toda la jerarquía de técnicas de recuperación de información. Por último, se describe dos de las técnicas más relevantes de recuperación de información en cada uno de estos modelos de coincidencia.

En la quinta sección, se profundiza sobre la anotación semántica. Esta sección comienza con el análisis de varias definiciones existentes en la literatura. Además, en esta sección se distinguen diversos tipos de anotación: anotación manual, anotación semiautomática y anotación automática. Por último, para concluir esta sección se presenta un estudio del arte de un conjunto de herramientas de anotación semántica. Este estudio del arte analiza cada una de las herramientas en profundidad explicando aspectos técnicos relacionados con su arquitectura y funcionalidad. Este análisis de herramientas es utilizado, en el apartado final de

esta sección, para confeccionar una tabla comparativa que recoge las herramientas de anotación analizadas junto con la que se propone en esta tesis.

Por último, la ultima sección I.11 explica el problema que se resuelve en esta tesis doctoral, haciendo hincapié en las carencias de las metodologías de anotación existentes y cómo la aproximación que se presenta en esta tesis cubre tales carencias. Además, esta sección recoge una descripción de la metodología que se ha seguido para lograr el desarrollo del sistema que se propone en esta tesis. En esta descripción se enumeran cada uno de los objetivos establecidos en las primeras fases de la investigación y también describe cada una de las validaciones que se aplicaron sobre cada objetivo establecido.

Capítulo II. ANOTACIÓN SEMÁNTICA.

II.1. INTRODUCCIÓN

Como se ha comentado anteriormente, la Web Semántica es una extensión de la Web orientada a subsanar los problemas de interoperabilidad y la ineficacia en la gestión de la información. Los pilares en los que la Web Semántica se sostiene son las anotaciones y las ontologías. Las ontologías permiten describir formalmente conceptualizaciones compartidas de un dominio utilizando distintos tipos de elementos, de entre los que podemos destacar los siguientes: conceptos, que representan a los tipos de objetos del mundo real; relaciones, que definen la forma en la que los conceptos interactúan dentro del dominio; axiomas, que son las reglas y restricciones que siempre son verdad en el dominio; e instancias, que representan a los objetos del dominio. Por otro lado, las anotaciones son metadatos que hacen referencia a conceptos e instancias en las ontologías y se asocian a texto en lenguaje natural, lo que supone ciertos beneficios para los motores de búsqueda. En concreto, las anotaciones permiten que estos motores sean capaces de entender el contexto de búsqueda definido por el usuario y de proporcionar un acceso a la información más rápido, eficiente y preciso.

Este capítulo representa la parte central de esta tesis doctoral. El objetivo del mismo es describir una nueva metodología para anotar recursos a partir de un modelo ontológico predefinido. El capítulo explica en profundidad la arquitectura del sistema de anotación semántica propuesto, que combina técnicas de extracción de información y evolución de ontologías.

El sistema recibe como entrada un conjunto de textos en lenguaje natural a anotar. Previamente al proceso de anotación, los recursos son analizados por el extractor de términos, que confecciona una lista de posibles términos a partir de técnicas estadísticas. Esta lista de términos contiene los candidatos potenciales a convertirse en conceptos dentro de la ontología durante su proceso de evolución. Después, los recursos son anotados semánticamente con respecto a una ontología del dominio. Estas anotaciones se almacenan en una base de datos relacional junto

con un valor numérico que representa la relevancia del concepto anotado con respecto al recurso particular y al grupo de recursos anotados. Todos los valores de las anotaciones de un documento simbolizan un índice semántico que representa el contenido del documento. El buscador semántico compara la búsqueda introducida por el usuario con respecto a este índice para obtener los recursos que cumplan con lo especificado en la búsqueda. Por otro lado, el trabajo presentado en esta tesis también incluye la definición de una metodología de evolución de ontologías que parte de una ontología. con un modelo semántico establecido, y que permite enriquecer la ontología mediante la incorporación de nuevos conceptos extraídos de los recursos analizados en la entrada del sistema utilizando la estructura jerárquica de categorías de la Wikipedia.

II.2. DESCRIPCIÓN DEL PROBLEMA Y OBJETIVOS

El objetivo de este capítulo es la definición de un nuevo método de anotación semántica basado en un modelo ontológico que integre la evolución de ontologías para la adaptación de la ontología a los cambios en el dominio. Esta metodología, además, puede anotar cualquier tipo de recurso, independientemente del tipo de información que contenga, del dominio que trate e incluso del idioma.

Este objetivo principal se puede desglosar en los siguientes sub-objetivos:

- Desarrollo de un método de extracción automática de términos a partir de textos en lenguaje natural independiente del idioma.
- Desarrollo de un método de evolución de ontologías a partir de estructuras jerárquicas.
- Desarrollo de un método de anotación y recuperación semántica que permita la anotación de cualquier documento escrito en lenguaje natural, independientemente de su formato, dominio o idioma. Este método incorporará un buscador semántico que permita recuperar los recursos anotados a partir de una búsqueda basada en palabras clave.

Además, se han tenido en cuenta los siguientes objetivos secundarios para mejorar las prestaciones de este método frente a las alternativas existentes actualmente:

- **Independencia del dominio y multilenguaje**. Uno de los aspectos más importantes que se persiguen en el diseño de este sistema es que no esté limitado a un solo dominio ni tampoco a un solo idioma. Por lo tanto, el sistema define una metodología de anotación con soporte de evolución del modelo semántico adaptable a cualquier dominio y a cualquier idioma.
- **Eficiencia y flexibilidad**. El software utilizado para el desarrollo del sistema se caracteriza por ser eficiente. Además, el modelo de diseño de software que se ha mantenido durante todo el desarrollo del sistema es lo suficientemente flexible para permitir el intercambio de elementos sin que ello afecte ni a la eficiencia ni al rendimiento del sistema.
- **Adaptabilidad y parametrización**. La adaptabilidad y parametrización son dos aspectos de relevante importancia durante todo el desarrollo del sistema. De ahí, que incluya un método que permita evolucionar el modelo ontológico subyacente para facilitar la adaptación dinámica del sistema. Además, el funcionamiento del sistema viene, en gran medida, determinado por ficheros de configuración que permiten configurar diversos parámetros de los distintos módulos y otros aspectos del sistema.

II.3. ARQUITECTURA DEL SISTEMA DE ANOTACIÓN SEMÁNTICA

El sistema de anotación semántica que presentamos en este capítulo se encuentra compuesto por cinco módulos. La Figura II.1 muestra una representación modular de la arquitectura del sistema.

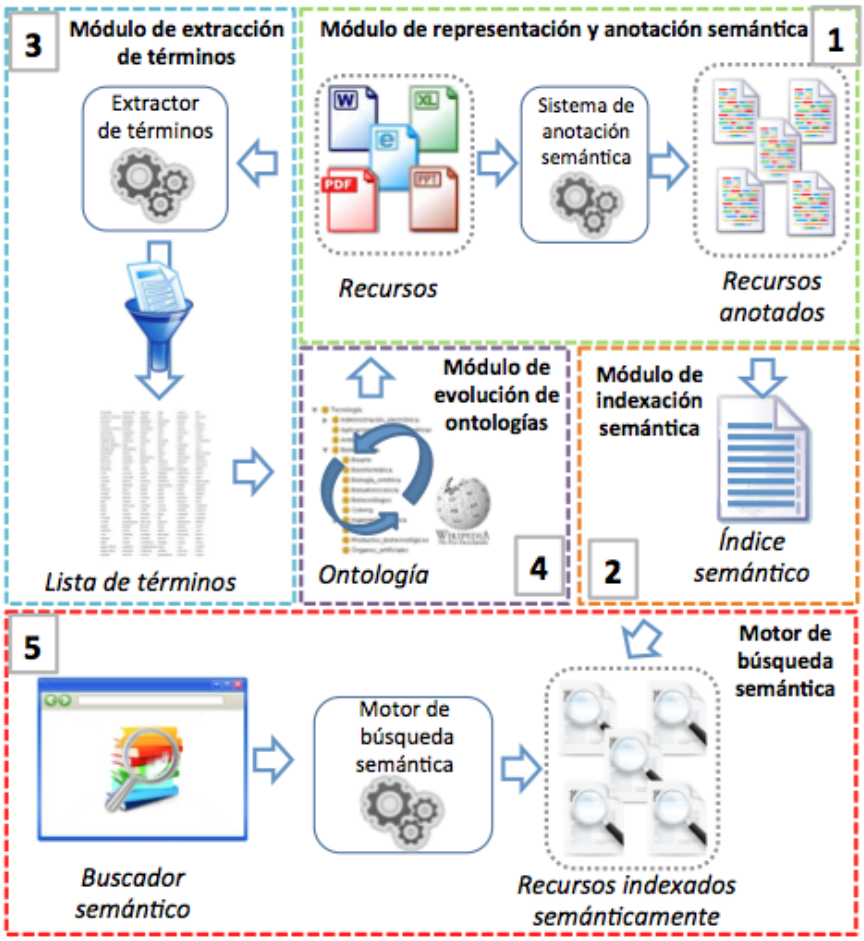


Figura II.1 Sistema de Anotación Semántica

El funcionamiento del sistema es como sigue. El módulo de representación y anotación semántica (1), basándose en un modelo ontológico predefinido, crea un conjunto de anotaciones basado en conceptos y propiedades "`rdfs:label`" de la ontología. Este conjunto de anotaciones se utilizará durante el proceso de anotación semántica para identificar las expresiones lingüísticas que el módulo intentará asociar con los conceptos definidos en el dominio ontológico. Después, el módulo de indexación (2) recupera las anotaciones para enriquecer cada anotación con información procedente del dominio ontológico. Este enriquecimiento aprovecha las relaciones taxonómicas de la ontología para incorporar nueva información a cada anotación recuperada. A continuación, el módulo de indexación semántica (2), mediante un modelo matemático extendido, obtiene un valor que define cómo de relevante es cada anotación con respecto al corpus de recursos anotados. Para un mismo recurso el conjunto de valores obtenidos relacionados con un mismo recurso constituirá un índice semántico. La

función de este índice semántico es organizar las anotaciones definidas por criterios de relevancia de información con el fin de facilitar su tarea al motor de búsqueda semántico (5).

Por otro lado, el módulo de extracción de términos (3) tiene la función de identificar los términos más relevantes en un recurso de texto. El funcionamiento de este módulo se basa en la metodología presentada en (Ochoa et al., 2011). Básicamente, esta metodología utiliza patrones lingüísticos y medidas estadísticas para obtener una lista de términos candidatos que pueden ser palabras simples o términos compuestos. Estos términos son los candidatos a ser definidos como conceptos dentro de la ontología. La lista de términos será remitida al módulo de evolución de ontologías (4). Para cada término, el módulo de evolución (4), utilizando la jerarquía de información dispuesta en Wikipedia, intentará encontrar una categoría de unión entre el término y cualquier concepto definido en la ontología.

Por último, el motor de búsqueda semántico (5) tiene como objetivo recuperar toda la información relacionada con consultas basadas en palabras clave. Este módulo se aprovecha de las anotaciones y los índices semánticos recopilados por el sistema para llevar a cabo la búsqueda.

II.3.1. MÓDULO DE REPRESENTACIÓN Y ANOTACIÓN SEMÁNTICA (1)

El módulo que se describe a continuación es el encargado de proporcionar las funciones de anotación, es decir, la función que desempeñará será insertar información lingüística interpretativa a un corpus compuesto por información no estructurada o semiestructurada. Esta información lingüística se obtiene del modelo ontológico definido anteriormente donde cada concepto era expresado en términos de conjuntos de sinónimos. Así, un concepto se puede definir como un conjunto de términos con un significado común (Reiterer et al., 2010). Es decir, un conjunto de términos sinónimos representarán un concepto. Para poder representar los sinónimos en nuestro modelo ontológico es necesario tener en cuenta varios tipos de anotaciones en los elementos de las ontologías representadas en formato RDF y OWL, a saber:

- "rdfs:comment": se utiliza para describir cada elemento de la ontología mediante una explicación entendible por un humano. Un mismo concepto podrá contener una o varias anotaciones de este tipo en el mismo o distintos idiomas.

- "rdfs:label": anotación que representa cómo se nombra el concepto en lenguaje natural. Por lo tanto, cada concepto en el modelo ontológico podrá tener tantas anotaciones de este tipo como términos relacionados disponga. Por ejemplo, si la ontología incluye el concepto "Procesamiento del lenguaje natural" que puede definirse mediante los términos "Procesamiento del lenguaje natural" y "PLN" en español, o "Natural Language Processing" y "NLP" en inglés, entonces, esto significa que este concepto puede incorporar cuatro propiedades "rdfs:label", una por cada término.

- "preferredTerm": anotación utilizada para definir el término preferido que identifica al concepto que lo contiene. Partiendo del ejemplo anterior, si tenemos el concepto "Procesamiento de Lenguaje natural" en la ontología, el "preferredTerm" podría ser "Procesamiento de Lenguaje natural". En este caso, el término "Procesamiento de Lenguaje natural" sería el término preferido para identificar el concepto "Procesamiento de Lenguaje natural" en la ontología.

- "URIresource": anotación que se emplea para almacenar la URI que relaciona el concepto definido en la ontología con el recurso de Wikipedia que ha sido utilizado para definir ese concepto.

La definición de estas anotaciones favorece la independencia del idioma. Por otro lado, cualquier dominio modelado con una ontología que defina una taxonomía de conceptos relacionados y que tenga estas anotaciones en cada concepto podrá ser utilizada como ontología del dominio dentro del sistema de anotación semántica propuesto en esta tesis.

La Figura II.2 representa una ontología que modela el dominio de las Tecnologías de la Información y la Comunicación (TIC). En esta representación se puede observar una estructura jerárquica bilingüe de conceptos relacionados.

Cada concepto en la jerarquía está representado por dos términos mediante la anotación "`rdfs:label`" en dos idiomas diferentes, español e inglés.

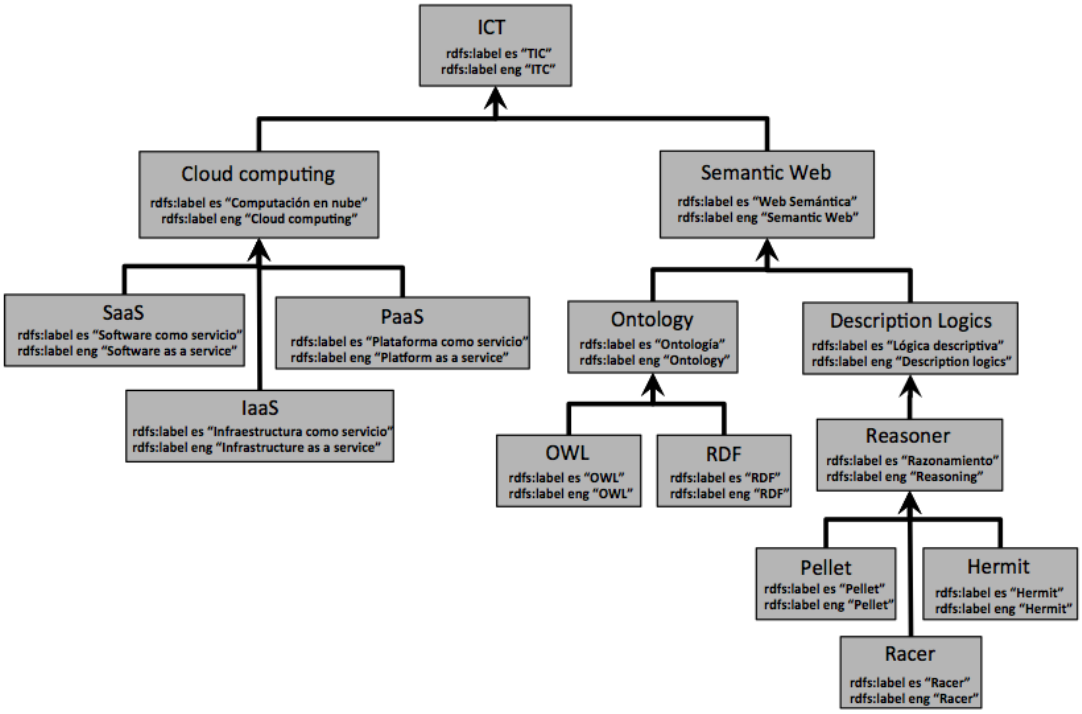


Figura II.2 Ejemplo de modelo ontológico

El módulo de representación de conocimiento se ha diseñado para definir cómo se representará la información procesada dentro del sistema de anotación. Además, incorpora funciones para extraer conocimiento de la ontología y generar diccionarios de términos, conocidos como gazetteers, que serán utilizados por la herramienta de anotación semántica integrada en el módulo.

II.3.1.1. Metodología de anotación semántica

El desarrollo de la metodología de anotación semántica ha requerido la integración de dos librerías: OWL API[11] y GATE. OWL API es un proyecto de código abierto bajo la licencia LGPL y Apache que ofrece una interfaz de programación de ontologías escrita en Java y que proporciona las estructuras de datos necesarias

[11] http://owlapi.sourceforge.net

para crear, manipular y serializar ontologías en formato OWL. Por otro lado, GATE es una arquitectura, un entorno de desarrollo y un framework que permite crear sistemas de procesamiento del lenguaje humano. En concreto, esta arquitectura presenta un sistema de reconocimiento de entidades que utiliza diccionarios de palabras denominados gazetteers. La integración de ambas herramientas facilita el desarrollo de un sistema de anotación semántica a partir de los conceptos definidos en la ontología y las anotaciones "`rdfs:label`" comentadas anteriormente.

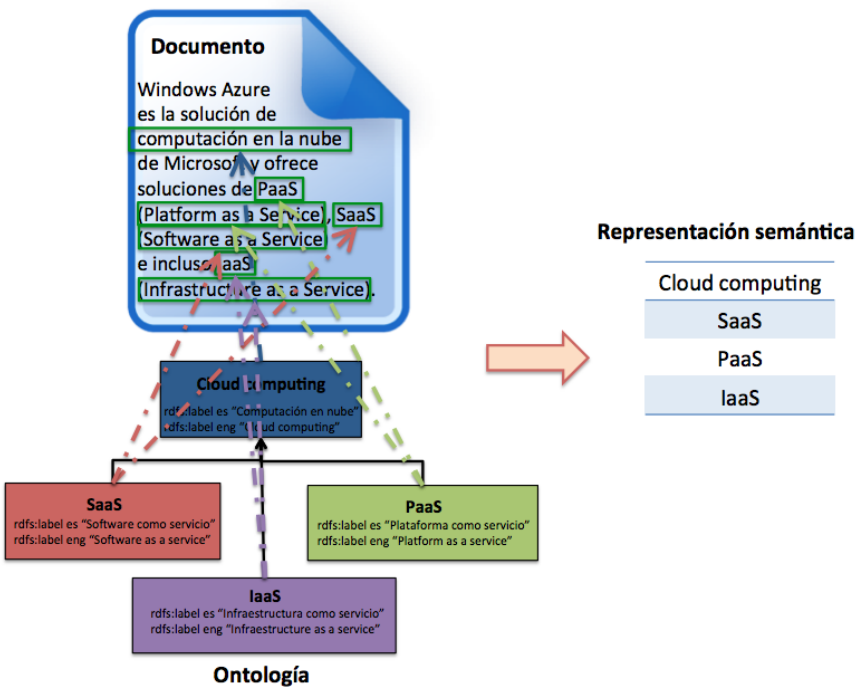


Figura II.3 Proceso de representación y anotación semántica

La Figura II.3 describe gráficamente la metodología de anotación semántica. En primer lugar, para que se lleve a cabo la anotación semántica se requiere de (i) una ontología del dominio que, en este escenario, modela el dominio de las tecnologías de computación en la nube, y (ii) un documento escrito en lenguaje natural relacionado con este dominio. Partiendo de la base de que el diccionario de palabras (gazetteer) ha sido previamente construido, la herramienta GATE utilizaría el sistema de reconocimiento de entidades con esta lista de palabras para detectar y anotar todas las palabras del texto que se encuentren definidas en la ontología. En la Figura II.3 se representan gráficamente las anotaciones mediante

la relación de los conceptos en la ontología con fragmentos del texto. Al final del proceso se obtiene una representación semántica en forma de vector que contiene el conjunto de conceptos extraídos del modelo ontológico que han sido utilizados durante el proceso de anotación.

En la Figura II.4 se describe la arquitectura del módulo de representación y anotación semántica. Esta imagen representa gráficamente la actividad que desarrolla el módulo, los elementos que requiere y las herramientas que integra para realizar la tarea de anotación. El funcionamiento de la arquitectura que se presenta en la figura comienza con la construcción de un gazetteer que contenga representaciones semánticas de cada concepto definido en la ontología. Estas representaciones se definen a partir de las propiedades de anotación "`rdfs:label`" y conceptos definidos en la ontología. Un gazetteer constituye, por tanto, un diccionario de palabras que contiene todos los conceptos de la ontología, así como los sinónimos que se definen para cada concepto, esto es, todos los términos que identifican a cada concepto dentro de la ontología de acuerdo con sus anotaciones "`rdfs:label`". El gazetteer representa la parte más importante del módulo de anotación debido a que define todos los metadatos que son utilizados durante la anotación. A partir del gazetteer definido y los recursos textuales proporcionados, el módulo de anotación semántica intenta identificar las expresiones lingüísticas en esos recursos textuales a las que se pueden asociar las anotaciones definidas en el gazetteer. Estas anotaciones son almacenadas en formato RDF en un repositorio semántico implementado con Virtuoso (Virtuoso, 2009).

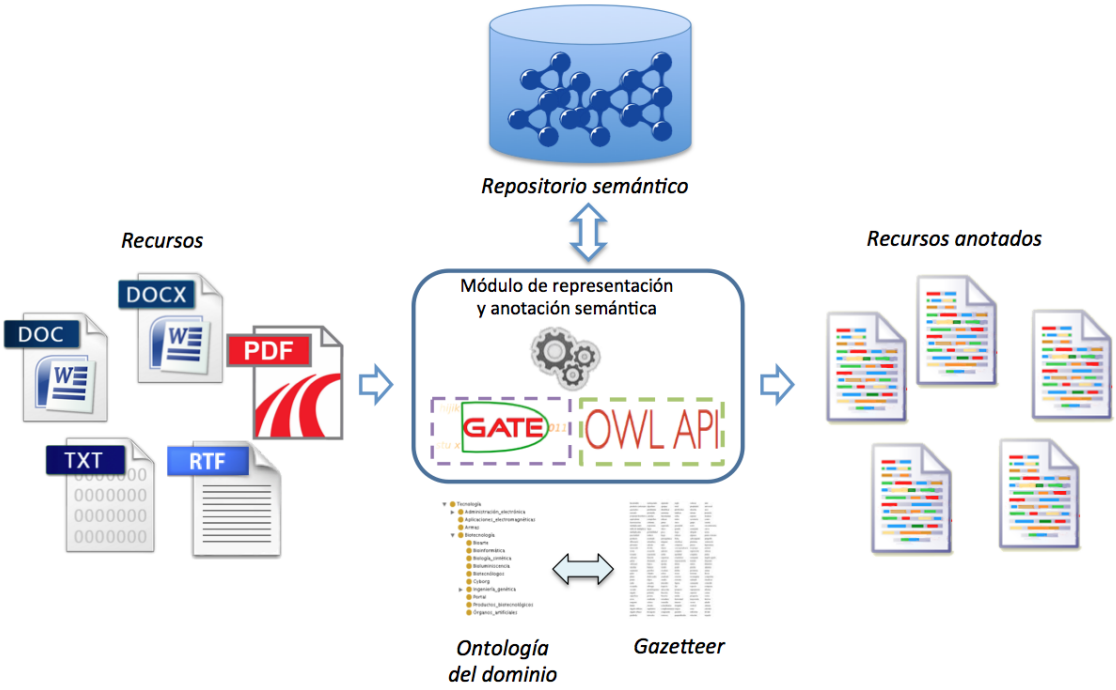


Figura II.4 Arquitectura del módulo de Representación y anotación semántica

A continuación, se muestra un ejemplo en el dominio de las TIC. La Figura II.5 muestra la ontología utilizada en este dominio. En ésta se definen los conceptos que el módulo de representación y anotación semántica empleará para crear el gazetteer con el que realizar las anotaciones. El sistema de reconocimiento de entidades de GATE utilizará esta lista de palabras para crear las anotaciones semánticas que relacionan conceptos del dominio ontológico con los fragmentos de texto que hagan referencia a estos conceptos.

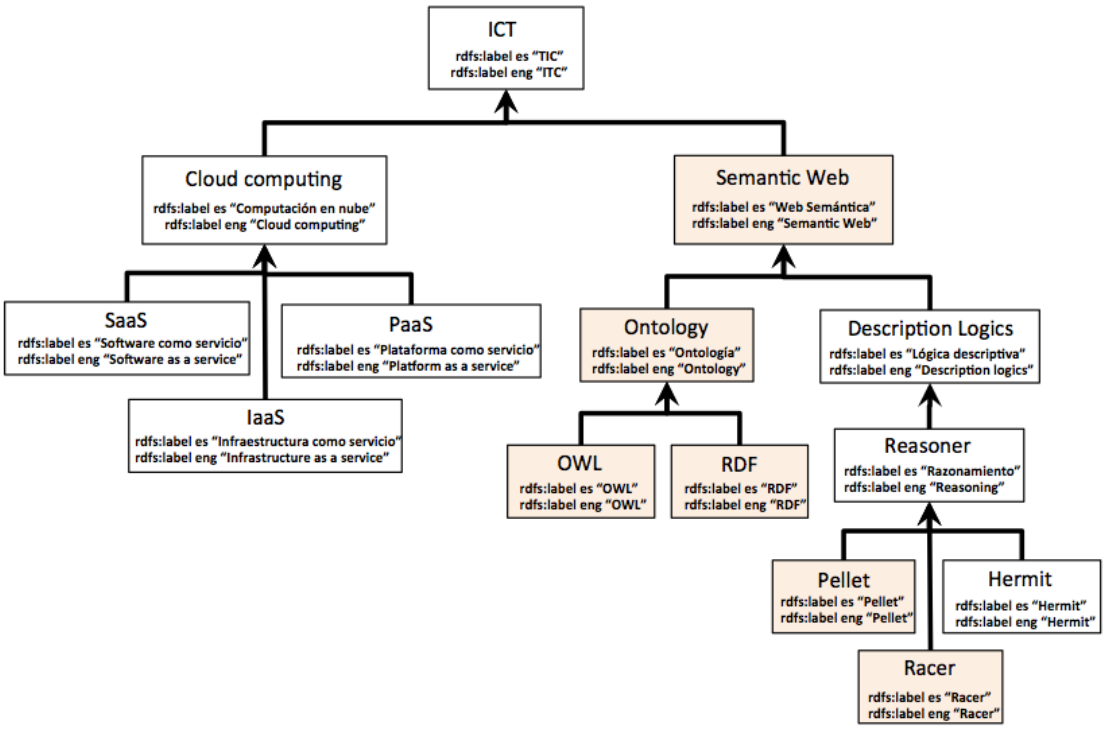


Figura II.5 Ontología de tecnologias de información y comunicación

Seguidamente, se presentan los tres documentos diferentes (véase Figura II.6, Figura II.7 y Figura II.8) que se van a utilizar en el ejemplo. Como se puede apreciar en las figuras, el tema de los documentos seleccionados debe coincidir con el dominio de la ontología para que el módulo sea capaz de detectar fragmentos de texto relacionados con la ontología del dominio y anotarlos semánticamente.

Sistema inteligente. Su misión principal será la de asistir a la Unidad de Gestión de la I+D+i en la toma de decisiones estratégicas sobre la Gestión de la I+D+i sirviéndose de las tecnologías descritas. Gracias al razonamiento y la inferencia de las ontologías se podrán detectar situaciones inconsistentes e inferir nuevo conocimiento útil para el sistema. Por ejemplo, se podrán detectar situaciones irregulares como que los recursos asignados a un proyecto de I+D+i no son los más idóneos, así como que las tareas o el presupuesto de un proyecto excede los límites permitidos en una determinada convocatoria de subvenciones de proyectos de I+D+i. Para ello se utilizarán los razonadores sobre OWL-DL existentes como Pellet o Hermit.

Figura II.6 Documento 1

Repositorio de ontologías. Este repositorio también permitirá la extracción de información semántica sobre los distintos aspectos que conforman el sistema de gestión de la I+D+i a partir de bases de datos relacionales que contengan información sobre los mismos. Actualmente los repositorios de ontologías y más concretamente de RDF no son tan eficientes como las bases de datos relacionales. Además, las distribuciones actuales de repositorios RDF de código libre presentan bastantes problemas de escalabilidad y eficiencia en la inserción y recuperación de tripletas con grandes cantidades de datos. La tecnología de almacenamiento y recuperación de información en bases de datos está mucho más consolidada y las distribuciones libres de bases de datos como MySQL y Postgres tienen muy buenos resultados de escalabilidad y eficiencia con grandes cantidades de datos.

Figura II.7 Documento 2

Sistema de extracción semántica desde BBDD. La Web Semántica está ganando madurez y existe una necesidad creciente para poder representar toda la información contenida en bases de datos en formatos semánticos como RDF u OWL. Algunas herramientas se han desarrollado actualmente para poder realizar mapeos entre los esquemas de las bases de datos relacionales y ontologías en OWL/RDF. Este módulo pretende extraer la semántica de las bases de datos para poder explotarla como ontologías OWL o grafos RDF para que puedan ser accedidas mediante SPARQL. Este módulo entonces nos servirá para poder extraer información semántica a partir de las BD del sistema de innovación y así poder tratar esos datos de manera semántica para poder realizar búsquedas más eficientes e inferir nuevo conocimiento. Para el desarrollo de este módulo se hará uso de la plataforma D2RQ para ayudar a la realización de esos mapeos. Este sistema será el que nos permita explotar las ventajas de las tecnologías semánticas sin las penalizaciones correspondientes, comentadas anteriormente.

Figura II.8 Documento 3

En la ontología de la Figura II.5 se ha coloreado cada concepto relacionado con alguna de las anotaciones detectadas, durante el proceso de anotación semántica. En las figuras Figura II.6, Figura II.7 y Figura II.8 aparecen resaltadas las anotaciones obtenidas por este módulo.

Como resultado del proceso de anotación, se obtiene una representación semántica de la información contenida en los documentos. Esta representación está compuesta por todas las anotaciones que han sido realizadas en cada uno de los documentos. La Tabla II.1 muestra las representaciones semánticas de cada documento anotado.

Tabla II.1 Representación semántica de los documentos

Representación semántica del documento		
Documento 1	**Documento 2**	**Documento 3**
Ontología	Web Semántica	Web Semántica
OWL	Ontología	RDF
Pellet	RDF	OWL
Hermit	Cloud computing	Ontología
TIC	TIC	Cloud computing
Web Semántica	OWL	TIC
RDF	Descripción lógica	Descripción lógica
Descripción lógica	Razonador	Razonador
Razonador		
Racer		

II.3.2. MÓDULO DE INDEXACIÓN SEMÁNTICA (2)

El módulo de indexación semántica recibe las anotaciones semánticas definidas por el módulo de representación y anotación semántica, para generar índices semánticos que faciliten el proceso de búsqueda y recuperación de información.

La generación de índices semánticos se lleva a cabo aplicando algoritmos que se han utilizado ampliamente en el área de extracción de información y que permiten explotar las anotaciones semánticas para calcular la relevancia de cada anotación dentro del documento. Hoy día existen diversos algoritmos que permiten calcular esta relevancia como, por ejemplo, TF-IDF (del inglés, "*Term Frequency – Inverse Document Frequency*") (Salton & McGill, 1983), Okapi BM25 (Robertson & Walker, 1999), o Kullbacl-Leiber divergence (Kullback & Leibler, 1951), entre otros.

En esta tesis, aunque se han provisto de las interfaces necesarias para utilizar cualquier algoritmo de relevancia, se ha seleccionado el algoritmo TF-IDF (Salton & McGill, 1983). Este algoritmo permite conocer cómo de discriminante es un término en relación a un documento, en particular, y a un conjunto de documentos, en general.

$$\mathrm{tf} * \mathrm{idf} = tf(t,d) * idf(t,D) \qquad \textbf{(II.1)}$$

Como se puede apreciar en la fórmula(II.1), la medida TF-IDF se compone de dos partes bien diferenciadas: la frecuencia de un término (TF, del inglés, "*Term Frequency*"), que obtiene la frecuencia de aparición de un término 't' en un documento 'd' (véase fórmula (II.2)), y la frecuencia inversa del documento (IDF, del inglés, "*Inverse Document Frequency*"), que obtiene la frecuencia de aparición de ese mismo término 't' en todos los documentos 'd' que forman parte de la colección que se ha anotado 'D' (véase fórmula (II.3)).

$$\mathrm{tf}(\mathrm{t},\mathrm{d}) = \frac{t}{d} \qquad \textbf{(II.2)}$$

$$\mathrm{idf}(\mathrm{t},\mathrm{D}) = log\ \frac{|D|}{|\{d \in D : t\ \in d\}|} \qquad \textbf{(II.3)}$$

Las ontologías organizan los conceptos basándose en jerarquías. Las jerarquías representan relaciones taxonómicas entre los conceptos. Esta organización nos permite introducir medidas de similitud semántica que tratan la interrelación existente entre dos conceptos en un contexto (Rada et al., 1989). La incorporación de esta medida en el cálculo de la relevancia de información provocó la

redefinición de la fórmula TF-IDF dando lugar a una versión extendida. La nueva versión de la fórmula TF-IDF no sólo calcula la relevancia de la información teniendo en cuenta las anotaciones semánticas que aparecen en el documento, sino que además obtiene un valor para los conceptos que, aunque no aparezcan explícitamente en el documento, se encuentran relacionados en la ontología con los conceptos que sí aparecen en el mismo. La fórmula (II.4) muestra los cálculos necesarios para generar el valor del TF-IDF extendido.

$$tf - idf_{extendido} = \sum_{j=1}^{n} \frac{tf - idf_{j,d}}{e^{dist(i,j)}} \qquad \textbf{(II.4)}$$

Donde '`dist(i,j)`' representa la distancia entre el concepto '`i`' y el concepto '`j`' en la ontología del dominio. El cálculo de esta distancia se basa en el algoritmo de caminos mínimos propuesto por Dijkstra (Dijkstra, 1959) para grafos. En este contexto, las aristas del grafo representan las relaciones taxonómicas del modelo ontológico y los nodos representan los conceptos definidos en la ontología. A diferencia del planteamiento en el algoritmo de Dijkstra, su aplicación en este ámbito no persigue encontrar el camino más corto entre dos vértices de un grafo, ya que todas las aristas tienen el mismo valor, la unidad. En su lugar, la utilización de este algoritmo permite calcular la distancia semántica existente entre dos conceptos definidos en la ontología. El objetivo de este cálculo es enriquecer cada concepto identificado durante el proceso de anotación semántica con todos los demás conceptos que se encuentren relacionados taxonómicamente a una distancia máxima preestablecida. Por ejemplo, supongamos que tenemos la ontología mostrada en la Figura II.9 y que el nodo verde, "Web Semántica", representa el concepto que ha sido identificado durante el proceso de anotación semántica. La fórmula TF-IDF extendida enriquecería esta anotación incorporando todos los conceptos que se encuentran relacionados taxonómicamente con éste proporcionando un valor más alto a aquellos conceptos que se encuentren más cercanos en la taxonomía y un valor más bajo para los más alejados.

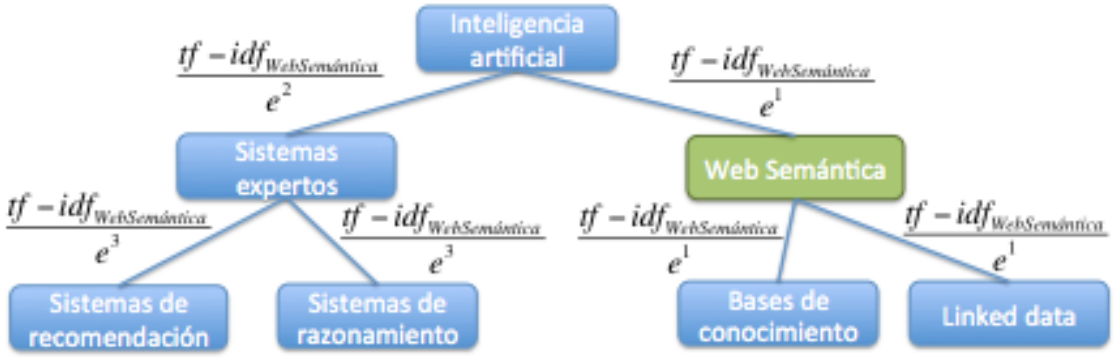


Figura II.9 Ejemplo de ontología y cálculo de dist (i,j)

Al final del proceso, cada documento tendrá asociado un índice semántico (véase Figura II.10) que viene representado como un vector con la medida TF-IDF extendido para cada concepto de la ontología. En otras palabras, el vector resultante tendrá una dimensión igual que el número de conceptos definidos en el dominio ontológico y a cada concepto se le asignará el valor obtenido por la aplicación de la fórmula TF-IDF extendido.

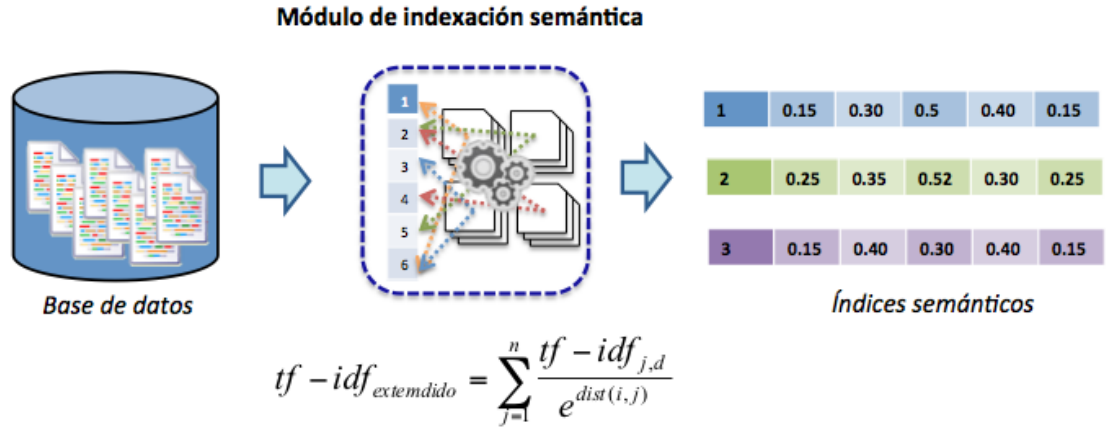


Figura II.10 Proceso de indexación semántica

Siguiendo con el escenario de ejemplo definido en el apartado anterior, a continuación se muestra cómo se generan los índices semánticos para cada uno de los documentos que han sido anotados. El procedimiento de cálculo de los valores de relevancia de información comienza con la obtención de los valores TF-IDF de cada anotación definida en cada uno de los documentos para, seguidamente, obtener los valores de los TF-IDF extendidos a partir de la estructura de la

ontología. La Tabla II.2 muestra los valores obtenidos al final del proceso de creación de índices semánticos. Como se puede observar en dicha tabla, existen términos que tienen un valor para el TF-IDF extendido superior a 0 aunque su valor para el TF-IDF sea 0. Esto significa que el documento trata sobre esos conceptos aunque no aparezcan explícitamente en el texto.

Tabla II.2 Calculo del tf-idf extendido

Docs.	Conceptos	TF	IDF	TF-IDF	TF-IDF extendido
1	Ontología	0,00862	0 (aprox. 0,1)	0,0008621	0,0017377
	OWL	0,00862	0,17609126	0,0023801	0,0026973
	Pellet	0,00862	0,47712126	0,0049752	0,0056485
	Hermit	0,00862	0,47712126	0,0049752	0,0056485
	TIC	0	0	0	0,0001167
	Web Semántica	0	0	0	0,0006393
	RDF	0	0	0	0,0006393
	Lógica descriptiva	0	0	0	0,0014633
	Razonador	0	0	0	0,0036605
	Racer	0	0	0	0,0013466
2	Web Semántica	0,00781	0,17609126	0,0013757	0,0020355
	Ontología	0,00781	0 (aprox. 0,1)	0,0007812	0,0022995
	RDF	0,01562	0,17609126	0,0027514	0,0032250
	Cloud computing	0	0	0	0,0001862
	TIC	0	0	0	0,0006118
	OWL	0	0	0	0,0008460
	Lógica descriptiva	0	0	0	0,0006118
	Razonador	0	0	0	0,0001862
3	Web Semántica	0,03030	0,17609126	0,0053361	0,0064257
	RDF	0,01818	0,17609126	0,0032017	0,0045801
	OWL	0,01818	0,17609126	0,0032017	0,0045801
	Ontología	0,00606	0 (aprox. 0,1)	0,0060606	0,0049248
	Cloud computing	0	0	0	0,0007222
	TIC	0	0	0	0,0020451
	Lógica descriptiva	0	0	0	0,0020451
	Razonador	0	0	0	0,0007222

Cualquier proceso de evolución, enriquecimiento o actualización del modelo ontológico subyacente implicaría la necesidad de actualizar los cálculos en los recursos informativos almacenados en nuestro dominio.

II.3.3. MÓDULO EXTRACTOR DE TÉRMINOS (3)

El módulo de extracción de términos se basa en el trabajo presentado en (Ochoa et al., 2011), que describe una metodología hibrida que combina información lingüística y estadística para extraer términos simples y compuestos. Esta metodología se basa en la obtención de patrones morfosintácticos a partir de análisis estadísticos de las palabras en un texto. Este método se compone de dos fases secuenciales que se conocen como (i) identificación y depuración de patrones, y (ii) optimización de patrones.

Durante la fase de identificación y depuración de patrones se etiqueta automáticamente cada elemento morfosintáctico con el fin de generar un vector de patrones candidatos. Este vector será filtrado mediante listas que contienen palabras no adecuadas. En la siguiente fase, se utilizan técnicas estadísticas y heurísticas para seleccionar el mejor patrón lingüístico que facilite la extracción de términos multipalabra. En la Tabla II.3 se muestra un ejemplo de la primera fase secuencial. El ejemplo utiliza la oración "Inteligencia Artificial" para representar el funcionamiento de la fase de identificación y depuración de patrones. En esta fase cada palabra de la oración es etiquetada morfosintácticamente. El proceso de etiquetado de la frase dará como resultado: "Inteligencia `NCFS000` Artificial `AQ0CS0`". Cada una de las letras que acompañan a las palabras tiene el significado morfosintáctico que se muestra en la Tabla II.3.

Tabla II.3 Etiquetado morfosintático

Término	Lema	Etiqueta morfosintáctica	Descripción.
Inteligencia	Inteligencia	NCFS000	N Nombre C Común F Femenino S Singular 0 Sin género 0 semántico 0 Sin grado
Artificial	Artificial	AQ0CS0	A Adjetivo Q Calificativo 0 Sin grado C Común S Singular 0 Sin función

El proceso de etiquetado morfosintáctico obtiene como resultado un conjunto de vectores que, posteriormente, son refinados mediante técnicas de filtrado de patrones. Estas técnicas de filtrado emplean listas "stopwords" para eliminar palabras que no se consideran adecuadas como, por ejemplo, preposiciones, pronombres, numerales, determinantes, conjunciones, adverbios, verbos e interjecciones. La segunda fase del proceso de obtención de patrones recibe como entrada los vectores de patrones generados y filtrados de la fase anterior, aplica sobre los mismos diferentes heurísticas y estadísticas, y obtiene patrones optimizados para detectar términos multipalabra.

Siguiendo con los ejemplos introducidos en los apartados anteriores, la Tabla II.4 contiene algunos de los patrones lingüísticos más frecuentes obtenidos durante el proceso de extracción de información. Como se puede ver, estos patrones están compuestos por una palabra o por varias palabras permitiendo la extracción de términos simples y compuestos, respectivamente. A partir de estos patrones, el

extractor de términos analiza el contenido del texto y extrae conjuntos de términos simples o compuestos que se ajustan a estos patrones morfosintácticos. Los "conceptos obtenidos", que se muestran en la tabla, constituyen la lista de conceptos candidatos, que serán proporcionados al módulo de evolución de ontologías, para enriquecer el modelo ontológico subyacente.

Tabla II.4 Ejemplos más frecuentes de patrones lingüisticos obtenidos

Patrón Lingüístico	Descripción	Conceptos obtenidos
A NC	Adjetivo + Nombre común	Nuevo conocimiento
NC A	Nombre común +Adjetivo	Web Semántica
NC SP NP	Nombre común + Preposición + Nombre propio	Ontologías en OWL
NC NP	Nombre común + Nombre propio	Repositorios RDF
NC SP NC	Nombre común + Proposición + Nombre común	Repositorios de ontologías
NC SP NC A	Nombre común + Preposición + Nombre común + Adjetivo	Bases de datos relacionales
NP CC NP	Nombre propio + Conjunción + Nombre propio	Pellet o Hermit

II.3.4. MÓDULO DE EVOLUCIÓN DE ONTOLOGÍAS (4)

Como se ha comentado anteriormente, la evolución de ontologías se puede definir como un proceso que permite la adaptación puntual de una ontología a los cambios surgidos y la propagación constante de estos cambios a los demás objetos dependientes. Un cambio en la ontología puede causar inconsistencias en otras partes de la ontología, así como en otros objetos dependientes como pueden ser los documentos anotados (Stojanovic et al., 2002).

El origen de la información de donde se va a extraer el nuevo conocimiento juega un papel esencial dentro del proceso de evolución de ontologías. La elección de la fuente de información, la organización de los datos y la transformación de estos datos en conocimiento son características que guían el desarrollo de la metodología de evolución de ontologías. Por ejemplo, el hecho de que una fuente de información almacene datos estructurados, no estructurados o semiestructurados cambia totalmente el proceso de extracción de información y, por lo tanto, afecta directamente al método desarrollado para evolucionar ontologías. Además, la información que se extrae de estas fuentes sólo constituye lo que debe ser incorporado a la ontología mediante la utilización del correspondiente lenguaje ontológico. La transformación a realizar sobre la ontología debe contemplar la creación de nuevos conceptos que representen la información extraída, su organización basada en relaciones taxonómicas del modelo ontológico, y el control de los posibles problemas de inconsistencia que pueden surgir al insertar nueva información en el modelo ontológico. Por lo tanto, como se puede atisbar ante estas consideraciones, la evolución de ontologías es una tarea extremadamente compleja donde la organización de la información juega un papel muy relevante.

II.3.4.1. Metodología de evolución de ontologías

El desarrollo del módulo de evolución de ontologías ha requerido de la utilización de dos librerías: OWL API, y Bliki Engine[12]. La librería OWL API ya ha sido descrita en apartados anteriores y nos permite el acceso y actualización de ontologías del dominio representadas en OWL. Por otro lado, la librería Bliki Engine es una librería escrita en Java que permite convertir wikitexto a HTML. El wikitexto es un texto elaborado mediante un lenguaje de marcado especial, que se utiliza en Wikipedia, para definir la mayor parte de su contenido. Wikipedia, además de la página Web, dispone de una API que ofrece un servicio Web de consulta de información a través de MediaWiki[13]. Este servicio Web de consulta admite diferentes llamadas con diferentes parámetros que facilitan la consulta de gran parte de la información disponible en la Web de Wikipedia. Además, la librería implementa procedimientos que permiten la traducción de las páginas de Wikipedia a diversos formatos digitales, tales como PDF, XML, y HTML. En concreto, una vez descritas las librerías, las operaciones que desempeñaran dentro del módulo de evolución son: facilitar la manipulación de las ontologías (OWL API) y proporcionar mecanismos de acceso al contenedor de información Wikipedia de donde se obtendrán los datos para enriquecer la ontología (Bliki Engine).

La Figura II.11 muestra gráficamente la descomposición funcional del módulo de evolución de ontologías. Como se puede observar en esta imagen, este módulo requiere de la lista de términos proporcionada por el módulo de extracción de términos para evolucionar la ontología. La librería OWL API es la encargada de proporcionar todas las funciones necesarias para manipular la ontología y la librería Bliki Engine es responsable de gestionar las conexiones con el servicio Web de consulta de Wikipedia.

[12] http://code.google.com/p/gwtwiki/

[13] http://www.mediawiki.org/wiki/API:Main_page

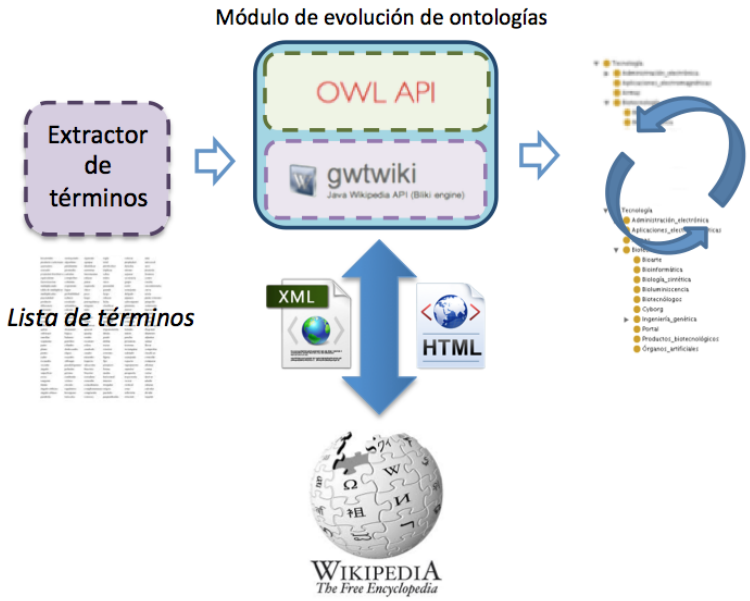


Figura II.11 Descomposición funcional de la evolución de ontologías

La metodología que se presenta en esta tesis se apoya en uno de los grandes contenedores multilingües de información en Internet como es Wikipedia. La utilización de este contenedor de información se debe a la organización de la información. Como ya destacamos anteriormente, la organización de la información es uno de los parámetros a tener en cuenta para el proceso de evolución de ontologías. En Wikipedia toda la información se encuentra organizada en categorías y artículos. Los artículos están agrupados por categorías y, éstas a su vez, se encuentran relacionadas jerárquicamente con otras categorías mediante hipervínculos. Cada categoría dispone de una categoría padre o superior y de una categoría inferior que representan relaciones taxonómicas entre los contenidos de Wikipedia. Por tanto, si extrapolamos el modelo de organización de información de Wikipedia a un modelo ontológico, las categorías más generales serían conceptos situados en los niveles más altos de la jerarquía y las categorías más concretas serían los nodos más cercanos al final de la jerarquía o los conceptos finales.

Una vez analizada la similitud entre ambos modelos de datos, falta la definición de un método capaz de transformar la información extraída de Wikipedia en conceptos que puedan ser incluidos en una ontología a través de un lenguaje ontológico. El método definido para tal fin se sustenta en la definición de concepto

propuesta por Reiterer y sus colegas (2010). Para definir un concepto se necesita un conjunto de términos que tengan el mismo significado. Wikipedia proporciona distintas formas para referenciar el mismo artículo a partir de las redirecciones. Según Wikipedia, una redirección es una página que redirige automáticamente a una página específica que ayuda a los usuarios a localizar información y mantener las wikis organizadas. Entre las diferentes utilidades que tienen, las redirecciones en Wikipedia permiten obtener variantes léxicas y ortográficas de un término. Por lo tanto, cada redirección sería un término sinónimo. Así, por ejemplo, buscando en Wikipedia los términos "lógica difusa", "lógica floja", "lógica borrosa" o "fuzzy logic", todos remiten al mismo artículo y se puede concluir, por tanto, que todos estos términos son sinónimos y constituyen un único concepto.

Para obtener todas las redirecciones en Wikipedia se utiliza una herramienta que se conoce con el nombre de "lo que enlaza aquí". Esta herramienta permite obtener todas las referencias sobre cada categoría y cada artículo presente en Wikipedia. Es decir, si buscamos el término "*Lógica difusa*" en Wikipedia aparecerá un artículo que proporciona la definición de lógica difusa y, utilizando la herramienta "lo que enlaza aquí", obtenemos una lista de redirecciones de todas las categorías y artículos en los cuales aparece el término "*Lógica difusa*". Si transformamos cada redirección de esta lista en un término, el conjunto de términos definirá un concepto que será almacenado en la ontología. La Figura II.12 representa gráficamente qué información de Wikipedia requiere el método utilizado para definir el concepto "Lógica difusa".

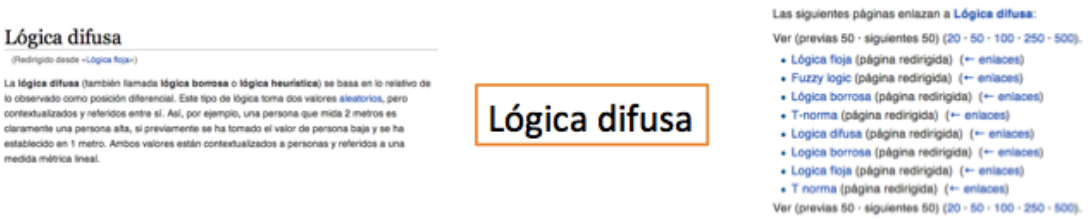


Figura II.12 Definición concepto "Lógica difusa"

Como se puede apreciar en la Figura II.12, al concepto creado se le asocia el nombre del artículo al que todos los términos redireccionan, en este ejemplo "*Lógica difusa*". Además, este concepto estará constituido por la lista de redirecciones que se extraen a través de la herramienta "lo que enlaza aquí" proporcionada por Wikipedia. En el caso de que esta herramienta no proporcione ninguna redirección, entonces se utilizaran técnicas estadísticas basadas en procesamiento de lenguaje natural para analizar y filtrar los enlaces, y obtener términos con los que definir el concepto.

Una vez definida la metodología que se va a utilizar para crear nuevos conceptos en la ontología a partir de la información de Wikipedia, el siguiente paso es analizar el algoritmo que se ha diseñado para evolucionar la ontología. Este algoritmo fue presentado en el trabajo (Rodríguez-García et al., 2012). La Figura II.13 recoge, en pseudocódigo, su implementación.

```
//Se obtienen todos los conceptos (class) de la ontología.
List<OWLClass>nodesOntology = ontology.getNodesOntology();
//Lista de términos obtenida del extractor de términos
List<String>termList = getExtractorList();
while(termList.hasMoreElements()){
      //El siguiente término es obtenido
      term = termList.nextElement();
      //Para cada término extraído el sistema comprueba si ese término se
      // encuentra definido en la ontología.
      if(!nodesOntology.contains(term)){
            while(nodesOntology.hasMoreElements()){
            //El concepto del nodo es obtenido
            nodeConcept = node.getConcept();
            //Se construye el primer concepto desde Wikipedia
            firstConcept = buildConceptFromWikipedia(nodeConcept);
            if(firstConcept==null){
                  //Si no existe el concepto en Wikipedia entonces se continua
                  //con el siguiente nodo.
                  continue;
            }else{
                  //Se construye el segundo concepto desde Wikipedia.
                  secondConcept = buildConceptFromWikipedia(term);
                  if(secondConcept==null){
                        //Si no existe el concepto en Wikipedia entonces se
                        //continúa con el siguiente término
                        break;
                  }else{
                        joiningConceptsFromWikipedia(firstConcept,
                        secondConcept);
                  }
            }
      }
}
```

Figura II.13 Algoritmo de Evolución de ontologías en pseudocódigo

El algoritmo de evolución de ontologías que presentamos en esta tesis está basado en un algoritmo de búsqueda en anchura (BFS, del inglés, "*Breadth First Search*") (Cormen et al., 2001). La función básica del algoritmo es unir dos conceptos utilizando como elemento de unión la jerarquía de categorías propuesta por Wikipedia. El primer paso que sigue el algoritmo es construir dos iteradores, uno sobre la lista de todos los conceptos de la ontología y otro sobre la lista de

términos extraídos. Para cada término extraído se comprueba, en primer lugar, que no se encuentre ya presente en la ontología. Si el término ya había sido definido, entonces se descarta y se pasa al siguiente término. En caso contrario, se inicia el proceso de evolución de ontologías. Este proceso se basa en obtener un concepto de la ontología y comprobar si éste existe en Wikipedia bien como artículo o como categoría. En el caso de que no exista, se descartaría ese concepto y se obtendría el siguiente. Si existe en Wikipedia el concepto extraído de la ontología, entonces se intenta realizar el mismo proceso para el término extraído, esto es, se comprueba si existe un artículo o categoría que lo define en Wikipedia. En el caso de que ambos, término extraído y concepto ontológico, tengan un artículo o una categoría que los definan en Wikipedia, se inicia el proceso de búsqueda de un camino entre ambos. El proceso de búsqueda se basa en un algoritmo de búsqueda en anchura que expande las categorías superiores de cada concepto hasta un número de niveles predefinido, parámetro del algoritmo de búsqueda. Si en la expansión de los niveles se encuentra una categoría común a ambos conceptos, entonces se realiza el recorrido en sentido contrario, partiendo de la categoría común al nodo de la ontología y al término extraído, transformando cada categoría en un concepto de la ontología. A partir de esta transformación se construye el camino de conceptos que representa la unión de ambos conceptos, concepto de la ontología y término extraído. Si, por el contrario, la expansión de los niveles superiores no proporciona ninguna categoría común, entonces el término se almacena en una parte de la ontología conocida como "cajón de sastre".

El "cajón de sastre" contiene todos aquellos términos extraídos que son candidatos a ser conceptos debido a que están presentes en Wikipedia, pero no han sido añadidos al modelo ontológico debido a que no se ha encontrado un camino en Wikipedia que contenga categorías compartidas entre este concepto y ninguno de los conceptos que constituyen el modelo ontológico. La eliminación de un concepto de este "cajón de sastre" se puede producir por dos razones. En primer lugar, puede deberse a que ha sido encontrado durante el proceso de búsqueda del camino entre dos conceptos, es decir, durante el proceso de expansión de las categorías superiores en cualquiera de los dos conceptos. La otra opción es que haya sido encontrado un concepto dentro de la ontología que comparta una categoría común con el concepto localizado en el cajón de sastre. De

hecho, cuando el proceso de evolución de ontologías ha analizado todos los términos extraídos, el siguiente paso es comprobar si cualquiera de los conceptos definidos en el "cajón de sastre" pueden seguir evolucionando la ontología.

Con objeto de permitir un mejor entendimiento del funcionamiento del módulo de evolución de ontologías, se explicará una iteración del algoritmo en un escenario real. Supongamos que el concepto "Inteligencia Artificial" es un término candidato proporcionado por el módulo de extracción de términos para evolucionar la ontología. Supongamos también que el concepto "Bases de conocimiento" ya se encuentra definido en la ontología y que este concepto supone el punto de partida a partir del cual el algoritmo de evolución va a intentar encontrar un punto de unión a través de la organización categórica en Wikipedia. La Figura II.17 representa gráficamente el escenario que aquí se describe. Por un lado el concepto "Inteligencia artificial" proveniente de la lista de términos extraídos y, por otro, el concepto "Bases de conocimiento" se encuentra definido en la ontología.

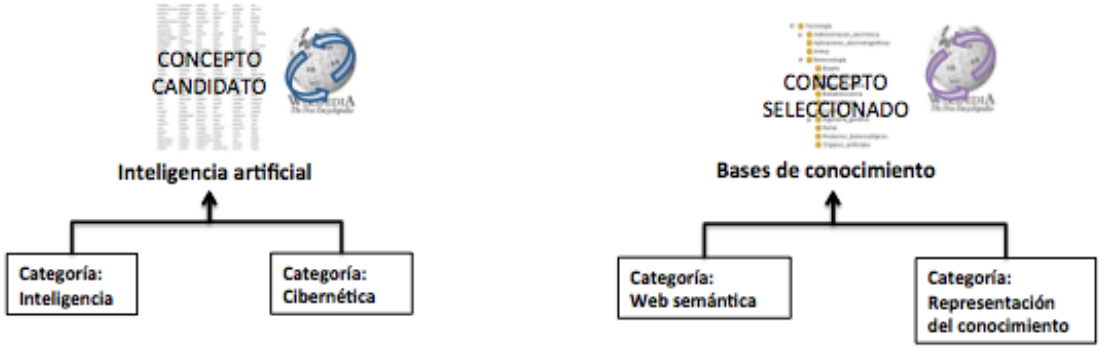
Figura II.14 Conceptos utilizados para iniciar el proceso de evolución

Utilizando este escenario como base, el proceso de evolución de ontologías comenzaría buscando el término "Inteligencia artificial" entre los artículos y las categorías de Wikipedia. La creación de un concepto difiere si el término buscado referencia en Wikipedia a un artículo o a una categoría. Esto se debe a que los artículos no pertenecen a la estructura organizativa en forma de jerarquía de Wikipedia, sino que sólo se encuentran relacionados mediante referencias entre ellos. En cambio, las categorías son las que tienen la función de organizar jerárquicamente la información. Incluso, existen artículos que, además de ser el artículo principal de la categoría, también cumplen el rol de categoría dentro de Wikipedia. Teniendo en cuenta esta distinción entre categorías y artículos, el

proceso de evolucionar la ontología y concretamente el definir un concepto, durante este proceso, se puede llevar a cabo de tres formas diferentes: (i) el término buscado se corresponde con un artículo, entonces solamente se utilizaría la información descrita en el artículo para definir el concepto en la ontología. (ii) el término buscado se corresponde con una categoría, por lo tanto sólo se utiliza la información que defina la categoría, normalmente más pobre que la proporcionada por el artículo. Por último (iii) el término buscado se corresponde con un artículo que a su vez es también una categoría dentro de la jerarquía de Wikipiedia. Este es el caso más favorable, donde el concepto en la ontología sería enriquecido a partir de la información obtenida de ambas fuentes. Al definir el nuevo concepto, será necesario recopilar de Wikipedia la información necesaria para cumplimentar los atributos descritos anteriormente, esto es, "`rdfs:comment`", "`rdfs:label`", "`preferredTerm`" y "`URIresource`". El atributo "`rdfs:comment`" se rellena con información de la definición extraída en el caso de que exista un artículo relacionado con el término buscado. El atributo "`rdfs:label`" debe completarse con todos los sinónimos que se generan a partir de las redirecciones obtenidas mediante la herramienta "lo que enlaza aquí" de Wikipedia. En el atributo "`preferredTerm`" se almacena el nombre que Wikipedia asigna al artículo principal o, en su defecto, a la categoría. Finalmente, "`URIresource`" recoge la dirección de Wikipedia del artículo o la categoría que referencia al término en cuestión.

Una vez que el término ha sido encontrado en Wikipedia, bien como artículo o como categoría, comenzaría el proceso de expansión de categorías superiores que, como se puede ver en el apartado a) de la Figura II.15, serían "Categoría: Inteligencia" y "Categoría: Cibernética". Posteriormente, se comprobaría si cualquiera de los términos que aparecen en las categorías, esto es 'Inteligencia' o 'Cibernética', se encuentran presentes en el modelo ontológico. Si se diera esta circunstancia el algoritmo de búsqueda acabaría y el proceso de evolución de ontologías recorrería el camino establecido a través de las categorías de Wikipedia. Para cada categoría se aplicarían técnicas de filtrado para quedarse sólo con el nombre del concepto y se actualizaría cada concepto en la ontología. Si, como ocurre en este escenario, no se diera esta situación, entonces se continuarían expandiendo los demás conceptos. La siguiente expansión se realizaría sobre el

concepto “Base de conocimiento” definido en la ontología. En primer lugar, el algoritmo trataría de localizar el artículo o categoría en Wikipedia relacionado con este concepto y, después, iniciaría la expansión de sus categorías superiores. Esta expansión, como se puede apreciar en el apartado b) de la Figura II.15, resultaría en las categorías “Categoría: Web Semántica” y “Categoría: Representación del conocimiento”. Ninguna de las categorías expandidas, ni las obtenidas a partir del concepto “Inteligencia artificial” ni las obtenidas a partir del concepto “Bases de conocimiento”, presentan una categoría común que proporcione un punto de unión entre ambos conceptos por lo que el proceso de evolución de ontologías seguiría ejecutándose.



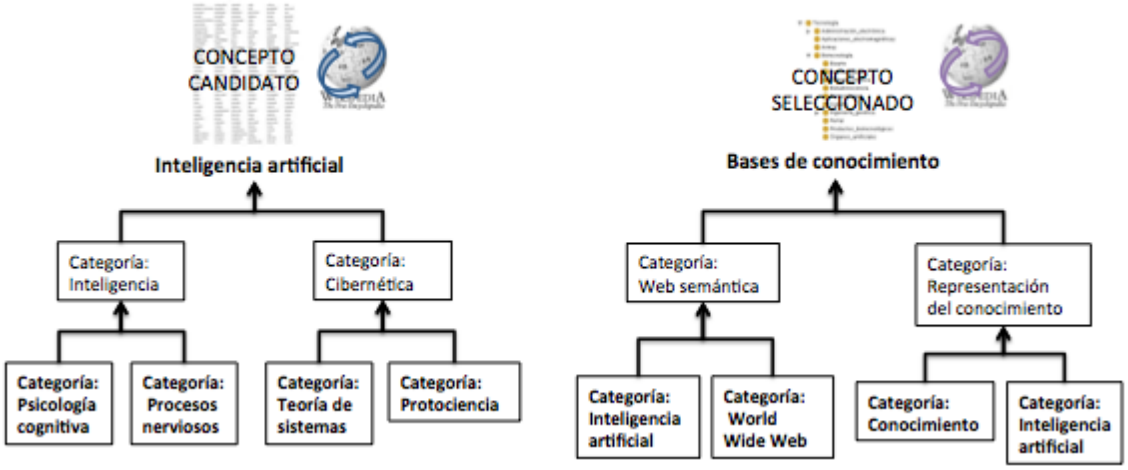
a)Expansión de las categorías padre de los conceptos “Inteligencia Artificial” en Wikipedia

b)Expansión de las categorías padre de los conceptos “Bases de conocimiento “ en Wikipedia

Figura II.15 Primera iteración del algoritmo de evolución de ontologías

Hasta aquí se habría simulado una primera iteración del algoritmo descrito anteriormente. A continuación, se iniciaría la segunda iteración que comienza con la búsqueda y expansión de los conceptos “Categoría: Inteligencia” y “Categoría: Cibernética” que generan cuatro categorías superiores, a saber, “Categoría: Psicología cognitiva”, “Categoría: Procesos nerviosos”, “Categoría: Teoría de sistemas” y “Categoría: Protociencia”, tal y como se muestra en el apartado a) de la Figura II.16. En este punto, teniendo en cuenta todas las categorías expandidas, no existe ninguna categoría que sirva de punto de unión entre ambos conceptos. Por lo tanto, el proceso de evolución continuará con la expansión de las categorías provenientes del concepto de la ontología “Bases de conocimiento”. La expansión

de este concepto proporcionaría las categorías superiores "Categoría: Inteligencia artificial", "Categoría: World Wide Web" y "Categoría: Conocimiento" representadas en el apartado b) de la Figura II.16. En esta ocasión, una de las categorías obtenidas, "Categoría: Inteligencia Artificial", se encuentra repetida en ambas agrupaciones, lo que culminaría con el proceso de evolución de ontologías.



a) Expansión de las categorías padre de los conceptos "Inteligencia" y "Cibernética" en Wikipedia

b) Expansión de las categorías padre de los conceptos "Web Semántica" y "Representación del conocimiento" en Wikipedia

Figura II.16 Segunda iteración del algoritmo de evolución de ontologías

Finalmente, una vez encontrada la categoría de unión entre ambos conceptos, el siguiente paso sería la reconstrucción de este camino de unión. Para este escenario, partiendo del concepto definido en la ontología "Bases de conocimiento", el camino estaría compuesto por los conceptos "Bases de conocimiento" → "Categoría: Web Semántica" → "Categoría: Inteligencia artificial". Durante la transformación de este camino de categorías en conceptos de la ontología se aplican técnicas para filtrar la información que finalmente se incluye en la ontología. Por ejemplo: las técnicas de filtrado recogen el nombre de la categoría (desechando la palabra "Categoría:") y obviar categorías genéricas como, por ejemplo, categorías que representan índices o categorías "ocultas" que utiliza Wikipedia para agrupar información, pero que no presentan ningún contenido semántico. Por lo tanto, el camino de conceptos final quedaría de la siguiente forma: "Bases de conocimiento"→"Web Semántica"→"Inteligencia Artificial". La

Figura II.17 representa gráficamente la reconstrucción de este camino que une ambos conceptos.

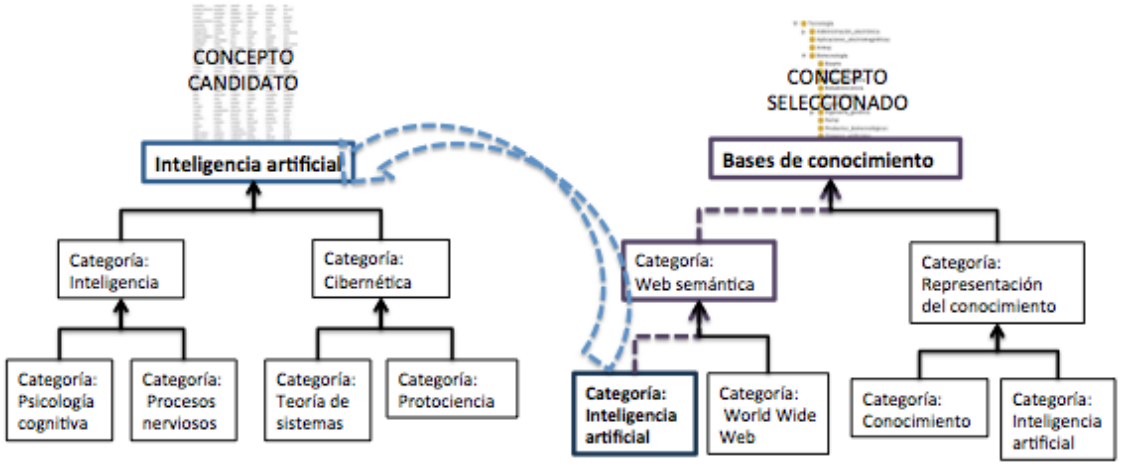


Figura II.17. Proceso de evolución de ontologías completado

II.3.5. MÓDULO MOTOR DE BÚSQUEDA SEMÁNTICA (5)

La función principal de este módulo es la implementación de un motor de búsqueda semántico que permita la localización de información previamente anotada e indexada semánticamente. La implementación de este motor sigue las directrices del modelo de espacio vectorial propuesto por (Salton & McGill, 1983), que proporciona un modelo algebraico utilizado para el filtrado, recuperación, indexado y cálculo de la relevancia de la información. La utilización de este modelo requiere de una representación formal de la información mediante el uso de vectores de identificadores que se encuentren relacionados con los términos de búsqueda en un espacio lineal y multidimensional.

Este modelo algebraico se emplea, en realidad, en los diferentes módulos que componen el sistema convirtiéndose, de este modo, en el esqueleto del sistema de anotación. Desde la representación formal de la información, que lleva a cabo el módulo de representación y anotación semántica, hasta el proceso de recuperación de información, que gestiona el motor semántico, y pasando por el módulo de indexación semántica, que implementa los procesos de indexación, filtrado y

cálculo de la relevancia de la información, todos utilizan el modelo de espacio vectorial.

En concreto, la función de recuperación de información del motor de búsqueda semántica se basa en la fórmula conocida como "función del coseno" (véase fórmula (II.5)), que permite calcular la proximidad angular que refleja el grado de parentesco o semejanza entre dos vectores que contienen los índices semánticos ('V1' y 'V2') de dos documentos.

$$\cos\theta = \frac{V1 * V2}{\left||V1|\right| \, ||V2||} \qquad \textbf{(II.5)}$$

El motor de búsqueda semántico que se propone en esta tesis utiliza como representación formal los índices creados a partir de las anotaciones semánticas definidas durante el proceso de anotación. Es decir, cada recurso anotado se representa mediante un vector de anotaciones semánticas que se ligan a conceptos definidos en la ontología. Estos conceptos junto con los valores TF-IDF calculados realizan la función de identificadores requeridos para poder aplicar la función del coseno y, de esta forma, poder calcular la proximidad angular entre dos recursos informativos.

La Figura II.18 representa gráficamente cómo funciona el motor de búsqueda semántica. En primer lugar, se deben establecer los criterios de búsqueda que van a ser aplicados, es decir, los conceptos que están relacionados con la información que se quiere extraer. A modo de ejemplo, supongamos que se busca un proyecto que utilice tecnologías semánticas y procesamiento de lenguaje natural. Al iniciar la búsqueda, el primer paso es anotar semánticamente el texto de la búsqueda para obtener una representación en forma de vector que contenga las anotaciones semánticas. Estas anotaciones semánticas se utilizan para construir una consulta que permita obtener de la base de datos todos aquellos vectores que contienen anotaciones semánticas relacionadas con las anotaciones de búsqueda. Los vectores de la base de datos descartados en este momento son aquellos en los que

no aparece ninguna de las anotaciones semánticas creadas a partir de la consulta de búsqueda. Los vectores extraídos representan documentos que previamente han sido anotados y que pueden contener información relacionada con la consulta. El siguiente paso es calcular el grado de proximidad entre el vector de la búsqueda y los vectores recopilados de la base de datos. Para ello, se utilizan los valores calculados durante el proceso de indexación semántica.

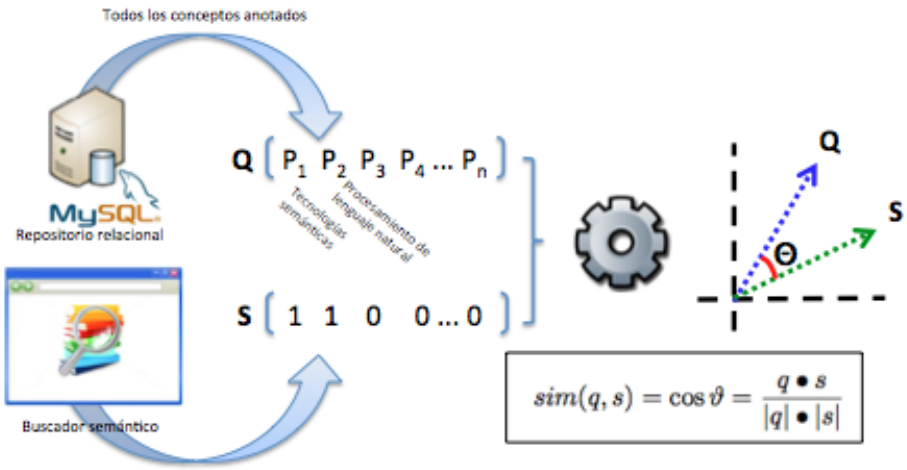


Figura II.18. Funcionamiento del búscador semántico

Para cada comparación entre dos vectores, la fórmula del coseno obtendrá un valor entre 0 y 1, que establece el grado de similitud entre ambos vectores. Un valor de 0 significa que ambos vectores son ortogonales uno del otro y, por lo tanto, se puede inferir que no hay coincidencia alguna entre ellos. Con un valor cercano a la unidad, el ángulo formado por los dos vectores será mínimo y, por lo tanto, la coincidencia entre ambos será máxima. En la Figura II.18, como resultado del proceso de búsqueda, se muestra a modo de ejemplo, un eje de coordenadas con la representación de dos vectores que están separados con un ángulo 'ϴ', que será el que indique el grado de parentesco o semejanza entre los dos índices semánticos comparados. Cuanto más amplio sea el ángulo 'ϴ', mayor será el grado de diferencia entre ambos documentos mientras que, por el contrario, un ángulo menor significará que ambos documentos tienen mayor grado de similitud.

Volviendo al escenario de ejemplo de los apartados anteriores compuesto por tres documentos (ver Figura II.6, Figura II.7, Figura II.8), en la Figura II.19 se

presenta el contenido de la base de datos relacional, una vez que los documentos han sido anotados e indexados semánticamente. Como se comentó anteriormente, cada documento se representa con un vector cuya cardinalidad es igual al número de conceptos definidos en el dominio ontológico y que contiene las medidas de TF-IDF extendido calculadas durante el proceso de indexación. Los ceros que aparecen en los vectores reflejan el hecho de que esos conceptos ni están definidos en los documentos ni están ubicados en la ontología lo suficientemente cercanos para que el TF-IDF extendido les asigne medida.

	TIC	Cloud computing	SaaS	PaaS	IaaS	Semantic Web	Ontology	OWL	RDF	Description logics	Reasoner	Pellet	Racer	Hermit
1	0,0001167	0	0	0	0	0,0006393	0,0017377	0,0026973	0,0006393	0,0014633	0,0036605	0,0056485	0,0013466	0,0056485
2	0,0006118	0,0001862	0	0	0	0,0020355	0,0022995	0,0008460	0,0032250	0,0006118	0,0001862	0	0	0
3	0,0020451	0,0007222	0	0	0	0,0064257	0,0049248	0,0045801	0,0045801	0,0020451	0,0007222	0	0	0

Figura II.19 Índices semánticos almacenados en una base de datos relacional.

Para los fines de este ejemplo, la consulta que se plantea es la siguiente: "*¿Qué es Pellet?*". Cuando el usuario ejecuta esta consulta en el buscador, la primera acción que llevará a cabo el módulo de búsqueda semántico es obtener la representación formal de esta consulta. Esto es, a partir del texto de la consulta, obtendrá un vector igual que los almacenados en la base de datos. Para ello, la consulta pasará por todo el proceso de anotación e indexado semántico. Primero, el buscador semántico utilizará al módulo de representación y anotación semántica para identificar qué expresiones lingüísticas existentes en la consulta se pueden asociar con entidades definidas en la ontología. Después, a partir de estas anotaciones, el indexador semántico construirá un vector idéntico a los que representan los documentos en la Figura II.19. La Figura II.20 muestra el vector resultante del proceso de representación. Si se analiza el contenido de este vector, se observa que sólo los cinco últimos elementos del vector tienen asignados un valor, mientras que todos los demás elementos tienen un valor de 0. Estos valores representan el valor TF-IDF extendido obtenido durante el proceso de anotación e indexación

semántica. Por lo tanto, cuando el valor de un elemento del vector es 0 significa que el concepto al que representa esa posición del vector ni aparece en la consulta ni tampoco tiene ninguna relación taxonómica en la ontología con un concepto que aparezca en la misma. En cambio, las posiciones cuyos valores son distintos de 0 representan conceptos con relaciones cercanas al concepto anotado. Por ejemplo, en este caso, el concepto "Pellet" tiene asignado un valor de uno ya que es el concepto que literalmente aparece en la consulta. Sin embargo, existen además otras posiciones con valores distintos de 0 como, por ejemplo, las posiciones que representan los conceptos "Description logics", "Hermit", "Racer" y "Reasoner". Estos conceptos no aparecen explícitamente en la consulta pero sí que se le asignan valores TF-IDF extendido por sus relaciones taxonómicas dentro del modelo ontológico.

Q	0	0	0	0	0	0	0	0	0	0,1353352	0,36787944	1	0,1353352	0,1353352

Figura II.20 Representación vectorial de la consulta

Una vez que la consulta ha sido transformada a un índice semántico, el siguiente paso es la comparación de este índice con todos los demás índices creados a partir de los documentos indexados semánticamente y que se encuentran almacenados en la base de datos. La comparación del índice semántico de la consulta con cada índice semántico de cada documento indexado da como resultado un valor que cuantifica su similitud obtenida a través de la fórmula del coseno (fórmula (II.5)). Como resultado de esta comparación, se obtiene una lista de documentos ordenados por el valor de similitud. En la primera posición de esta lista se encontrará el documento más relacionado con los conceptos definidos en la consulta. La Tabla II.5 presenta el listado para este ejemplo en concreto con los documentos ordenados por grado de similitud. Si analizamos el contenido de esta tabla, se puede concluir que, para la consulta "*¿Qué es Pellet?*", de todos los documentos indexados el más similar a la consulta es el documento 1 debido a que el valor de similitud obtenido es el más elevado con respecto a los demás, le sigue el documento 3 y el documento 2, siendo este último el menos relacionado.

Tabla II.5 Listado ordenado de recursos obtenidos

Documento	Grado de similitud
1	0,828145785
3	0,229248314
2	0,216850581

II.4. RESUMEN

Este capítulo describe en profundidad la parte central de esta tesis doctoral, el sistema de anotación semántica. El capítulo comienza por una introducción donde se proporciona una breve descripción de las diferentes tecnologías que han sido utilizadas para el desarrollo del sistema. A continuación, en la sección II.2 se proporciona un análisis sobre el problema a resolver y cuáles son los objetivos que se han marcado en el desarrollo del sistema.

Por último, en la sección II.3, la más extensa de este capítulo, se analiza en profundidad la funcionalidad del sistema de anotación. Esta sección comienza con una descripción breve de la arquitectura del sistema de anotación. Posteriormente, la sección se descompone en cinco grandes apartados que describen cada uno de los módulos que constituyen el sistema de anotación semántica. En la sección II.3.1 se describe el módulo de representación y anotación semántica, se definen las propiedades y anotaciones del modelo ontológico, y la metodología de anotación semántica utilizada para anotar la información. Seguidamente, en la sección II.3.2 se presenta el módulo de indexación semántica que tiene la labor de crear índices semánticos mediante la aplicación de fórmulas matemáticas que calculan la relevancia de información de cada anotación semántica. La función básica de estos dos módulos es cambiar la representación de la información de modo que pueda ser comparable. De este modo, el módulo de representación y anotación semántica

convierte cada documento en un vector de anotaciones semánticas y el segundo módulo asigna a cada anotación un valor que define cómo de relevante es esa anotación para un documento dentro de una colección de documentos.

Por otro lado, en la sección II.3.3 se describe el funcionamiento del módulo de extracción de términos que utiliza algoritmos híbridos para obtener patrones léxicos que faciliten la labor extracción de términos multipalabra. En el siguiente apartado, sección II.3.4, se describe el módulo de evolución de ontologías que es el encargado de actualizar el dominio ontológico. Este módulo utiliza Wikipedia como repositorio de información y la lista de términos proporcionada por el módulo de extracción de términos. Finalmente, en la sección II.3.5 se describe el módulo de búsqueda semántica cuya función es la de recuperar información almacenada en el sistema utilizando como fuente los índices semánticos obtenidos a partir de los documentos anotados.

Capítulo III. VALIDACIÓN DEL SISTEMA DE ANOTACIÓN SEMÁNTICA

III.1. INTRODUCCIÓN

Este capítulo se centra en la validación de la metodología de anotación desarrollada en esta tesis. Para la validación se han utilizado las medidas de ”precisión”, “exhaustividad” y “medida-F”. Estas medidas se utilizan ampliamente en la validación de sistemas de procesamiento del lenguaje natural y recuperación de información (Rodríguez-García et al., 2014).

El proceso de validación se ha llevado a cabo en dos dominios bien diferenciados, a saber, el dominio de la descripción de los servicios de computación en la nube (del inglés, '*Cloud Computing*') y el dominio de la gestión de la I+D+i, con el objetivo de demostrar la utilidad de la metodología desarrollada, pudiéndose adaptar a cualquier ámbito.

El primer experimento se ha realizado en el dominio de la computación en la nube. Para plantear este escenario ha sido necesaria la intervención de expertos en el dominio que validasen los resultados obtenidos por el sistema, así como, la recopilación de un corpus de descripciones técnicas sobre servicios de computación en la nube. Este corpus de descripciones técnicas no sólo ha sido utilizado con fines de validación, sino que también se ha utilizado para el modelado del dominio ontológico. Es éste, quizás, uno de los problemas más arduos en el proceso de evaluación del sistema, encontrar una ontología que modele el dominio objetivo y que además proporcione la información semántica necesaria al módulo de anotación semántica para crear las anotaciones. Al final se solucionó seleccionando una ontología que modelaba información de las TIC y la computación en nube, después de realizar un estudio previo, y enriqueciendo la ontología con el corpus creado.

El segundo experimento se realizó en el dominio de la gestión de la I+D+i. Dada la pluralidad de este dominio, este experimento se centró en la aplicación de las TIC en la I+D+i. Para realizar esta validación se utilizaron muchos de los recursos

generados en el escenario anterior como, por ejemplo, el modelo de ontología original. Esta ontología, aunque incorporaba ciertos aspectos relacionados con las TIC, tuvo que ser enriquecida utilizando la funcionalidad de evolución de ontologías desarrollada. La evolución de la ontología se llevó a cabo a partir del corpus confeccionado para evaluar la metodología. Este corpus se recopiló a partir de documentos que son utilizados en cualquier proceso de gestión de proyectos de I+D+i como, por ejemplo, memorias de proyectos de innovación, descripciones de ideas, propuestas, currículos vitae, etc. En cuanto a la metodología de evaluación utilizada, para este experimento fue similar a la anterior. Se precisó de la colaboración de cuatro expertos en la materia para llevar a cabo la evaluación de los resultados obtenidos. La función que desempeñaron se ceñía a producir resultados procesados manualmente que, posteriormente, se contrastaban con los obtenidos por el sistema.

En los siguientes apartados se analiza, en primer lugar, las medidas de evaluación que se emplearán para evaluar el sistema. Una vez conocidas las herramientas de evaluación que se van a utilizar, en las sucesivas secciones se llevará a cabo la evaluación de la metodología de anotación semántica en el dominio de la computación en la nube y en el dominio de gestión de proyectos I+D+i.

III.2. MEDIDAS DE EVALUACIÓN

La evaluación del rendimiento del sistema propuesto en esta tesis doctoral ha sido llevada a cabo mediante la utilización de un conjunto de métricas de evaluación estándar: “precisión”, “exhaustividad” y “medida-F”. Estas métricas, que son de aplicación muy común en los procesos de evaluación de sistemas de procesamiento del lenguaje natural y de recuperación y extracción de información, fueron propuestas inicialmente por Salton en 1983 (Salton & McGill, 1983). Aunque estas métricas estaban destinadas a la medida de rendimiento de sistemas de búsqueda y recuperación de información y reconocimiento de patrones, actualmente son varias las áreas de investigación que las han adaptado para medir el rendimiento en otros contextos como, por ejemplo, oftalmología (Milios et al.,

2003), biomedicina (Krauthammer & Nenadic, 2004) y bioquímica (Alexopoulou et al., 2008), entre otros.

Según Salton y McGill (1983) la precisión se define como una medida de exactitud y determina la fracción de artículos relevantes de todos los artículos recuperados en un sistema de extracción de información. Acorde con esta definición, la `precisión` se puede calcular a partir de la fórmula (III.1).

$$precisión = \frac{|\{documentos\ relevantes\} \cap \{documentos\ recuperados\}|}{|\{documentos\ recuperados\}|} \quad \textbf{(III.1)}$$

Por otro lado, la `exhaustividad` es una medida de la integridad y determina la fracción de artículos relevantes recuperados de todos los artículos relevantes. Acorde con esta definición, se puede calcular a partir de la fórmula (III.2).

$$exhaustividad = \frac{|\{documentos\ relevantes\} \cap \{documentos\ recuperados\}|}{|\{documentos\ relevantes\}|} \quad \textbf{(III.2)}$$

Por último, la `medida-F` es la media armónica de los valores de precisión y exhaustividad (Yang & Liu, 1999). En concreto, la fórmula de la medida-F (del inglés, '*F-Measure*, *balanced F-Score* o F_1 *measure*') se utiliza para evaluar el rendimiento global de las dos métricas propuestas anteriormente. Acorde con esta definición, la medida-F se puede obtener a partir de la fórmula (III.3). El resultado de la fórmula proporciona un valor real entre 0 y 1.

$$medida - F = 2 \cdot \frac{precision \cdot exhaustividad}{precision + exhaustividad} \quad \textbf{(III.3)}$$

El sistema de anotación semántica expuesto en esta tesis doctoral se evaluó utilizando estas tres métricas sobre varios módulos que lo constituyen: (i) motor de búsqueda semántica que se evaluó en función de la precisión de las sugerencias recuperadas por el motor semántico y a partir del índice de exhaustividad sobre

estas sugerencias. (ii) módulo de extracción de términos que fue evaluado en función de la precisión de los términos que se extrajeron durante los experimentos y también fue evaluado utilizando el índice de exhaustividad sobre estos términos extraídos. (iii) Por último, módulo de evolución de ontologías que se evaluó de dos formas diferentes. En primer lugar, validando la precisión que medía cuantos de los términos candidatos enriquecían finalmente la ontología y validando la exhaustividad para obtener conclusiones acerca de la relevancia de los conceptos. En segundo lugar, el módulo de evolución de ontologías también se evaluó en función de la precisión de las relaciones taxonómicas generadas durante el proceso de evolución, así como el análisis de relevancia de las relaciones taxonómicas generadas. La evaluación comenzará por aplicar estas métricas sobre el motor de búsqueda semántica. En esta evaluación se utilizarán las métricas con las variables originales: '`sugerencias recuperadas`', que representa todas las sugerencias que se recuperan en un proceso de búsqueda y '`sugerencias relevantes`', que son aquellas que se encuentran realmente relacionadas con los conceptos buscados (véase fórmulas (III.4) y (III.5)).

$$precision = \frac{|\{sugerencias\ relevantes\} \cap \{sugerencias\ recuperadas\}|}{|\{sugerencias\ recuperadas\}|} \quad \textbf{(III.4)}$$

$$exhaustividad = \frac{|\{sugerencias\ relevantes\} \cap \{sugerencias\ recuperadas\}|}{|\{sugerencias\ relevantes\}|} \quad \textbf{(III.5)}$$

El siguiente modulo que se va a evaluar es el que desarrolla la función de extracción de términos. En este caso, la evaluación no se aplicará sobre la metodología de extracción, ya que ésta se corresponde con un aporte científico propio. Por el contrario, la evaluación trata de validar esta técnica de extracción como recurso de información del módulo de evolución de ontologías. El objetivo de esta evaluación es determinar el porcentaje de términos válidos para el proceso de evolución de ontologías a partir de los términos candidatos. Para ello, se utilizaron las métricas definidas de acuerdo a la cantidad de '`términos candidatos`', que

representan el número de términos extraídos y los '`términos seleccionados`', referido a los conceptos a agregar en la ontología seleccionados manualmente por el experto (véase fórmulas (III.6) y (III.7)).

$$precision = \frac{|\{términos\ candidatos\} \cap \{términos\ seleccionados\}|}{|\{términos\ seleccionados\}|} \quad \textbf{(III.6)}$$

$$exhaustividad = \frac{|\{términos\ \ candidatos\} \cap \{términos\ seleccionados\}|}{|\{términos\ \ candidatos\}|} \quad \textbf{(III.7)}$$

También utilizando estas métricas se pretende evaluar el módulo de evolución de ontologías. El objetivo de la validación de este módulo es obtener (i) un valor cuantitativo de la cantidad de nuevos conceptos insertados durante los experimentos y (ii) las relaciones taxonómicas incluidas en las ontologías. Para satisfacer el primero de estos objetivos, se redefinen las métricas para que utilicen términos seleccionados frente a conceptos definidos (véase fórmulas (III.8) y (III.9)). Los '`términos candidatos`' representan aquellos términos que son candidatos a ser conceptos en la ontología y los '`conceptos seleccionados`' representan los términos que han sido finalmente insertados en la ontología.

$$precision = \frac{|\{términos\ candidatos\} \cap \{conceptos\ seleccionados\ \}|}{|\{conceptos\ seleccionados\ \}|} \quad \textbf{(III.8)}$$

$$exhaustividad = \frac{|\{términos\ candidatos\ \} \cap \{conceptos\ seleccionados\ \}|}{|\{términos\ \ candidatos\}|} \quad \textbf{(III.9)}$$

Para la segunda evaluación de la metodología de evolución, se van a tener en cuenta las relaciones taxonómicas creadas a partir de la incorporación de los nuevos conceptos. En esta evaluación las métricas quedarían adaptadas de la

siguiente manera (véase fórmulas (III.10) y (III.11)). En estas fórmulas las '`relaciones candidatas`' representan aquellas relaciones taxonómicas que se han creado durante el proceso de evolución de ontologías y que son candidatas a establecer relaciones entre dos conceptos dentro de la ontología que se está evolucionando, y las '`relaciones seleccionadas`' representan aquellas relaciones que finalmente han sido incluidas en la ontología después del proceso de evolución.

$$precision = \frac{|\{relaciones\ candidatas\} \cap \{relaciones\ seleccionadas\}|}{|\{relaciones\ candidatas\}|} \quad \textbf{(III.10)}$$

$$exhaustividad = \frac{|\{relaciones\ candidatas\} \cap \{relaciones\ seleccionadas\}|}{|\{relaciones\ seleccionadas\}|} \quad \textbf{(III.11)}$$

III.3. VALIDACIÓN EN EL DOMINIO DE COMPUTACIÓN EN LA NUBE

III.3.1. INTRODUCCIÓN

Actualmente las empresas han sufrido un decrecimiento económico considerable lo que les ha llevado a reducir costes en Tecnologías de la Información y Comunicación (TIC). Ante esta problemática situación los expertos proponen como solución externalizar partes del negocio para que sean gestionados por servicios de terceros (Rhoton, 2010). Con esto se consigue liberar recursos invertidos en los procesos de negocio externalizados y centrar sus inversiones en los procesos de negocio propios de la organización (Rodríguez-García et al., 2014).

Actualmente, los servicios en Internet están enriqueciendo funcionalmente la Web para que ésta evolucione de un simple repositorio de información a una nueva plataforma de transacciones comerciales e intercambio de información en dominios tales como la e-ciencia, la educación y el comercio electrónico (Schubert et al., 2012). Esta evolución tecnológica está influyendo para que grandes

organizaciones expongan sus procesos de negocio a través de la tecnología de servicios Web, tanto para el desarrollo a gran escala de software como el intercambio de servicios dentro y fuera de la organización. La utilización masiva de estos servicios ha facilitado la generación de nuevos paradigmas de computación que ofrecen nuevos niveles de eficiencia a través de la distribución a gran escala de recursos informáticos como, por ejemplo, la computación en la nube, primer dominio que va a ser utilizado para validar nuestro sistema de anotación y recuperación semántica de información.

La computación en la nube puede verse como un cambio de paradigma tecnológico que permite ofrecer servicios de computación a través de Internet utilizando un modelo de pago por uso que hace que sea especialmente atractivo. El paradigma de computación en la nube distingue tres modelos de prestación de servicios principales que son(Mell & Grance, 2011):

- Software como Servicio (SaaS, del inglés "*Software-as-a-Service*"): es un modelo de distribución de software escalable y personalizable de aplicaciones que se ejecutan en una infraestructura de la nube. Las aplicaciones y los recursos computacionales son accedidos a través de Internet y se alojan en servidores de una compañía de TIC. Por lo tanto, es el proveedor y no el cliente el que se encarga de administrar y controlar la infraestructura de la nube subyacente par que el cliente pueda disfrutar del servicio. Es decir, el usuario no necesita gestionar ningún tipo de recurso como redes, servidores, sistemas operativos, almacenamiento, entre otros que sean necesarios para la ejecución de las aplicaciones o los recursos que él precisa. De esta forma, este modelo de distribución facilita las tareas de gestión y administración de software convirtiéndose en un modelo cada vez más extendido de distribución de software.
- Plataforma como Servicio (PaaS, del inglés "*Platform-as-a-Service*"): en la literatura suele identificarse como una evolución del software como servicio. Este modelo reduce bastante la complejidad de despliegue y mantenimiento de aplicaciones. El modelo está constituido por la encapsulación de un conjunto de servicios de aplicación, frameworks de desarrollo y sistemas de monitorización que se ofrecen de manera integral en la Web. Este modelo proporciona todas

las aplicaciones necesarias para permitir el ciclo completo de construcción de software y servicios disponibles desde Internet.

- Infraestructura como Servicio (IaaS, del inglés "*Infraestructure-as-a-Service*"): conocido en algunos casos hardware como servicio (HaaS, del inglés "*Hardware-as-a-Service*"), es un modelo que se caracteriza por proveer infraestructuras escalables totalmente externas y básicas de cómputo como servidores, software y equipamiento de red como un servicio bajo demanda. Este modelo ofrece a las organizaciones la capacidad de poder hacer uso del hardware sin la necesidad de enormes inversiones en servidores, sistemas de almacenamiento, enrutadores, entre otros.

Aunque el objetivo de cada modelo de prestación de servicios en la nube queda bien definido, no todos los servicios pertenecientes al mismo modelo poseen las mismas características. Por ejemplo, dentro del modelo IaaS, una organización puede demandar un servicio Web basado en Apache Tomcat[14] y, dentro de este mismo nivel de prestación de servicios, la organización puede requerir o una versión especifica del servidor o de la base de datos o de la máquina virtual de Java[15]. Es decir, pueden existir diversas combinaciones para las organizaciones lo que, a la hora de implantar cualquiera de estos servicios, puede ser un gran inconveniente debido a que la búsqueda de servicios apropiados manualmente requiere mucho tiempo y esfuerzo. Este hecho fue el causante que motivó la selección de este dominio para validar el sistema de anotación. La validación del sistema consistió en la anotación semántica de una colección de documentos que describían servicios en la nube. Las anotaciones semánticas generadas eran aprovechadas por el buscador semántico, adaptado al dominio de los servicios en la nube, para automatizar la búsqueda de servicios ayudándose de las tecnologías descritas en el apartado II.3.

[14] http://tomcat.apache.org

[15] http://www.java.com/es/

III.3.2. ESCENARIO DE EVALUACIÓN

La evaluación de esta herramienta en este contexto ha requerido el desarrollo de una ontología que modela el dominio de las TIC. El desarrollo de una ontología desde cero para un dominio particular supone una tarea que requiere mucho tiempo y recursos, además que dicha implementación supone la búsqueda de vocabulario consensuado para su desarrollo. Por consiguiente, se realizó un estudio sobre las ontologías existentes actualmente que modelaban el dominio de las TIC para determinar qué modelo del dominio era más adaptable a los requisitos de nuestra propuesta de validación. Es posible destacar algunas de las soluciones analizadas. En primer lugar, el trabajo de Velasco y sus colegas (2009), donde se desarrolla una ontología que modela los requisitos de seguridad en los documentos de especificación de requisitos. También resultó interesante la propuesta de Happel y sus colegas (2006), donde la ontología que presentan constituye el sistema Kontor, encargado de proporcionar descripciones semánticas de componentes de software que son almacenados en bases de conocimiento. Estas descripciones pueden ser consultadas utilizando el lenguaje de consulta semántico SPARQL. Por último, otra investigación relevante es la que se describe en (Hartig et al., 2008), donde se presenta el proyecto DESWAP, construido sobre una base de conocimiento que almacena descripciones semánticas integrales de software y sus funciones desarrolladas.

Entre todos los modelos ontológicos estudiados y atendiendo a las características funcionales del sistema de anotación a evaluar, se seleccionó la ontología desarrollada por Hartig y sus colegas (2008) en el proyecto DESWAP como ontología semilla. Esta ontología se extendió para representar las características y propiedades funcionales de los servicios en la nube. La Figura III.1 representa gráficamente un breve extracto de la ontología y en la Tabla III.1 se muestran algunas de las métricas relacionadas con esta ontología.

Figura III.1 Extracto de la ontología de las TIC

Tabla III.1 Características de la ontología de las TIC

	Ontología de las TIC
Clases	2569
Instancias	1725
Atributos	29
'*Relaciones*'	285
Axiomas de clases	272
Axiomas *de relaciones*	197
Axiomas *de atributos*	58

La adaptación de la ontología llevó consigo la incorporación de nuevo conocimiento relacionado con la computación en la nube. Este enriquecimiento se llevó a cabo mediante la inserción en el sistema de 500 descripciones en lenguaje natural de distintos servicios en la nube. Cada uno de estos servicios fue anotado semánticamente y almacenado en un repositorio de ontologías implementado a

través de Virtuoso. Durante la anotación semántica, las descripciones eran analizadas para confeccionar una lista de términos que posteriormente eran utilizados para enriquecer el dominio ontológico. Después de ser anotados, se calcularon los índices semánticos para facilitar los procesos de búsqueda de servicios en la nube. La Tabla III.2 muestra un ejemplo de algunos de los servicios en la nube que se insertaron en el sistema.

Tabla III.2 Ejemplo de servicios de computación en la nube

Nombre	Tipo de servicio	Proveedor	Tecnología
Amazon SimpleDB	PaaS	Amazon	Api; Multi-platform
Amazon Relational Database Service	PaaS	Amazon	Api; Multi-platform
Amazon SQS	PaaS	Amazon	Api; Multi-platform
Amazon Elastic Beanstalk	PaaS	Amazon	Api; Multi-platform
Appian Anywhere	PaaS	Amazon	Api; Multi-platform
Google App Engine	PaaS	Google	Python; Java; Go app engine; Api
Gmail	SaaS	Google	Api; Python; Ajax

III.3.3. BUSCADOR SEMÁNTICO

El objetivo de la evaluación en este escenario fue comprobar la utilidad del motor de búsqueda semántica propuesto en la plataforma de anotación. Para realizar la validación se seleccionaron cuatro expertos en servicios de computación en la nube que definieron cinco consultas sobre diez tipos de servicios preestablecidos en la nube. Los temas establecidos para realizar las búsquedas en este proceso de validación fueron: “J2EE”, “servidor de aplicaciones”, “Ajax”, “bases de datos”, “sistemas de información empresarial”, “bases de datos y servidor de aplicaciones”, “J2EE y bases de datos”, “bases de datos y sistemas de información empresarial”, “J2EE y Ajax” and “J2EE y servidor de aplicaciones”. Posteriormente, para cada consulta los expertos debían seleccionar manualmente qué servicios eran los más adecuados en función de la consulta realizada. Del mismo modo, se lanzaron estas consultas en el motor de búsqueda semántico para comprobar si los servicios sugeridos por el sistema coincidían con los servicios seleccionados por

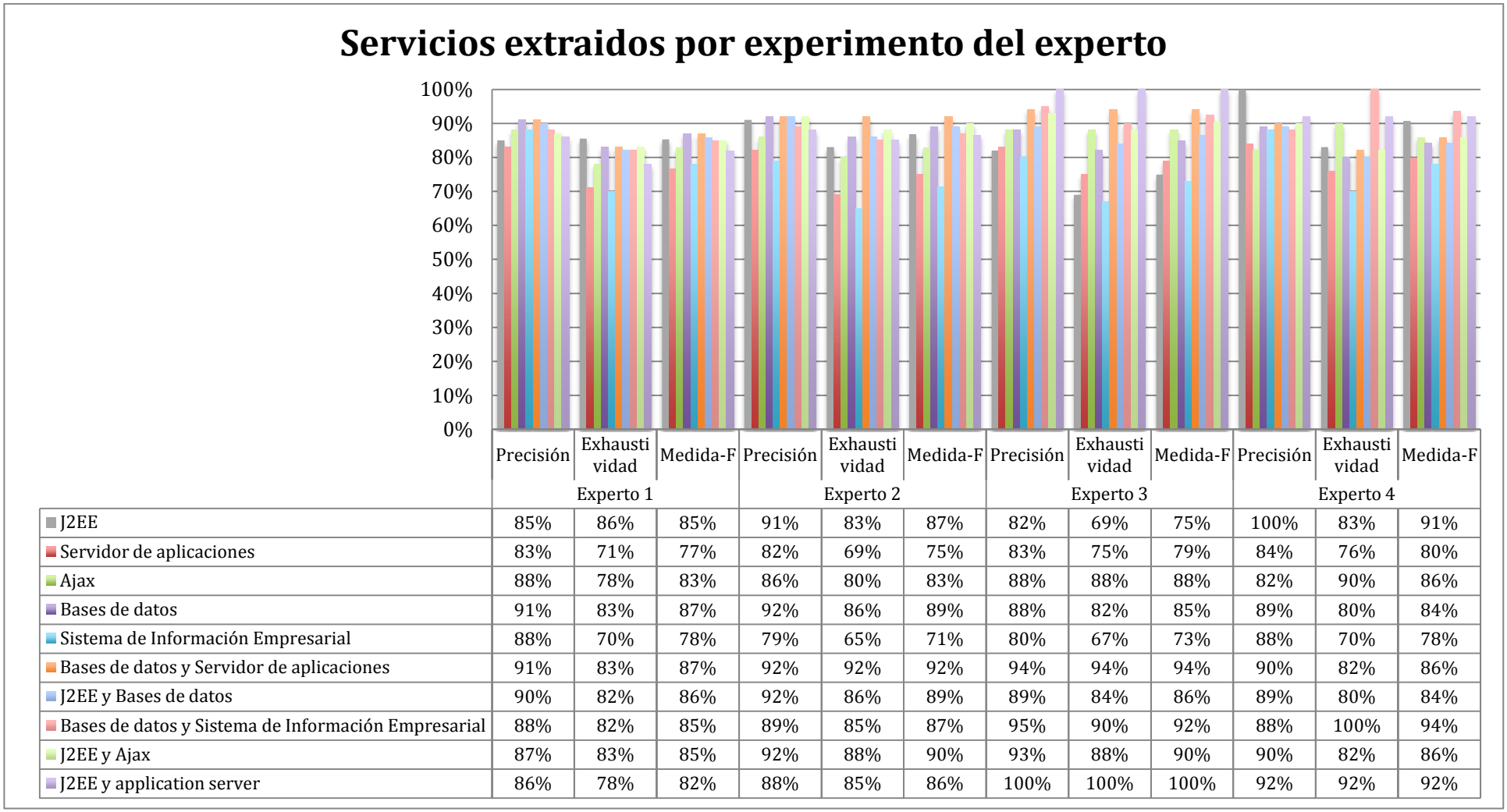
parte de los expertos. La Tabla III.3 muestra los resultados obtenidos en el experimento para cada uno de los expertos en términos de servicios acertados ('A'), servicios extraídos ('E') y servicios relevantes ('R'). A partir de estos valores se ha construido la Tabla III.4, que contiene los valores calculados utilizando las fórmulas de precisión ('P'), exhaustividad ('E') y medida-F ('F') definidas al principio del capítulo. Por último, la Figura III.2 proporciona una representación gráfica de estos valores obtenidos.

Tabla III.3 Valores de sugerencias acertadas (A), Extraidas (E) y Relevantes en el dominio de la computación en la nube

	Experto 1			Experto 2			Experto 3			Experto 4		
Temas	**A**	**E**	**R**	**A**	**E**	**R**	**A**	**E**	**R**	**A**	**E**	**R**
J2EE	43	50	50	36	39	43	32	39	46	42	42	51
Servidor de aplicaciones	50	61	71	53	65	77	54	65	72	48	57	63
Ajax	33	37	42	30	35	38	31	35	35	36	44	40
Bases de datos	75	82	90	78	85	91	81	92	99	71	80	89
Sistema de Información Empresarial(SIE)	56	64	80	52	66	80	54	67	80	56	64	80
Bases de datos y servidor de aplicaciones	25	27	30	29	31	31	33	35	35	24	26	29
J2EE y bases de datos	21	24	26	25	27	29	18	20	21	19	22	24
Base de datos y Sistema de Información Empresarial	19	21	23	19	21	22	17	18	19	25	28	25
J2EE y Ajax	16	18	19	18	20	21	20	22	23	16	18	20
J2EE y servidor de aplicaciones	16	19	21	19	22	23	25	25	25	25	27	27

Tabla III.4 Valores de Precisión (P), Exhaustividad (E) y Medida-F (F) obtenidos en el experimento en el dominio de la computación en la nube

	Experto 1			Experto 2			Experto 3			Experto 4			Media Experto		
Temas	**P**	**E**	**F**	**P**	**E**	**F**	**P**	**E**	**F**	**P**	**E**	**F**	**P**	**E**	**F**
J2EE	0,85	0,85	0,85	0,91	0,83	0,87	0,82	0,69	0,75	**1,00**	**0,83**	**0,91**	0,89	0,80	0,84
Servidor de aplicaciones	0,83	0,71	0,77	0,82	0,69	0,75	0,83	0,75	0,79	**0,84**	**0,76**	**0,80**	0,83	0,73	0,78
Ajax	0,88	0,78	0,82	0,86	0,80	0,83	**0,88**	**0,88**	**0,88**	0,82	0,90	0,86	0,86	0,84	0,85
Bases de datos	0,91	0,83	0,87	**0,92**	**0,86**	**0,89**	0,88	0,82	0,85	0,89	0,80	0,84	0,90	0,83	0,86
Sistema de Información Empresarial (SIE)	**0,88**	**0,70**	**0,78**	0,79	0,65	0,71	0,80	0,67	0,73	**0,88**	**0,70**	**0,78**	0,83	0,68	0,75
Total tema simple	*0,87*	*0,77*	*0,82*	*0,86*	*0,77*	*0,81*	*0,84*	*0,76*	*0,80*	*0,88*	*0,80*	*0,84*	*0,86*	*0,78*	*0,82*
Bases de datos y Servidor de aplicaciones	0,91	0,83	0,87	0,92	0,92	0,92	**0,94**	**0,94**	**0,94**	0,90	0,82	0,86	0,92	0,88	0,90
J2EE y bases de datos	0,90	0,82	0,86	**0,92**	**0,86**	**0,89**	0,89	0,84	0,86	0,89	0,80	0,84	0,90	0,83	0,86
Bases de datos y SIE	0,88	0,82	0,85	0,89	0,85	0,87	0,95	0,90	0,92	**0,88**	**1,00**	**0,93**	0,90	0,89	0,89
J2EE y Ajax	0,87	0,83	0,85	0,92	0,88	0,90	**0,93**	**0,88**	**0,90**	0,90	0,82	0,86	0,90	0,85	0,88
J2EE y servidor de aplicaciones	0,86	0,78	0,82	0,88	0,85	0,87	**1,00**	**1,00**	**1,00**	0,92	0,92	0,92	0,92	0,89	0,90
Total tema múltiple	*0,88*	*0,82*	*0,85*	*0,91*	*0,87*	*0,89*	*0,94*	*0,91*	*0,93*	*0,90*	*0,87*	*0,88*	*0,91*	*0,87*	*0,89*
Total	0,87	0,80	0,83	0,88	0,82	0,85	0,89	0,84	0,86	0,89	0,84	0,86	**0,88**	**0,82**	**0,85**



	Experto 1			Experto 2			Experto 3			Experto 4		
	Precisión	Exhaustividad	Medida-F	Precisión	Exhaustividad	Medida-F	Precisión	Exhaustividad	Medida-F	Precisión	Exhaustividad	Medida-F
J2EE	85%	86%	85%	91%	83%	87%	82%	69%	75%	100%	83%	91%
Servidor de aplicaciones	83%	71%	77%	82%	69%	75%	83%	75%	79%	84%	76%	80%
Ajax	88%	78%	83%	86%	80%	83%	88%	88%	88%	82%	90%	86%
Bases de datos	91%	83%	87%	92%	86%	89%	88%	82%	85%	89%	80%	84%
Sistema de Información Empresarial	88%	70%	78%	79%	65%	71%	80%	67%	73%	88%	70%	78%
Bases de datos y Servidor de aplicaciones	91%	83%	87%	92%	92%	92%	94%	94%	94%	90%	82%	86%
J2EE y Bases de datos	90%	82%	86%	92%	86%	89%	89%	84%	86%	89%	80%	84%
Bases de datos y Sistema de Información Empresarial	88%	82%	85%	89%	85%	87%	95%	90%	92%	88%	100%	94%
J2EE y Ajax	87%	83%	85%	92%	88%	90%	93%	88%	90%	90%	82%	86%
J2EE y application server	86%	78%	82%	88%	85%	86%	100%	100%	100%	92%	92%	92%

Figura III.2 Resultados obtenidos por Experto en términos de precisión, exhaustividad y medida-F

La Tabla III.4 clasifica los resultados que se han obtenido durante el experimento en diferentes categorías. En primer lugar, organiza los resultados obtenidos para cada uno de los expertos en diferentes columnas. La última columna representa una media aritmética de los resultados obtenidos por todos los expertos en cada experimento. En segundo lugar, organiza la información en dos grupos de filas que contienen resultados obtenidos por consultas simples, que solo hacen referencia a un tema, y las que contienen resultados obtenidos por consultas múltiples, que describen más de un tema. Por ejemplo, siguiendo esta clasificación, la primera fila representa los resultados que obtuvieron los expertos en el experimento con la consulta simple "J2EE". De esta fila es posible resaltar que el "Experto 4" obtiene el mejor resultado, con una precisión de 1. Esto significa que todas las sugerencias obtenidas sobre servicios en la nube están en la lista que el experto había seleccionado en relación con esta temática. La exhaustividad en este tema para el "Experto 4" es de 0.83, lo que significa que sólo el 17% de los servicios relevantes almacenados no son recuperados por el sistema. La media aritmética de resultados obtenidos de acuerdo al experimento de todos los expertos sobre esta primera temática según las métricas de precisión, exhaustividad y medida-F son 0.89, 0.80 y 0.84, respectivamente.

El siguiente tema a analizar de entre los utilizados en este proceso de validación es el de "Servicio de aplicaciones", que se encuentra ubicado en la siguiente fila. El mejor resultado es el obtenido por el experimento del "Experto 4" cuyos valores resultantes son 0.84 de precisión, 0.76 de exhaustividad y 0.80 de medida-F. En cuanto a la media aritmética obtenida, los resultados no son tan destacables como los obtenidos en el experimento con el tópico "J2EE", pero aun así dignos de mencionar con una precisión de 0.83, una exhaustividad de 0.73 y un valor de medida-F de 0.78.

Las consultas realizadas asociadas al tema "Ajax" obtienen mejores resultados para el "Experto 3", con una precisión y exhaustividad de 0.88. En este experimento, la media aritmética de todos los resultados incluye una precisión de 0.86, una exhaustividad 0.84 y una medida-F 0.85.

En el caso de las consultas simples, el mejor resultado global es el obtenido para el tema "Bases de datos", donde la precisión media obtenida es de 0.90, la exhaustividad de 0.83 y la medida-F de 0.86. En este mismo tema, destacan los

resultados obtenidos en el experimento del "Experto 2", donde el valor de precisión es de 0.92, la exhaustividad 0.86 y la medida-F 0.89. Por otro lado, el último tema simple, "Sistemas de Información Empresarial", obtiene los peores resultados con una precisión media de 0.83, exhaustividad de 0.68 y medida-F de 0.75. La obtención de unos resultados tan lejanos a los óptimos en este tema se justifica por la diferencia existente entre los sistemas de gestión seleccionados por los expertos y los devueltos por el buscador semántico. A pesar de este resultado, en general, la media agregada de resultados obtenidos para las consultas sobre un sólo tema, que se muestran en la sexta fila de la Tabla III.4, son bastante buenos, con una precisión media de 0.86, una exhaustividad de 0.78 y una medida-F de 0.82.

En el caso de las consultas múltiples, las consultas sobre los temas "Bases de datos y Servicio de aplicaciones" obtuvieron excelentes resultados con una precisión media de 0.92, exhaustividad de 0.88 y medida-F de 0.90. En este tipo de consultas, el experimento llevado a cabo por el "Experto 3" es el que mejores resultados ofrece, con valores de precisión y exhaustividad muy próximos a la unidad, más concretamente, de 0.94. Si comparamos estos valores con los obtenidos por las consultas sobre los temas "J2EE y Bases de datos", se observa que los valores obtenidos en términos de precisión media (esto es, 0.90) son bastante cercanos a los mencionados para los temas "Bases de datos y Servicios de aplicaciones". Sin embargo, en el caso del valor obtenido para la exhaustividad, éste es sustancialmente menor (a saber, 0.83), dejando entrever que un porcentaje de los servicios relevantes no son recuperados.

Por otro lado, aunque las consultas para los temas "Bases de datos y Sistemas de Información Empresarial" y "J2EE y Ajax" obtienen un valor de precisión 0.90, similar a los experimentos analizados anteriormente, el valor de exhaustividad es superior, de 0.89 y 0.85, respectivamente. Este incremento cuantitativo remarca la capacidad del sistema en encontrar un 89% de los servicios en la nube asociados con las consultas sobre "Bases de datos y Sistemas de Información empresarial" y un 85% de los servicios relacionados con las consulta sobre "J2EE y Ajax".

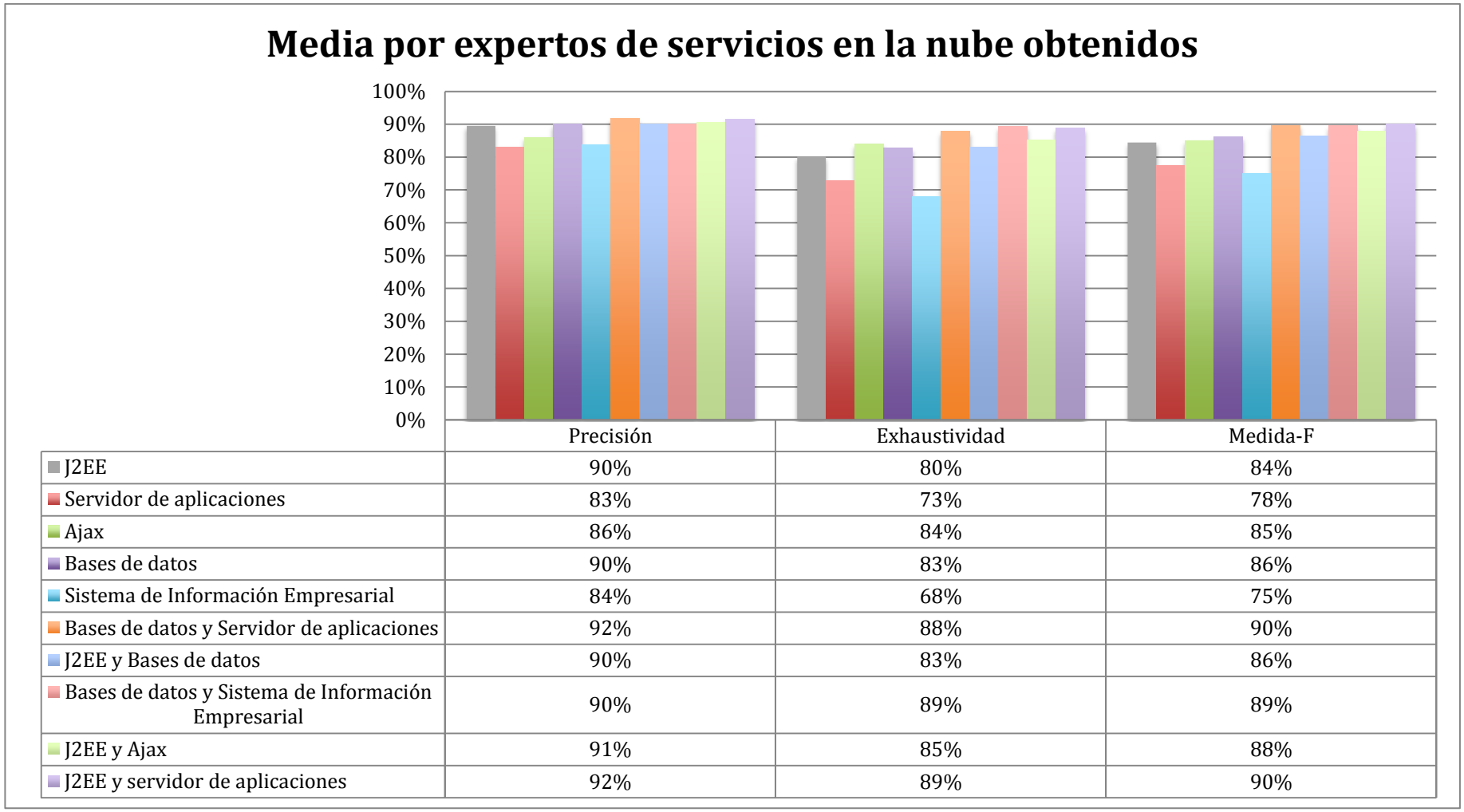
Si analizamos detalladamente la Tabla III.4, los mejores resultados en las consultas múltiples se obtienen para las búsquedas relacionadas con los temas "J2EE y Servidor de aplicaciones", donde el experimento del "Experto 3" logra una

precisión y una exhaustividad del 100%. La media aritmética obtenida para el conjunto de experimentos en estos temas es de una precisión de 0.92%, exhaustividad de 0.89% y medida-F de 0.90%.

Es de destacar que el buscador obtiene mejores valores de precisión en las búsquedas complejas con un 91% de media, frente a las búsquedas simples que sólo obtienen un 86% para este parámetro. Además, en lo referente a la exhaustividad, el buscador obtiene varios resultados bajos para las búsquedas simples, resultando en una media del 78%, frente a las búsquedas complejas en las que el sistema proporciona un 87%. Probablemente, esta diferencia de exhaustividad se deba al hecho de que las consultas múltiples son más específicas que las consultas simples, por lo que la puntuación para el parámetro de exhaustividad mejora significativamente.

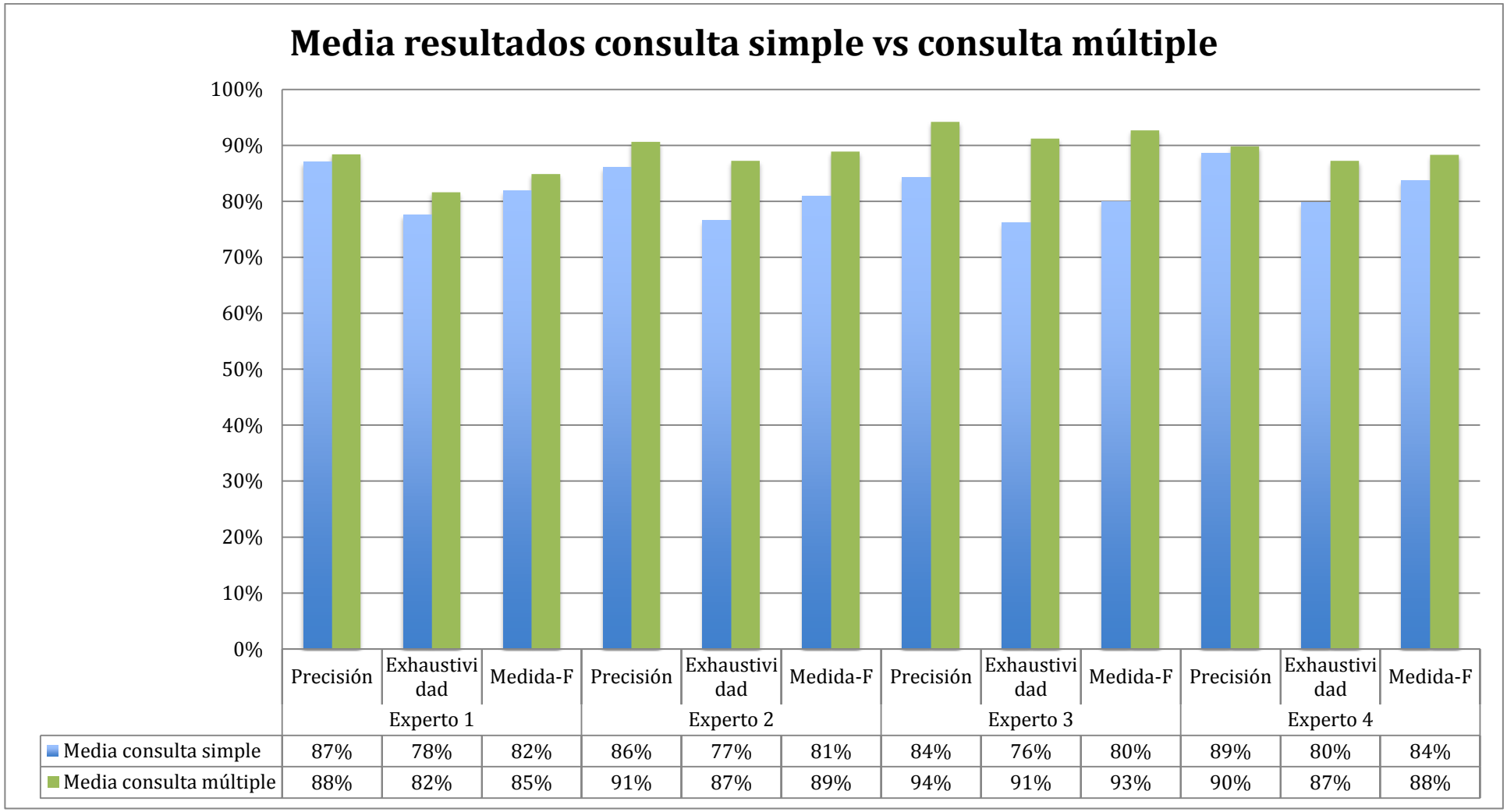
Por último, para concluir esta evaluación se han elaborado unas gráficas que expresan las medias aritméticas de los resultados obtenidos agregando los experimentos de los cuatro expertos colaboradores (véase Figura III.3). Esta figura refleja la media de valores de precisión, exhaustividad y medida-F obtenida por los expertos en los distintos experimentos. Los porcentajes de media obtenidos, en general, son bastante favorables, y las medias más altas son resultado de las búsquedas compuestas de servicios relacionados con los temas “Bases de datos y Servicios de aplicaciones”, con un 92% de precisión, un 88% de exhaustividad y un 90% en la medida-F, y “J2EE y servidor de aplicaciones”, obteniendo un 92% en la precisión, 89% en la exhaustividad y un 90% de medida-F. Sin embargo, como se puede apreciar en la gráfica, las medias más bajas se obtienen para las búsquedas simples de servicios como, por ejemplo, las asociadas al tema “Servidor de aplicaciones”, con un 83% de precisión, 73% de exhaustividad y 78% de medida-F, o al tema “Sistema de Información Empresarial”, con un 84% de precisión, 68% de exhaustividad y 75% de medida-F. Este contraste en términos de precisión entre búsquedas compuestas y búsquedas simples se debe, en gran medida, a la mayor concreción de las consultas compuestas que reducen el espacio de búsqueda y, por lo tanto, son más precisas obteniendo mejores resultados. Las consultas simples, por el contrario, son más genéricas, no concretan tanto el espacio de búsqueda y, por tanto, obtienen mayores cantidades de servicios perdiendo precisión. En la Figura III.4, que representa una media comparativa entre las búsquedas simples y

complejas, se puede comprobar gráficamente esta circunstancia. En esta comparativa se observa que los valores de precisión, exhaustividad y medida-F obtenidos para las búsquedas complejas son bastante más altos que para las búsquedas simples.



	Precisión	Exhaustividad	Medida-F
J2EE	90%	80%	84%
Servidor de aplicaciones	83%	73%	78%
Ajax	86%	84%	85%
Bases de datos	90%	83%	86%
Sistema de Información Empresarial	84%	68%	75%
Bases de datos y Servidor de aplicaciones	92%	88%	90%
J2EE y Bases de datos	90%	83%	86%
Bases de datos y Sistema de Información Empresarial	90%	89%	89%
J2EE y Ajax	91%	85%	88%
J2EE y servidor de aplicaciones	92%	89%	90%

Figura III.3 Media comparativa de servicios en la nube obtenidos



	Experto 1			Experto 2			Experto 3			Experto 4		
	Precisión	Exhaustividad	Medida-F	Precisión	Exhaustividad	Medida-F	Precisión	Exhaustividad	Medida-F	Precisión	Exhaustividad	Medida-F
Media consulta simple	87%	78%	82%	86%	77%	81%	84%	76%	80%	89%	80%	84%
Media consulta múltiple	88%	82%	85%	91%	87%	89%	94%	91%	93%	90%	87%	88%

Figura III.4 Media de resultados obtenidos Consulta simple VS Consulta múltiple

III.3.4. EXTRACTOR DE TÉRMINOS

La metodología de validación seguida para evaluar el módulo de extracción de términos, se ha basado en una extracción manual previa de términos de las 500 descripciones en lenguaje natural de los servicios en la nube confeccionando una lista de términos que fue verificada manualmente. Seguidamente, el extractor de términos analizó de forma automática cada una de estas descripciones de servicios creando otra lista de términos. La validación del módulo consiste en comprobar qué términos de los extraídos automáticamente aparecen de igual manera en la lista de términos extraídos en el proceso manual. Los resultados de esta comparación se muestran en la Tabla III.5, donde (i) los `términos acertados` representan los términos validados, es decir, términos extraídos por el extractor de términos que han sido identificados como `términos relevantes` según el experto; (ii) los `términos extraídos` que, cómo su nombre indica, representan todos los términos extraídos por el módulo de extracción de términos; y, por último, (iii) los `términos relevantes` seleccionados subjetivamente por el experto en el dominio de la computación en la nube. En la Figura III.5 se representan gráficamente estos resultados.

Tabla III.5 Resultados de evaluación del extractor de términos en el dominio de la computación en la nube

Términos acertados	Términos extraídos	Términos relevantes
2322	4147	4300

Figura III.5 Medidas de precisión, exhaustividad y medida-F obtenidas en la evaluación del extractor de términos

Según Tabla III.5, el módulo es capaz de extraer grandes cantidades de términos, de los que sólo un poco más de la mitad 56% son realmente los términos que son utilizados para evolucionar la ontología (véase el valor para la precisión en la Figura III.5). El índice bajo de exhaustividad refleja, por su parte, el bajo nivel de extracción de términos relevantes en los documentos que han sido analizados durante el proceso de extracción.

III.3.5. EVOLUCIÓN DE ONTOLOGÍAS

El módulo de evolución de ontologías se validó empleando dos metodologías de evaluación alternativas. En el contexto de la primera de estas metodologías, se realizó la validación a partir del número de nuevos conceptos incluidos en la ontología. Es necesario tener en cuenta que esta metodología de validación sólo tiene en cuenta aquellos términos proporcionados por el módulo de extracción que han sido encontrados en Wikipedia, descartando todos los demás. Consecuentemente, el número de términos candidatos se reduce ostensiblemente frente al número de términos obtenido originalmente por el extractor de términos. Para la aplicación de esta metodología de evaluación ha sido necesario contar con

un experto que validara manualmente el proceso de evolución de la ontología. La validación consistió en proporcionar al experto ambas versiones, ontología semilla y evolucionada para que éste comprobase manualmente el efecto de los cambios introducidos en la ontología durante el proceso de evolución. Los resultados obtenidos en este proceso de evaluación se definen en la Tabla III.6, donde se muestran (i) los `conceptos relevantes`, que representan aquellos conceptos seleccionados por el experto y que, subjetivamente, representan el dominio de la computación en la nube; (ii) los `conceptos candidatos`, que, eliminando los que no han sido encontrados en Wikipedia, representan los conceptos con los que el módulo de evolución de ontologías ha intentado enriquecer la ontología; y (iii) los `conceptos definidos`, que finalmente han enriquecido la ontología y se encuentran entre la lista de `conceptos relevantes` seleccionada por el experto. A partir de estos valores se ha confeccionado la gráfica que se muestra en la Figura III.6, que presenta estos resultados en términos de precisión, exhaustividad y medida-F.

Tabla III.6 Resultados de evaluación de inserción de conceptos dentro de la ontología en el dominio de la computación en la nube

Conceptos definidos	Conceptos candidatos	Conceptos relevantes
173	274	258

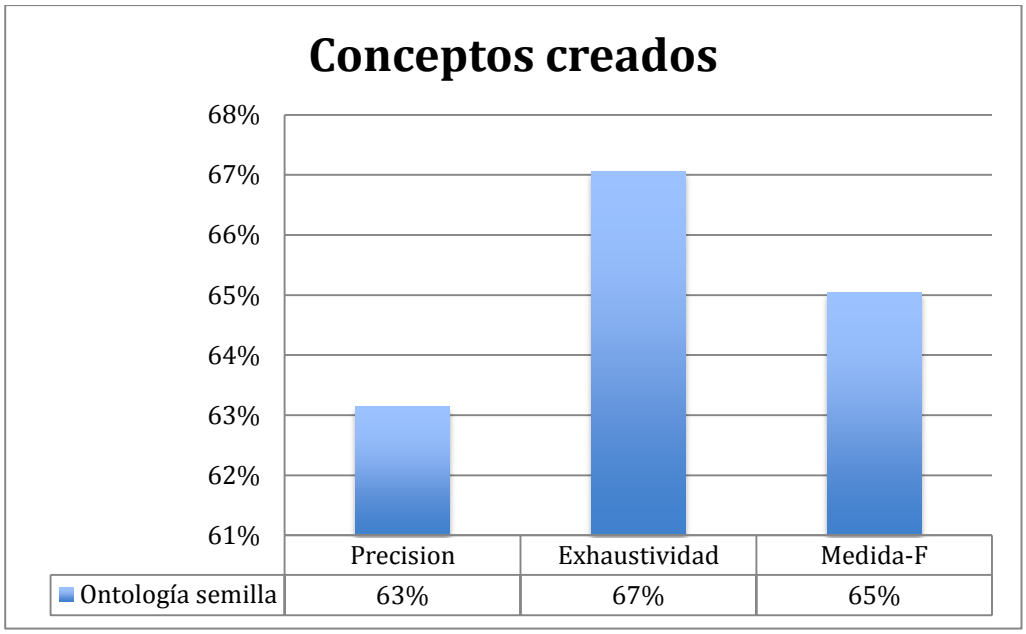


Figura III.6 Medidas de precisión, exhaustividad y medida-F obtenidos durante el proceso de inserción de conceptos en la ontología

Según los datos resultantes del experimento recogido en la Tabla III.6 no existe gran diferencia entre los "`conceptos candidatos`" y los "`conceptos relevantes`". Este hecho se debe a que, como se ha comentado anteriormente, de los conceptos candidatos, obtenidos como resultado de la extracción de términos, sólo se han tenido en cuenta los conceptos que se han recuperado en Wikipedia, descartando todos los demás. Sin embargo, la diferencia se hace más significativa entre estos valores y el número de conceptos que realmente han sido definidos en la ontología ("`conceptos definidos`"). Esta situación refleja la delicada labor del algoritmo de búsqueda en anchura implementado en el módulo de evolución. Como se muestra en la Tabla III.6, muy pocos de los conceptos relevantes son finalmente definidos en la ontología. Esto se debe fundamentalmente a la incapacidad del módulo de evolución para encontrar un camino común de categorías entre los nuevos conceptos a añadir y los conceptos pre-existentes en la ontología. Esta ineficacia se ve reflejada en la Figura III.6, donde el valor de la precisión se ve claramente afectado con respecto a los demás índices.

Por otro lado, la segunda metodología de evaluación que se empleó para analizar las prestaciones del módulo de evolución de ontologías fue la validación de las relaciones taxonómicas. En este caso, la evaluación consistía en validar el número

de relaciones taxonómicas que se crean durante el proceso de evolución de ontologías. Para este propósito, sólo se han contabilizado las relaciones taxonómicas de aquellos conceptos que finalmente han enriquecido el dominio ontológico. La Tabla III.7 recoge los valores obtenidos como resultado del experimento, donde (i) las `relaciones relevantes` representan aquellas relaciones taxonómicas que fueron seleccionadas por el experto; (ii) las `relaciones candidatas` representan las relaciones taxonómicas que el módulo de evolución creo durante los procesos de evolución de ontologías realizados; y (iii) las `relaciones acertadas` representan aquellas relaciones que finalmente han sido definidas en la ontología y se encuentran entre el grupo de `relaciones relevantes` seleccionadas por el experto. En la Figura III.7 se muestra una representación gráfica de estos resultados en términos de los parámetros de precisión, exhaustividad y medida-F.

Tabla III.7 Resultados de evaluación de inserción de relaciones dentro de la ontología En el dominio de la computación en la nube

Relaciones acertadas	Relaciones candidatas	Relaciones relevantes
234	251	260

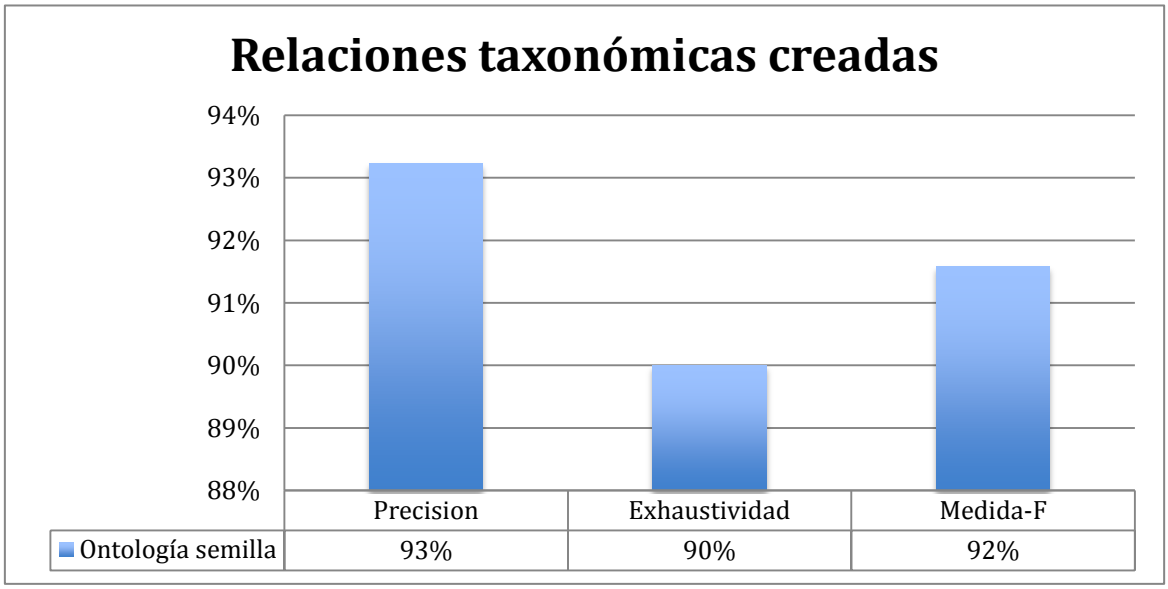


Figura III.7 Medidas de precisión, exhaustividad y medida-F obtenidas durante el proceso de evolución de ontologías

La Tabla III.7 muestra los resultados obtenidos en la creación de relaciones taxonómicas en términos de precisión, exhaustividad y medida-F. Como se puede ver en la gráfica, los resultados representados son bastante buenos, muy cercanos al 100%. Por lo tanto, aunque según el estudio anterior, el número de conceptos que finalmente evolucionan la ontología es bastante bajo, el enriquecimiento taxonómico que se produce después de cada proceso de evolución es muy alto, lo que significa que la cantidad de conceptos que son extraídos de Wikipedia e incorporados a la ontología durante cada proceso de evolución es significante. De esta forma, se contrarrestan los bajos resultados obtenidos en la evaluación del número de conceptos insertados en la ontología, con el número elevado de conceptos insertados en cada proceso de evolución, de ahí los resultados tan elevados de evaluación obtenidos. En la gráfica que se recoge en la Figura III.7 se observan los valores de precisión, exhaustividad y medida-F que hacen patentes los buenos resultados obtenidos por el módulo de evolución de ontologías.

III.3.6. CONCLUSIÓN

Teniendo en cuenta los resultados obtenidos a través de los métodos de evaluación propuestos para la validación individual de la metodología de anotación y recuperación semántica de información sobre el dominio de la computación en la nube, se puede concluir que la metodología desarrollada en esta tesis doctoral es eficiente.

La exhaustiva evaluación realizada sobre el motor de búsqueda semántico ha resultado en unos valores agregados medios de los parámetros de precisión, exhaustividad y medida-F de 0.88, 0.82 y 0.85, respectivamente. Pueden considerarse estos resultados como positivos, sobre todo para las búsquedas compuestas, que destacan con valores superiores a los obtenidos con las búsquedas simples. En concreto, los valores medios agregados para las búsquedas compuestas fueron de un 92% de precisión, un 89% de exhaustividad y un 90% de medida-F. Los resultados obtenidos en las búsquedas simples son mucho más discretos con unos valores de precisión del 86%, de exhaustividad del 78% y de medida-F del 82%. Esta diferencia tan notable refleja el buscador semántico tiene un rendimiento superior cuanto más precisa es la consulta.

En cuanto al módulo de extracción de términos, las medidas de rendimiento obtenidas son bastante más moderados, obteniendo valores de precisión del 56% y de exhaustividad del 54%. Los valores que se utilizan en el cálculo de estas medidas reflejan un alto número de términos extraídos durante el proceso de análisis. Sin embargo, esta caída tan abultada en la precisión se debe a la diferencia tan importante que existe entre la cantidad de términos que se extrajeron y los términos acertados, es decir, los términos relevantes que fueron extraídos.

Por último, el módulo de evolución de ontologías fue analizado desde dos perspectivas diferentes que arrojan resultados complementarios. Por un lado, en términos de la inserción de conceptos en la ontología se obtiene un porcentaje de precisión modesto, del orden del 63%, derivado fundamentalmente de la gran diferencia entre los conceptos candidatos y los que realmente acaban enriqueciendo la ontología. Por otro lado, en lo que concierne a la definición de nuevas relaciones taxonómicas durante el proceso de evolución de ontologías, el resultado es mucho más favorable, con un porcentaje de precisión del 96%. Estos valores obtenidos evidencian que a pesar del limitado número de conceptos que finalmente enriquecen la ontología, la incorporación de nuevos conceptos durante el proceso de evolución es bastante alto, de ahí, que se hayan obtenido valores tan altos en los resultados de validación de las relaciones taxonómicas.

Por otro lado, para completar este estudio, resultaría de gran interés establecer comparaciones entre los enfoques que fueron analizados en el apartado I.6.3. Sin embargo, este tipo de comparación resulta bastante compleja debido a que ni las aplicaciones software analizadas se encuentran disponibles ni tampoco la batería de ejemplos que fueron utilizados en sus validaciones. También se puede hacer notar que para realizar una comparación razonable, debería de tenerse en cuenta tanto la batería de ejemplos como las características relacionadas con el modelo ontológico utilizado por el enfoque, recursos que no se encuentran disponibles. Además, otra problemática agregada a esta comparación son las métricas, algunas de los enfoques analizados no proporcionan ningún tipo de métrica estadística relacionada con la que se utiliza en esta tesis doctoral. Por lo tanto, esta situación limita bastante la posibilidad de establecer comparaciones con cualquiera de las aproximaciones analizadas.

III.4. VALIDACIÓN EN EL DOMINIO DE LA INVESTIGACIÓN, DESARROLLO E INNOVACIÓN (I+D+I)

III.4.1. INTRODUCCIÓN

La innovación consiste en producir, asimilar y explotar con éxito una novedad en el campo económico y social, en un intento de aportar nuevas soluciones a los problemas y facilitar el cumplimiento de las necesidades de los individuos y la sociedad (COTEC, 2004). En general, la necesidad de innovación siempre ha existido en la industria, pero esta necesidad ha cogido más relevancia debido al contexto económico actual donde los clientes exigen cada vez mejores servicios y productos con precios más bajos. Por lo tanto, el hecho de estar a la vanguardia de la producción y la comercialización permite a las organizaciones disponer de una ventaja competitiva con respecto a sus competidores. En la práctica, para una organización significa impulsar su productividad mejorando la calidad y el precio de sus productos.

En las organizaciones, el concepto de innovación se encuentra profundamente relacionado con la gestión de la Investigación y el Desarrollo (I+D). Esta gestión se define como un conjunto general de procesos y procedimientos que garantizan el éxito de la empresa en todos sus objetivos. Por lo general, esta gestión afecta a diferentes fases del proceso de desarrollo de proyectos como, por ejemplo, la entrega de propuestas, selección de los proyectos, administración de los proyectos, difusión, etc. En cada una de estas fases el intercambio de información entre los grupos involucrados en el desarrollo del proyecto es crucial para poder analizar toda la información y facilitar una toma de decisiones sobre cualquier tarea relacionada con la gestión de la I+D+i dentro de un proyecto.

Hoy en día, la gestión de la I+D está estrechamente relacionada con la integración y la colaboración entre los diferentes grupos de interés (Nobelius, 2004). Por lo tanto, la I+D se describe como una red de actores tales como competidores, proveedores, distribuidores, que se centran en la colaboración dentro de un entorno empresarial. El entorno colaborativo requerido por la gestión I+D y la transferencia de conocimiento necesaria para que esta gestión funcione correctamente fueron las características que hicieron emerger la idea de

utilizar este dominio como herramienta de validación del sistema propuesto en esta tesis doctoral.

En este dominio, el propósito de la evaluación es, a parte de comprobar el rendimiento del sistema propuesto en el citado dominio, demostrar el poder de adaptación del sistema propuesto y analizar cómo afectaría la utilización del sistema propuesto en la gestión de proyectos de I+D+i. Sobre todo, se desea analizar qué beneficios supondría para los procesos de búsqueda de información que normalmente requieren de estos procesos de gestión.

III.4.2. ESCENARIO DE EVALUACIÓN

Para plantear el escenario de evaluación, como ya se comentó en dominio de evaluación anterior, se requiere de una ontología que modele el dominio a partir del cual se crearán los metadatos necesarios para el proceso de anotación semántica. En concreto, en este caso el dominio que se va a utilizar para evaluar el sistema es el de la I+D+i en el sector de las TIC. Por lo tanto, se reutilizará la ontología del proyecto DESWAP que fue empleada para la evaluación del sistema en el dominio de la computación en nube.

La validación del sistema en este dominio ha requerido de la compilación de un corpus de 100 documentos relacionados con el área de la investigación y el desarrollo. Entre estos documentos se encuentran: proyectos de I+D+i, memorias de proyectos, artículos de investigación relacionados con proyectos, entregables redactados para justificaciones, y propuestas que describen las bases de futuros proyectos, entre otros. Cada uno de estos documentos fue anotado semánticamente y almacenado en un repositorio de ontologías implementado a través de Virtuoso. Después de realizar las anotaciones se calcularon los índices semánticos y se almacenaron en una base de datos relacional.

III.4.3. MOTOR DE BÚSQUEDA SEMÁNTICO

La metodología de evaluación que se va a utilizar para validar el motor de búsqueda semántica es idéntica a la que se empleó para evaluar este componente

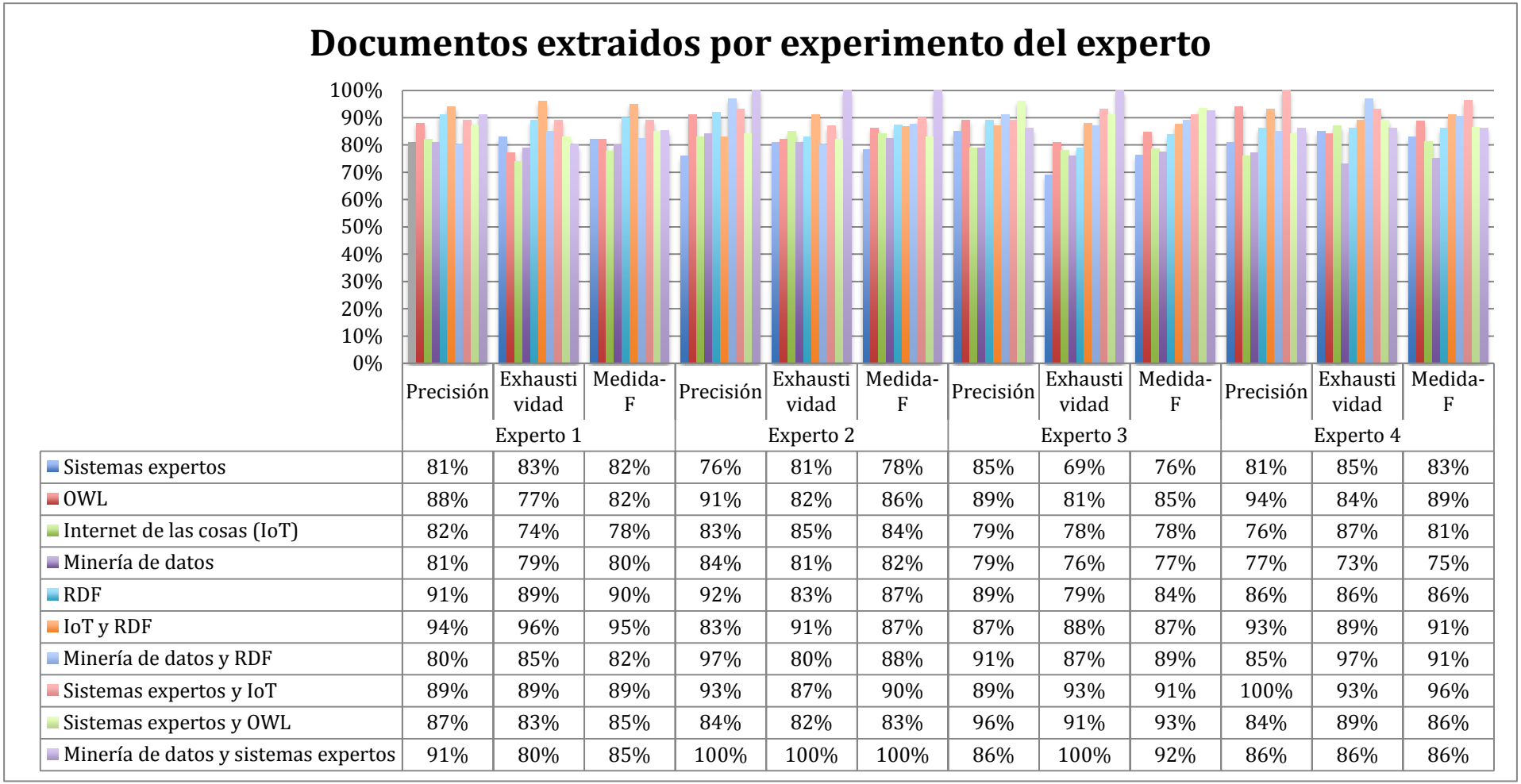
en el dominio de la computación en nube. Por tanto, fue necesario contar con cuatro expertos en I+D+i que hicieran cinco consultas sobre diez temas predefinidos relacionados con tecnologías de la I+D+i. Los temas que fueron establecidos son los siguientes: "Sistemas expertos", "OWL", "Internet de las cosas", "Minería de datos", "RDF", "Internet de las cosas y RDF", "Minería de datos y RDF", "Sistemas expertos y Internet de las cosas", "Sistemas expertos y OWL" y "Minería de datos y sistemas expertos". Para cada consulta, los expertos seleccionaron manualmente los documentos más adecuados en función del tema de la consulta realizada. Seguidamente, se ejecutaron las mismas consultas en el motor de búsqueda semántico para comprobar cuántos de los documentos seleccionados por los expertos son sugeridos también por el motor de búsqueda. La Tabla III.8 muestra los resultados obtenidos en el experimento para cada uno de los expertos en términos de sugerencias acertadas ('A'), sugerencias extraídas ('E') y sugerencias relevantes ('R'). Las sugerencias acertadas ('A') representan aquellas sugerencias recuperadas por el sistema que han sido validadas por el experto, es decir, que se encuentran entre las `sugerencias relevantes` seleccionadas por el experto; las sugerencias extraídas ('E') representan las sugerencias que son recuperadas por el motor de búsqueda semántico; y por último, las sugerencias recuperadas ('R') representan aquellas sugerencias que fueron seleccionadas por los expertos. A partir de estos valores se construye la Tabla III.9, que recoge los valores de precisión ('P'), exhaustividad ('E') y medida-F ('F'). La Figura III.2 proporciona una representación más gráfica y expresiva de los resultados obtenidos en los experimentos.

Tabla III.8 Valores de sugerencias acertadas (a), Extraidas (E) y relevantes (R) En el dominio de las I+D+i

	Experto 1			Experto 2			Experto 3			Experto 4		
Temas	**A**	**E**	**R**	**A**	**E**	**R**	**A**	**E**	**R**	**A**	**E**	**R**
Sistemas expertos	31	38	37	24	32	30	23	27	33	32	40	38
OWL	11	13	15	15	16	18	11	13	14	9	10	11
Internet de las cosas (IoT)	20	24	27	29	36	35	19	25	25	27	35	31
Minería de datos	35	44	45	31	38	39	31	39	41	26	34	36
RDF	17	18	19	17	19	21	19	21	24	15	17	17
IoT y RDF	10	11	11	13	16	15	8	9	9	11	12	13
Minería de datos y RDF	10	13	12	12	12	15	9	10	11	12	14	12
Sistemas expertos y IoT	16	18	18	13	14	15	14	16	15	15	15	16
Sistemas expertos y OWL	4	5	5	3	4	4	4	4	4	5	6	6
Minería de datos y sistemas expertos	18	20	23	19	19	19	27	31	27	18	21	21

Tabla III.9 Valores de precisión (P), exhaustividad (E) y medida-F (F) obtenidos durante la evaluación en el dominio de las I+D+i

	Experto 1			Experto 2			Experto 3			Experto 4			Media Experto		
Temas	**P**	**E**	**F**	**P**	**E**	**F**	**P**	**E**	**F**	**P**	**E**	**F**	**P**	**E**	**F**
Sistemas expertos	0,81	0,83	0,82	0,76	0,81	0,78	0,85	0,69	0,76	**0,81**	**0,85**	**0,83**	0,81	0,79	0,80
OWL	0,88	0,77	0,82	0,91	0,82	0,86	0,89	0,81	0,85	**0,94**	**0,84**	**0,89**	0,90	0,81	0,85
Internet de las cosas (IoT)	0,82	0,74	0,78	**0,83**	**0,85**	**0,84**	0,79	0,78	0,78	0,76	0,87	0,81	0,8	0,81	0,80
Minería de datos	0,81	0,79	0,80	**0,84**	**0,81**	**0,82**	0,79	0,76	0,77	0,77	0,73	0,75	0,80	0,77	0,79
RDF	**0,91**	**0,89**	**0,90**	0,92	0,83	0,87	0,89	0,79	0,84	0,86	0,86	0,86	0,89	0,84	0,87
Total tema simple	0,85	0,80	0,82	0,85	0,82	0,84	0,84	0,76	0,80	0,83	0,83	0,83	0,84	0,81	0,82
IoT y RDF	**0,94**	**0,96**	**0,95**	0,83	0,91	0,87	0,87	0,88	0,87	0,93	0,89	0,91	0,89	0,91	0,90
Minería de datos y RDF	0,8	0,85	0,8	**0,97**	**0,8**	**0,88**	0,91	0,87	0,89	0,85	0,97	0,91	0,88	0,87	0,87
Sistemas expertos y IoT	0,89	0,89	0,89	0,93	0,87	0,90	0,89	0,93	0,91	**1**	**0,93**	**0,96**	0,93	0,90	0,91
Sistemas expertos y OWL	0,87	0,83	0,85	0,84	0,82	0,84	**0,96**	**0,91**	**0,93**	0,84	0,89	0,86	0,88	0,86	0,87
Minería de datos y sistemas expertos	0,91	0,8	0,85	**1**	**1**	**1**	0,86	1	0,92	0,86	0,86	0,86	0,91	0,91	0,91
Total tema compuesto	0,88	0,87	0,87	0,91	0,88	0,90	0,90	0,92	0,91	0,90	0,91	0,90	0,90	0,89	0,89
Total	0,86	0,83	0,85	0,88	0,85	0,86	0,87	0,84	0,85	0,86	0,87	0,86	**0,87**	**0,85**	**0,86**



	Experto 1			Experto 2			Experto 3			Experto 4		
	Precisión	Exhaustividad	Medida-F	Precisión	Exhaustividad	Medida-F	Precisión	Exhaustividad	Medida-F	Precisión	Exhaustividad	Medida-F
Sistemas expertos	81%	83%	82%	76%	81%	78%	85%	69%	76%	81%	85%	83%
OWL	88%	77%	82%	91%	82%	86%	89%	81%	85%	94%	84%	89%
Internet de las cosas (IoT)	82%	74%	78%	83%	85%	84%	79%	78%	78%	76%	87%	81%
Minería de datos	81%	79%	80%	84%	81%	82%	79%	76%	77%	77%	73%	75%
RDF	91%	89%	90%	92%	83%	87%	89%	79%	84%	86%	86%	86%
IoT y RDF	94%	96%	95%	83%	91%	87%	87%	88%	87%	93%	89%	91%
Minería de datos y RDF	80%	85%	82%	97%	80%	88%	91%	87%	89%	85%	97%	91%
Sistemas expertos y IoT	89%	89%	89%	93%	87%	90%	89%	93%	91%	100%	93%	96%
Sistemas expertos y OWL	87%	83%	85%	84%	82%	83%	96%	91%	93%	84%	89%	86%
Minería de datos y sistemas expertos	91%	80%	85%	100%	100%	100%	86%	100%	92%	86%	86%	86%

Figura III.8 Resultados obtenidos por experto en términos de precisión, exhaustividad y medida-F

La Tabla III.9 recoge los resultados que se han obtenido durante todos los experimentos de evaluación realizados ofreciendo diferentes agrupaciones que facilitan su análisis. Por un lado, la organización en columnas permite asilar los experimentos realizados por experto separando los resultados obtenidos. Además, la última columna incluye los valores de media aritmética de los resultados obtenidos de todos los expertos. Por otro lado, la organización en filas permite diferenciar los experimentos de consultas simples que hacen referencia a un tema y experimentos de consultas múltiples que están constituidas por varios temas. En la transición de un tipo de experimento a otro se ha añadido una nueva fila que recoge la media aritmética obtenida, por experto, de todas las consultas de ese tipo (simples o múltiples) realizadas. Finalmente, la última fila de la tabla refleja la media aritmética agregada de todos los resultados obtenidos, tanto para experimentos de consultas simples como múltiples. Por ejemplo, continuando con esta clasificación, en la primera fila se representan los resultados obtenidos por los expertos en la consulta simple "Sistemas expertos". En esta agrupación de resultados, es posible destacar que el "Experto 4" obtiene el mejor resultado, con una precisión de 0.81. Esto significa que en el 81% de los casos las sugerencias obtenidas sobre las tecnologías de I+D+i se encuentran en al lista que el experto había, previamente, seleccionado en relación a la temática "Sistemas expertos". La exhaustividad obtenida por el "Experto 4" es de 0.85, lo que significa que sólo el 15% de los recursos relevantes almacenados no son recuperados por el sistema. La media aritmética de los resultados obtenidos a partir de los resultados agregados de todos los expertos en esta temática según las métricas de precisión, exhaustividad y medida-F en este primer análisis son de 0.81, 0.79 y 0.80, respectivamente.

Para las consultas sobre el tema "OWL", es nuevamente el "Experto 4", con valores de precisión del 94%, exhaustividad del 84% y medida-F del 89%, quien obtiene mejores resultados. Comparando estos resultados con los obtenidos para el tema anterior, "Sistemas Expertos", se puede apreciar que el valor de precisión aumenta considerablemente obteniendo resultados sobresalientes. Este alto valor para la precisión indica que el buscador semántico es más concreto a la hora de recuperar información. Se puede concluir que el buscador es más preciso cuanto más específicas son las consultas. De ahí, que los resultados de precisión sean más

bajos para el experimento anterior donde se define un tema genérico como "Sistemas expertos" en contraste con un tema tan específico como es "OWL".

Como se ha destacado anteriormente, la Tabla III.9 distingue entre dos tipos de consultas, a saber, consultas simples, aquellas definidas por un solo concepto a buscar y consultas múltiples, aquellas en las que se utiliza más de un concepto. En el caso de las consultas simples, los valores de precisión más altos que se han logrado se corresponden con las búsquedas relacionadas con los temas "OWL" y "RDF", cuyos valores medios de precisión son 90% y 89%, respectivamente. En particular, los mayores valores de precisión entre todos los expertos para estos temas los obtienen el "Experto 4", para el caso de "OWL" con un 94%, y el "Experto 1", para el caso de "RDF" con un 91%.

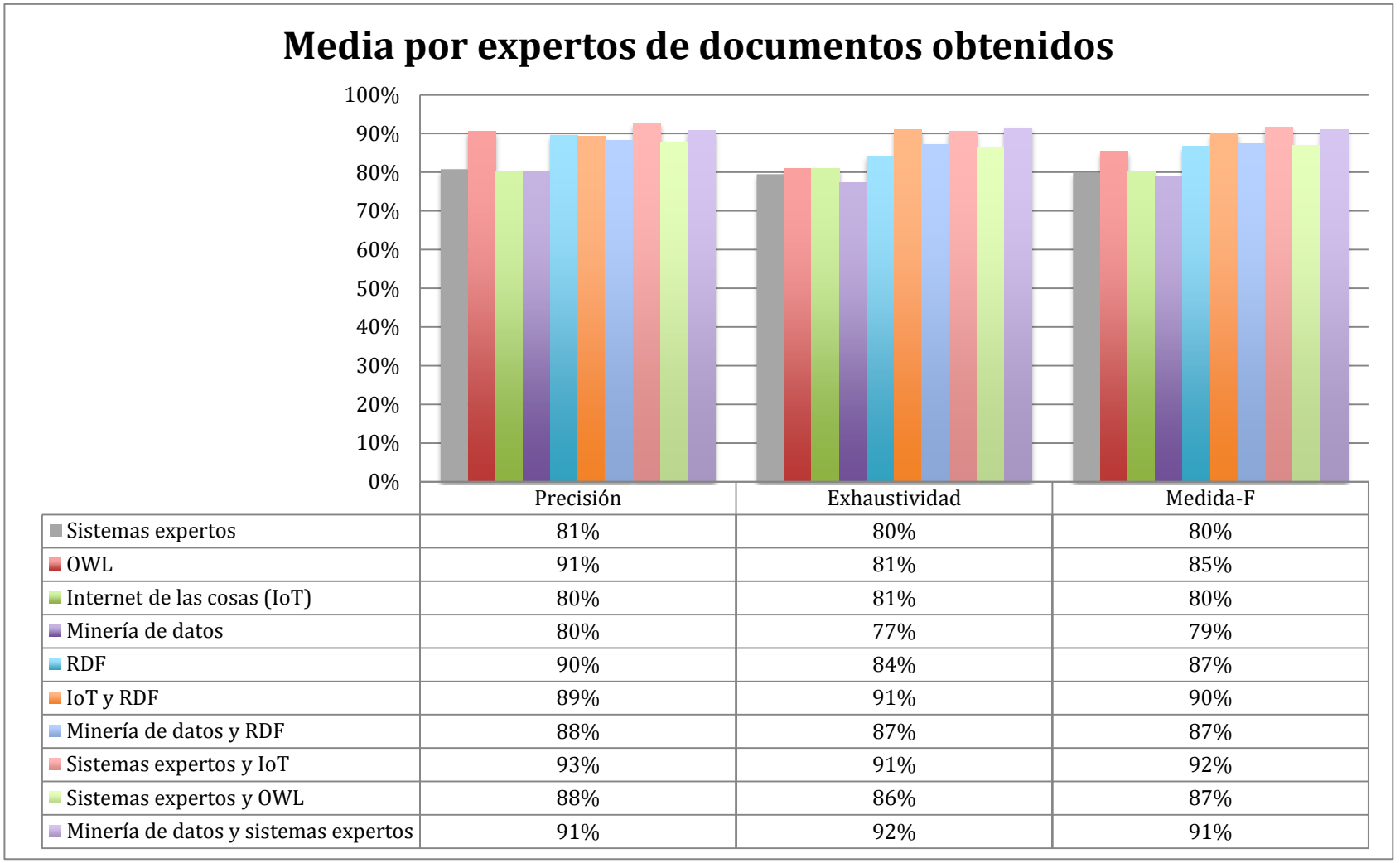
En el caso de las consultas múltiples, destacamos los tópicos "Minería de datos y sistemas expertos" donde los experimentos obtenidos por el "Experto 2" arrojan unos valores excelentes de precisión de 1, exhaustividad de 1 y medida-F de 1. En este mismo tema, destacaremos los resultados obtenidos por el experimento llevado a cabo por el Experto 1, que con un valor de 91% de precisión obtiene el segundo valor más alto de precisión de entre todos demás experimentos en este tema. Si comparamos estos valores con los obtenidos por el Experto 3 en el tema "Minería de datos y RDF" se observa que los valores obtenidos en términos de precisión (91%) obteniendo el mismo valor de precisión que "Minería de datos y sistemas expertos". Sin embargo, en el caso del valor obtenido para la exhaustividad es más alta 87% que la obtenida por el "Experto 1" 80% lo que significa que el experimento realizado por el "Experto 1" obtuvo un bajo porcentaje de documentos relevantes que no son recuperados.

Por otro lado, se destacan los resultados de las consultas "Internet de las cosas y RDF" y "Sistemas expertos y Internet de las cosas" debido a su significante valor de exhaustividad, obteniendo un 96% en el experimento realizado por el "Experto 1"y 93% en el experimento realizado por el "Experto 4". Estos incrementos cuantitativos recalcan la capacidad del sistema en encontrar un 96% de los documentos asociados a la consulta "Internet de las cosas y RDF" y un 93% de los documentos relacionados con la consulta "Sistemas expertos y Internet de las cosas".

De forma agregada, los valores obtenidos para los experimentos con consultas simples se pueden considerar muy positivos, con una precisión media del 84%, una exhaustividad media del 81% y una medida-F del 82%. Sin embargo, al igual que sucedía para el caso de estudio en el dominio de la computación en la nube, los resultados asociados a las consultas múltiples son mucho más prometedores. Como se discutió para el anterior escenario, este hecho se debe a que las búsquedas con más términos dotan a la consulta de más precisión que las consultas simples, de ahí que los valores medios de precisión obtenidos sean mayores para las consultas compuestas que en las consultas simples. En otras palabras, las consultas múltiples tienden a proporcionar más información al incorporar más de un concepto, lo que reduce el campo de búsqueda. Esta reducción mejora los valores de precisión de las consultas múltiples con respecto a las simples.

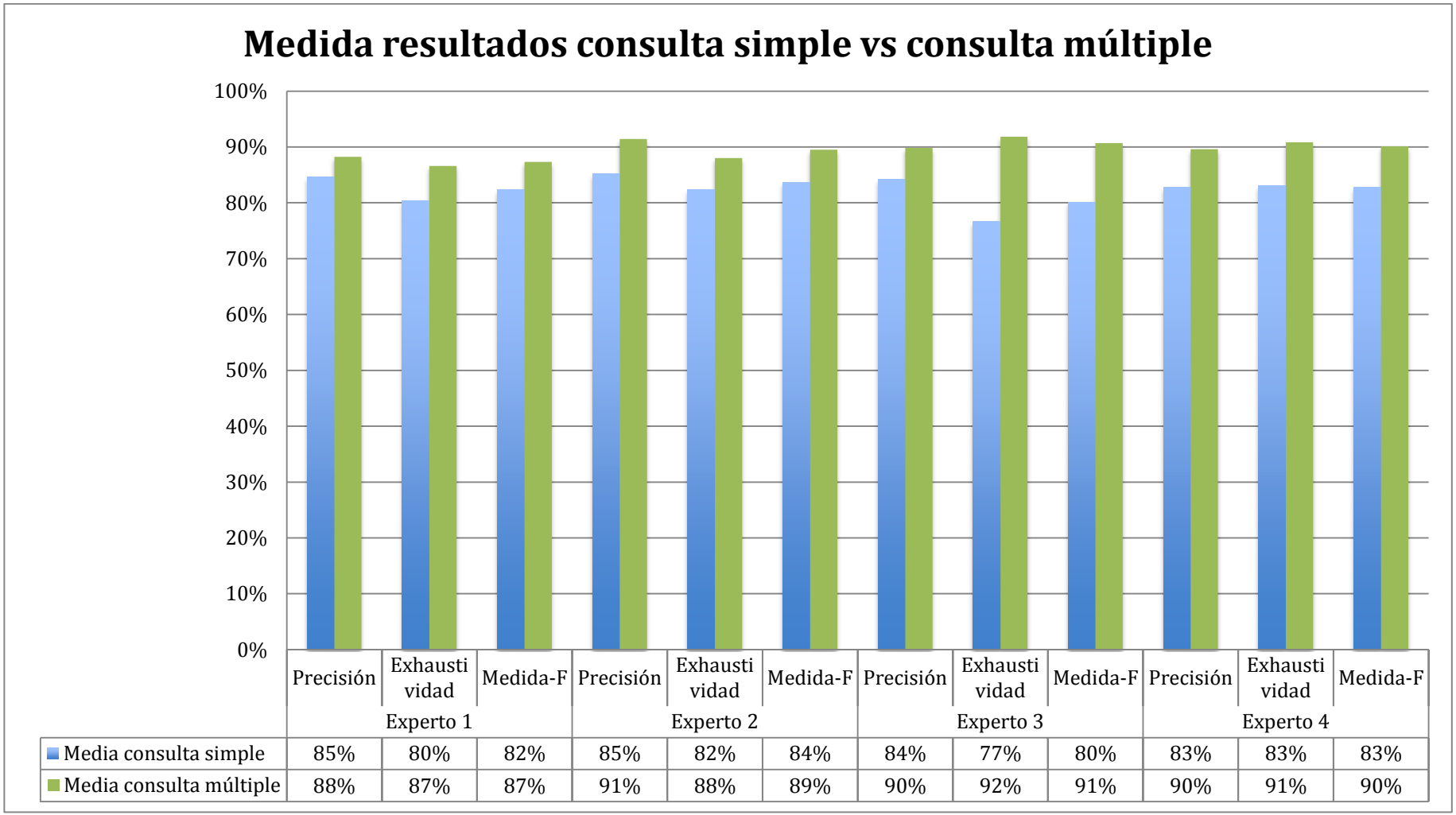
Para finalizar la evaluación del buscador semántico en este dominio, se ha incluido una gráfica que representa las medias aritméticas de los resultados recogidos en la Tabla III.9 (véase Figura III.9). En esta figura se muestra gráficamente la media de los valores de precisión, exhaustividad y medida-F obtenidos por los expertos en todos los experimentos ejecutados. Los porcentajes de media obtenidos son bastante favorables, destacando los resultados obtenidos para los temas “Sistemas expertos e Internet de las cosas” y “Minería de datos y Sistemas expertos”, con valores de precisión: 93%, exhaustividad: 90%, medida-F 91% para el caso del tópico “Sistemas expertos e Internet de las cosas” y, para el caso de “Minería de datos y Sistemas expertos”, se obtienen los valores siguientes de precisión 91%, exhaustividad 91% y medida-F 91%. Sin embargo, como se puede ver en la gráfica, en el caso de las consultas simples como, por ejemplo, “OWL”, los valores de precisión: 90% exhaustividad: 81% y medida-F: 85% o “RDF” los valores de precisión: 89%, exhaustividad: 84% y medida-F: 87% son mucho más bajos que los proporcionados por las consultas múltiples pero a la vez, si se observa detenidamente, son las consultas que mejores resultados han obtenido en la agrupación de consultas simples. Por lo tanto, esta situación ratifica la hipótesis, que también ocurría en la evaluación anterior, y es que el motor de búsqueda semántico obtiene mejores resultados cuando realiza búsquedas concretas como en el caso de los tópicos simples “RDF” o “OWL” en los que se

especifica la tecnología concreta, a diferencia de cuando se realizan experimentos con búsquedas generales como por ejemplo “Sistemas expertos”, donde no se realiza ninguna concreción del tema. Por lo tanto, esta situación nos lleva a concluir que, cómo es lógico, cuanta más concreta sea la consulta, es decir, más información se le proporcione al buscador semántico, más concreto es el campo de búsqueda y por lo tanto mejor resultados obtendrá. La Figura III.10 refleja gráficamente esta suposición donde se puede ver la diferencia considerable entre los valores de evaluación obtenidos para los temas “RDF” y “OWL” con respecto al tema “Sistemas expertos”, en el caso de las consultas simples. Además, si se compara los resultados obtenidos entre las consultas simples y múltiples se observan las desigualdades significantes entre ambos tipos de consulta.



	Precisión	Exhaustividad	Medida-F
Sistemas expertos	81%	80%	80%
OWL	91%	81%	85%
Internet de las cosas (IoT)	80%	81%	80%
Minería de datos	80%	77%	79%
RDF	90%	84%	87%
IoT y RDF	89%	91%	90%
Minería de datos y RDF	88%	87%	87%
Sistemas expertos y IoT	93%	91%	92%
Sistemas expertos y OWL	88%	86%	87%
Minería de datos y sistemas expertos	91%	92%	91%

Figura III.9 Media comparativa de documentos recuperados



	Experto 1			Experto 2			Experto 3			Experto 4		
	Precisión	Exhaustividad	Medida-F	Precisión	Exhaustividad	Medida-F	Precisión	Exhaustividad	Medida-F	Precisión	Exhaustividad	Medida-F
Media consulta simple	85%	80%	82%	85%	82%	84%	84%	77%	80%	83%	83%	83%
Media consulta múltiple	88%	87%	87%	91%	88%	89%	90%	92%	91%	90%	91%	90%

Figura III.10 Medias de resultados obtenidos consulta simple vs Consulta múltiple

III.4.4. EXTRACTOR DE TÉRMINOS

La metodología de validación que ha sido utilizada para evaluar el extractor de términos en el dominio de las I+D+i ha sido la misma que la definida en el domino de los servicios en la nube. A los efectos de esta validación, ha sido preciso llevar a cabo un proceso laborioso de análisis y extracción de términos en los 100 documentos relacionados con el I+D+i de los que se compone el corpus. De la misma forma, el módulo de extracción confeccionó automáticamente una lista de términos a partir del análisis de estos 100 documentos. Una vez que se disponía de ambas listas de términos, la validación ha consistido en comprobar cuántos de los términos extraídos manualmente se encontraban también entre los extraídos automáticamente. Los resultados de esta comparación han sido recopilados en la Tabla III.10, donde (i) los `términos acertados` representan los términos que han sido extraídos por el módulo extractor y que se encuentran en la lista de términos seleccionados por el experto; (ii) los `términos candidatos` representan la lista de términos extraídos por el extractor de términos durante los experimentos realizados; y (iii) los `términos seleccionados` representan los términos que según los expertos se encuentran relacionados con el dominio de las I+D+i. La Figura III.11 muestra una representación gráfica de los resultados obtenidos.

Tabla III.10 Resultados obtenidos en la evaluación del extractor de términos en el dominio de las I+D+i

Términos acertados	Términos candidatos	Términos seleccionados
2883	5545	5653

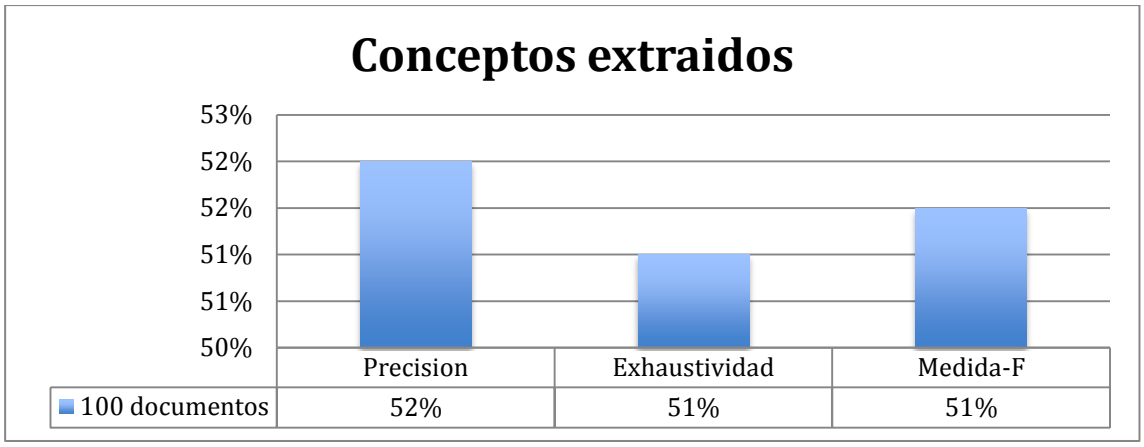


Figura III.11 Resultados de la evoluación de inserción de conceptos en la ontología

Los resultados representados en la Tabla III.10 reflejan el elevado número de términos extraídos, que se acerca bastante al número de términos relevantes. Sin embargo, el número de términos acertados no se aproxima a ninguno de estos valores, lo que se traduce en los bajos valores de precisión y exhaustividad que se representan en la Figura III.11. Esta situación deja patente el alto poder de extracción de términos que tiene el módulo, pero también evidencia la baja precisión del mismo. En otras palabras, el módulo de extracción es capaz de analizar y extraer grandes cantidades de términos que no pertenecen al conjunto de términos relevantes, lo que afecta notablemente a los valores de precisión como se aprecia en la gráfica.

III.4.5. EVOLUCIÓN DE ONTOLOGÍAS

El método utilizado para evaluar el módulo de evolución de ontologías en este dominio, es igual al que se explicó en el dominio de la computación en la nube. Este método requiere de la participación de un experto para validar los conceptos insertados en la ontología y las relaciones taxonómicas creadas durante el proceso de evolución. La Tabla III.11 contiene los datos referentes a la inserción de nuevos conceptos en la ontología. La Figura III.12 representa gráficamente estos datos en términos de precisión, exhaustividad y medida-F.

Tabla III.11 Resultados de evaluación de inserción de conceptos en la ontología en el dominio de las I+D+i

Conceptos acertados	Términos candidatos	Conceptos seleccionados
274	370	356

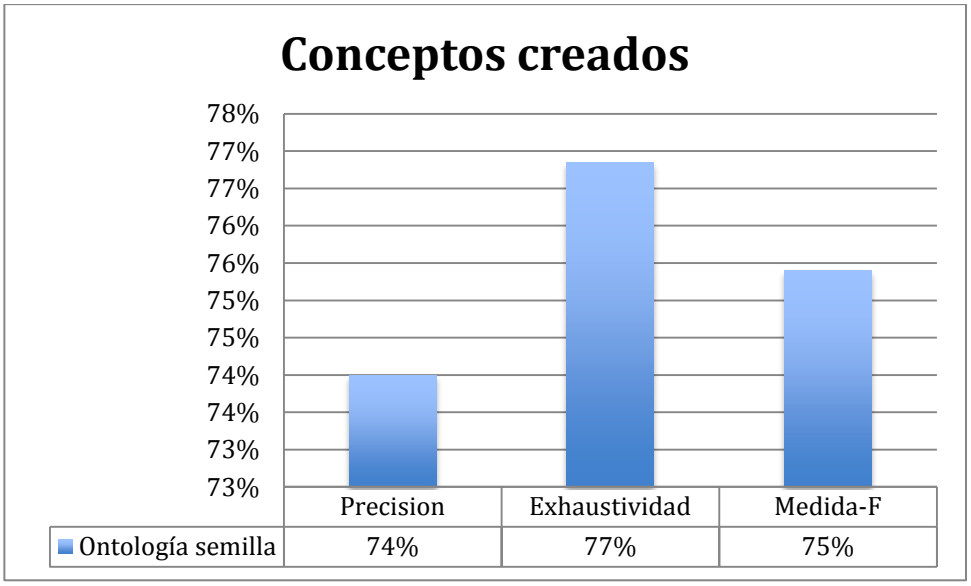


Figura III.12 Medidas de precisión, exhaustividad y medida-F obtenidas en el proceso de inserción de conceptos en la ontología

A partir de los números recogidos en la Tabla III.11, es posible destacar la escasa diferencia en términos cuantitativos entre los valores de "`términos candidatos`" y "`conceptos seleccionados`". Sin embargo, como ocurrió en la anterior validación, la brecha se incrementa cuando entra en juego el número de "`conceptos acertados`", lo que resulta en una merma en la medida de precisión. Este hecho constata que un alto porcentaje de los conceptos candidatos no acaban evolucionando la ontología debido a que el módulo de evolución es incapaz de encontrar una categoría común entre los conceptos de la ontología y los candidatos en función de las relaciones jerárquicas en Wikipedia. Esta imprecisión, se ve reflejada en la Figura III.12 donde la medida de precisión obtenida cae con respecto a la medida de exhaustividad.

Por otro lado, y como se hizo para el dominio anterior, el segundo mecanismo empleado para analizar las prestaciones del módulo de evolución de ontologías consiste en contabilizar el número de relaciones taxonómicas que se generan

durante un proceso de evolución de ontologías completo. Durante este proceso, el sistema busca un camino viable entre un concepto de la lista de candidatos y los conceptos de la ontología a partir de las categorías comunes presentes en Wikipedia. En caso de encontrar un camino de unión, entonces se definen los conceptos necesarios en la ontología para hacer efectiva la unión entre los conceptos mediante relaciones taxonómicas. Los resultados que se han obtenido en este proceso de evaluación se muestran en la Tabla III.12, donde, (i) las `relaciones acertadas` representan la agrupación de relaciones taxonómicas que han sido creadas durante un proceso de evolución de ontologías y se encuentran validadas por el experto, es decir, que se encuentran entre las `relaciones seleccionadas` escogidas por el experto; (ii) las `relaciones candidatas` representan todas las relaciones taxonómicas generadas durante la evolución de la ontología; y (iii) `relaciones seleccionadas` hacen referencia a las relaciones taxonómicas seleccionadas que, para el experto, se encuentran relacionadas con la temática de la I+D+i. Estos valores han sido utilizados para elaborar la gráfica de la Figura III.13 donde se representan las medidas de precisión, exhaustividad y medida-F.

Tabla III.12 Resultados de evaluación de inserción de relaciones en la ontología en el dominio de las I+D+i

Relaciones acertadas	Relaciones candidatas	Relaciones seleccionadas
370	397	425

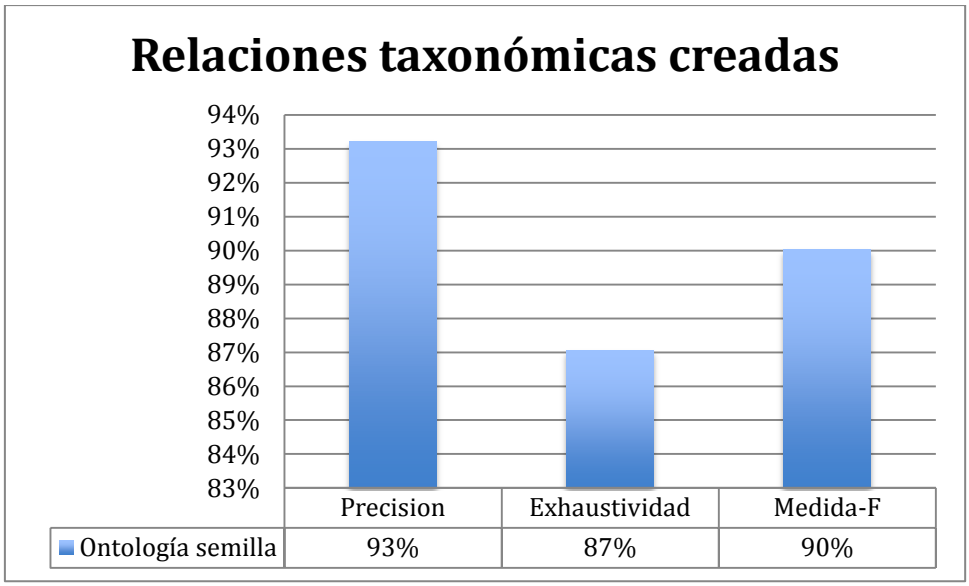


Figura III.13 Medidas de precisión, exhaustividad y medida-F obtenidas en el proceso de evolución de la ontología

Los valores de la Tabla III.12 constituyen buenos resultados en lo referido a la creación de relaciones taxonómicas durante el proceso de evolución de ontologías. La diferencia entre los tres valores representados es mínima, lo que indica que el nivel de enriquecimiento taxonómico durante el proceso de evolución de ontologías es cercano al óptimo. Esto supone un claro contraste frente a los resultados obtenidos en el estudio anterior, donde se ha analizado este mismo módulo en términos de creación de conceptos. Por lo tanto, estos resultados suavizan los malos resultados detectados en el anterior estudio. La Figura III.13 refleja gráficamente la gran efectividad del módulo de evolución de ontologías sobre la base de los altos valores en las medidas de precisión, exhaustividad y medida-F.

III.4.6. CONCLUSIÓN

La evaluación de la metodología de anotación en el dominio de la I+D+i ha proporcionado resultados positivos que constatan la efectividad de la metodología y, sobre todo, la capacidad de ésta para obtener buenos resultados en los dos dominios en los que ha sido evaluada. Estos resultados refuerzan la idea que se ha defendido durante toda la evaluación de la metodología de anotación, la capacidad

de adaptación y rendimiento de la metodología que se describe en esta tesis doctoral.

El método de evaluación que se ha seguido es el mismo que el utilizado en el dominio de computación en la nube. En primer lugar, se ha evaluado el motor de búsqueda semántica. Los resultados que se han obtenidos son muy favorables, siguiendo el mismo comportamiento que en la evaluación anterior. Las medias obtenidas en términos de precisión, exhaustividad y medida-F relativas a las búsquedas simples son bastante discretas, más aún cuando se compara con los resultados de los experimentos asociados a las consultas compuestas. Este hecho lleva a alcanzar las mismas conclusiones que para el escenario de evaluación anterior. Los valores de media obtenidos en las consultas simples son del 78% para la precisión, 81% de exhaustividad y 79% de medida-F. En contraste, los valores obtenidos en las consultas compuestas ascienden a una precisión del 89%, una exhaustividad del 88% y una medida-F del 88%. A pesar de la diferencia significativa entre ambos tipos de consultas, los resultados de media agregados que se obtienen son bastante buenos (83% de precisión, 84% de exhaustividad y 84% de medida-F).

El siguiente módulo validado es el de extracción de términos, obteniendo índices de precisión del 52%, de exhaustividad del 51% y de medida-F del 51%. El bajo rendimiento de este módulo viene originado por una elevada extracción de términos candidatos muy cercana al número de términos seleccionados por el experto, sin embargo, muy alejada del número de términos realmente útiles, términos acertados. De ahí, que los resultados que muestran las métricas de precisión, exhaustividad y medida-F reflejen la imprecisión del método de extracción. Estos valores obtenidos demuestran, por tanto que el módulo de extracción es capaz de obtener grandes cantidades de términos, pero un alto porcentaje de los mismos no coinciden con los términos seleccionados manualmente por los experto, lo que provoca que la precisión disminuya bastante.

Por último, en el módulo de evolución de ontologías se aplican dos métodos de evaluación basados en la contabilidad de conceptos y relaciones taxonómicas creadas durante la evolución de la ontología, respectivamente. Las medidas obtenidas en la evaluación de los conceptos creados son de un 74% de precisión, un 77% de exhaustividad y un 75% de medida-F. Por otro lado, los resultados en

términos de las relaciones taxonómicas creadas son mucho más favorables, alcanzando un 96% de precisión, un 94% de exhaustividad y un 95% de medida-F. El análisis de estos resultados nos lleva a la conclusión de que aunque muchos de los conceptos sean desechados en el proceso de evolución, aquellos conceptos que logran enriquecer la ontología definen una cantidad de relaciones taxonómicas muy alto. Esto supone un alto enriquecimiento de la ontología en cada proceso de evolución.

III.5. CONCLUSIÓN GLOBAL

En este apartado se lleva a cabo un análisis comparativo, entre los resultados en los dos dominios contemplados y se discuten las principales conclusiones extraídas del proceso de validación del sistema de anotación semántica propuesto en esta tesis doctoral.

El método de comparación, que se va a seguir en este apartado, se basa en contrastar los resultados de las evaluaciones llevadas a cabo en sendos dominios módulo a módulo. En primer lugar, se comparan los resultados agregados obtenidos por el motor de búsqueda semántica. La Figura III.14 muestra gráficamente los valores medios de los parámetros de precisión, exhaustividad y medida-F obtenidos en ambos dominios.

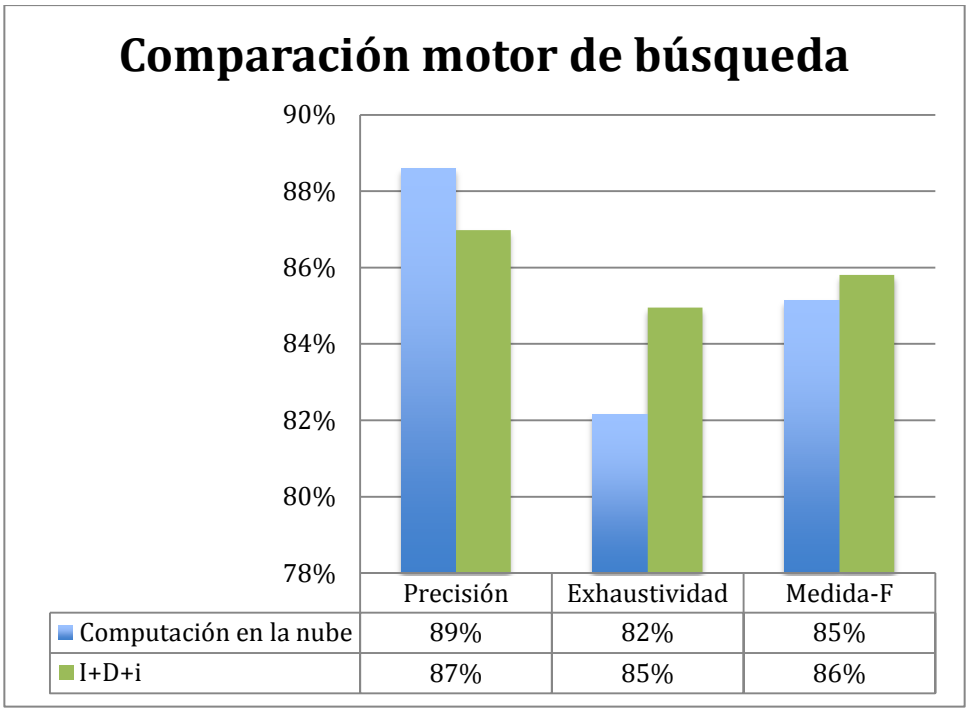


Figura III.14 Comparación de resultados de evaluación del motor de búsqueda semántico para Computación en la nube vs. I+D+i

Los resultados de la Figura III.14 muestran que en el dominio de la computación en la nube el motor de búsqueda tuvo una precisión más elevada que en el dominio de la I+D+i. Estos valores de precisión sugieren que la longitud del texto puede afectar a la efectividad del motor de búsqueda. Concretamente, las descripciones de los servicios en la nube anotados semánticamente tenían menor longitud en términos de palabras que los documentos relacionados con el dominio de la I+D+i. Por otro lado, para el índice de exhaustividad ocurre al contrario y el resultado obtenido por el dominio de la I+D+i es ligeramente superior al de computación en la nube. Esto refleja que el motor de búsqueda realiza una función más eficiente obteniendo toda la información relevante en cada búsqueda para el caso de la I+D+i. Ambos resultados, precisión y exhaustividad, resaltan la relevancia del corpus como elemento importante que repercute sobre los resultados obtenidos.

En segundo lugar, con respecto al módulo de extracción de términos, en la Figura III.15 se representan gráficamente los resultados obtenidos durante los experimentos realizados.

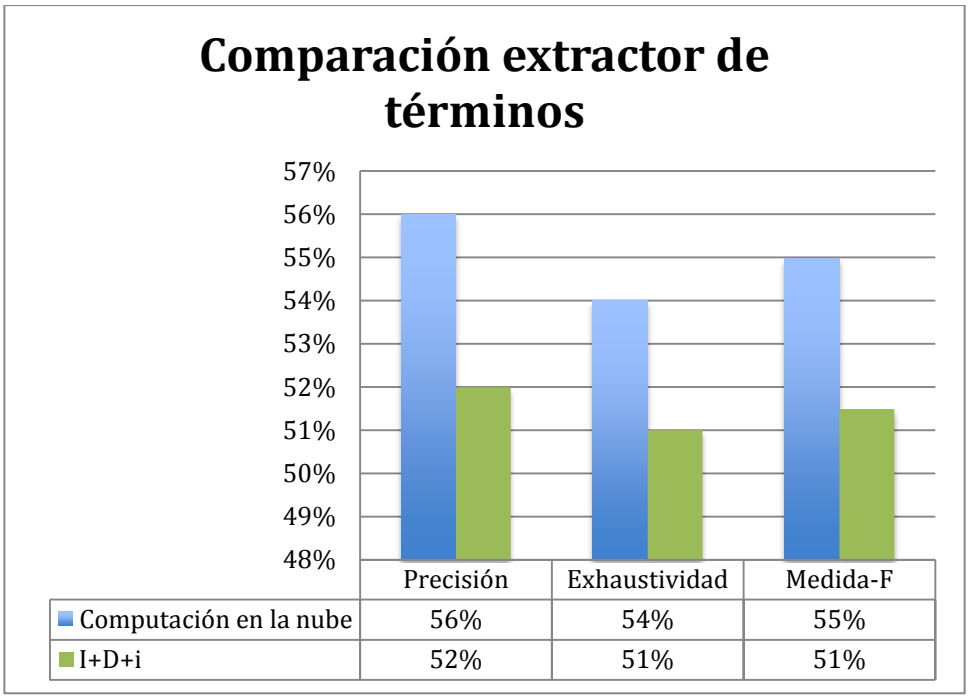


Figura III.15 Comparación de resultados de evaluación del extractor de términos para computación en la nube vs. I+D+i

Los resultados representados en la Figura III.15 son bastante similares. Analizando la gráfica es posible comprobar un comportamiento parecido en la evolución de los valores de los distintos parámetros en sendos dominios. La razón por la que los valores de precisión sean más altos en el dominio de la computación en la nube con respecto al dominio de la I+D+i se debe a la longitud del corpus utilizado por el módulo extractor de términos. En el dominio de la computación en la nube la longitud de los corpus es inferior al empleado en el dominio de la I+D+i. Por lo tanto, al tener descripciones de los servicios en la nube con una longitud menor al de los documentos en el corpus de la I+D+i, esto hace que la precisión del extractor de términos en el dominio de la computación en la nube sea mayor que en el dominio de la I+D+i.

Por último, a continuación se compara el módulo de evolución de ontologías desde las dos perspectivas en que ha sido analizado a lo largo del documento: la creación de conceptos y la creación de relaciones taxonómicas. En primer lugar, la Figura III.16 representa las medidas de precisión, exhaustividad y medida-F que se han obtenido en la evaluación asociada con la cantidad de conceptos creados en proceso de evolución ontológica.

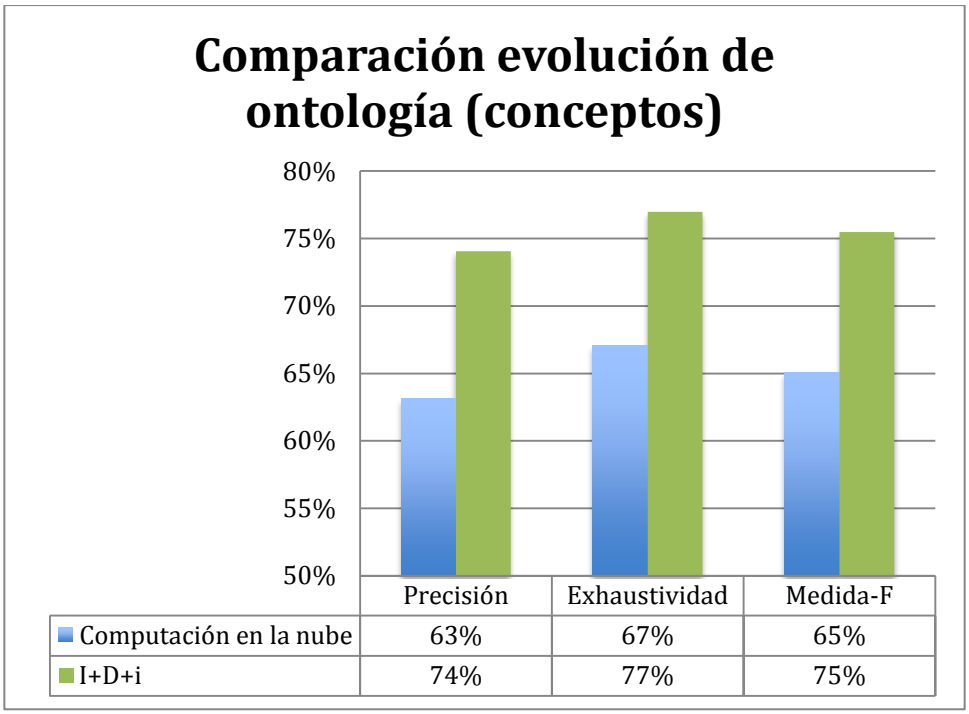


Figura III.16 Comparación de resultados de evaluación de la creación de conceptos en la evolución de ontología para Computación en la nube VS. I+D+i

La Figura III.16 representa gráficamente el mismo comportamiento en ambos dominios. El dominio con mejores resultados es el de la I+D+i. Este hecho refleja la importancia de la selección del corpus, que en el caso de la I+D+i es más rico terminológicamente que en el caso de la computación en la nube. De esta forma se posibilita un ligero incremento en los resultados de los parámetros utilizados en la evaluación del módulo.

En segundo lugar, la Figura III.17 representa los resultados obtenidos en las evaluaciones llevadas a cabo sobre la creación de relaciones taxonómicas durante el proceso de evolución de las ontologías.

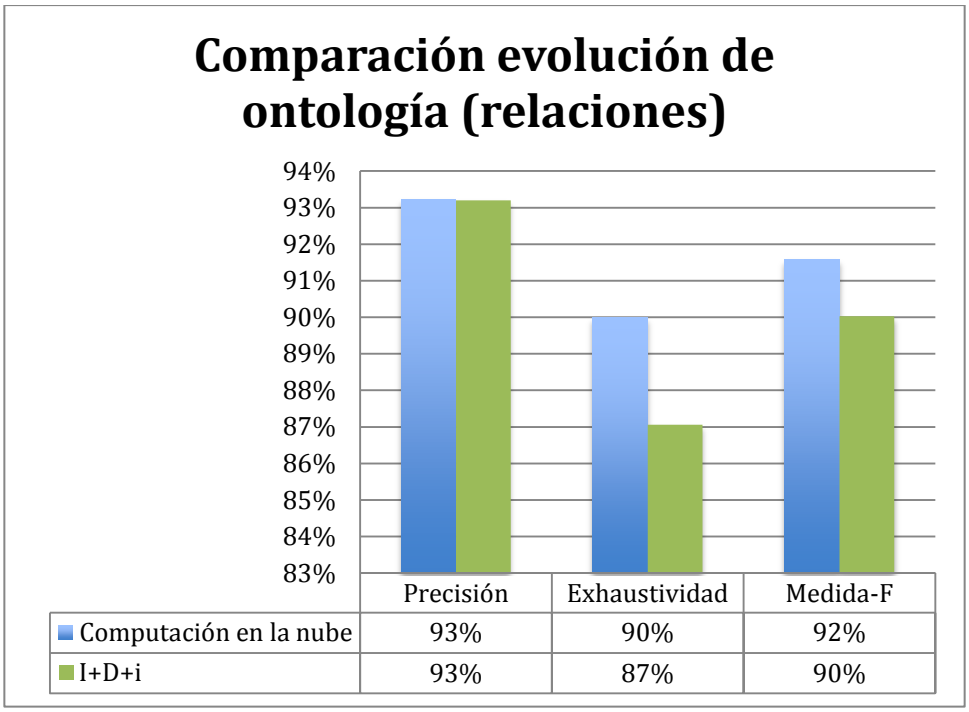


Figura III.17 Comparación de resultados de evaluación de la creación de relaciones en la evolución de ontología para Computación en la nube VS. I+D+i

Los resultados que se muestran en la Figura III.17 presentan el mismo comportamiento en ambos dominios. Este hecho deja patente que en ambos dominios el enriquecimiento de la ontología durante cada proceso de evolución es máximo a pesar de, como revela el análisis de resultados de creación de conceptos, se pierdan muchos conceptos durante este proceso.

III.6. RESUMEN

En este capítulo se presenta una evaluación de la metodología de anotación descrita en el Capítulo II. Esta evaluación se ha llevado a cabo en dos dominios diferentes: el dominio de la computación en la nube y el dominio de la I+D+i. La utilización de distintos dominios para evaluar tiene por objetivo demostrar la adaptabilidad y extensibilidad de la metodología diseñada en esta tesis doctoral.

En el epígrafe III.2 de este capítulo se realiza una breve introducción sobre las diferentes medidas de evaluación que van a ser utilizadas durante el proceso de evaluación de la metodología, a saber, precisión, exhaustividad y medida-F. Posteriormente, puntualiza qué módulos de la metodología van a ser evaluados y

cómo se van a evaluar, utilizando extensiones de las medidas descritas al inicio del epígrafe. En concreto, los módulos evaluados son: el buscador semántico, el extractor de términos y el sistema de evolución de ontologías.

En el epígrafe III.3 se evalúa la metodología en el dominio de la computación en la nube. En primer lugar, se realiza una breve introducción en la que se describe el efecto tecnológico de la computación en la nube en la actualidad y la arquitectura del paradigma, analizando cada capa de computación en la nube. Posteriormente, se analiza el escenario de evaluación, donde se describen los elementos más relevantes que van a formar parte del proceso de validación de la metodología. A continuación, se muestra la validación de los diferentes módulos de la metodología. En primer lugar se evalúa el buscador semántico, después el extractor de términos y, por último, se evalúa el módulo de evolución de ontologías desde dos perspectivas diferentes. Por un lado, se evalúa el número de conceptos insertados en los procesos de evolución y, por otro lado, el número de relaciones taxonómicas creadas durante los procesos de evolución de ontologías. Para concluir este apartado, se presenta una conclusión global de los resultados obtenidos en las diferentes evaluaciones realizadas.

Seguidamente, en el epígrafe III.4 se evalúa la metodología de anotación en el dominio de la I+D+i, más concretamente, desde el punto de vista de las TIC. Al igual que la evaluación anterior, en primer lugar se realiza una breve introducción acerca de la importancia de las I+D+i en la actualidad, sobre todo en ambientes empresariales. Después, se describe el escenario de evaluación en el que se reutilizan algunas descripciones y recursos del proceso de evaluación anterior. A continuación, se evalúa el buscador semántico a partir de los resultados obtenidos durante los experimentos. Posteriormente, se evalúa el extractor de términos y, por último, el módulo de evolución de ontologías utilizando el método de conceptos definidos y relaciones taxonómicas creadas durante un proceso de evolución. A modo de conclusión se presenta un análisis de los resultados obtenidos durante todo el proceso de validación.

Por último, en la sección III.5 se realiza un análisis comparativo entre los resultados obtenidos en los dos dominios contemplados, y se discuten las principales conclusiones extraídas del proceso de validación del sistema de anotación semántica propuesto en esta tesis doctoral. Este análisis presentado,

recoge una comparación módulo a módulo de los resultados obtenidos en las evaluaciones realizadas sobre los dos dominios seleccionados. Al final de cada comparación se presenta una breve conclusión donde se discuten los resultados obtenidos a partir de cada validación realizada.

Capítulo IV. APLICACIÓN DE ANOTACIÓN SEMÁNTICA PARA EL CÁLCULO DE SIMILITUD

IV.1. INTRODUCCIÓN

En este capítulo se describe un modelo de cálculo de la similitud entre dos entidades. Este modelo utiliza la metodología de anotación semántica descrita en el Capítulo II como base para el cálculo de similitud. La utilización de la anotación semántica junto con tecnologías de representación del conocimiento ha facilitado la definición de una metodología capaz de calcular la similitud entre cualquier par de entidades, independientemente del tipo de entidad e, incluso, de la fuente de información.

El diseño de esta metodología para el cálculo de la similitud se inició a partir de un proyecto de investigación en el dominio de la gestión de la I+D+i en el que se requería la implementación de un motor de inferencia semántico de recursos relacionados con el dominio de la I+D+i. Para este caso, la función del motor no se restringía a la búsqueda de recursos a partir de palabras clave, sino que tenía que incorporar, además, funciones que facilitaran la comparación entre los recursos almacenados en el sistema de información.

El diseño de la metodología para el cálculo de la similitud se inició con la creación de perfiles semánticos que definieran los diferentes atributos de cada una de las entidades almacenadas en el sistema de información. Por entidad, se entiende a una "cosa" u "objeto" del mundo real con existencia independiente que puede ser identificado de forma unívoca (Beynon-Davies, 2000). En esta aplicación de anotación semántica para cálculo de similitud y, más concretamente, en el dominio de la I+D+i, por entidad se entiende a cualquier objeto con existencia física (entidad concreta) como una casa, persona, animal o un objeto; o con existencia conceptual (entidad abstracta) como tecnologías, proyectos o un concepto. Concretamente, en el dominio I+D+i una entidad representará un objeto que define una serie de atributos relacionados con la I+D+i y que se encuentra almacenado en el sistema de información. Por ejemplo, Tecnología u Organización

constituirían ejemplos de entidades abstractas. A partir de estas entidades se definieron los perfiles semánticos que, básicamente, consistían en representaciones semánticas de cada entidad. Es decir, cada entidad fue modelada mediante la utilización de los diferentes elementos que dispone una ontología para representar la información de la misma. A partir de estos perfiles semánticos se consiguió disponer de un modelo de información homogéneo, sobre el cual se definió la metodología que permitía el cálculo de matrices de similitud. Las matrices de similitud reflejan cuantitativamente el grado de similitud entre las diferentes entidades almacenadas en el repositorio de información.

El sistema recibe como entrada un fichero de configuración, donde se definen los perfiles semánticos de las entidades requeridas por el sistema de gestión de I+D+i. Sobre la base de estas descripciones, se generan tantas matrices de similitud como perfiles contenga el fichero de configuración. Entonces, cualquier operación de inserción, actualización y eliminación sobre los atributos definidos en los perfiles semánticos provoca el recálculo de los valores afectados sobre el o los atributos modificados. Por último, el motor de inferencia semántico hará uso de estas matrices durante los procesos de búsqueda y comparación.

IV.2. DESCRIPCIÓN DEL PROBLEMA Y OBJETIVOS

El objetivo de este capítulo es la definición de un método, para el cálculo de la similitud entre perfiles semánticos modelados en RDF a partir de una ontología. La metodología diseñada supone la aplicación del método de anotación descrito en el Capítulo II sobre un conjunto específico de entidades almacenadas en un sistema de gestión I+D+i. Por lo tanto, en el desarrollo de esta metodología se persiguen algunos de los objetivos ya planteados en dicho capítulo. No obstante, a continuación se enumeran los objetivos específicos que se pretenden con este sistema:

- Desarrollo de un método que facilite la definición de perfiles semánticos, a partir de propiedades definidas en un modelo ontológico.
- Desarrollo de un método para el cálculo de la similitud entre perfiles semánticos con independencia del dominio de aplicación.

- Desarrollo de un método de recuperación de perfiles semánticos, que proporcione la posibilidad de buscar entidades en el sistema de información a partir de palabras clave y la posibilidad comparar perfiles semánticos.

Además, se han tenido en cuenta una serie de objetivos secundarios con los que se pretende que el sistema sea fácilmente extensible, integrable e independiente del dominio de aplicación.

- **Integrable e extensible**. Facilitar la integración del sistema en cualquier repositorio de información es un aspecto que ha sido tenido en cuenta durante todo el proceso de desarrollo del sistema. Además, la estrategia de desarrollo basado en módulos ha permitido la utilización de interfaces que permiten extender fácilmente las funciones del sistema.
- **Eficiencia y flexibilidad**. Conseguir un entorno operativo eficiente es otro de los aspectos que se han tenido presentes en el desarrollo del sistema. Además, gracias a la utilización de un modelo de software durante todas las fases de desarrollo del sistema se permite el intercambio de elementos software sin que la eficiencia ni el rendimiento del sistema se vean afectados.
- **Adaptabilidad y parametrización**. El funcionamiento del sistema requiere de ficheros de configuración, que permiten definir diferentes aspectos relacionados con los perfiles semánticos y otros aspectos del sistema. Esta parametrización permite, además, la fácil adaptación del sistema a cualquier otro dominio diferente.

IV.3. ARQUITECTURA DEL SISTEMA DE CÁLCULO DE SIMILITUD

El sistema de cálculo de la similitud entre perfiles semánticos que presentamos en este capítulo está basado en el trabajo (García-Moreno et al., 2013) y se compone de cinco módulos funcionales. La Figura IV.1 muestra de forma esquemática la arquitectura del sistema.

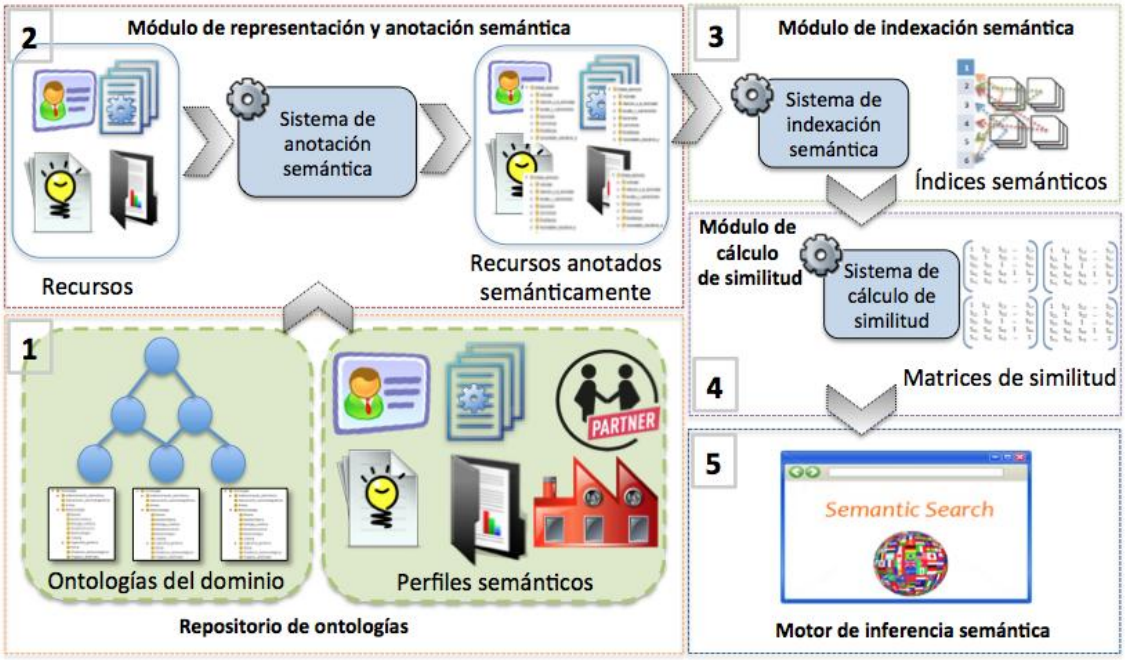


Figura IV.1 Arquitectura del sistema de comparación semántico

El módulo de representación y anotación semántica (2) se sostiene sobre tres tipos de ontologías que se encuentran almacenadas en el repositorio de ontologías (1): (i) la ontología del dominio, que se utiliza como fuente de etiquetas para realizar la anotación semántica; (ii) la ontología de las entidades, que se utiliza para modelar los atributos de las entidades a comparar; y (iii) el conjunto de ontologías que se utiliza para representar semánticamente la información y conocimiento de las entidades del sistema en formato RDF. El módulo de representación y anotación semántica (2) recibe como entrada información sobre entidades definidas en el sistema de información y construye perfiles semánticos para cada entidad basándose en diferentes modelos ontológicos predefinidos y almacenados en el repositorio de ontologías. Posteriormente, las anotaciones semánticas se recuperan por parte del módulo de indexación semántica (3) para construir los índices semánticos. La construcción de índices semánticos se basa en la misma metodología que la explicada en el II.3.2 y su objetivo es enriquecer las anotaciones semánticas definidas con los conceptos que se encuentren relacionados taxonómicamente con los que han sido anotados. Después de construir los índices semánticos, el módulo de cálculo de similitud (4), a través de una de las nuevas funciones incorporadas al sistema, construye las matrices de similitud utilizando como fuente de información los perfiles semánticos generados

previamente. Para la construcción de estas matrices de similitud, el módulo toma como partida un fichero de configuración que define la metodología de cálculo de similitud entre cada par de entidades a comparar. Las matrices almacenan valores reales que representan los grados de similitud entre cada par de entidades. Por último, el motor de inferencia semántico (5) utilizará esta información para ofrecer diferentes opciones de búsqueda, incluyendo búsquedas basadas en palabras clave y la comparación de perfiles semánticos.

A continuación se describe en detalle cada uno de estos módulos.

IV.3.1. REPOSITORIO DE ONTOLOGÍAS (1)

El principal objetivo de este módulo es almacenar todos los modelos ontológicos que son utilizados por el sistema. El repositorio almacenará tres tipos de ontologías: (i) la ontología que representa la estructura de las entidades a comparar, (ii) ontologías del dominio que se utilizan para la anotación semántica, y (iii) los perfiles semánticos basados en RDF de las entidades a comparar (se trata de instancias de la primera ontología pero que se almacenan por separado).

IV.3.1.1. Ontología que representa la estructura de las entidades a comparar

Esta ontología describe las entidades que van a ser comparadas en el sistema y definen su estructura en forma de clases, atributos y relaciones. Como se ha comentado con anterioridad, el dominio donde se ha desarrollado esta aplicación fue el de la I+D+i y, en consecuencia, esta ontología contiene la descripción de ideas, proyectos, propuestas, tecnologías, y recursos humanos, entre otros.

IV.3.1.2. Ontologías del dominio para el sistema de anotación semántica

Estas ontologías describen el vocabulario específico en uno o más dominios relevantes donde la plataforma va a ser utilizada. Concretamente, la aplicación en la que se utilizó este sistema fue en el dominio de las TIC. El modelo de representación de conocimiento de esta ontología ya se describió en el Capítulo II, capítulo dedicado a la descripción de la metodología de anotación semántica. Para

el desarrollo de esta ontología se utilizó Wikipedia como fuente de información. La utilización de este repositorio de información solucionó uno de los grandes problemas que se produce cuando se modela un dominio, la estructuración de la información. Por lo tanto, para resolver este problema se dispuso de esta fuente a modo de organizador de la información en el dominio de las TIC.

IV.3.1.3. Perfiles semánticos

Los perfiles semánticos describen las instancias de las entidades modeladas en la ontología descrita en el primer apartado. Estos perfiles describen semánticamente los principales recursos del sistema de información y se emplean para su comparación. Más concretamente, para la aplicación de esta metodología en la gestión de la I+D+i, se distinguen dos tipos principales de perfiles que representan (i) las propuestas, proyectos o ideas innovadoras, y (ii) los recursos humanos como, por ejemplo, profesionales, trabajadores o directivos, es decir, todas las personas que se encuentran involucradas en la compañía.

Las descripciones semánticas de proyectos han sido realizadas utilizando como base el vocabulario definido por DOAP (del inglés "*Description Of A Project*") (Dumbill, 2012). DOAP es un esquema RDF y un vocabulario XML para describir proyectos de software, concretamente, proyectos de software libre. Este modelo de descripción semántica fue diseñado para facilitar la transmisión de información semántica asociada a los proyectos de software de código abierto. El problema de DOAP es que está muy centrado en describir este tipo de software presentando algunas carencias para la representación de proyectos de I+D+i. Un ejemplo claro de esta carencia es la escasez de vocabulario para representar los roles de los distintos participantes que forman parte en los proyectos de investigación. Esta carencia, junto con otras, han sido las causas por las que se ha definido un nuevo esquema RDF, extendiendo DOAP, con el que se ha enriquecido el vocabulario para permitir la descripción semántica de todas las características de los proyectos de I+D+i. La Figura IV.2 muestra, a modo de ejemplo, la descripción semántica de un proyecto de I+D+i.

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
        xmlns:foaf="http://xmlns.com/foaf/0.1/"
        xmlns:doap="http://usefulinc.com/ns/doap#"
        xmlns:doap_ext="http://www.innovation-labs.com/ns/doap_ext#">
        <doap:Project>
          <doap:name>OPEN IDEA</doap:name>
          <doap:shortdesc xml:lang="es">
El proyecto Open Idea Este proyecto pretende incluir tecnologías propias del
conocimiento, como son el desarrollo de ontologías de dominio, esto es,
esquemas conceptuales para la representación del conocimiento en un dominio
concreto (automoción, turismo, bancario, etc.) y la relación entre los conceptos
de esos dominios, o la inferencia a partir de estas ontologías, de modo que se
pueda realizar un razonamiento sobre dichas ontologías; persigue el objetivo de
automatizar o simplificar diferentes tareas en el proceso de gestión de la ideas y
su ciclo de vida.
  </doap:shortdesc>
        <doap_ext:project_participant>
              <foaf:Organization>
                    <foaf:name>Universidad de Murcia</foaf:name>
              </foaf:Organization>
              <foaf:Organization>
                    <foaf:name>Universidad Carlos III de
Madrid</foaf:name>
              </foaf:Organization>
        </doap_ext:project_participant>
        <doap_ext:project_manager>
              <foaf:Person>
                    <foaf:name>Miguel Ángel Rodríguez García</foaf:name>
              </foaf:Person>
    </doap_ext:project_manager>
        </doap:Project>
  </rdf:RDF>
```

Figura IV.2 Ejemplo de descripción de un proyecto a través de doap

En la Figura IV.2 todas las etiquetas que tienen como prefijo el término "`doap_ext`" hacen referencia al vocabulario definido como extensión del proporcionado por DOAP. Entre las nuevas etiquetas es posible destacar "`project_coordinator`", "`project_participant`" y "`project_manager`", definidas para describir semánticamente información acerca de los participantes, el coordinador y los diferentes roles que existen en los proyectos de investigación, desarrollo e innovación.

Por otro lado, se crearon perfiles para representar los roles y competencias de los participantes en un proyecto de I+D+i, esto es, los recursos humanos. Este

vocabulario representa las funciones desempeñadas por el personal dentro de la empresa, junto con otra información semántica. Para ello se han empleado FOAF (del inglés, "*Friend-of-a-Friend*") (Brickley & Miller, 2012), ResumeRDF (Bojārs & Breslin, 2007) y perfiles semánticos descritos utilizando la extensión del vocabulario proporcionado por DOAP que se describió anteriormente. FOAF es un vocabulario descriptivo que utiliza RDF y OWL para describir personas, actividades y relaciones con otras personas y objetos. ResumeRDF es una ontología desarrollada para representar la información contenida en un currículum vitae en la Web Semántica. Esta ontología permite describir la experiencia laboral y académica, habilidades, publicaciones, certificaciones y otra información relevante relacionada con el dominio académico.

Finalmente, los perfiles semánticos representan la experiencia de los usuarios en temas relacionados con el dominio de la I+D+i. El perfil semántico incluirá información relacionada con: i) proyectos de desarrollo en los que ha estado involucrado, ii) experiencia laboral en otras compañías, y iii) información personal. En la Figura IV.3 se muestra, a modo de ejemplo, la descripción semántica de un trabajador.

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
   xmlns:foaf="http://xmlns.com/foaf/0.1/"
   xmlns:doap="http://usefulinc.com/ns/doap#"
   xmlns:doap_ext="http://www.innovation-labs.com/ns/doap_ext#">
   xmlns:cv=" http://rdfs.org/resume-rdf/#">
  <cv:CV>
   <cv:aboutPerson>
    <foaf:Person>
     <foaf:name>Miguel Ángel Rodríguez García</foaf:name>
    </foaf:Person>
   </cv:aboutPerson>
   <cv:hasEducation>
    <cv:Education>
     <cv:studiedIn>Universidad de Murcia</cv:studiedIn>
     <cv:degreeType>Computer Science MSC</cv:degreeType>
    </cv:Education>
   </cv:hasEducation>
   <cv:hasWorkHistory>
    <cv:WorkHistory>
     <cv:employedIn>TECNOMOD</cv:employedIn>
     <cv:jobTitle>Researcher</cv:jobTitle>
     <cv:isCurrent/>
    </cv:WorkHistory>
   </cv:hasWorkHistory>
   <doap_ext:workedIn>
     <doap:Project>
      <doap:name>OPEN IDEA</doap:name>
     <doap:Project>
   </doap_ext:workedIn>
  </cv:CV>
</rdf:RDF>
```

Figura IV.3 Ejemplo de una descripción semántica de un trabajador

En esta descripción semántica (véase Figura IV.3) se pueden identificar varios prefijos en las etiquetas utilizadas para describir al trabajador. En particular, el prefijo "`cv`" hace referencia al vocabulario utilizado de ResumeRDF para expresar información del trabajador relacionado con su Curriculum Vitae, el prefijo "`foaf`" hace referencia al vocabulario de FOAF para expresar información relacionada con las actividades del trabajador y el prefijo "`doap_ext`" hace referencia al vocabulario de DOAP, utilizado para la descripción de proyectos en los que el trabajador haya intervenido.

IV.3.2. MÓDULO DE REPRESENTACIÓN Y ANOTACIÓN SEMÁNTICA (2)

En términos generales, la labor que desempeña este módulo dentro de este escenario es la misma que el módulo explicado en el Capítulo II. Particularmente, en este caso el módulo extiende las funciones de representación y anotación semántica para que no sólo permita crear las anotaciones semánticas sino que, además, pueda describir las distintas entidades del sistema a través del modelo ontológico predefinido. A partir de este modelo se definen los perfiles semánticos que se almacenarán en el repositorio de información semántica. El objetivo principal de este módulo es traducir la información a RDF para homogeneizar y facilitar su manipulación. La Figura IV.4 muestra gráficamente la nueva función incorporada dentro del módulo de representación y anotación semántica.

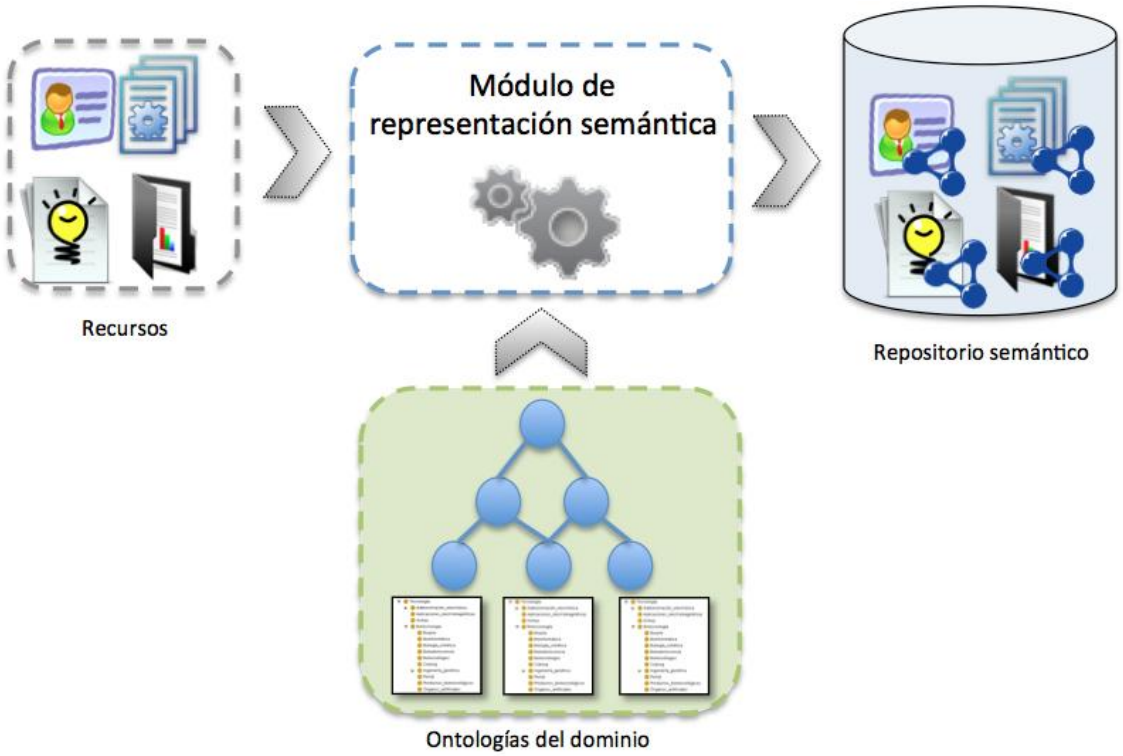


Figura IV.4 Funcionamiento del módulo de representación semántica

El módulo recibe como entrada recursos de información relacionados con las entidades del sistema de gestión de I+D+i (véase Figura IV.4). Para cada tipo de recurso, el módulo de representación semántica, a partir del modelo ontológico predefinido, extrae la información y la traduce a tripletas RDF, creando los perfiles

que se insertarán en el repositorio semántico. Además de realizar esta transformación, el módulo de representación semántica también se encarga de anotar semánticamente cada una de las descripciones textuales existentes de cada perfil semántico. De este modo, la traducción de información a tripletas RDF y anotaciones semánticas normaliza la información, lo que facilita la labor desempeñada por el módulo de cálculo de similitud semántica.

IV.3.3. MÓDULO DE INDEXACIÓN SEMÁNTICA (3)

Como se analizó en el apartado II.3.2 del Capítulo II, la finalidad de este sistema es enriquecer las anotaciones semánticas creadas por el módulo de representación y anotación semántica utilizando la fórmula del TF-IDF extendida. El resultado de este módulo es un vector de orden igual al número de conceptos definidos en el modelo ontológico que contiene una representación semántica de cada documento anotado. En particular, cada concepto del modelo ontológico dispone de una dimensión dentro del vector que contiene un índice de relevancia calculado a partir de la fórmula TF-IDF. Este índice expresa la relevancia de ese concepto para un documento dentro de una colección.

IV.3.4. MÓDULO DE CÁLCULO DE SIMILITUD (4)

El módulo de cálculo de similitud proporciona una herramienta de comparación entre perfiles semánticos. La labor del sistema es la administración de un conjunto de matrices que expresan la similitud entre pares de perfiles semánticos. Es decir, dentro del sistema de cálculo de similitud cada matriz representa la comparación entre dos tipos de perfiles semánticos. Tanto la comparación como los tipos de perfiles y la metodología de comparación a seguir se establecen a partir de un fichero de configuración.

El procedimiento de cálculo de similitud es un proceso en continua actualización. Cualquier proceso que implique la inserción, eliminación o actualización de información provoca la actualización de aquellas matrices de similitud que estén relacionadas con la entidad modificada. La Figura IV.5 representa gráficamente la

actualización continua a la que las matrices de similitud están sometidas. En el centro del flujo de información, entre la plataforma de gestión de información y el repositorio de información semántico, se encuentran ubicadas las matrices de similitud. Por lo tanto, la actualización de cualquier elemento de información dentro del repositorio de información semántico también afectará a los valores almacenados en las matrices de similitud.

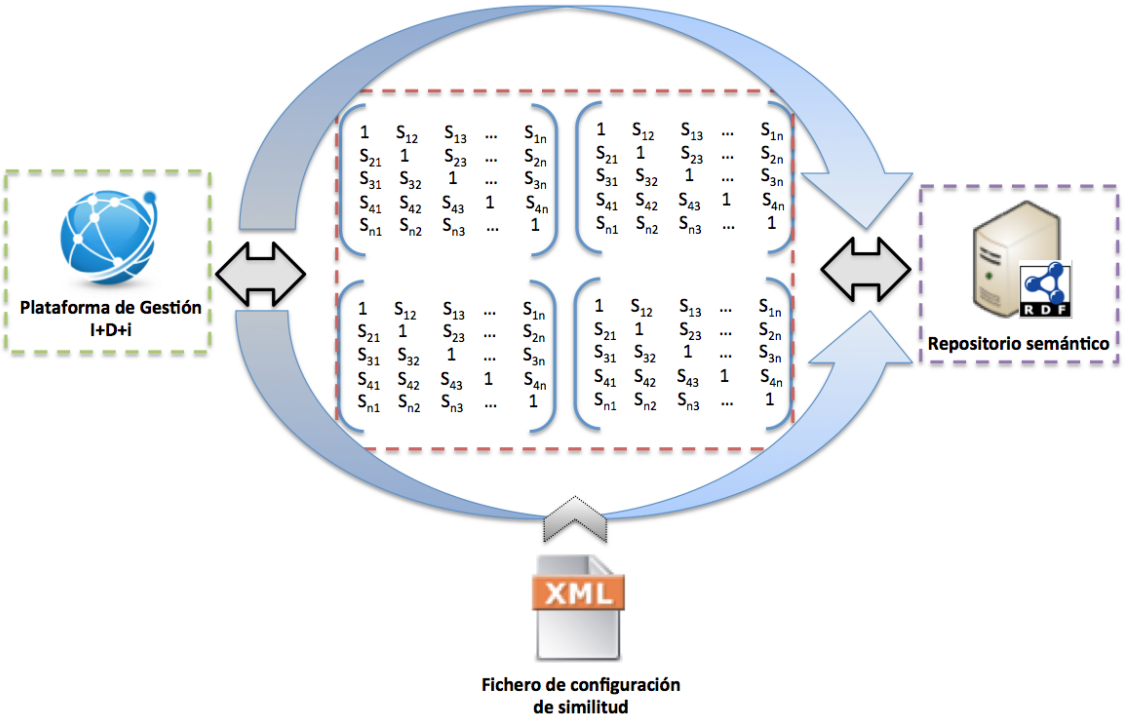


Figura IV.5. Arquitectura del módulo de cálculo de similitud

La Figura IV.5 muestra una representación del sistema de cálculo de similitud como una capa intermedia entre la plataforma de gestión de información y el repositorio semántico. La plataforma de gestión proporciona las interfaces que el usuario utilizará para insertar, modificar o eliminar entidades dentro del sistema de gestión de información. El repositorio semántico, por su parte, será el encargado de almacenar toda esta información en formato de tripletas RDF. Entre uno y otro, el sistema de cálculo de similitud tiene la responsabilidad de detectar cualquier modificación de información y actualizar las matrices de similitud.

A continuación, se describe la metodología empleada para el cálculo de la similitud por medio de la cual se generan los valores con los que se rellenan las

matrices. Esta metodología define, en función del tipo de dato del atributo, algoritmos matemáticos alternativos que se aplican sobre los atributos de las entidades. Como resultado de la aplicación de cada uno de estos algoritmos, se obtiene un valor que refleja el grado de similitud para cada par de entidades comparadas.

IV.3.4.1.1. Metodología de cálculo de similitud

La metodología de cálculo de similitud se basa en la creación de matrices, que facilitan la comparación binaria ente dos entidades almacenadas en el sistema. Es decir, cada matriz estará compuesta por un conjunto de números, que expresará el grado de similitud entre dos entidades cualesquiera. El cálculo de estos valores de similitud se lleva a cabo a partir de unos algoritmos que permiten realizar comparaciones entre instancias de la misma entidad o, incluso, entre instancias de entidades de distinto tipo. La metodología para el cálculo de la similitud toma como entrada un fichero descrito en lenguaje de marcas XML que permite configurar qué par de entidades van a ser comparadas y cómo se van a comparar. El método de comparación que se define en este fichero de configuración es de granularidad muy fina facilitando las comparaciones a nivel de atributo de concepto en la ontología. El único requisito que impone este mecanismo de comparación es que los atributos a comparar sean del mismo tipo de datos.

Por ejemplo, supongamos un escenario en el que un sistema de gestión de I+D+i modela la entidad “proyectos” y la entidad “ideas de innovación” que tienen definidos los atributos de tipo cadena de texto “nombre del proyecto” y “nombre de la idea”, respectivamente. En este contexto, el método de comparación que se establecería para estas entidades es la comparación a través de los nombres. Por otro lado, los valores con los que se rellenarían las matrices de similitud serían aquellos que se obtienen de calcular la similitud entre ambas cadenas. De esta forma, el método permite establecer comparaciones entre entidades que pertenezcan al mismo perfil semántico o entidades que pertenezcan a distintos perfiles. Para cada comparación definida en el fichero de configuración, el sistema de cálculo de similitud creará una matriz de similitud independiente. Por lo tanto,

existirán tantas matrices de similitud como definiciones recoja este fichero de configuración.

En resumen, se podría decir que el sistema de cálculo de similitud es un administrador de matrices, donde las matrices representan contenedores de información que almacenan conjuntos de valores que definen la similitud entre instancias de entidades del mismo o diferente tipo. La única restricción que debe cumplirse en cada comparación es que los atributos de cada entidad a comparar pertenezcan al mismo tipo de datos.

A continuación, se detallan cada uno de los algoritmos que hacen posible el cálculo de similitud entre instancias.

IV.3.4.1.2. Algoritmos de similitud

El cálculo de la similitud entre dos entidades se define a través de una fórmula general, que establece cómo se construye la matriz y cómo se obtienen los valores de similitud cuando dos entidades se comparan. La fórmula (IV.1) ejemplifica una matriz de similitud entre proyectos de I+D+i.

$$\text{similitud proyectos} = \begin{bmatrix} 1 & SP_{12} & \cdots & SP_{1n} \\ SP_{21} & 1 & \cdots & SP_{2n} \\ \cdots & \ldots & 1 & \cdots \\ SP_{n1} & SP_{n2} & \cdots & 1 \end{bmatrix} \qquad \textbf{(IV.1)}$$

La matriz tiene definida una fila y columna por cada entidad que se compara. De esta forma, cualquier matriz de similitud será una matriz cuadrática, ya que tendrá el mismo número de filas que de columnas, además de ser una matriz simétrica, debido a que el triangular inferior es idéntico al triangular superior. En el caso de la matriz en la fórmula (IV.1), esta matriz de similitud simboliza la comparación entre proyectos de I+D+i. Por lo tanto, cada valor 'SP_{ij}' representa un valor de similitud entre el proyecto '`i`' y el proyecto '`j`'. El cálculo de este valor, como se ha indicado anteriormente, se rige por una fórmula general (véase fórmula (IV.2)) que especifica cómo se calcula la similitud entre dos entidades cualesquiera.

$$SP_{ij} = \sum \alpha_k * SimilitudPropiedad(a,b) \quad \textbf{(IV.2)}$$

En la fórmula (IV.2) 'α_k' es un valor entre `0` y `1`, y representa el peso que se asigna a cada algoritmo de similitud para establecer cómo de relevante es ese valor de similitud calculado con respecto a los demás valores. La función '`SimilitudPropiedad(a,b)`' representa el cálculo de similitud entre cada par de propiedades entre ambas entidades. De este modo, '`a`' y '`b`' representan dos atributos de las entidades a comparar. Por último, el sumatorio de los resultados obtenidos al aplicar los algoritmos de similitud y multiplicar los resultados por sus correspondientes pesos, proporciona el valor 'SP_{ij}' que indica el grado de similitud entre ambos proyectos.

A continuación, se describen los diferentes algoritmos de similitud que han sido implementados para comparar atributos de los distintos tipos de datos.

IV.3.4.1.2.1 Similitud entre atributos de tipo cadena de texto

Este cálculo de similitud se utiliza cuando se comparan dos atributos de tipo cadena de caracteres. El algoritmo implementado para calcular el grado de similitud entre dos cadenas de texto ha sido el algoritmo de distancia propuesto por Levenshtein (1966). Este algoritmo permite obtener la distancia de edición o distancia entre palabras, entendiendo por distancia al número mínimo de operaciones requeridas para transformar una cadena de caracteres en otra. Para normalizar a `1` el resultado, se ha diseñado la siguiente fórmula matemática (ver fórmula (IV.3)), donde la '`cadena1`' y '`cadena2`' hacen referencia a las cadenas sobre las que quiere conocerse el grado de similitud; '`distanciaLevenshtein`' referencia la función que define el algoritmo de Levenshtein que será utilizado para comparar ambas cadenas; y, por último, '`máximaLongitud`' referencia la función que obtiene la longitud máxima de '`cadena1`' y '`cadena2`'.

$$\text{similitud texto} = 1 - \frac{distanciaLevenshtein\ (cadena1, cadena2)}{máximaLongitud(cadena1, cadena2)} \quad \textbf{(IV.3)}$$

Por ejemplo, si partimos de las palabras "`casa`" y "`calle`", la distancia de Levenshtein entre ellas sería de `3` debido a que se necesitan al menos tres ediciones para transformar una palabra en la otra. Ambas palabras poseen la misma longitud por lo que en este caso la longitud sería de `4`. Por lo tanto, la fórmula (IV.3) quedaría de la siguiente forma: `1 - (3/4)` obteniendo el valor de `0.25`.

IV.3.4.1.2.2 Similitud entre atributos con valores numéricos

Este mecanismo de similitud se utiliza para comparar dos propiedades numéricas. En este caso, el algoritmo empleado es la media aritmética normalizada para obtener un valor comprendido entre `0` y `1`. Para ello, se ha diseñado la fórmula (IV.4), donde '`valor1`' y '`valor2`' referencian los valores sobre los que se quiere obtener el valor de similitud, y '`máxima`' la función que se encarga de obtener el valor máximo entre '`valor1`' y '`valor2`'.

$$\text{similitud numérica} = \frac{valor1 + valor2}{2 * (máxima(valor1, valor2))} \quad \textbf{(IV.4)}$$

Por ejemplo, supongamos que partimos de dos atributos numéricos de dos entidades cualesquiera con valores '`1000 €`' y '`1500 €`', respectivamente, representando el saldo de dos empleados de una organización. El resultado de aplicar la formula (IV.4) sería tal que así: `(1000+1500)/2*1500`. Este cálculo resultaría en un `0.83` de grado de similitud entre ambos valores.

IV.3.4.1.2.3 Similitud entre relaciones

Normalmente, dentro de un sistema de información las entidades que lo constituyen no se encuentran totalmente aisladas sino que existe cierta dependencia que favorece la aparición de relaciones entre ellas. Al igual que en los modelos entidad-relación, las relaciones en ontologías OWL vienen asociadas a un dominio y un rango. Para los propósitos de esta medida de similitud, se utilizan las instancias presentes en el rango de la relación. Así, la similitud se calcula obteniendo la intersección de ambos conjuntos rango, dividiendo por aquel que tenga una mayor longitud con el objetivo de normalizar el valor obtenido entre `0` y `1`. Para el cálculo de esta similitud se ha diseñado la fórmula (IV.5), donde '`rango1`' y '`rango2`' referencian los conjuntos de relaciones de los que se quiere obtener el grado de similitud; '`intersección`' hace referencia a la función que obtiene el conjunto intersección creado a partir de ambos conjuntos; y, por último, '`máx`' referencia a la función que obtiene el módulo máximo entre '`rango1`' y '`rango2`'.

$$\text{similitud relación} = \frac{intersección(rango1, rango2)}{máx(|rango1|, |rango2|)} \qquad \textbf{(IV.5)}$$

Por ejemplo, supongamos que en nuestro sistema de gestión de I+D+i se tienen las entidades "Organización", "Proyecto" y "Tecnología", y que existe, además, una relación entre "Organización" y "Tecnología" y otra entre "Proyecto" y "Tecnología" representando que una organización puede ser experta en una o varias tecnologías y un proyecto puede, a su vez, hacer uso de una o varias tecnologías. Entonces, si una organización "`O`" es experta en las tecnologías "`Web Semántica`" y "`Ontologías`" y un proyecto "`P`" hace uso de las tecnologías "`Web Semántica`", "`Ontologías`" y "`Procesamiento del Lenguaje Natural`", entonces el valor obtenido por el cálculo de similitud a partir de la fórmula (IV.5) será de `0.67`. En efecto, la intersección del grupo de tecnologías en ambas entidades es de `2` y la máxima longitud del grupo es de `3`, por lo que la formula descrita queda

como sigue: `2/3`. Este valor, `0.67`, representa la similitud entre las relaciones de las entidades "`O`" y "`P`".

IV.3.4.1.2.4 Similitud entre los índices semánticos

Los índices semánticos se crean a partir de la anotación semántica de la información textual de cada perfil semántico almacenado en el repositorio de ontologías. El cálculo de la similitud entre los índices semánticos de cada entidad se realiza a través de la función del coseno, utilizada en el motor de búsqueda semántico descrito en el Capítulo II y que se representa en la ecuación (II.5).

$$\cos\theta = \frac{V1 * V2}{||V1||\,||V2||} \qquad \textbf{(IV.6)}$$

IV.3.4.1.2.5 Configuración de la similitud semántica

El proceso de creación de las matrices de similitud requiere, como se ha comentado anteriormente, de una configuración previa donde se establezca qué entidades van a ser comparadas y cómo van a ser comparadas. Esta configuración se define a través de unas etiquetas en un fichero XML elaborado a partir de un DTD. Estas etiquetas proporcionan el vocabulario suficiente para describir el conjunto de propiedades a emplear para llevar a cabo el cálculo de similitud semántica. Entre este conjunto de propiedades se destacan las siguientes:

- **First_instance**. Esta etiqueta representa la primera instancia de la comparación.
- **Second_instance.** Esta etiqueta hace referencia a la segunda instancia de la comparación.
- **Tipos de datos**. Los tipos de datos sobre los que van a ser utilizados en el cálculo de similitud se clasifican en tres grandes grupos: (i) *datatype properties*, que son los atributos de los conceptos que pueden tomar valores simples o complejos; (ii) *object properties*, que representan las relaciones no taxonómicas existentes entre los conceptos de la ontología; y (iii) *annotations*, que

representan a las anotaciones semánticas que han sido obtenidas durante el proceso de anotación realizado por el módulo de representación y anotación semántica.

- **Property_name**. Etiqueta que referencia a la propiedad de la primera instancia que va a ser comparada.
- **Compare_with**. Etiqueta que hace referencia a la propiedad de la segunda instancia que se va a comparar y que deberá pertenecer al mismo tipo que la propiedad definida en la etiqueta 'property_name'. Con esta restricción el sistema se asegura que se comparan dos propiedades pertenecientes al mismo tipo (p.ej. cadenas de caracteres con cadenas de caracteres o propiedades numéricas con propiedades numéricas).
- **Weight**. Esta etiqueta permite la asignación de pesos de manera individual a cada comparación realizada. Los valores que puede tomar esta etiqueta van de 0 a 100. La utilización de pesos proporciona un método para resaltar la relevancia de una comparación frente a las demás comparaciones. De este modo, se puede contemplar el caso de que en un sistema de gestión, la comparación entre las anotaciones semánticas sea más relevante que, por ejemplo, la comparación numérica entre los sueldos de los empleados para el cálculo de las matrices de similitud. En este caso, se pueden asignar pesos diferentes a las comparaciones para unas que adquieran mayor notabilidad que las otras.
- **Comparison**. A través de esta etiqueta se especifica qué algoritmo de comparación se va a utilizar para comparar las propiedades definidas en las etiquetas 'property_name' y 'compare_with'. Por lo tanto, esta etiqueta permite desacoplar completamente el modelo de comparación a utilizar de las propiedades que se van a comparar. De esta forma, cualquier par de propiedades pueden ser comparadas utilizando cualquiera de los algoritmos de similitud implementados.

A modo de aclaración, y siguiendo con uno de los ejemplos descritos anteriormente, se presenta un extracto de un fichero de configuración de similitud (véase Figura IV.6). Este extracto define cómo se comparan las entidades "Organización" y "Proyecto" a través de los atributos "`nombre`" y

"organización". En este escenario, el atributo "nombre" representa el nombre de la organización y el atributo "organización" representa el nombre de la organización donde se ha llevado a cabo el proyecto. Además, también se utilizará en esta comparación los atributos "Tiene_tecnología" de sendas entidades, que representan las relaciones que éstas tienen con la entidad "Tecnología" dentro del repositorio semántico. Por tanto, la metodología de comparación definida en este escenario permite comparar las entidades "Organización" y "Proyecto" a través de los atributos "nombre" y "organización", que son del tipo *datatype*, y el atributo "Tiene tecnología", definido en ambas entidades, que es del tipo *object property*. Como se puede apreciar en la Figura IV.6, cada comparación se delimita por la etiqueta que hace alusión al tipo de propiedad comparada. En este ejemplo, los tipos *datatype* y *object property*.

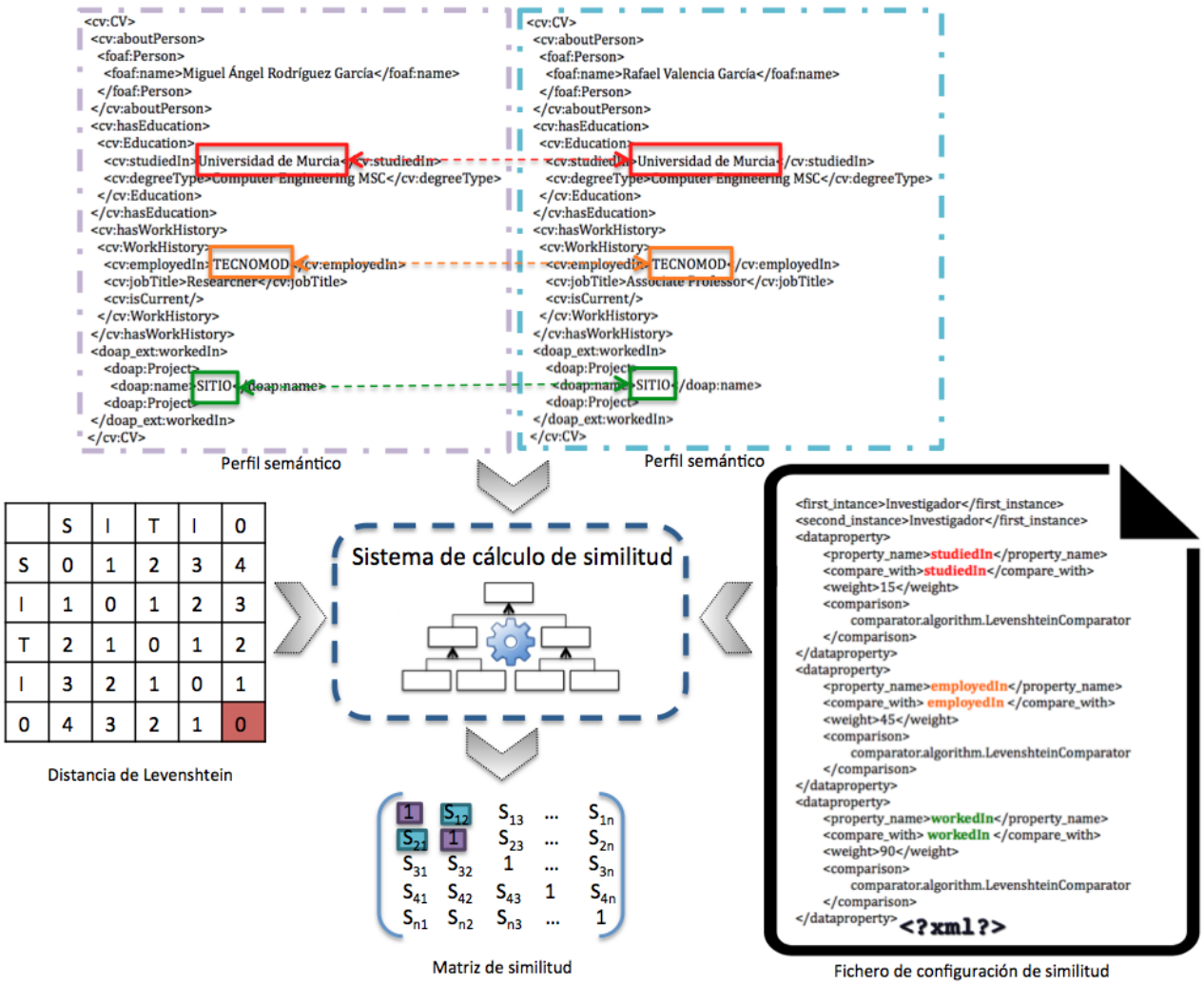
Por otro lado, analizando en profundidad el extracto de la Figura IV.6, el peso asignado al resultado de la comparación de los atributos de tipo *datatype property* es de 15, mientras que la ponderación para los atributos de tipo *object property* es de 30. Esta configuración revela que, en este escenario de aplicación, las similitudes calculadas a partir del atributo "Tiene_tecnología" son más relevantes que las similitudes obtenidas a partir de la comparación entre los atributos "nombre" y "organización". Además, el fichero de configuración especifica el algoritmo que se va a aplicar en cada tipo de comparación. En este caso, como se trata, por un lado, de propiedades del tipo cadena de texto, se utiliza el algoritmo que implementa la distancia de Levenshtein y, por otro lado, de propiedades del tipo relación, se utiliza el algoritmo de comparación de relaciones.

```
<first_intance>Organización</first_instance>
<second_instance>Proyecto</first_instance>
<dataproperty>
    <property_name>nombre</property_name>
    <compare_with>organización</compare_with>
    <weight>15</weight>
        <comparison>LevenshteinComparator</comparison>
</dataproperty>
<objectproperty>
    <property_name>Tiene_tecnología</property_name>
    <compare_with>Tiene_tecnología</compare_with>
    <weight>30</weight>
    <comparison>ObjectPropertyComparator</comparison>
</objectproperty>
```

Figura IV.6. Extracto de fichero de configuración de similitud

En otras palabras, el extracto que se muestra en la Figura IV.6 define un método de comparación entre la entidad "Organización" y la entidad "Proyecto" a través de las comparaciones de la propiedad "`nombre`" de la entidad "Organización" y la propiedad "`organización`" de la entidad "Proyecto", y también de la propiedad "`Tiene_tecnología`". Además, ambas comparaciones tienen asignados pesos distintos, en el caso de las comparaciones en tipos de datos *datatype property* la relevancia asignada es de `15` y en el caso de las *object property* la relevancia es de `30`. Por último, destacar la configuración del algoritmo de similitud que operará con ambos tipos de propiedades para obtener el valor conjunto que simbolice la similitud entre ambas entidades.

Para concluir este apartado, en la Figura IV.7 Figura IV.6 se muestra gráficamente el modo en que se construyen y calculan las matrices de similitud. El escenario que se propone como ejemplo parte de dos perfiles semánticos que expresan información profesional sobre dos trabajadores y un fichero de configuración de similitud. En esta figura también se puede observar una representación gráfica del algoritmo de similitud utilizado y la matriz de similitud generada.



	S	I	T	I	0
S	0	1	2	3	4
I	1	0	1	2	3
T	2	1	0	1	2
I	3	2	1	0	1
0	4	3	2	1	0

Figura IV.7 Ejemplo de la construcción de matrices de similitud

En el escenario representado por la Figura IV.7, el primer elemento que se puede destacar es el fichero de similitud semántica. Este fichero, como ya ha sido mencionado anteriormente, establece qué propiedades del perfil semántico serán utilizadas para aplicar los algoritmos de cálculo de similitud y obtener de esta forma un valor que represente el grado de similitud entre ambos perfiles. En este ejemplo, los atributos que se utilizan son: "`studedIn`", "`employedIn`", "`workedIn`". Todos estos atributos tienen una característica común y es que todos tienen el tipo de dato de cadenas de caracteres. Por lo tanto, el algoritmo de comparación que se utilizará será el de la distancia de Levenshtein que viene representado en la parte izquierda de la Figura IV.7. Entonces, el sistema de cálculo de similitud recibirá como entrada los perfiles y, después, extraerá estos atributos que aparecen indicados en el fichero de configuración de similitud. Por último, aplicará los algoritmos de similitud indicados y rellenará las matrices con los valores de similitud obtenidos. Para el ejemplo concreto presente en la Figura IV.7, se insertarían dos valores en la matriz de similitud simbolizando el grado de similitud entre estos dos perfiles. De ahí que los marcos de los perfiles semánticos estén coloreados con el mismo color con el que se resaltan los valores insertados en la matriz de similitud.

IV.3.5. MOTOR DE INFERENCIA SEMÁNTICA (5)

El motor de inferencia semántica es un módulo que se encarga de ofrecer servicios de búsqueda y recomendación a partir de la metainformación generada por los demás módulos. Es decir, el motor de inferencia se abastece de información almacenada en el repositorio semántico como perfiles, anotaciones semánticas, etc. y de las matrices de similitud creadas y gestionadas por otros módulos para ofrecer el servicio de búsqueda y recomendación semántica.

El servicio de búsqueda semántica se basa en el módulo "motor de búsqueda semántica" descrito en el apartado II.3.5 del Capítulo II, donde se utiliza una implementación del modelo de espacio vectorial, para representar los recursos anotados como índices semánticos, y que puedan compararse a través de la función coseno para obtener un valor de similitud que indique el grado de parentesco entre dos recursos. La diferencia en la aplicación de la metodología de

anotación en este caso es que los recursos anotados no se restringen a documentos, sino que se puede tratar de cualquier recurso relacionado con un sistema de información, desde un perfil de usuario al contenido de las filas de tablas en bases de datos relacionales.

El servicio de recomendación es otro de los servicios que proporciona el motor de inferencia semántica. Su funcionamiento se basa en las matrices de similitud creadas y cumplimentadas por el sistema de cálculo de similitud. La función de este servicio es, dada una descripción semántica, proporcionar todo tipo de información relacionada con esta descripción semántica. La Figura IV.8 muestra gráficamente las dos funciones que desempeña el motor de inferencia.

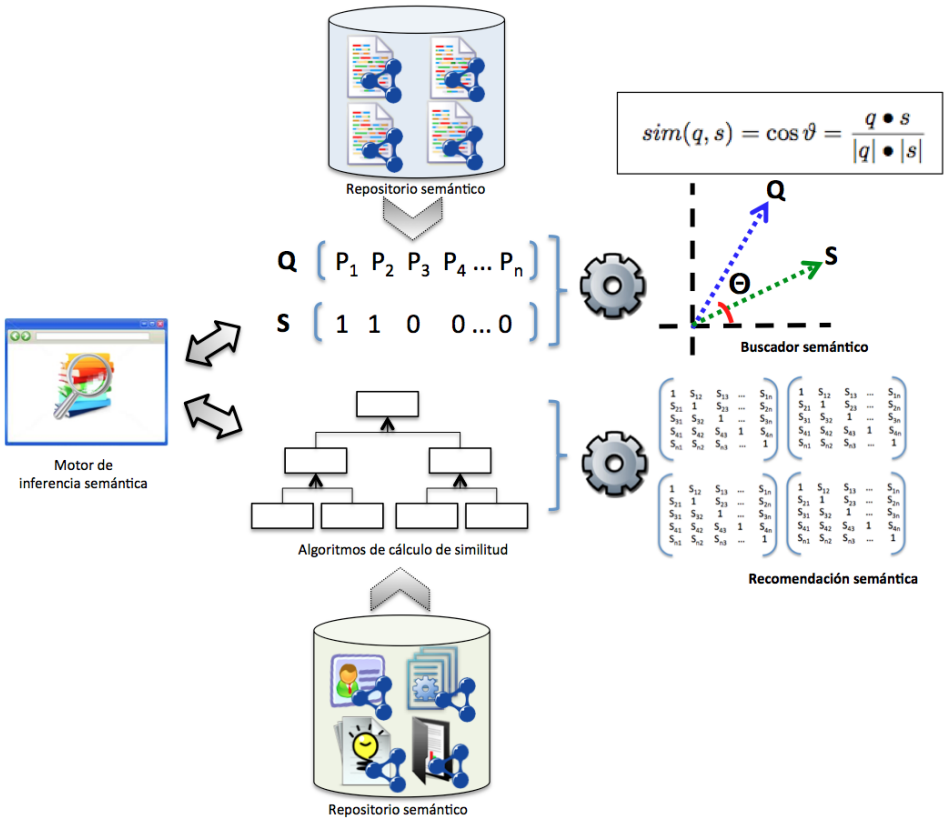


Figura IV.8 Arquitectura del motor de inferencia semántica

La función de búsqueda semántica ya fue descrita en el Capítulo II. En el siguiente ejemplo se explica el funcionamiento del servicio de recomendación. Como ya se mencionó anteriormente, este servicio hace uso de los valores de similitud almacenados en las matrices para proporcionar las recomendaciones. La Figura IV.9 incluye una representación gráfica de cómo funciona el servicio y qué

información de las matrices de similitud emplea para proporcionar recomendaciones.

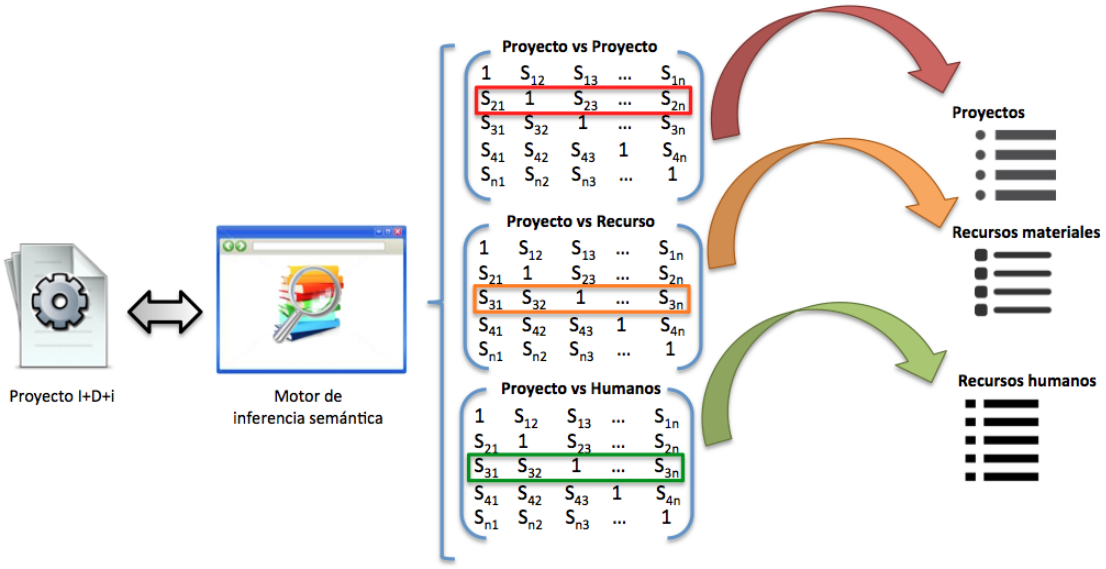


Figura IV.9 Representación gráfica del funcionamiento del servicio de recomendación

En el ejemplo de la Figura IV.9, el servicio de recomendación recibe como entrada un proyecto de I+D+i, aunque podría utilizarse cualquier entidad del sistema de gestión previamente descrita semánticamente y que perteneciera a algunos de los perfiles configurados en el fichero de configuración de similitud. A partir del identificador asociado en el repositorio semántico, se consultan todas las tablas de similitud y se extraen todos los valores de similitud que relacionan este perfil semántico con cualquier perfil existente en el repositorio. Como resultado, se obtiene la información de la lista de entidades relacionadas con el proyecto de I+D+i de entrada. La Figura IV.10 muestra una captura de pantalla de la interfaz Web de la herramienta con un ejemplo del sistema de cálculo de similitud descrito en este capítulo. En este ejemplo, las entidades comparadas son "Idea", "Organización" y "Objetivo", aunque sólo se muestra información relacionada con la entidad "Idea". La captura de pantalla presenta, en la parte superior, la idea seleccionada y, en la parte inferior, un listado de entidades similares a esta idea agrupado por tipo de entidad en pestañas. La lista de entidades mostrada en la figura representa el conjunto de ideas, dentro del repositorio semántico, similares a la idea proporcionada como entrada. En este caso concreto, la idea de partida es "SONAR2: SISTEMA DE RECOMENDACIÓN Y SOPORTE A DECISIONES

FINANCIERAS CORPORATIVO BASADO EN TECNOLOGÍAS SEMÁNTICAS" y la idea más similar resultado del proceso de cálculo de similitud es "SONAR. BUSCADOR FINANCIERO CORPORATIVO BASADO EN TECNOLOGÍAS SEMÁNTICAS". Además de ofrecer el nombre de la idea similar, el sistema ofrece una breve descripción en lenguaje natural de los atributos que se han utilizado en la comparación comentando, para cada uno de ellos, el porcentaje de similitud obtenido.

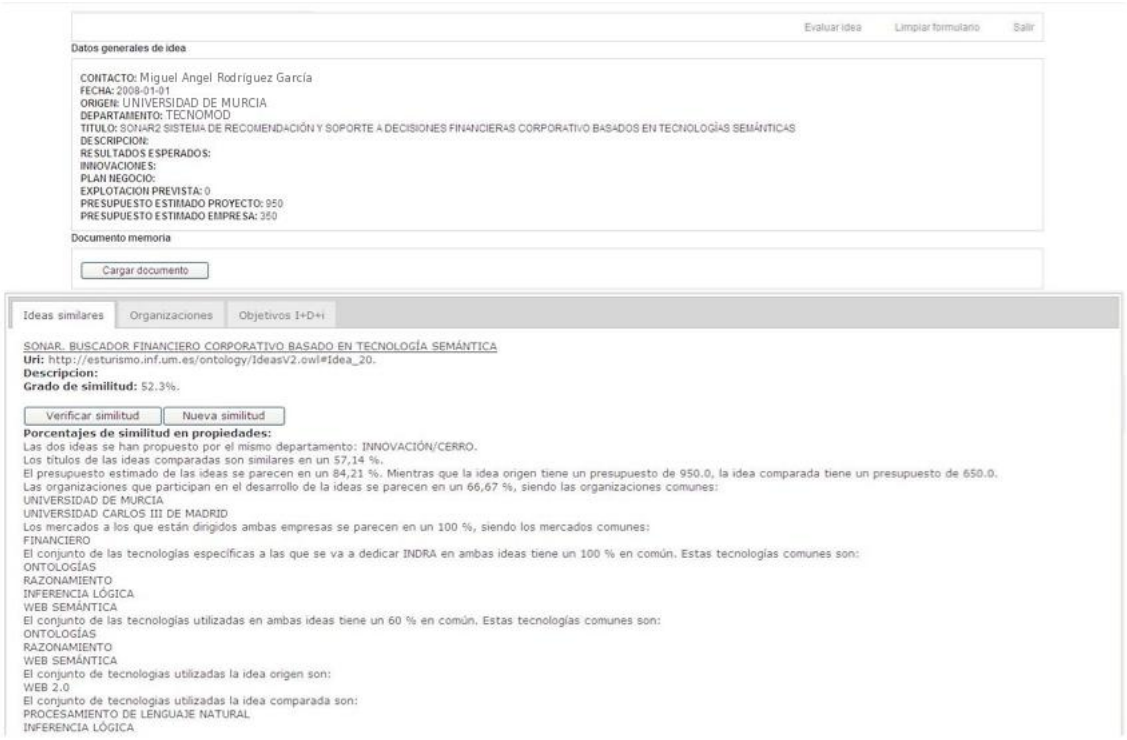


Figura IV.10 Captura de pantalla de la aplicación

IV.4. RESUMEN

En este capítulo se ha descrito la aplicación práctica de la metodología de anotación semántica presentada en el Capítulo II en el ámbito del cálculo de similitud entre instancias. Durante la primera sección del capítulo se realiza una breve introducción sobre la aplicabilidad de esta metodología, sobre todo en el dominio de la I+D+i, y cuáles fueron las motivaciones que hicieron emerger la posibilidad de desarrollar esta aplicación dentro de la metodología de anotación implementada.

En la siguiente sección, epígrafe IV.2, se describe el problema a solucionar, así como el conjunto de objetivos y subobjetivos que se establecieron como camino a seguir para dar solución al reto planteado.

Por último, en el epígrafe IV.3 se describe la arquitectura del sistema de comparación semántica y cada uno de los módulos y subsistemas que lo constituyen. En el primer apartado de esta sección se describe el repositorio de ontologías, es decir, la base de conocimiento que almacenará toda la parte semántica del sistema. En el siguiente apartado se describe brevemente el módulo de representación y anotación semántica, ya que se trata de un módulo que forma parte de la metodología de anotación descrita en el Capítulo II. El tercer apartado incluye la descripción del módulo de indexación semántica, módulo también tratado en el Capítulo II encargado de enriquecer las anotaciones semánticas utilizando las relaciones taxonómicas como fuente de información. A continuación, el siguiente apartado presenta en detalle el módulo de cálculo de similitud, que constituye la parte más importante del sistema y donde se definen todas las funciones y operaciones matemáticas que se llevan a cabo en el cálculo de la similitud. Por último, para terminar la descripción de este sistema, se presenta el motor de inferencia semántica que abarca los servicios de consulta y recomendación semántica. Finalmente, para concluir este último apartado se incluye un ejemplo gráfico de funcionamiento del módulo.

Capítulo V. VALIDACIÓN DE LA APLICACIÓN SEMÁNTICA PARA SIMILITUD

V.1. INTRODUCCIÓN

Este capítulo se centra en la validación de la aplicación que emplea la metodología para el cálculo de similitud presentada en el Capítulo IV. Esta validación se realizó utilizando las mismas métricas que se han descrito en el Capítulo III, a saber, "precisión", "exhaustividad" y "medida-F".

La validación de esta aplicación se ha llevado a cabo en el dominio de la gestión de la I+D+i, con el objetivo de demostrar la fácil integración y aplicabilidad de la metodología de anotación y recuperación semántica de información desarrollada en el Capítulo II.

La validación se ha realizado dentro de una organización orientada al desarrollo de proyectos de I+D+i relacionados con el dominio de las TIC. Para ello, fue necesaria la obtención de recursos utilizados por esta organización. A partir de estos recursos se generaron las entidades necesarias con las que se nutrió el sistema y se generaron los resultados que se analizan en este capítulo.

Antes de comenzar con el análisis de los resultados obtenidos por los experimentos, se presenta un breve estudio de las diferentes métricas que se utilizarán para analizar el rendimiento del sistema. Después, en las sucesivas secciones se llevará a cabo la evaluación de la aplicación semántica para la similitud en el dominio de la gestión de proyectos de I+D+i.

V.2. MEDIDAS DE EVALUACIÓN

La evaluación del rendimiento de la aplicación de cálculo de similitud se ha llevado a cabo a través de la utilización de métricas de evaluación estándar utilizadas en sistemas de procesamiento del lenguaje natural y de recuperación, y extracción de información como son: "precisión" (véase fórmula (III.4)), "exhaustividad" (véase fórmula (III.5)) y "medida-F" (véase fórmula (III.3)).

Estas mismas métricas fueron aplicadas para la validación de la metodología de anotación presentada en el Capítulo II.

La propuesta del sistema de cálculo de similitud descrito en el Capítulo IV se evaluó utilizando estas tres métricas sobre el motor de inferencia semántica en el dominio de la gestión de la I+D+i. La evaluación se fundamentó en validar la precisión de las entidades recuperadas por el motor de inferencia semántico y en comprobar el índice de exhaustividad sobre estas entidades. Para ello, se redefinieron las métricas para que utilizasen '`entidades relevantes`' frente a '`entidades recuperadas`'. Las '`entidades recuperadas`' representan el número de entidades que se recuperan en un proceso de búsqueda, mientras que las '`entidades relevantes`' son aquellas que se encuentran realmente relacionadas con la entidad a comparar en el proceso de búsqueda de entidades similares. De forma que, si los experimentos obtienen un buen número de entidades relevantes, esto indicaría que los métodos de similitud que están siendo aplicados son acertados y, por lo tanto, el rendimiento del motor de inferencia semántico es óptimo. Al mismo tiempo se verifica el número de entidades recuperadas dado que, si este valor es muy alto, significa que los métodos de similitud son muy genéricos y no proporcionan información de valor para el usuario.

$$precision = \frac{|\{entidades\ relevantes\} \cap \{entidades\ recuperadas\}|}{|\{entidades\ recuperadas\}|} \quad \textbf{(V.1)}$$

$$exhaustividad = \frac{|\{entidades\ relevantes\} \cap \{entidades\ recuperadas\}|}{|\{entidades\ relevantes\}|} \quad \textbf{(V.2)}$$

$$medida - F = 2 \cdot \frac{precision\ \cdot exhaustividad}{precision + exhaustividad} \quad \textbf{(V.3)}$$

Además, también se utiliza la métrica de "precisión promedio" con el fin de evaluar, no sólo el ranking de entidades después de cada consulta, sino también el orden en el que son presentados en el sistema. Esta métrica se representa en la

fórmula (V.4) como MAP (del inglés, "*Mean Average Precision*") donde, 'P' representa el número de consultas realizadas y 'p' referencia una consulta especifica del total. Así, el sumatorio del numerador se encarga de agregar los valores de precisión obtenidos en todas las consultas realizadas dividiéndolo entre el número total de consultas. La finalidad de esta métrica es aplicar un algoritmo que consiste en promediar las distintas precisiones obtenidas para cada consulta en cada conjunto de consultas realizadas y, de esta forma, trazar una curva de precisión.

$$MAP = \frac{\sum_{p=1}^{P} Precision(p)}{P} \quad \textbf{(V.4)}$$

En los siguientes apartados se describe el proceso de validación aplicado, empezando por la configuración del escenario de validación y terminando con un apartado de conclusiones acerca de los resultados obtenidos.

V.3. VALIDACIÓN EN EL DOMINIO DE LA I+D+I

V.3.1. INTRODUCCIÓN

En un ambiente empresarial, la gestión I+D+i supone una de las mayores preocupaciones de las empresas, convirtiéndose en uno de los aspectos más relevantes en las organizaciones que buscan soluciones a las necesidades de los clientes (Quélin, 2000). La correcta gestión de la I+D+i en una organización supone una ventaja competitiva en esta búsqueda de soluciones para que las organizaciones sean capaces de satisfacer las demandas de sus clientes potenciales, sin necesidad de sufrir ningún tipo de riesgo que proporcione una ventaja a sus competidores (Alves et al., 2005). Esta necesidad empresarial ha estimulado la aparición de varios proyectos con el propósito de ofrecer herramientas que faciliten la gestión del conocimiento en proyectos I+D+i mejorando su competitividad con respecto a sus adversarios en el sector (García-Moreno et al., 2013).

Actualmente, uno de los mayores problemas a los que se enfrentan las empresas y los gestores de la innovación es la necesidad de una plataforma que facilite herramientas que permitan la explotación del conocimiento en los proyectos de I+D+i. La mayoría de estas herramientas refuerzan la importancia de la integración y la colaboración entre los distintos grupos de interés como fuente de recursos a tener en cuenta en el desarrollo de este tipo de plataformas (Nobelius, 2004). Esta consideración ha propiciado la aparición de un nuevo paradigma de innovación considerado como antítesis del modelo de integración vertical tradicional, donde las actividades internas de I+D dan lugar a productos desarrollados internamente por las organizaciones (Chesbrough et al., 2006). En concreto, la innovación abierta, nombre que recibe este nuevo paradigma, fomenta el uso de flujos de información internos y externos como fuente de conocimiento para acelerar la innovación de la organización (Chesbrough, 2003).

En el contexto del trabajo asociado al desarrollo de esta tesis doctoral, se realizó un análisis en profundidad acerca de la explotación de estos flujos internos y externos por parte de las organizaciones. Este análisis derivó en la conveniencia de aplicar la metodología de anotación en el campo de la similitud. El objetivo de esta aplicación es el desarrollo de una sistema que, además de proporcionar un servicio de anotación y recuperación semántica, proporcione un servicio de recomendación basado en la similitud que facilite la recuperación de recursos de información relacionados con el mundo de la I+D+i de una manera altamente eficaz y rentable. Por ejemplo, entre estas recomendaciones podrían incluirse las organizaciones más adecuadas para colaborar en el proyecto o encontrar los evaluadores con más experiencia en las tecnologías del proyecto.

V.3.2. ESCENARIO DE EVALUACIÓN

El escenario de evaluación configurado para validar la herramienta utiliza gran parte de los recursos que han sido descritos en el Capítulo III en el contexto de las validaciones desarrolladas para la metodología de anotación y recuperación semántica de información. En primer lugar, la ontología que se utiliza para anotar semánticamente los recursos se basa en el trabajo presentado en (Rodríguez-García et al., 2012). Esta ontología fue creada a partir de recursos Wikipedia. Para

este proceso de evaluación se construyó una ontología de forma manual que definía un conjunto reducido de conceptos. Después, esta ontología fue enriquecida y ampliada mediante la herramienta de evolución de ontologías descrita en el Capítulo II. Concretamente, la ontología tenía aproximadamente unos 250 conceptos y 400 relaciones taxonómicas. La Tabla V.1 muestra algunos de los atributos de la ontología de las TIC creada.

Tabla V.1 Caracteristicas de la ontologia TIC utilizada en la validación

	Ontología TIC
Clases	1747
Propiedades `dataproperty`	39
Propiedades `objectproperty`	155
Relaciones `Subclass_of`	5379
Clases con múltiple herencia	259
Máx. profundidad de la clase árbol	11
Min. profundidad de la clases árbol	2
Promedio profundidad de la clases árbol	5,9
Máx. factor de bifurcación de las clases árbol	113
Min. factor de bifurcación de la clase árbol	1
Promedio factor de bifurcación de la clase árbol	6

Una vez definida la ontología, el siguiente paso fue la inserción de proyectos I+D+i y personal en el sistema. Este proceso se realizó en el marco de un proyecto de investigación y fue necesaria la colaboración del departamento de I+D+i de una organización de desarrollo software. Esta organización proporcionó un conjunto de proyectos de I+D+i relacionados con el dominio de las TIC. Concretamente, se introdujeron 40 proyectos de I+D+i relacionados con el desarrollo de software. En cada proyecto fue seleccionado un promedio de 5 personas implicadas en el proyecto. La inserción de esta información en el sistema dio lugar a más de 100 descripciones semánticas que fueron insertadas en el repositorio de ontologías que se implementó a través de Virtuoso (Virtuoso, 2009).

La inserción de los perfiles basados en ResumeRDF (Bojārs & Breslin, 2007), FOAF (Brickley & Miller, 2012) y DOAP (Dumbill, 2012) se realizó a través de una aplicación Web incorporada en el sistema que facilitaba la inserción y administración de descripciones semánticas en el repositorio de la ontología. En primer lugar, se crearon los perfiles semánticos relacionados con participantes del proyecto y, después, se crearon manualmente las descripciones semánticas de los proyectos. Cada perfil semántico fue anotado utilizando el módulo de representación y anotación semántica, y se calculó el índice semántico correspondiente utilizando el módulo de indexación semántica. Por último, a partir de estos índices y junto con los perfiles semánticos, construidos a partir de la transformación de información a RDF, se utilizó el módulo de cálculo de similitud para aplicar los algoritmos que calculasen las matrices de similitud. En este escenario, las entidades que han sido identificadas para generar estos perfiles semánticos son: "Proyecto", "Trabajador", "Organización" y "Tecnología". A partir de estas entidades, se han elaborado los perfiles semánticos empleados para la construcción de las matrices de similitud, que proporcionaron los resultados que a continuación serán analizados.

V.3.3. RESULTADOS

La validación del sistema ha sido desarrollada a partir de las métricas definidas al inicio del capítulo. Para realizar la validación, aparte de la información insertada en el repositorio de ontologías, se incluyeron 10 proyectos nuevos de software de I+D+i relacionados con las TIC. Para cada proyecto, un experto generó la descripción semántica correspondiente para insertarla en el repositorio de ontologías. Posteriormente, se seleccionó manualmente un conjunto de recursos humanos de la organización para cada proyecto, en función de la temática del mismo, y del perfil y experiencia de los recursos. Esto se contrastó con los datos devueltos por la aplicación para verificar la efectividad del sistema y para sugerir el equipo de trabajo de la empresa a cada uno de estos proyectos. Además, el experto seleccionó los proyectos existentes en el sistema más similares a cada uno de esos 10 proyectos. Simultáneamente, se utilizó el motor de inferencia para que realizase la misma tarea automáticamente. Por último, ambos resultados, los

obtenidos manualmente por los expertos y los obtenidos automáticamente por el motor de inferencia semántica, fueron comparados.

A raíz de esta comparación se obtuvieron los resultados que se muestran en la Tabla V.2, donde se recogen los resultados en términos de recomendaciones acertadas ('A'), recomendaciones extraídas ('E') y recomendaciones relevantes ('R'). A partir de estos valores se elaboró la Tabla V.3, que proporciona información sobre la efectividad en la recomendación de proyectos y grupos de trabajo en términos de las medidas de precisión ('P'), exhaustividad ('E') y medida-F ('F'), tal cual fueron definidas al principio del capítulo.

En la Tabla V.3 se presentan varias clasificaciones que se enumeran a continuación. En primer lugar, se agrupan los resultados por temas, mostrando las métricas obtenidas por cada experimento realizado. También es posible relacionar las métricas de los proyectos recomendados con las métricas del grupo de trabajo recomendado para ese proyecto. Además, se ofrecen las medias aritméticas de los resultados obtenidos en los experimentos agrupadas por temática. Por último, los resultados de la Tabla V.3 se utilizan para calcular la métrica de "precisión promedio", última métrica descrita que se va a utilizar para evaluar el sistema y cuyos valores se muestran en la Tabla V.4. Este cálculo permite no sólo tener en cuenta el ranking de proyectos recomendados sino que, además, evalúa el orden en que éstos son devueltos.

Tabla V.2 Valores de Recomendaciones Acertadas (A), Extraidas (E) y Relevantes (R)

	Pruebas	Proyectos recomendados			Grupos de trabajo recomendados		
Temas		**A**	**E**	**R**	**A**	**E**	**R**
Ingeniería del software (4)	**1**	7	9	10	1	2	2
	2	8	9	11	2	2	3
	3	7	9	10	3	4	4
	4	8	11	13	1	1	2
Interacción persona-computador (2)	**5**	5	7	8	1	2	2
	6	6	9	9	1	1	1
Vida cotidiana asistida por el entorno (2)	**7**	11	12	14	2	3	3
	8	11	13	17	2	3	4
Servicios de internet (2)	**9**	2	3	4	1	2	2
	10	3	4	5	1	1	1

Tabla V.3 Valores de Precisión (P), Exhaustividad (E) y Medida-F (F) obtenidos en el experimiento

	Pruebas	Proyectos recomendados			Grupos de trabajo recomendados		
Temas		**P**	**E**	**F**	**P**	**E**	**F**
Ingeniería del software (4)	**1**	0,75	0,66	0,7	0,71	0,63	0,67
	2	0,84	0,7	0,76	0,76	0,64	0,69
	3	0,81	0,72	0,76	0,74	0,68	0,71
	4	0,78	0,65	0,71	0,73	0,58	0,65
Media		**0,80**	**0,68**	**0,73**	**0,74**	**0,63**	**0,68**
Interacción persona-computador (2)	**5**	0,69	0,62	0,65	0,68	0,6	0,64
	6	0,72	0,7	0,71	0,73	0,66	0,69
Media		0,71	0,66	0,68	0,71	0,63	0,67
Vida cotidiana asistida por el entorno (2)	**7**	0,88	0,76	**0,82**	0,77	0,68	0,72
	8	0,82	0,64	0,72	0,7	0,53	0,6
Media		0,85	0,70	0,77	0,74	0,61	0,66
Servicios de internet (2)	**9**	0,71	0,61	0,66	0,65	0,58	0,61
	10	0,65	0,59	0,62	0,63	0,55	0,59
Media		0,68	0,60	0,64	0,64	0,57	0,60
Total		**0,77**	**0,66**	**0,71**	**0,71**	**0,61**	**0,66**

Tabla V.4 Valores de la métrica Precisión Promedio obtenidos

Tema del proyecto	Precisión promedio Proyectos recomendados	Precisión promedio Grupos de trabajo recomendados
Ingeniería del software	0,80	0,74
Interacción persona-computador	0,71	0,71
Vida cotidiana asistida por el entorno	0,85	0,74
Servicios de internet	0,68	0,64
Total	**0,77**	**0,71**

Como se puede observar en la Tabla V.3, los experimentos que obtienen mejores resultados son aquellos asociados a la recomendación de proyectos que describen soluciones para los temas de "Vida cotidiana asistida por el entorno" y para "Ingeniería del software". La media aritmética de los valores de precisión, exhaustividad y medida-F es de 85%, 70% y 77%, respectivamente, en el caso de los experimentos sobre proyectos relacionados con el tema de "Vida cotidiana asistida por el entorno", y 80%, 68% y 73%, respectivamente, en el caso de las pruebas sobre proyectos relacionados con la "Ingeniería del Software". Estos resultados positivos para la recomendación de proyectos coinciden con aquellos en el ámbito de grupos de trabajo recomendados, donde los experimentos sobre el tema "Vida cotidiana asistida por el entorno" han obtenido una media aritmética de precisión del 74%, de exhaustividad del 61% y de medida-F del 66%, similar a la obtenida en la recomendación de grupos de trabajos en el tema de "Ingeniería del Software", con media de precisión del 74%, de exhaustividad del 63% y de medida-F del 68%.

Por otro lado, en las otras dos temáticas de proyectos empleadas en la validación, "Interacción persona-computador" y "Servicios de internet", los experimentos en recomendación de proyectos obtuvieron unos índices de precisión del 71% y 68%, respectivamente, de exhaustividad del 66% y 60%, respectivamente, y de medida-F del 68% y 64%, respectivamente. Este leve empeoramiento también se refleja en los valores obtenidos para la recomendación de grupos de trabajo en estas temáticas. En concreto, en las recomendaciones sobre el tema "Interacción persona-computador" los resultados obtenidos son de

un 71% de precisión, un 63% de exhaustividad y un 67% de medida-F. En el caso de los experimentos para el tema "Servicios de Internet" los índices obtenidos son del 64% de precisión, 57% de exhaustividad y 60% de medida-F.

El análisis de estos resultados, tanto en el ámbito de la recomendación de proyectos como en el ámbito de la recomendación de grupos de trabajo, pone de manifiesto que los bajos valores de efectividad en los ámbitos de "Interacción persona-computador" y "Servicios de Internet" se deben a la escasa experiencia que la organización participante en el experimento dispone en proyectos en estas temáticas. Esta escasa experiencia afecta directamente a la efectividad del sistema debido a que, el hecho de disponer de menos proyectos, currículos, tecnologías, en general, recursos, sobre esta temática, hace que el sistema no esté enriquecido en este dominio. En otras palabras, si la organización carece de experiencia en proyectos relacionados con estas temáticas, esto implica que en el sistema de información de la organización existirán pocos proyectos relacionados con estas temáticas. Por lo tanto, si el número de proyectos sobre esta temática es escaso implica, haciendo alusión al funcionamiento del sistema evaluado, que la extracción de términos sobre esta temática es escasa y por lo tanto no se enriquece el dominio ontológico subyacente con los nuevos conceptos extraídos. Si la ontología no es enriquecida, la precisión del motor de inferencia se verá afectada debido al hecho de que disponer de un dominio ontológico escaso repercute negativamente sobre los metadatos asociados a los perfiles semánticos durante la etapa de anotación semántica. Estos metadatos representan uno de los recursos que se utilizan para la elaboración de las matrices de similitud y que el motor de búsqueda semántica utiliza para la recuperación de información. Por lo tanto, cuanto menos metadatos disponga cualquiera de los dos motores sobre un tema, menos precisión existirá en los resultados obtenidos. De ahí, que para los experimentos sobre los proyectos relacionados con "Interacción persona-computador" y "Servicios de Internet" los índices de precisión obtenidos sean bajos.

Si se analizan con detenimiento los detalles de la Tabla V.3, es posible apreciar que las pruebas '9' y '10' han obtenido un resultado de precisión bajo, del 71% y 65%, respectivamente. Estos valores menos destacados para los índices de precisión en las pruebas '9' y '10' se deben, fundamentalmente, a la escasa

experiencia de la organización en este tipo de proyectos. Como se puede observar en la Tabla V.2, existen muy pocos proyectos anteriores similares en el sistema. En cambio, las recomendaciones realizadas para las pruebas '7' y '8' resultan en altos índices de precisión, concretamente, del 88% y 82%, respectivamente. En este caso ocurre al contrario, los resultados obtenidos reflejan que la organización se encuentra centrada en el desarrollo de tecnologías sobre la "Vida cotidiana asistida por el entorno", de ahí que exista una amplia base de proyectos anteriores en el repositorio de la ontología y la mejora experimentada en los resultados.

Continuando con el análisis de las pruebas '9' y '10', el índice de exhaustividad en los experimentos llevados a cabo sobre la recomendación de proyectos similares es el más bajo, con un 61% y 59%, respectivamente. Esto se debe, nuevamente, al reducido número de proyectos similares con la temática de "Servicios de Internet". Los mejores resultados de todos los experimentos realizados se han obtenido en las pruebas sobre proyectos relacionados con el tema de "Vida cotidiana asistida por el entorno". Estos resultados tan favorables se deben a la especificidad de la temática y, por ende, desambiguación que tienen este tipo de proyectos. Por último, recalcar que el experimento número '7´ es el que obtiene el mejor valor de medida-F, con una puntuación del 82%. Este hecho es fiel reflejo de los altos porcentajes obtenidos en las métricas de precisión y exhaustividad en comparación con los resultados de los demás experimentos.

La media total de los resultados obtenidos, en cuanto a los experimentos sobre recomendación de proyectos, han arrojado una precisión del 77%, una exhaustividad de un 66% y una medida-F del 71%, lo que indica que son buenos resultados y, por tanto, se considera un éxito el experimento realizado y la validación de la metodología que se propuso en el Capítulo IV. Posteriormente, al final de este análisis, se realiza una comparación de los resultados obtenidos por otros enfoques propuestos.

En cuanto a las recomendaciones de los grupos de trabajo, el sistema obtiene peores resultados que los alcanzados en las pruebas de sugerencias de proyectos similares. De acuerdo con los resultados de la Tabla V.3, las pruebas '9' y '10' obtienen los menores índices de precisión. Al igual que se indicó anteriormente, estos bajos valores de precisión se deben a que la organización no cuenta con muchos empleados que posean experiencia en proyectos sobre la temática de

"Servicios de Internet". En cambio, las mejores puntuaciones en cuanto a la métrica de precisión se obtuvieron en las pruebas relacionadas con "Ingeniería del software" y "Vida cotidiana asistida por el entorno". En este caso, los empleados de la organización presentan una amplia experiencia relacionada en esta temática de proyectos. En concreto, la mejor puntuación se obtiene para la prueba '7', con una precisión del 77%. En términos de exhaustividad, el resultado más discreto es el obtenido por la prueba '8', siendo ésta del 53%. Esto se debe a que algunos de los trabajadores con experiencia en este tipo de proyectos no fueron sugeridos por el sistema durante el experimento. Los malos resultados cosechados en los experimentos relacionados con "Servicios de Internet" se deben a la carencia de trabajadores por parte de la empresa con experiencia en esta temática. En cuanto a los resultados globales, los obtenidos para las recomendaciones de personal son prometedores, con un índice de precisión del 71%, exhaustividad del 61% y medida-F del 66%.

Por otro lado, la Tabla V.4 muestra la métrica de precisión promedio de los experimentos realizados. Si se analizan los resultados de la tabla es posible destacar los obtenidos para las pruebas relacionadas con la temática "Vida cotidiana asistida por el entorno", que han obtenido en la recomendación de proyectos y grupos de trabajo una precisión promedio de 85% y 74%, respectivamente.

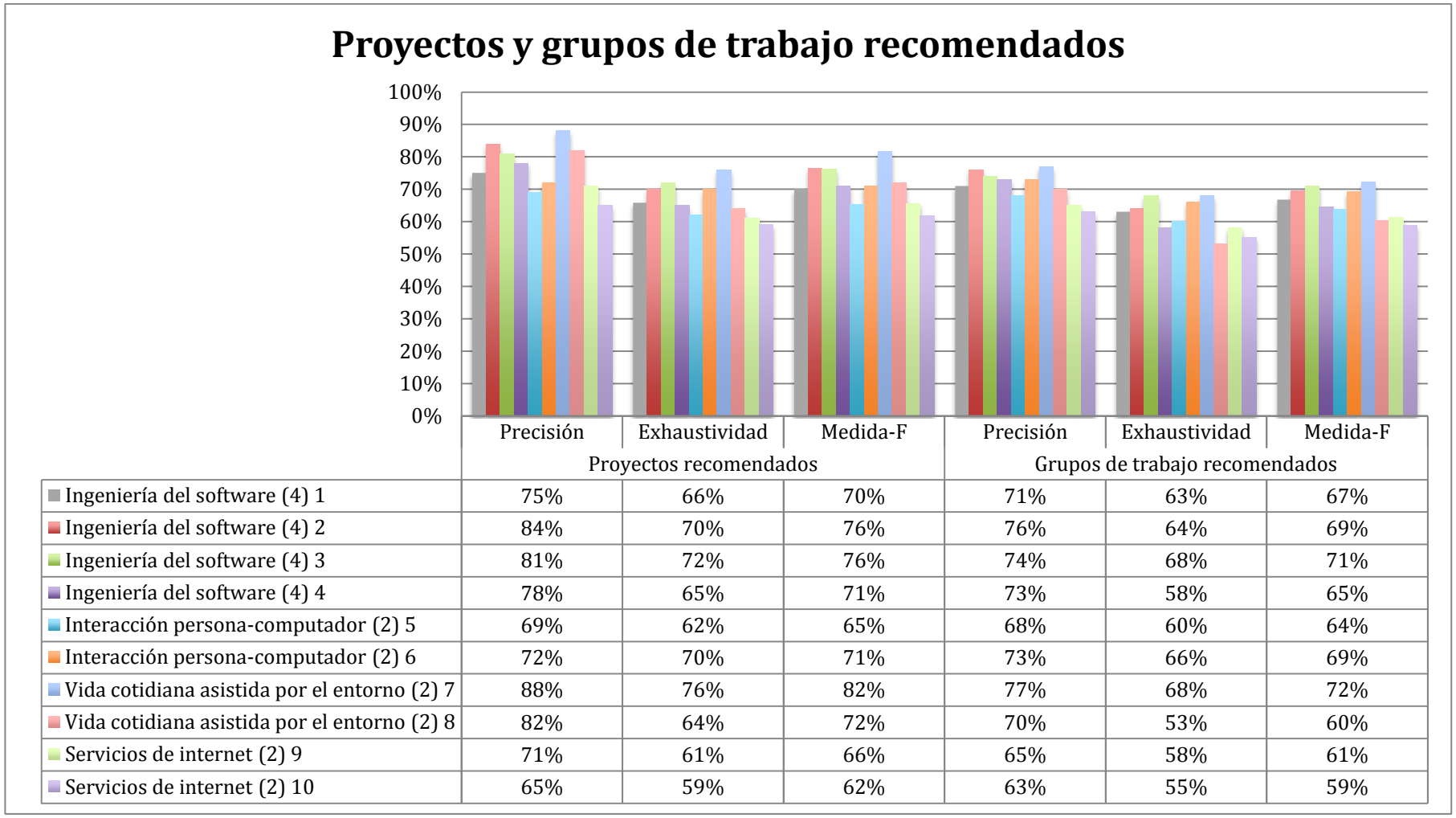
Siguiendo con este análisis de la métrica "precisión promedio", si analizamos la Tabla V.4 se observa que los resultados que se representan hacen alusión al análisis extraído a partir de los demás valores analizados en las otras tablas. Las pruebas relacionadas con temas de "Ingeniería del software" y "Vida cotidiana asistida por el entorno" alcanzan porcentajes de precisión destacable en comparación con las pruebas cuya temática se encuentra relacionada con la "Interacción persona-computador" y "Servicios de Internet".

Para finalizar este análisis, se dispone de un conjunto de gráficos que representan los valores obtenidos en cada una de las tablas analizadas. Así, la Figura V.1 representa los resultados obtenidos a partir de la aplicación de las métricas de precisión, exhaustividad y medida-F representados en la Tabla V.2. En esta representación es posible observar de forma más nítida la diferencia existente en los valores de precisión y exhaustividad de las pruebas relacionadas con las

temáticas de “Ingeniería del software” y “Vida cotidiana asistida por el entorno” frente a las pruebas con los otros dos temas. Además, para contrastar esta información, la Figura V.2 representa los promedios de los resultados obtenidos clasificados por temas de proyectos. Esta representación remarca visiblemente la superioridad en valores de precisión y exhaustividad para los proyectos con temáticas en las que la organización y sus empleados tienen más experiencia, en contraste con los proyectos relacionados con temas en los que ni los empleados ni la organización han trabajado anteriormente.

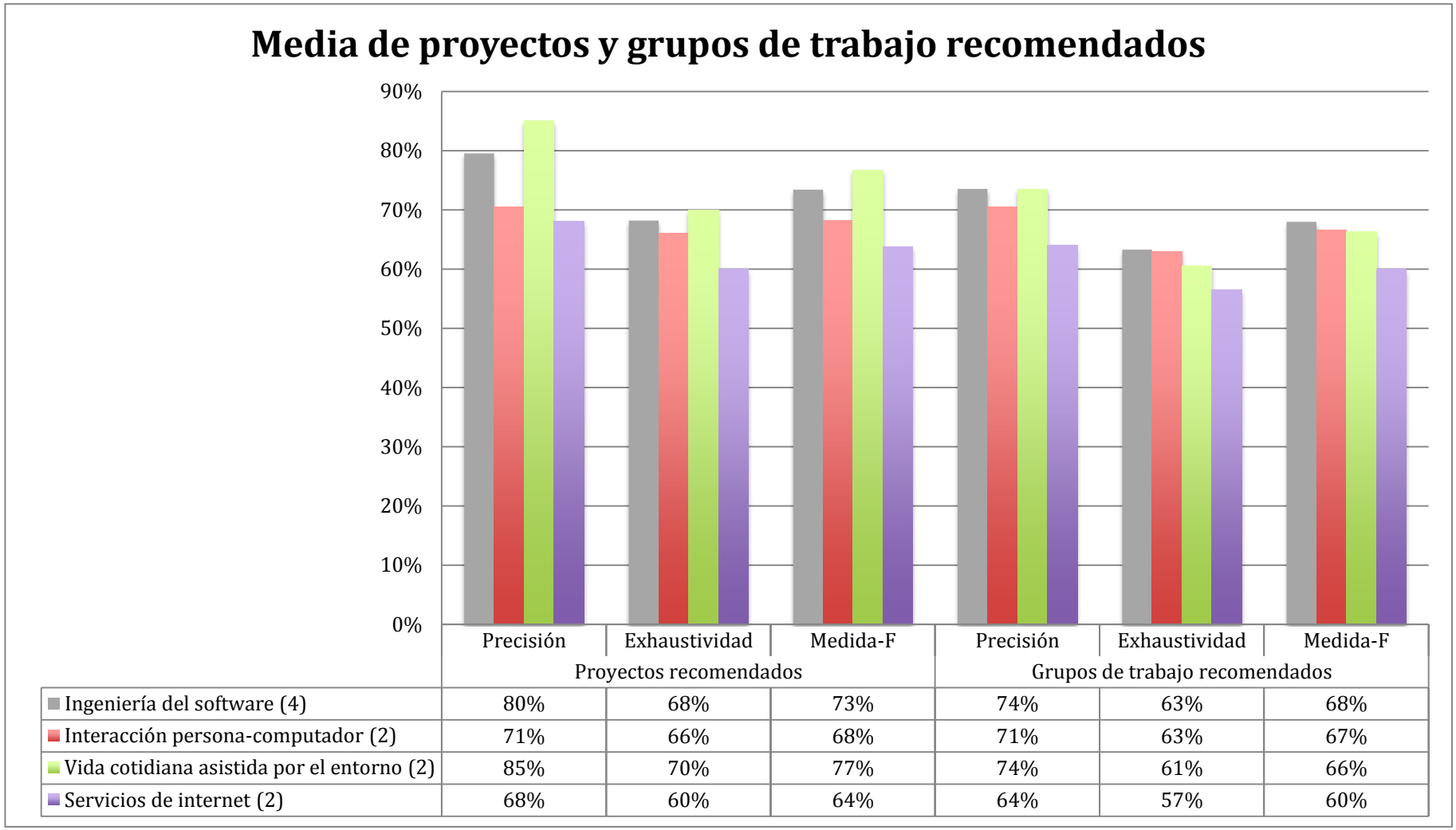
Por último, para concluir este análisis de resultados, la Figura V.3 proporciona una representación gráfica del promedio total logrado en todos los experimentos realizados. Este resultado promedio proporciona una visión más completa y significativa sobre la media de resultados obtenidos que la métrica “precisión promedio”, representada en la Figura V.4, donde solo se representa el promedio de los valores de precisión.

En cuanto a este estudio, los resultados dejan patente el buen rendimiento de la aplicación semántica para el cálculo de similitud, proporcionando una nueva metodología inteligente de explotación de información en un ambiente organizacional.



	Proyectos recomendados			Grupos de trabajo recomendados		
	Precisión	Exhaustividad	Medida-F	Precisión	Exhaustividad	Medida-F
Ingeniería del software (4) 1	75%	66%	70%	71%	63%	67%
Ingeniería del software (4) 2	84%	70%	76%	76%	64%	69%
Ingeniería del software (4) 3	81%	72%	76%	74%	68%	71%
Ingeniería del software (4) 4	78%	65%	71%	73%	58%	65%
Interacción persona-computador (2) 5	69%	62%	65%	68%	60%	64%
Interacción persona-computador (2) 6	72%	70%	71%	73%	66%	69%
Vida cotidiana asistida por el entorno (2) 7	88%	76%	82%	77%	68%	72%
Vida cotidiana asistida por el entorno (2) 8	82%	64%	72%	70%	53%	60%
Servicios de internet (2) 9	71%	61%	66%	65%	58%	61%
Servicios de internet (2) 10	65%	59%	62%	63%	55%	59%

Figura V.1 Resultados obtenidos en medidas de precisión, exhaustividad y medida-F



	Proyectos recomendados			Grupos de trabajo recomendados		
	Precisión	Exhaustividad	Medida-F	Precisión	Exhaustividad	Medida-F
Ingeniería del software (4)	80%	68%	73%	74%	63%	68%
Interacción persona-computador (2)	71%	66%	68%	71%	63%	67%
Vida cotidiana asistida por el entorno (2)	85%	70%	77%	74%	61%	66%
Servicios de internet (2)	68%	60%	64%	64%	57%	60%

Figura V.2 Media de resultados obtenidos en términos de precisión, exhaustividad y medida-F

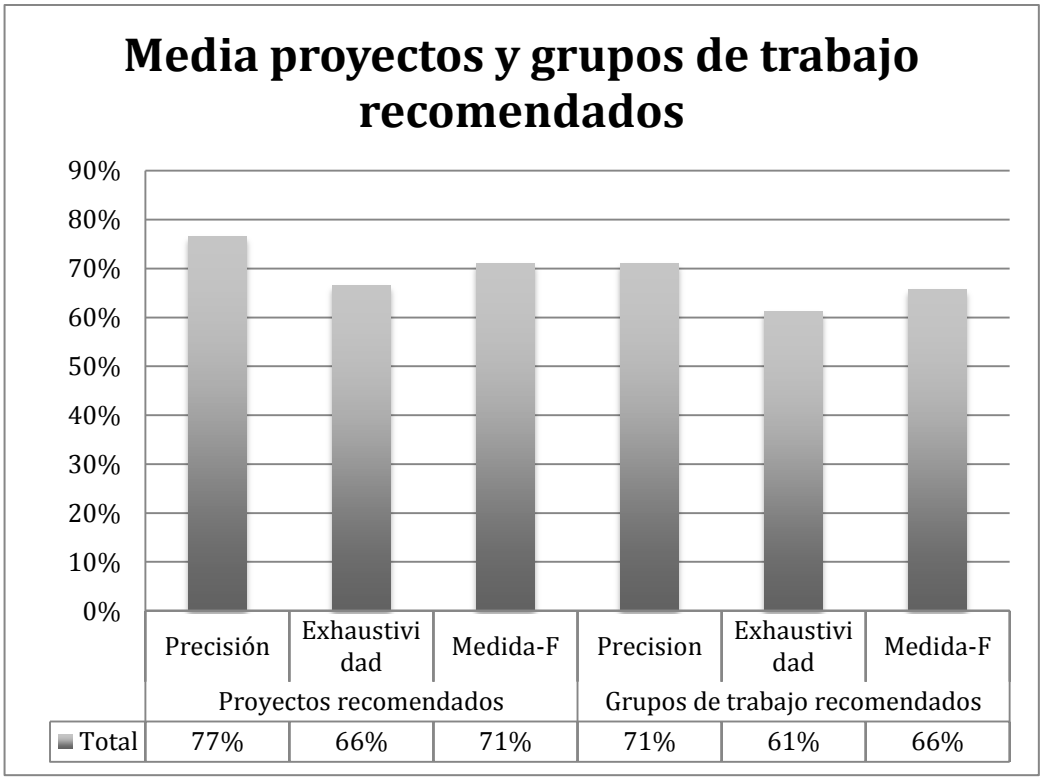


Figura V.3 Media total de resultados obtenidos en términos de precisión, exhaustividad y medida-F

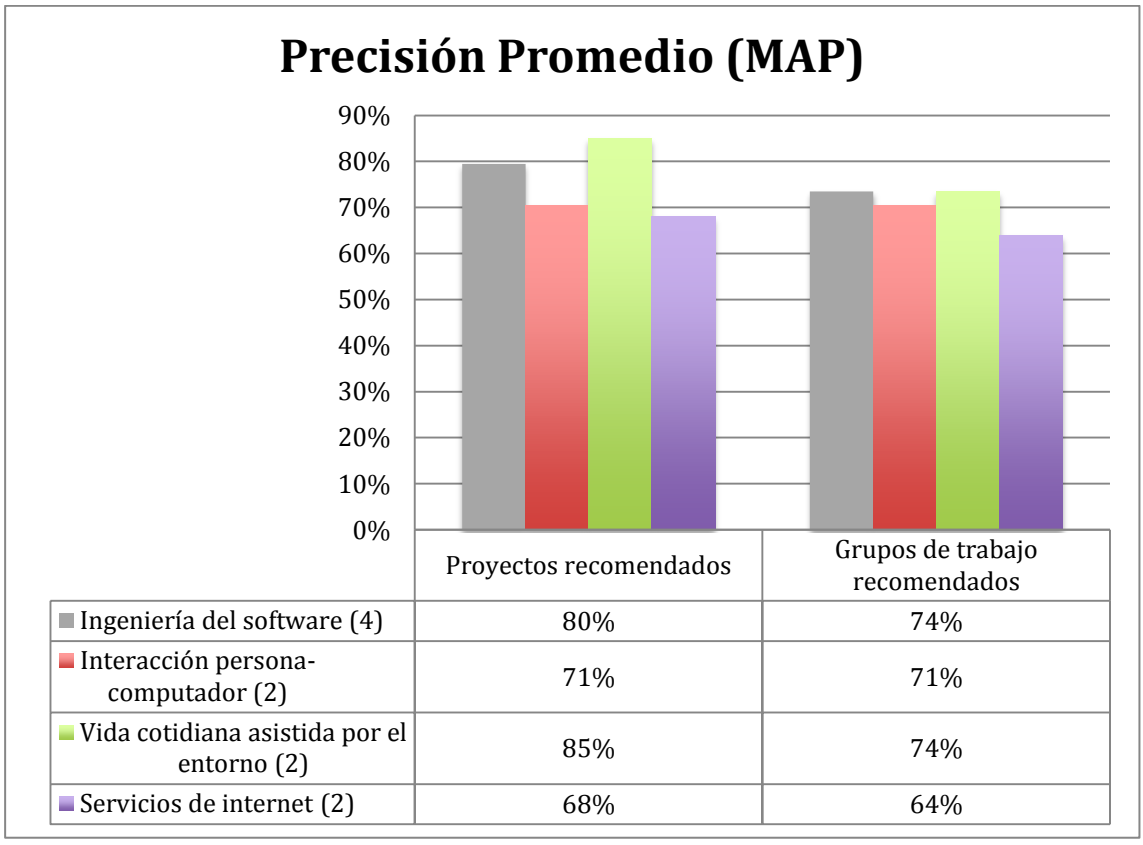


Figura V.4 Precisión promedio resultante

La comparación de los resultados obtenidos en estos experimentos con los alcanzados por otras aproximaciones existentes resulta compleja debido a que ni las aplicaciones software están disponibles, ni tampoco la batería de ejemplos que fueron utilizados para su validación. Cabe destacar que los ejemplos utilizados en los experimentos realizados con estas aproximaciones similares difieren significativamente en cuanto a contenido y tamaño. Los conceptos y las relaciones se tratan de manera diferente, y también difieren los dominios, las relaciones, los atributos, y el modo en que se tratan los errores. Además, algunas de las aproximaciones en la literatura no proporcionan ningún tipo de métrica estadística y, en cualquier caso, para hacer una comparación razonable entre la propuesta aquí presentada y las aproximaciones similares sería necesaria la utilización de los mismos ejemplos de ensayo, así como la misma ontología. Con todo, únicamente es posible realizar una comparativa con sistemas de búsqueda de expertos tales como el presentado en (Stankovic, 2010), donde las puntuaciones de medida-F obtenidos fueron entre el 49% y el 71%. Como se puede ver, en términos cuantitativos nuestro enfoque mejora estos resultados debido a la obtención de un medida-F entre el 59% y el 72%. Por último, algunos trabajos utilizan la métrica de "precisión promedio" para evaluar el rendimiento de las aproximaciones descritas en estos trabajos. Por ejemplo, el trabajo presentado en (Zhu et al., 2010) es el que obtiene, de entre todas las aproximaciones analizadas, una mejor puntuación con respecto a esta métrica, con un valor del 65.64%, mientras que nuestro enfoque obtiene una puntuación del 71%.

V.3.4. CONCLUSIÓN

El análisis de los resultados de la prueba experimental que se ha llevado a cabo ha revelado información congruente sobre las hipótesis que se planteó al inicio del experimento. Por un lado, las áreas de I+D+i en las que la organización colaboradora no tenía experiencia previa iban a obtener unos resultados peores, como es en el caso de los resultados obtenidos a partir de experimentos llevados a cabo sobre proyectos relacionados con "Interacción persona-computador" y "Servicios de Internet". Este comportamiento se diferencia bastante de las pruebas relativas a las temáticas de proyectos en los que sí tenía experiencia, donde los

resultados son bastante buenos. Este es el caso, por ejemplo, de los experimentos relacionados con los proyectos de "Ingeniería del software" y "Vida cotidiana asistida por el entorno". Además, también se destaca la ambigüedad como una de las principales dificultades que hay que tener en cuenta. Aquellos temas que se incluyeron en la batería de proyectos a analizar, que hacen referencia a temáticas muy genéricas como, por ejemplo, "Interacción persona-computador" o "Servicios de Internet" obtienen peores resultados que aquellos obtenidos en experimentos sobre proyectos donde la concreción del dominio se encuentra mejor definida y sin ningún tipo de ambigüedad. Esto se debe a que las tecnologías que se utilizan en esa temática de proyecto quedan especificadas con todo detalle. Ejemplo de esta circunstancia es el caso de los experimentos sobre proyectos relacionados con el tema "Vida cotidiana asistida por el entorno", que se encuentran centrados en la temática I+D+i, concretamente, en el área de domótica. En este dominio las tecnologías que se utilizan están bien definidas, a diferencia de lo que ocurre en la temática de "Servicios de Internet", donde las tecnologías vinculadas a este tipo de proyectos están más dispersas y, por lo tanto, los resultados son menos explícitos y concretos.

Por otro lado, el estudio realizado proporciona resultados prometedores sobre la aplicación del cálculo de similitud semántica, ofreciendo un sistema de búsqueda de recursos eficiente y capaz de relacionar distintos tipos de recursos a partir de un mismo tema. La capacidad que ofrece esta aplicación multiplica las probabilidades de éxito de las empresas proporcionando, en este caso, una metodología de explotación más eficiente que le permite poder afrontar los retos tecnológicos de los proyectos de I+D+i de una manera más inteligente. Como se ha podido comprobar, proporciona un sistema de ayuda a la decisión que, en el dominio seleccionado, sería capaz de proporcionar información extra para facilitar la asignación del grupo de trabajo más capacitado para afrontar el reto de proyectos I+D+i. Esto lo consigue atendiendo a la temática de los proyectos, y teniendo en cuenta las habilidades y experiencias de los trabajadores de la organización. Por lo tanto, dados los resultados positivos obtenidos y la amplia cantidad de variantes en lo relativa a su aplicabilidad, se concluye que la metodología y la aplicación de cálculo de similitud han sido validadas con éxito y suponen un avance frente al estado del arte.

V.4. RESUMEN

En este capítulo se presenta una evaluación de la metodología de anotación y recuperación semántica de información descrita en el Capítulo II aplicada para el cálculo de similitud entre instancias. Esta evaluación se ha llevado a cabo en el dominio de la I+D+i y se aplica sobre el motor de inferencia semántica. El objetivo de esta evaluación es validar el rendimiento de la aplicación, así como demostrar la aplicabilidad de la metodología de anotación desarrollada en un ambiente organizacional real.

La validación de la aplicación comienza con una breve introducción, que describe los puntos más importantes en los que se centra el desarrollo del capítulo. A continuación, en el epígrafe V.2 se describen las diferentes métricas que han sido empleadas para la evaluación. Por último, en el epígrafe V.3 se lleva a cabo la evaluación del motor de inferencia semántico. Previo al análisis de los resultados obtenidos, se realiza una breve contextualización en la que se describe el dominio seleccionado utilizado para la evaluación y también se enumeran las características más relevantes del escenario de evaluación que se ha utilizado para validar la aplicación de la metodología.

Capítulo VI. CONCLUSIONES Y LÍNEAS FUTURAS.

VI.1. CONCLUSIONES

El trabajo presentado en esta tesis describe una metodología innovadora de extracción y recuperación de información que utiliza, como base de conocimiento, una ontología. A partir de esta ontología se generan los metadatos con los que se anota la información. La incorporación de estos metadatos a la información facilita los procesos de recuperación de información minimizando el tiempo de respuesta del sistema y mejorando la precisión de la información recuperada. Además, la implementación de esta metodología ha propiciado el desarrollo de una novedosa aplicación para el cálculo de similitud entre entidades almacenadas en repositorios de información.

En la actualidad, existen diferentes tipos de propuestas basadas en ontologías que proporcionan sistemas de anotación y recuperación semántica. La mayoría de ellas presentan algunas desventajas que dificultan su establecimiento como solución estándar. Las principales desventajas que afectan a este tipo de sistemas son (i) la gestión de la consistencia de la metainformación generada y (ii) la evolución de la información. La evolución está relacionada con las continuas modificaciones a los que los repositorios de información están sometidos. Los dominios de estos repositorios tienen una naturaleza evolutiva que no todos los sistemas de anotación basados en ontologías soportan. Además, la capacidad evolutiva de la información dentro de un repositorio afecta directamente a la consistencia de la metainformación. Así, cualquier modificación o actualización de dicha información debe de estar constantemente reflejada en los metadatos, manteniendo así la consistencia tanto de la información almacenada como de los metadatos generados. A pesar de estas características significativas analizadas anteriormente, no existe un claro modelo de comparación que se pueda utilizar para contrastar metodologías de anotación semántica (Uren et al., 2006). No obstante, en I.6.3 de esta memoria se recoge un análisis que revela la existencia de varias tendencias tecnológicas que se resumen en función de una serie de

características diferenciadoras predefinidas tales como: soporte de múltiples ontologías, formatos de documentos, métodos de anotación implementados, soporte de evolución de ontologías, entre otras (Rodríguez-García et al., 2014).

Entre los enfoques analizados, tomando como base la tendencia de proporcionar soporte para el manejo de múltiples ontologías, se han destacado las herramientas KIM (Popov et al., 2003), CREAM (Handschuh & Staab, 2003), Armadillo (Chapman et al., 2005), GoNTogle (Bikakis et al., 2010) y la descrita en esta tesis, que proporcionan soporte de múltiples ontologías, frente a otros enfoques como CERNO (Kiyavitskaya et al., 2009), S-CREAM (Handschuh et al., 2002), MnM (Vargas-Vera et al., 2002) y EVONTO (Tissaoui et al., 2011), que no comparten esta característica. Por el contrario, el análisis correspondiente a los métodos de anotación implementados permite concluir que pocos enfoques analizados incorporan todas las características analizadas. Sólo la aproximación GoNTogle (Bikakis et al., 2010) y el enfoque descrito en esta tesis doctoral, permiten a los expertos en el dominio manipular las anotaciones ya que ambos enfoques soportan métodos manuales de anotación.

La metodología de anotación y recuperación semántica de información propuesta en esta tesis proporciona un entorno estable y escalable a la vez que maduro y formal. El objetivo de esta metodología es la creación de una plataforma que, automáticamente, facilite la labor de anotación semántica, así como la búsqueda de recursos de una manera fácil y sin apenas la intervención de ningún humano. Para alcanzar este objetivo a partir de la plataforma propuesta, se ha dividido el sistema en un conjunto de módulos que, aportando funciones muy específicas, permiten la consecución del objetivo marcado. Entre las funciones que desempeñan cada uno de los módulos se pueden destacar las siguientes: representación y anotación semántica, indexación semántica, extracción de información, evolución de ontologías y motor de búsqueda semántico.

A partir de las herramientas analizadas y los criterios estipulados, las principales aportaciones de esta tesis son las siguientes:

- **Desarrollo de una metodología de anotación y recuperación semántica de información**. Esta metodología satisface cada una de las características que han sido analizadas anteriormente. En particular, entre las características de que dispone esta metodología se destacan las siguientes: formato OWL para

representar el conocimiento, soporte de información no estructurada y semiestructurada, anotación manual y automática, mecanismos que evitan incoherencias e inconsistencias, y soporte de evolución tanto a nivel de modelo ontológico como a nivel de recurso anotado.

- **Desarrollo de un modelo ontológico formal**. Este modelo representa semánticamente conceptos que definen un conjunto de atributos específicos y anotaciones que se utilizan en la etapa de anotación semántica. Este modelo ontológico formal ha sido diseñado para facilitar la adaptación de la metodología a cualquier dominio, ya que utiliza atributos y anotaciones que son propiedades estándares de las ontologías. Esta simplificación de diseño también está orientada a facilitar el proceso de evolución de ontologías que, al disponer de un modelo de representación de información basado en estas propiedades estándares, simplifica al máximo la evolución de ontologías a un proceso de incorporación de conceptos, sin la inclusión de ningún tipo de lógicas descriptivas o axiomas que puedan dificultar la posibilidad de enriquecer la ontología. Además, este sencillo modelo ontológico mejora el rendimiento del sistema de anotación acelerando el proceso de anotación semántica. Capacitándolo para trabajar sobre cantidades ingentes de información en tiempo reducido.
- **Validación del sistema de anotación semántica**. El proceso de validación del sistema de anotación semántica se realizó en dos dominios diferenciados: la computación en la nube y la I+D+i. El llevar a cabo esta validación supuso un considerable esfuerzo, debido a la inexistencia de modelos ontológicos del dominio que definieran la información requerida para, a partir de ellos, generar los metadatos necesarios para anotar semánticamente los recursos. Además, este proceso de validación requirió de la participación de varios expertos en ambos dominios para validar los resultados obtenidos por el sistema de anotación desarrollado. Las conclusiones extraídas tras el proceso de validación en estos dos dominios se alinearon por completo con los objetivos definidos al inicio de la investigación.
- **Aplicación de la anotación semántica en el cálculo de similitud entre instancias**. Los resultados satisfactorios cosechados durante la evaluación de la metodología de anotación semántica y, en particular, del buscador semántico

hicieron emerger varias líneas de aplicación, entre ellas, el cálculo de similitud semántico. Este cálculo permite obtener la similitud entre dos instancias independientemente del dominio/rango al que pertenezcan. La metodología de cálculo se basa en obtener la similitud entre dos instancias a partir de la comparación de los atributos que tengan definidos. Para establecer cómo se comparan las dos instancias, es decir, qué atributos de una instancia se van a comparar con qué atributos de la otra instancia, se utiliza un fichero de configuración donde se define qué instancias se van a comparar y cómo se va a llevar a cabo esta comparación. A partir de esta configuración, el sistema genera tantas matrices de similitud como comparaciones hayan sido definidas. Cada matriz de similitud representa la comparación entre dos instancias y los números que hay en ella representan la similitud entre cada par de instancias comparadas según la manera definida en el fichero de configuración. El cálculo de similitud se realiza a partir de unos algoritmos matemáticos implementados para cada tipo de atributo.

- **Validación de la aplicación de cálculo de similitud en un entorno real**. El dominio de evaluación que fue seleccionado para validar la aplicación de la metodología de anotación semántica en el contexto del cálculo de similitud fue el de proyectos de I+D+i relacionados con el dominio de las TIC. En este proceso validación también fue necesario contar con la participación de expertos y de un departamento de recursos humanos de la organización para validar la metodología de cálculo de similitud en la recuperación de proyectos y empleados. La evaluación se basaba en la inserción en el sistema de un conjunto de proyectos y antecedentes técnicos de un conjunto de empleados. El motor de inferencia semántico, partiendo de un proyecto concreto dado, retornaba un listado de proyectos similares y de empleados con el historial tecnológico más idóneo para llevarlo a cabo. Los resultados de la validación se pueden considerar positivos y favorables, sobre todo en contraste con los resultados de otras herramientas similares que han sido divulgados.

Una vez analizadas las diversas aportaciones de la tesis, a continuación se enumerarán algunas de las limitaciones que presenta la metodología de anotación descrita. Muchas de estas limitaciones constituyen el punto de partida para algunas de las líneas futuras que se comentarán para concluir este capítulo.

- **Lenguaje de consulta basado en palabras clave**. El lenguaje de consulta que se utiliza en el motor de búsqueda semántico está basado en palabras clave. Esta manera de definir las consultas limita bastante la expresividad de las mismas afectando directamente a la precisión de los resultados obtenidos. Esto se debe, como ya se analizó en la validación del motor de búsqueda, a que cuanto más expresivas y específicas sean las consultas, mejores y más precisos serán los resultados obtenidos.
- **Problema de ambigüedad**. El soporte de múltiples ontologías en la plataforma aporta un sinfín de ventajas con respecto a los sistemas que no incluyen esta característica. Sin embargo, este soporte también supone algunos inconvenientes como el de la ambigüedad. En particular, al soportar múltiples ontologías se puede dar el caso de que un concepto se encuentre definido en ambas ontologías y que éste tenga definiciones totalmente diferentes en cada conceptualización. Esta situación implicaría la inconsistencia de la metodología de anotación. Además, en esta situación también se vería afectada el proceso de indexación semántica, que seleccionaría la ontología para calcular el índice semántico a partir de los demás conceptos que han sido anotados.
- **Rendimiento del algoritmo de indexación semántica**. El cálculo de los índices semánticos utilizando la versión extendida del algoritmo TF-IDF, que se ha propuesto en esta tesis, requiere excesivo tiempo de ejecución debido a las dependencias existentes en el cálculo del algoritmo. Para agilizar este cálculo sería recomendable, rediseñar el algoritmo para que se pudieran obtener los valores utilizando un paradigma de computación paralela.
- **Dependencia de Wikipedia**. El proceso de evolución de ontologías se basa en la información de Wikipedia para recopilar nuevo conocimiento a incluir en la ontología. El hecho de sólo disponer de un repositorio de información resulta, en muchos casos, insuficiente para la búsqueda de un concepto concreto con el que evolucionar la ontología. Por lo tanto, la incorporación de nuevos repositorios de información como respaldo, incrementaría las posibilidades de encontrar esos conceptos y, finalmente, podrían ser definidos en la ontología sin la necesidad de que fueran desechados.

VI.2. LÍNEAS FUTURAS

En lo que respecta a las líneas futuras, en esta tesis doctoral se citan varios temas que no han sido considerados como parte relevante en el desarrollo de la misma, pero que proporcionan nuevas líneas de investigación que a continuación se señalan como trabajos futuros:

- *Extensión del módulo de búsqueda semántica que permita enriquecer el lenguaje de búsqueda.*

Actualmente, el motor que se encarga de realizar la búsqueda semántica se encuentra limitado por la utilización de un conjunto de palabras clave. Sin embargo, algunas propuestas actuales (Bast et al., 2007);(Hogan et al., 2011);(Patel et al., 2003) de buscadores semánticos utilizan lenguajes más ricos como SPARQL (Prud'Hommeaux & Seaborne, 2008), RDQL (Seaborne, 2004) y OWLQL (Fikes et al., 2004), que enriquecen el vocabulario de búsqueda sin hacer uso de palabras clave. El problema que plantea la utilización de este tipo de lenguajes semánticos en los sistemas de búsqueda es que requiere de usuarios con un cierto conocimiento en tecnologías de la Web Semántica para poder ser explotados con éxito. Por lo tanto, esta línea de investigación se basaría en la incorporación de una interfaz en lenguaje natural o vocabulario controlado (Valencia-García et al., 2011) que permita al usuario final, de forma guiada, construir las consultas aprovechando toda la riqueza de los lenguajes semánticos. El motor de búsqueda semántico parte de la descripción en lenguaje natural, proporcionada por el usuario y que se encuentra mapeada con los lenguajes semánticos, para realizar búsquedas más precisas. Así, se pondría a disposición de los usuarios inexpertos los beneficios que ofrecen este tipo de lenguajes.

- *Corrección de ambigüedades provocadas por el soporte de múltiples ontologías.*

Soportar múltiples ontologías en la anotación semántica puede ser muy beneficioso ya que, entre otras ventajas, permite al sistema cubrir diferentes dominios. Además, esta característica evita que la aplicación tenga que, forzosamente, depender de ontologías gigantescas que puedan afectar

negativamente al rendimiento del sistema. Sin embargo, las implicaciones que conlleva proporcionar soporte a múltiples ontologías no son ventajosas en todos los casos. En concreto, la posible introducción de diferentes dominios provoca la aparición de inconsistencias en las anotaciones así como que las anotaciones se vean afectadas por la ambigüedad de los términos. La utilización de técnicas de desambiguación proporcionaría una sólida solución a los problemas de inconsistencia y ambigüedad derivados del soporte de múltiples ontologías por parte del sistema de anotación.

- *Mejorar el rendimiento del algoritmo de indexación semántica.*

En un importante porcentaje de los casos, el algoritmo de indexación semántica tiene un impacto negativo sobre el rendimiento del sistema debido a la utilización de tiempo y recursos computacionales en los procesos de indexación semántica. Rediseñar el algoritmo de indexación para que explote el paralelismo existente en los procesos de indexación permitiría mitigar la pérdida de rendimiento de la aplicación. La paralelización de los procesos asociados a la indexación semántica posibilitaría mejorar el rendimiento y la eficiencia en términos del tiempo de ejecución.

- *Extender la metodología de anotación semántica a otros tipos de información.*

Actualmente, todas las validaciones a las que han sido sometidos los sistemas planteados en esta tesis han utilizado información no estructurada o semiestructurada. Sin embargo, la extensión que se propone como línea futura sería la utilización de esta metodología de anotación, o incluso la aplicación semántica para el cálculo de similitud, sobre información estructurada como, por ejemplo, bases de datos relacionales, hojas de cálculo, facturaciones, recibos, etc. Dentro de esta categoría también se puede incluir a los sistemas de almacenamiento de grandes cantidades de datos, como ocurre con Big Data, y los sistemas que manipulan grandes conjuntos de datos. Además, el impacto asociado a la utilización de la metodología presentada en este tipo de información es doble. En primer lugar, gracias a las anotaciones semánticas sería posible aplicar lenguajes de búsqueda más ricos que los asociados a las bases de datos

relacionales. Por otro lado, las anotaciones semánticas proporcionarían una descripción semántica en formato RDF de la información anotada. Esta descripción semántica favorece la utilización de herramientas tales como los razonadores, que incorporan métodos para generar nuevo conocimiento implícito a partir del conocimiento explicito obtenido durante el proceso de anotación de la información.

- *Incorporación de otros repositorios de información para evolucionar la ontología.*

El procedimiento diseñado para evolucionar ontologías se apoya en Wikipedia como proveedor de información. La utilización de éste como único repositorio de información aumenta las posibilidades de que existan términos de entre los extraídos que no se encuentren y que, por lo tanto, no puedan ser utilizados para evolucionar la ontología. Sin embargo, la disponibilidad de diferentes repositorios de información permitiría disminuir esta probabilidad haciendo mucho más efectiva la búsqueda de información. Esta sería una de las líneas de investigación futuras, incorporar otros repositorios de información para que el proceso de evolución de ontologías no dependa solamente de Wikipedia. En caso de incluirse nuevos repositorios, y si se diera la circunstancia de tratar con dominios muy específicos (por ejemplo, medicina, biología o botánica), entonces sería incluso posible desechar directamente Wikipedia como proveedor de información.

- *Utilización de métricas que permitan validar la calidad ontológica como medida de evaluación en el proceso de evolución de ontologías.*

Las métricas de evaluación de ontologías proporcionan un conjunto de herramientas que aportan confianza a la hora de compartir y reutilizar las ontologías (Brank et al., 2005). Actualmente, existen diferentes tipos de métricas, que pueden ser utilizadas para validar propiedades muy interesantes, relacionadas con el modelo de representación de conocimiento definido en el enfoque de anotación semántica propuesto. Entre las distintas métricas existentes se destacan: número de relaciones entre clases, número de anotaciones por clase, número de

clases descendientes o ascendientes por clase, profundidad de la taxonomía, entre otras (Duque-Ramos et al., 2011).

En esta línea de investigación, se propone realizar un estudio del estado del arte de las diferentes métricas de evaluación disponibles. El objeto de este estudio es seleccionar qué métricas pueden ser aplicables en el modelo ontológico formal desarrollado. El fin último sería, por tanto, la incorporación de este tipo de métricas como medidas de evaluación de la ontología resultante del proceso de evolución de ontologías. Así, se conseguiría pulir el proceso de enriquecimiento para obtener ontologías que puedan ser compartidas y reutilizables en otras aplicaciones y dominios tecnológicos.

- *Integración de un sistema de recomendación basado en minería de opiniones.*

La aplicación de la metodología de anotación para el cálculo de la similitud abre un amplio espectro de líneas de investigación innovadoras que se basan en la integración de tecnologías basadas en el procesamiento de información. Entre estas nuevas líneas podría destacarse la incorporación de tecnologías de minería de opiniones. La utilización de estas tecnologías facilitaría el desarrollo de sistemas de recomendación guiados por las preferencias de los usuarios. Empleando estas tecnologías sería posible definir perfiles que representasen las preferencias de los usuarios. Estos perfiles proporcionarían restricciones en los procesos de búsqueda con el objetivo de obtener resultados más personalizados y estrechamente relacionados con las preferencias de los usuarios.

En resumen, tanto la metodología de anotación semántica como la aplicación de la semántica en el cálculo de la similitud que se proponen en esta tesis doctoral, se pueden enmarcar en el campo de la Ingeniería Ontológica, los Sistemas de Recuperación de Información y los Sistemas de Anotación Semántica. Por un lado, la metodología de anotación que se presenta se basa en una ontología para representar el dominio. A partir de esta ontología del dominio se generan los metadatos con los que se anota la información textual. Por otro lado, en esta tesis también se propone una metodología de evolución de ontologías completamente automática. Por último, se incluye un motor de búsqueda semántico para la recuperación de información anotada semánticamente.

Si bien puede parecer de reducida complejidad, la definición de la metodología de anotación semántica no sólo ha consistido en la concepción de un mecanismo que facilite la incorporación de metadatos que hagan entendibles los recursos a las máquinas, sino que ha sido necesario llevar a cabo un exhaustivo análisis de los diversos problemas que presentan las aproximaciones propuestas hasta el momento. Entre las limitaciones más importantes de las que adolecen los sistemas que constituyen el estado del arte se encuentran, por ejemplo, la actualización, modificación o eliminación de los recursos, la evolución de modelo ontológico, o la gestión de ciclo de vida de las anotaciones semánticas, entre otros.

La metodología de anotación semántica propuesta en este trabajo da respuesta a estos problemas mediante el planteamiento de un conjunto de módulos reutilizables y compartibles que definen las funciones fundamentales del sistema de anotación presentado en esta tesis.

Capítulo VII. CONTRIBUCIONES CIENTÍFICAS.

VII.1. PUBLICACIONES JCR.

1. Miguel Ángel Rodríguez-García, Rafael Valencia-García, Francisco García-Sánchez, José Javier Samper-Zapater (2014) Creating a semantically-enhanced cloud services environment through ontology evolution. Future Generations in Computer Systems. vol. 32, pp. 295–306.
2. Miguel Ángel Rodríguez-García, Rafael Valencia-García, Francisco García-Sánchez y José Javier Samper-Zapater (2014) Ontology-based annotation and retrieval of services in the Cloud. Knowledge-Based Systems. vol. 56, pp. 15-25.
3. Carlos García-Moreno, Yolanda Hernández-González, Miguel Ángel Rodríguez-García, Jose Antonio Miñarro-Giménez, Valencia-García R. y Angela Almela (2013) A Semantic based Platform for Research and Development Projects Management in the ICT Domain. Journal of Universal Computer Science. vol. 19, no. 13 , pp 1914-1939.

VII.2. CONGRESOS INTERNACIONALES.

1. Miguel Angel Rodríguez-García, Rafael Valencia-García y Francisco García-Sánchez (2012) An Ontology Evolution-Based Framework for Semantic Information Retrieval. In Proceedings of the 2nd Workshop on Industrial and Business Applications of Semantic Web Technologies (INBAST 2012), Roma Italia. Septiembre 11.
2. Miguel Ángel Rodríguez-García, Rafael Valencia-García, Francisco García-Sánchez, José Javier Samper-Zapater y Isidoro Gil-Leiva (2013) Semantic annotation and retrieval of services in the Cloud. In Proc DCAI 2013.

Capítulo VIII. RESUMEN EXTENDIDO EN INGLÉS / EXTENDED SUMMARY IN ENGLISH

VIII.1. INTRODUCTION

The Semantic Web is an extension of the current Web in which information is given well-defined meaning, better enabling computers and people to work in cooperation. It is based on the idea of having data on the Web defined and linked such that it can be used for more effective discovery, automation, integration, and reuse across various applications (Berners-Lee et al., 2001).

The Semantic Web relies heavily on the formal ontologies that structure its underlying data for comprehensive and transportable machine understanding (Maedche & Staab, 2001). An ontology can be defined as "a formal and explicit specification of a shared conceptualization" (Studer et al., 1998). Ontologies provide a formal, structured knowledge representation, with the advantage of being reusable and shareable. Knowledge in ontologies is mainly formalized using five kinds of components: classes, relations, attributes, axioms and instances. Nowadays, it is well established that ontologies are needed for the Semantic Web, knowledge management and annotating specific information.

Semantic annotation provides a more precise description of the knowledge contained in a document and its semantics in the domain (Steels, 1992). It helps bridge the ambiguity of natural language and its computational representation in a formal language through ontologies. The process basically consists in inserting tags in a document. These tags represent links between text fragments and ontological elements (attributes, concepts, relationships and instances). As a result of this process, documents are created that can be processed not only by humans but also by automated agents. The addition of metadata enables an easier and faster information retrieval process minimizing the response time and improving its precision.

The main goal of this thesis is the development of an innovative methodology for the extraction and retrieval of information based on semantic technologies. This

methodology makes use of ontologies as knowledge base. The methodology promotes the generation of metadata by analysing natural language textual resources and makes use of this metadata to add semantic annotations to these information resources. Besides, it is also desired to explore the application of the semantic annotation methodology in a number of domains for different purposes. In particular, among the applications developed by leveraging this methodology, it is possible to highlight the application devoted to calculate the semantic similarity between various types of entities stored in a data repository.

This summary is organised as follows. First, the state-of-the-art of the different technologies leveraged in this work is analysed. Then, the main components of the proposed annotation methodology along with its overall architecture are described. In Section 4, the process followed to validate the semantic annotation system is presented. The application of the annotation methodology to similarity calculation is analysed in Section 5 and its validation process is described in Section 6. Finally, conclusions and future work are put forward in Section 7.

VIII.2. STATE OF THE ART

In this section, the state-of-the-art of the main technologies involved in this research is thoroughly revised. The Semantic Web, ontologies and their evolution, natural language processing techniques, information extraction and retrieval, and semantic annotation mechanisms are among the topics covered here.

VIII.2.1. THE SEMANTIC WEB AND ONTOLOGIES

The information contained in Web pages was originally designed to be human-readable. As the Web grows in both size and complexity, there is an increasing need for automating some of the time consuming tasks related to Web content processing and management. In 2001, Berners-Lee and his colleagues defined the Semantic Web as an extension of the current Web, in which information is given well-defined meaning, better enabling computers and people to work in cooperation(Berners-Lee et al., 2001). The Semantic Web vision is based on the

idea of explicitly providing the knowledge behind each Web resource in a manner that is machine processable. Ontologies (Studer et al., 1998) constitute the standard knowledge representation mechanism for the Semantic Web. During the last few years, a number of approaches have appeared with the purpose of structuring non-structured and semi-structured data sources. In particular, some approaches try to automatically associate data and semantic notes with HTML documents (Yang, 2009). Other approaches focus on giving structure to semi-structured documents (Park et al., 2008). There are also approaches that attempt to automatically create an ontology from unstructured HTML documents (Du et al., 2009).

Ontologies can be used to structure information. The formal semantics underlying ontology languages enables the automatic processing of the information in ontologies and allows the use of semantic reasoners to infer new knowledge. In this research work, an ontology is seen as "a formal and explicit specification of a shared conceptualisation" (Studer et al., 1998). Ontologies provide a formal, structured knowledge representation, with the advantage of being reusable and shareable. They also provide a common vocabulary for a domain and define, with different levels of formality, the meaning of the terms and the relations between them. Knowledge in ontologies is mainly formalized using five kinds of components: classes, relations, functions, axioms, and instances (Gruber, 1993). Classes in the ontology are usually organized into taxonomies. Sometimes the definition of ontologies has been diluted, in the sense that taxonomies are considered to be full ontologies (Studer et al., 1998). In this thesis, the second version of OWL (Web Ontology Language), i.e. OWL 2, which is the de facto Semantic Web standard language, has been used to represent the knowledge.

Creating and populating ontologies manually is a very time-consuming and labour-intensive task. Several methodologies have been designed in order to assist in building ontologies (Noy & Musen, 2003); (Staab et al., 2001); (Wahl & Sindre, 2009). However, in order to overcome the bottleneck created by manually constructing ontologies (Shamsfard & Barforoush, 2004), several (semi-)automatic approaches are being researched. In this regard, it is necessary to differentiate between ontology learning (Maedche & Staab, 2004) and ontology population (Ruiz-Martínez et al., 2012). Ontology learning is about acquiring new knowledge

in the form of concepts and relations to be added to an ontological model. As a consequence of this process, the inner structure of the ontology is modified. The goal of Ontology Population, on the other hand, is to extract and classify instances of the concepts and relations defined in an ontology from a particular data source. The process of Ontology Population does not change the structure of an ontology; what changes is the instances of concepts and relations in the domain. Instantiating ontologies with new knowledge is a relevant step towards the provision of valuable ontology-based knowledge services.

VIII.2.2. ONTOLOGY EVOLUTION

Ontology evolution can be defined as the timely adaptation of an ontology to changed business requirements, as well as the consistent management/propagation of these changes to dependent elements (Stojanovic, 2004). In fact, the evolution and modification in one part of the domain ontology can produce inconsistencies in the whole ontology (Stojanovic et al., 2002). In this context, it is important to distinguish between ontology evolution and ontology versioning. While ontology evolution allows access to all data only through the newest ontology, the ontology versioning allows access to data through different versions of the ontology (Stojanovic, 2004). The use of ontology versioning in platforms is gaining success and some platforms such as SHOE (Luke et al., 1997) and KAON (Bozsak et al., 2002) support multiple versions of ontologies and enable to declare whether the new version is backward-compatible with an old version.

Ontologies evolve continuously during their life cycle to adapt to new requirements and necessities. Ontology-based information retrieval systems use semantic annotations that are also regularly updated to reflect new points of view. In order to provide a general solution and to minimize the users' effort in the ontology enriching process, a methodology for extracting terms and evolve the domain ontology from Wikipedia is proposed in this thesis. The framework presented here combines an ontology-based information retrieval system with an ontology evolution approach in such a way that it simplifies the tasks of updating concepts and relations in domain ontologies.

VIII.2.3. NATURAL LANGUAGE PROCESSING

Natural Language Processing (NLP) can be defined as "a theoretically motivated range of computational techniques for analysing and representing naturally occurring texts at one or more levels of linguistic analysis for the purpose of achieving human-like language processing for a range of tasks or applications" (Liddy, 2001). According to Feldman (1999), people extract meaning from text or spoken language on at least seven levels: phonological, morphological, syntactic, semantic, discourse and pragmatic. In order to understand Natural Language Processing, it is important to be able to distinguish among these, since not all NLP systems use every level. Moreover, for each linguistic analysis level, NLP provides a set of tools in order to analyse the language. Concretely, the semantic annotation methodology subject of this thesis makes use of NLP techniques in a term extraction module. The goal of this module is to extract terms from natural texts in order to elaborate a list of the most important terms on the analysed text.

VIII.2.4. INFORMATION EXTRACTION AND RETRIEVAL

A continuously growing amount of natural language information is available in electronic format online. The need for intelligently processing this information makes information extraction (IE) a fundamental task given that it allows to locate and in some way identify the specific pieces of data needed from each document (Califf & Mooney, 1999). Computational linguistic techniques and theories are playing a strong role in this emerging technology, which should not be confused with the more mature technology of Information Retrieval (IR), which selects a relevant subset of documents from a larger set. IE extracts information from the actual text of documents (Cowie & Lehnert, 1996). According to Cunninghan (2005) an IR system finds relevant texts and presents them to the user; an IE application analyses texts and presents only the specific information from them that the user is interested in.

The use of ontologies may help overcome the limitations of traditional methods in information extraction and retrieval. In this thesis, an ontology-based methodology for resources retrieval is proposed in order to take advantage from

ontologies in providing a more powerful search service. The framework presented here combines an ontology-based information retrieval system and a term extraction module. The information retrieval system helps users to obtain better results by taking advantage of the knowledge representation in the form of ontologies. On the other hand, the term extraction system is used to update the domain through the lexical and morphosyntactic analysis of the stored resources. Both components enable an enhanced information search in domain ontologies.

VIII.2.5. SEMANTIC ANNOTATION

Leech (1997) defined annotation as "the practise of adding interpretative linguistic, information to a corpus". More concretely, annotation or tagging is a process that permits the mapping of concepts, relationships, comments, or descriptions to a document or to a fragment in a text. In general, annotations can be seen as metadata that are associated with a particular text fragment in a document or another piece of information. Semantic annotations help to bridge the ambiguity of the natural language and their computational representation in a formal language through ontologies. This process consists of inserting tags in a document that represents ontological elements (concepts, relationships, attributes and instances) to text fragments, thus allowing the creation of documents that can be processed not only by humans but also by automated agents (Kiyavitskaya et al., 2005). Although several systems for ontology-based annotation have been proposed over the last decade, there is no standard approach for semantic annotation (Uren et al., 2006). The following table (see Table VIII.1) shows some of the best known semantic annotation systems. The classification is based on that presented in (Uren et al., 2006).

Table VIII.1 summary of the main semantic annotation approaches

Work	Standard format	Ontology support	Document Formats	Document evolution	Annotation storage	Automation	Ontology evolution
Armadillo (Chapman et al., 2005)	RDF	Multiple ontologies	Web and text documents	Yes	Semantic Model	Automatic	No
CERNO (Kiyavitskaya et al., 2009)	RDF OWL	No multiple ontologies	Structured text documents	No	Semantic Model	Semiautomatic	No
CREAM (Handschuh & Staab, 2003)	RDF	Multiple ontologies	Web pages	Yes	Semantic Model	Automatic	Yes
S-CREAM (Handschuh et al., 2002)	RDF	No multiple ontologies	Web pages	No	Semantic Model	Semiautomatic	Yes
KIM (Popov et al., 2003)	RDF (OWL extensible)	Multiple ontologies	Text documents	Yes	Semantic Model	Automatic	Yes
EVONTO (Tissaoui et al., 2011)	RDF OWL	No multiple ontologies	Text documents	Yes	Semantic Model	Automatic	Yes
GoNTogle (Bikakis et al., 2010)	RDF OWL	Multiple ontologies	Text documents	No	Semantic Model	Manual and automatic	No
MnM (Vargas-Vera et al., 2002)	RDF	No multiple ontologies	Documents	No	Semantic Model	Semiautomatic	No
Our approach	OWL	Multiple ontologies	Web and text documents	Yes	Semantic Model	Manual and automatic	Yes

The parameters selected for their representation in the table are the following: 'standard format', 'ontology support', 'support of heterogeneous document format', 'document evolution', 'annotation storage model', 'automation' and 'ontology evolution'. These parameters are explained in detail as follows, and the differences between the proposed annotation method and the current state-of-the-art mechanisms are also highlighted.

Standard formats

Most of the annotation tools developed up to 2006 use the RDF format for the annotations, such as Armadillo (Chapman et al., 2005), MnM (Vargas-Vera et al., 2002), S-CREAM (Handschuh et al., 2002) and CREAM (Handschuh & Staab, 2003) or RDFa for embedding rich metadata within web documents. The tendency in the last few years, however, has been to utilize OWL ontologies as the annotation format, as described in CERNO (Kiyavitskaya et al., 2009), EVONTO (Tissaoui et al., 2011), GoNTogle (Giannopoulos et al., 2010), and KIM (Popov et al., 2003). Finally, there exist other approaches, such as the work presented in (Zeni et al., 2007), that make use of alternative annotation schemas based on text files generated by the TXL programming language (Cordy et al., 1988).

In this thesis, the second version of OWL, OWL 2 (Grau et al., 2008), has been used. Its formal model supports a number of important automatic DL inference services. These inference services can be provided by different DL reasoners including HermiT, Pellet2, Fact++ or Racer (Sirin & Parsia, 2004).

Ontology support (multiple ontologies)

One property that is often desired in the scope of semantic annotation is multiple ontologies support. The tools that offer such a feature can support several ontologies in different domains. The main advantage of semantic annotation systems supporting multiple ontologies is that they can cover different domains. While KIM (Popov et al., 2003), CREAM (Handschuh & Staab, 2003), GoNTogle (Bikakis et al., 2010), or Armadillo (Chapman et al., 2005) support the use of multiple ontologies, CERNO (Kiyavitskaya et al., 2009), S-CREAM (Handschuh et al., 2002), MnM (Vargas-Vera et al., 2002) or EVONTO (Tissaoui et al., 2011) do not include this feature.

The methodology proposed in this thesis provides support for this feature. Besides enabling the system to deal with different domains, supporting multiple

ontologies does also allow the system to boost its performance since it can manage various smaller ontologies with limited scope.

Support for heterogeneous document formats

The main approaches for semantic annotation focus on dealing with texts available on the Web, and so the documents being annotated are in Web-native formats such as HTML or XML. For example, S-CREAM (Handschuh et al., 2002) and CREAM (Handschuh & Staab, 2003), among others, provide support for these document formats. However, there exist other approaches, such as KIM (Popov et al., 2003), MnM (Vargas-Vera et al., 2002) and Armadillo (Chapman et al., 2005), that permit to analyse and annotate several types of documents in different formats, and others, such as CERNO (Kiyavitskaya et al., 2009), GoNTogle (Bikakis et al., 2010) and EVONTO (Tissaoui et al., 2011), are specifically designed for dealing with textual document formats.

The system presented here provides support for different types of document formats, including the most important textual document formats and Web-native formats.

Document evolution (document and annotation consistency)

One of the main problems of annotation systems is the preservation of the consistency between the annotated documents and the ontology repository in which the annotations are saved. Here, consistency is seen from a textual point of view, i.e., it refers to the maintenance of correct pointers from the annotations to the linguistic expressions in the text. Specifically, all changes (such as, for example, adding or deleting information) in an annotated document have to be reflected in the annotation system's repositories. Otherwise, if the system does not notice the modification that has taken place in a part of a document, there will be an inconsistency in the annotations in that document. Moreover, if the annotation system provides support for multiple ontologies, it must maintain the coherence and consistence of the annotations between the changing documents and all the ontologies.

Almost all the current semantic annotation methods provide support for document evolution. However, while Armadillo (Chapman et al., 2005), CREAM (Handschuh & Staab, 2003), KIM (Popov et al., 2003) and EVONTO (Tissaoui et al., 2011) update the annotations if a change is made in one or more documents, S-

CREAM (Handschuh et al., 2002), GoNTogle (Bikakis et al., 2010), MnM (Vargas-Vera et al., 2002), and CERNO (Kiyavitskaya et al., 2009) do not. The solution proposed in this thesis can cope with any kind of change in the annotated documents.

Annotation storage model

Two means of storing annotations can be distinguished in the most recent semantic annotation systems (Devedzic & Gaseviceds, 2009): (1) the semantic model approach and (2) the document-centric approach. The first approach, based on a semantic Web-guided model, establishes that the annotations should be stored separately from the original document. The latter approach, which is based on a document-centric (word processor) vision, states that annotations should be stored as an integral part of the document.

As can be seen in Table VIII.1, the vast majority of the annotation methods that have been analysed in this study make use of semantic Web models to store the annotations. Resource annotations are consequently introduced into data containers such as databases and are thus isolated from the source documents. The system presented here also leverages the benefits of the semantic model approach.

Automation

This parameter helps classify the annotation methods into three different categories (Oren et al., 2006): manual (performed by one or more people), semiautomatic (based on automatic suggestions) or fully automatic (based on computer annotation processes).

Both the semi-automated and fully automated approaches are used more frequently because manual annotation is considered to be a time-consuming task (Ciravegna et al., 2002). The tendency in the current ontology-based annotation systems is therefore to provide a semi-automated tool such as CERNO (Kiyavitskaya et al., 2009), GoNTogle (Bikakis et al., 2010) and S-CREAM (Handschuh et al., 2002), or a fully automated tool such as Armadillo (Chapman et al., 2005), CREAM (Handschuh & Staab, 2003), MnM (Vargas-Vera et al., 2002), KIM (Popov et al., 2003) and EVONTO (Tissaoui et al., 2011).

The integration of fully automated knowledge extraction technologies into semantic annotation approaches is necessary when huge document collections

have to be analysed and annotated. In this research, a fully automated method, which also enables the manual manipulation and verification of the annotations, is proposed.

Ontology evolution

Ontology evolution can be defined as the timely adaptation of an ontology and consistent propagation of changes to dependent artefacts (Stojanovic et al., 2002). More concretely, it refers to the process of changing the ontologies over time by, for example, adding new classes or instances, modifying classes and instances or removing knowledge. The semantic annotation approaches that support ontology evolution must ensure the consistency of the annotations against the ontologies that are being modified.

The enablement of ontology evolution entails that the annotation system must maintain the consistence and coherence between the annotations and ontologies. EVONTO (Tissaoui et al., 2011), KIM (Popov et al., 2003), S-CREAM (Handschuh et al., 2002) and CREAM (Handschuh & Staab, 2003) implement an ontology evolution approach. On the contrary, CERNO (Kiyavitskaya et al., 2009), GoNTogle (Bikakis et al., 2010), MnM (Vargas-Vera et al., 2002) and Armadillo (Chapman et al., 2005) do not cover this feature.

The system presented in this thesis provides full support for ontology evolution from Wikipedia.

In conclusion, the majority of the approaches included in this study are fully automatic but do not provide support for either multiple ontologies or ontology evolution. Like other approaches such as Armadillo (Chapman et al., 2005), CREAM (Handschuh & Staab, 2003), KIM (Popov et al., 2003), the method proposed here is able to manage multiple ontologies. However, it is the only one that is able to cope with both features, multiple ontologies and ontology evolution. Furthermore, most of these approaches are focused on resolving a specific annotation problem in a concrete and structured application domain (see, for example, Armadillo (Chapman et al., 2005), S-CREAM (Handschuh et al., 2002), CERNO (Kiyavitskaya et al., 2009) and KIM (Popov et al., 2003)), while the proposed semantic annotation tool, along with others such as EVONTO (Tissaoui et al., 2011) and CREAM (Handschuh & Staab, 2003), implements a more general perspective based on resolving the semantic annotation problem from unstructured documents.

VIII.3. METHODOLOGY FOR SEMANTIC ANNOTATION

The architecture of the proposed approach is shown in Figure VIII.1. The system is composed of five main modules: the semantic representation and annotation module (1), the semantic indexing module (2), the term extraction module (3), the ontology evolution module (4), and the semantic search engine (5). In a nutshell the system works as follows. First, using a predefined ontology model, a set of annotations based on the concepts and the "`rdfs:label`" properties of the ontology are built. During the semantic annotation process, these annotations are used to identify linguistic expressions, which are associated to concepts defined on the ontology model. Then, the annotations are enriched using the taxonomic relations from the ontology. For that purpose, from these annotations, a semantic index is created using the classic vector space model. At the same time, the term extractor obtains the terms appearing in the texts that are not present in the domain ontologies. Then, the ontology evolution module checks whether the most important new terms previously gathered can be added to the domain ontologies. Finally, a semantic search engine permits to retrieve the matching resources from keyword-based searches. Next, these components are described in detail.

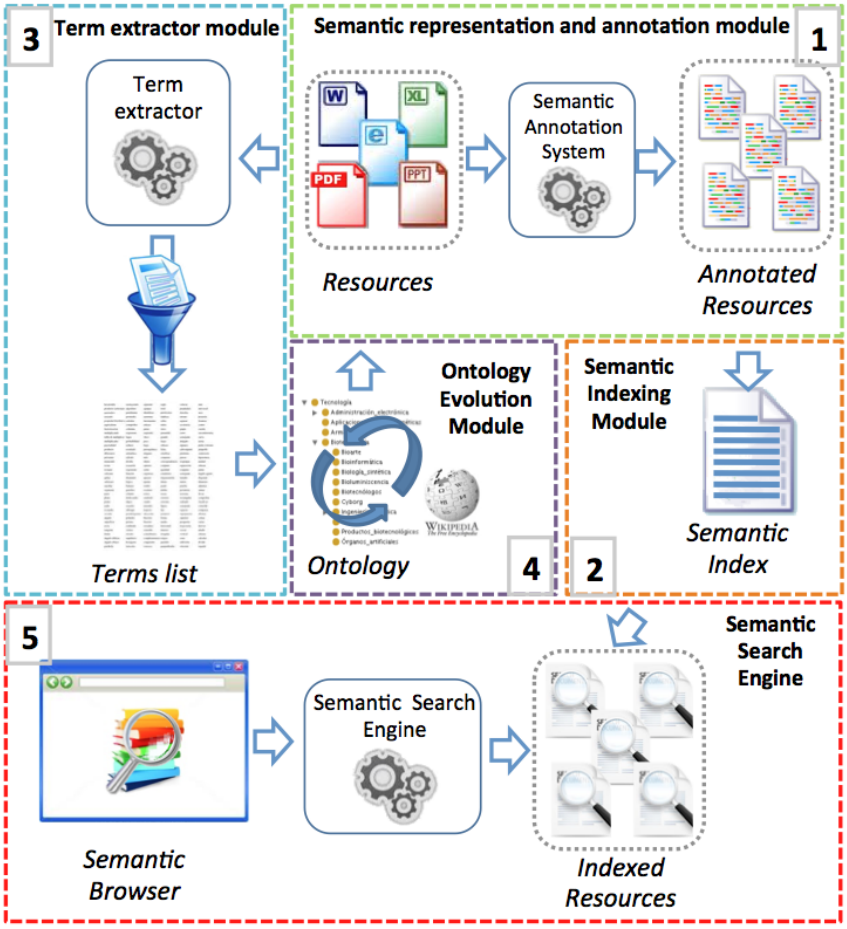


Figure VIII.1 Platform architecture

VIII.3.1. SEMANTIC REPRESENTATION AND ANNOTATION MODULE (1)

This module is responsible for providing the annotation function, that is, inserting interpretative linguistic information, which is obtained from the ontology model, in a corpus. For this, it receives both the domain ontologies and the natural-language, non-structured or semi-structured texts as inputs. The linguistic information is obtained from the ontology model, where each concept is expressed in terms of a set of synonyms. Thus, a concept is defined as a set of terms related by a meaning (Reiterer et al., 2010). In order to represent these synonyms in our ontology model, different kinds of annotations have been defined: "`rdfs:comment`", "`rdfs:label`", "`preferredTerm`" and "`URIresource`". For implementing this module, the GATE framework and OWL API have been employed. GATE is an infrastructure for developing and deploying software components that process human language. OWL API, on the other hand, is a

reference implementation for creating, manipulating and serialising OWL ontologies. As a result of this first phase, a set of annotations representing the syntactic structure of the text is obtained.

During the semantic annotation phase, texts are annotated with the classes of the domain ontologies by following the process described next. First, a Named-Entity Recognition (NER) tool is used in order to recognize the most important linguistic expressions based on gazetteers, which represents a dictionary of words. Then, for each linguistic expression the system tries to determine whether such expression is a class of any of the domain ontologies. If so, the linguistic expression is related with the URI (Uniform Resource Identifier) of the class or instance of such domain ontology.

VIII.3.2. SEMANTIC INDEXING MODULE (2)

In this module, the system retrieves all the annotated knowledge from the previous module and tries to create fully-filled annotations with this knowledge. Each annotation of each document is stored in a database and has a weight assigned, which reflects how relevant the ontological entity is for the document meaning. Weights are calculated by using the TF-IDF algorithm (Salton & McGill, 1983), which uses the following equation (see equation (II.1)).

$$\mathrm{tf} * \mathrm{idf} = tf(t,d) * idf(t,D) \qquad \textbf{(VIII.1)}$$

where '`tf(t,d)`' represents the Term Frequency function (see equation (II.2)). This function obtains the frequency of occurrence of a term '`t`' in a document '`d`'. On the other hand '`idf(t,D)`' represents the Inverse Document Frequency (see equation (II.3)). This function obtains the frequency of occurrence of this term '`t`' in the document '`d`' that belong to the set of annotated documents '`D`'.

$$\mathrm{tf}(\mathrm{t},\mathrm{d}) = \frac{t}{d} \qquad \text{(VIII.2)}$$

$$\mathrm{idf}(\mathrm{t},\mathrm{D}) = log\ \frac{|D|}{|\{d \in D : t\ \in d\}|} \qquad \text{(VIII.3)}$$

In this scenario, the content descriptions are the documents to be analysed. For each description, a semantic index is calculated based on the semantic similarity metrics of taxonomic relations, which are concerned with the interrelation between two concepts in a context (Rada et al., 1989). Each description is represented as a vector in which each dimension corresponds to a separate ontological concept of the domain ontology. The value of each ontological concept dimension is calculated as follows (see equation (II.4)).

$$tf - idf_{extended} = \sum_{j=1}^{n} \frac{tf - idf_{j,d}}{e^{dist(i,j)}} \qquad \text{(VIII.4)}$$

where '`dist(i,j)`' is the semantic distance between the concept '`i`' and concept '`j`' in the domain ontology. This distance is calculated using the shortest path algorithm (Dijkstra, 1959) developed to graphs. This algorithm uses the taxonomic relationships of concepts in the domain ontology in order to enrich each identified concept during annotation process with all concepts that are related taxonomically over a pre-established minimal distance. So, the distance between a concept and itself is 0, the distance between a concept and its taxonomic parent or child is 1 and so on.

VIII.3.3. TERM EXTRACTOR MODULE (3)

Through this module, the most significant terms from the text documents are identified. This module is based on previous works of our research group (Ochoa et al., 2011). It is assumed that there exist both multiword and single word terms. The method employed is composed by two sequential phases: (i) pattern identification and depuration and (ii) pattern optimization. As a result of these phases a list of the most important terms appearing in the texts that are not part of the domain ontologies, is taken as the input of the ontology evolution module.

VIII.3.4. ONTOLOGY EVOLUTION MODULE (4)

The main objective of this module is to maintain and evolve ontologies using the information available in Wikipedia. The terms list gathered by the term extraction module is used to enrich, enhance and increase the knowledge represented on the domain ontologies. Wikipedia is a free encyclopaedia where thousands of concepts are classified in a taxonomy by taking into account Wikipedia's categories. The main idea of this method is to keep domain ontologies up-to-date by leveraging the structure of Wikipedia, that is, the assignment of articles to categories, the subcategory relation and the cross-language relations linking equivalent articles across languages. The Wikipedia structure is an important and necessary condition since it allows to establish groups of terms defining the semantic concepts.

Each relevant term in the list produced by the term extractor module (i.e. terms not currently present in the ontologies) is looked up in Wikipedia. If a Wikipedia article or category is found that matches the term, a new concept is created containing all the term's synonyms and subsequently added into the domain ontology by using the algorithm described in Figure VIII.2. This algorithm has been implemented by using the OWL API framework and the Java Wikipedia API (Bliki engine), which is a parser library for converting Wikipedia wikitext notation into HTML. The module has been designed to allow the access to online wikitext using the Hypertext Transfer Protocol (HTTP). Therefore, the extracted content is always up-to-date since it is gathered from the online version of Wikipedia. Additionally, this API permits to collect outdated wikitext content if desired.

Figure VIII.2 shows the pseudocode of the algorithm designed to solve the ontology evolution problem. The input of the algorithm is the list of terms produced by the term extraction module. Each term in the list represents a candidate concept that can be used to enrich the ontology knowledge. The proposed evolution algorithm has been designed to find a path in the Wikipedia hierarchy. This path assists the system to generate relationships between the candidate concepts and the elements already available in the ontology. The algorithm attempts to find the shortest path between each term in the list and the ontology concepts based on the Wikipedia categories. An ontology class is defined during the search process for each category that is obtained from the Wikipedia hierarchy. These classes are built by extracting the synonymous terms from Wikipedia using the "WhatLinksHere" tool. This tool obtains the references of each category and article for each given term. These references are used to define the concept. In particular, the concept is comprised by the list of redirections that the "WhatLinksHere" tool provides. If the tool does not return any redirection for a given term, statistical techniques based on natural language processing are used to analyse and filter the links. Through this process, a number of terms are retrieved, which are used to define the concept. Finally, if a common category is found, a path between the term of the list and the concept of the ontology is built. For each Wikipedia category located in the path, a class is defined on the ontology. It is worth pointing out that it can be the case that the process ends without finding a connecting path. In such situations, the searched term is stored in a part of the ontology known as "hodgepodge". This part of the ontology contains all these terms that were candidates to be concepts but the evolution algorithm has not been able to find a path between the term and an ontology concept.

```
   //The ontology is analyzed and the ontology concepts (classes) are obtained from
the 'ontology' resource
   List<OWLClass> nodesOntology = ontology.getNodesOntology();
   //The list of terms is created by the term extractor module
   List<String>termList = getExtractorList();
   while(termList.hasMoreElements()){
      //The next term is obtained
      term = termList.nextElement(i);
      //For each extracted term, the system first checks whether the term is already
in the ontology
      if(!nodesOntology.contains(term)){
         for(nodesOntology.hasMoreElements()){
            //The node concept is obtained
            nodeConcept = node.getConcept();
            //The first concept is built from Wikipedia
            firstConcept = buildConceptFromWikiipedia(nodeConcept);
            if(firstConcept==null){
               //If there is not any concept in the Wikipedia, then it has to continue
with the next node
               continue;
            }else{
               //The second concept is built from Wikipedia
               secondConcept = buildConceptFromWikiipedia (term);
               if(secondConcept==null){
               //If there is not any term in the Wikipedia, then it has to continue to the
next term
                  continue;
               }else{
                  joiningConceptsFromWikipedia(firstConcept,secondConcept);
               }
            }
         }
      }
   }
```

Figure VIII.2 Ontology evolution algorithm

VIII.3.5. SEMANTIC SEARCH ENGINE (5)

This module is in charge of finding semantic concepts in different kinds of textual resources from a keyword-based query. This process takes advantage of the semantic content and annotations previously gathered by the system. First, users introduce a series of keywords and the system identifies which concepts in the domain ontology are referred by them. Then, each resource is represented as a vector in which each dimension corresponds to a separate concept of the domain

ontology. Then, the semantic search engine calculates a similarity (Xu et al., 2011) value between the query 'V1' and each semantic index 'V2'. In order to do that, the cosine similarity is used (see equation (II.5)):

$$\cos\theta = \frac{V1 * V2}{\left||V1|\right| \, ||V2||} \qquad \textbf{(VIII.5)}$$

A ranking of the most relevant semantic concepts that are related to the topics referenced in the query is then defined by using the similarity function showed in equation (II.5). The vector space model, 'V1', is calculated by using equation 2 for each service description, and the second vector, 'V2', is the one created from the concepts extracted from the search engine query. The θ symbol is the angle that separates both vectors and represents the similitude grade between two documents.

VIII.4. VALIDATION OF THE SEMANTIC ANNOTATION SYSTEM

The platform described in the previous section has been implemented and tested in two domains, namely, ICT cloud computing and R&D management. Given this, the first step to providing a complete use case has been to develop a domain knowledge base. Then, descriptions in natural language have been selected and inputted to the evolution module in order to adapt the domain ontologies to the particularities of each use case. Additionally, these descriptions have been annotated by the system.

During a first stage, experts are required to input a set of descriptions. Then, these descriptions along with their natural language descriptions were semantically annotated and stored in the ontology repository. The Virtuoso repository (Virtuoso, 2009) has been used to implement the ontology repository. By means of the semantic annotation module, these descriptions were also automatically annotated. Next, their semantic indexes were calculated by using the semantic indexing module.

Once the semantic indexes have been created, the experiments can start. This experimental evaluation aims at elucidating whether the semantic search engine module of the proposed platform is useful. The experts were asked to issue a number of queries concerning ten different topics in each evaluation domain. For each query, a set of resources was manually selected. At the same time, the semantic search engine was asked to perform the same task, in an automatic way. These results were then compared to those produced by the manual selection.

The evaluation of the results has been done through the precision and recall scores. The precision score is obtained by dividing the number of resources suggested automatically by the system that had also been selected in the manual process by the experts, and the total number of resources suggested automatically by the system. The recall score is obtained by dividing the number of resources suggested automatically by the system that have been also selected in the manual process, by the total amount of resources selected manually. Finally, the F-measure is the weighted harmonic mean of the precision and recall scores.

Since neither the software applications nor the ontologies and textual resources used in the experiments are available, the comparison between our proposed method and the different approaches described in Chapter 2 is very difficult. Many of these approaches were either unavailable for download or were designed for specific domains (different from the annotation and retrieval of ICT cloud services and R&D management domains). The results of the respective experiments are not conclusive either, given that (i) the corpora, ontologies and resources used differ significantly in content and size, and (ii) concepts, attributes and relationships are treated differently.

In general terms, annotation methods perform better when dealing with small, very specific domains than with large, general domains. This is mainly due to the fact that the broader the domain, the larger the ontology should be and the more the information is required for the annotation tools to provide accurate annotations. Our approach has achieved a total average precision score of 89%, a recall score of 82% and an F-measure score of 85% in a broad domain such as the ICT-related cloud services field. However, in a more concrete domain such as the R&D management our approach has obtained a total average precision score of 87%, a recall score of 85% and an F-measure score of 86%.

VIII.5. APPLICATION OF SEMANTIC ANNOTATION TO SIMILARITY CALCULATION

The encouraging results obtained during the evaluation of the semantic annotation methodology and, specifically, the semantic search engine module fostered its application in a number of different scenarios. A particularly notorious application is the use of the semantic application methodology in order to calculate the similarity between ontological instances. This measurement represents the similarity between two instances, regardless of their domain or range. The similarity methodology is based on identifying how similar two instances are by comparing their constituent attributes. The architecture of the proposed approach is shown in Figure VIII.3. The system is composed by five main modules: the ontology repository (1), the semantic representation and annotation module (2), the semantic indexing module (3), the similarity calculation module (4), and the semantic inference engine (5).

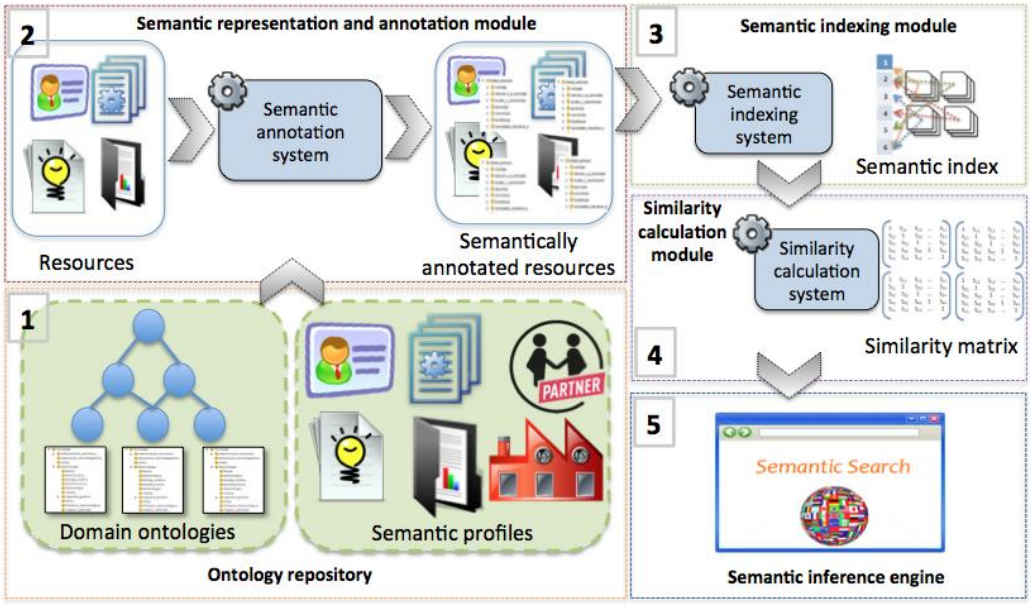


Figure VIII.3 Similarity calculation architecture

The semantic representation and annotation module (2) is sustained by three types of ontologies that are stored in the ontology repository (1): (i) the domain ontology, which is used as the source for tags in the semantic annotation process; (ii) the entities ontology, which is used to model the attributes of the entities to be compared; and (iii) the set of ontologies that are employed to semantically

represent the entities information and the knowledge in RDF format. The representation and annotation format (2) receives information about the entities defined in the information system as input and builds semantic profiles for each entity based on the different ontology models stored in the repository (1). Later, the semantic annotations are retrieved by the semantic indexing module (3) in order to build the semantic indexes. The elaboration of these indexes is based on the methodology described above in Section VII.3.2 and its goal is to enrich the semantic annotations defined with the concepts taxonomically related with the ones annotated. After specifying the semantic indexes, the similarity calculation module (4) builds the similarity matrixes by using the previously generated semantic profiles as information source. A configuration file is used to establish how the instances are to be compared, that is, which attributes are going to be used in the comparison process. A similarity matrix is generated for each comparison. Each numerical element in the matrix represents the similarity score between two instances of each compared entity. Once the matrix has been generated, it can be used by the semantic inference engine (5) to identify matching instances from each entity according to their similarity.

VIII.6. VALIDATION OF THE APPLICATION OF SEMANTIC ANNOTATION TO SIMILARITY CALCULATION

The platform described in the previous section has been implemented and tested in the R&D domain. The validation took place in the scope of an R&D software company. The experiment is focused on evaluating platform's semantic inference engine. The ultimate goal of this experiment is to demonstrate the usefulness of the semantic annotation and recovery for a number of different domains and how easy is to integrate and apply it in those domains.

During the first stage of this validation, managers of the company were requested to input data about some previous software projects of the company and to describe them from a semantic perspective. This information was stored in the ontology repository. The Virtuoso repository (Virtuoso, 2009) was used to implement the ontology repository. Information about several R&D-based software projects was semantically described and introduced into the system.

Information about the staff involved in those projects was also semantically described and stored in the repository. The FOAF- and ResumeRDF-based profiles were inputted into the ontology repository through a Web application. Once the semantic information was manually introduce, the semantic annotations and indexes were calculated automatically from the project profile natural language descriptions. For this purpose, a Web application for managing these DOAP-based profiles in the ontology repository was developed.

In order to test the semantic inference engine of the proposed platform, a number of new R&D-related projects within the context of ICT were proposed. For each project, experts generated the corresponding semantic description, which was subsequently included into the repository. Then, the experts also selected manually the historical projects and the members of the staff that were most related for every given new R&D project. At the same time, the proposed tool performed the same task automatically. The results obtained by the inference engine were then compared to those provided by the human experts.

Again, the evaluation of the results has been done through the precision, recall and F-measure scores. Additionally, the "Mean Average Precision" (MAP) score has been used to not only analyse the entities ranking but also the order in which they are retrieved by the system. The overall results concerning projects seem promising, obtaining a precision score of 77%, a recall score of 67% and an F-measure score of 71%. The results related to team suggestions are also quite positive, obtaining a precision score of 71%, a recall score of 61% and an F-measure score of 66%. Finally, in terms of the MAP measure, the results are 85% for projects suggested and 74% for team suggested. These results, which have been compared against state-of-the-art tools with a satisfactory outcome, are very promising and are aligned with the objectives of this thesis.

VIII.7. CONCLUSIONS AND FUTURE WORK

The impact of new information system integration and management paradigms has been dramatic in most of society's day-to-day life scenarios. The revolution undergone and provoked by Information Technologies (IT) over the last few years was unforeseeable, particularly the transformation of the Web from a mere

repository of information to a vehicle for business transactions. Emerging approaches such as semantic technologies are changing the current business landscape, as are domains such as e-Science, education and e-Health, particularly as regards their adaptability and interoperability, and the incorporation of cognitive and reasoning capabilities has resulted from both of these characteristics. The semantic annotation and retrieval of text documents and Web resources is a challenging task and it addresses the general issue of making computers aware of the content of informational resources so as they can be of better assistance for users.

In this thesis, a semantic platform for text resources annotation and retrieval from their source documents has been proposed. The system presented here automatically annotates natural language documents, which may be available in a number of formats such as XML, HTML or PDF. The proposed platform has additionally been implemented, considering multi-ontology environments (with OWL 2 ontologies) in order to be able to cope with several domains. It also supports the evolution of the source documents, thus maintaining the coherence between the natural language descriptions and the annotations, which are stored using a semantic Web-based model.

A new methodology is presented that comprises three commonly used but enhanced stages in the scope of semantic search processes. The first phase involves the automatic semantic annotation of natural language texts containing the description of the cloud services available. This annotation process, which is an improved version of the methodology presented in (Valencia-García et al., 2008), results in the source documents being annotated with the classes of domain ontologies. During the second stage, a semantic index is created from the previously retrieved semantic annotations. The classic vector space model has been used in order to build the semantic index. This step is based on an adaptation of the very well-known TF-IDF algorithm as described in (Castells et al., 2007). Finally, the third phase refers to the semantic search itself, during which keyword-based queries are modelled in the form of a vector and compared with the original textual resources' vectors by using cosine similarity. The proposed platform has been evaluated two domains, namely, ICT-related cloud computing and R&D management, with very promising results.

Using the semantic annotation methodology a new application of similarity calculation has been proposed. The methodology been conceived with the aim of assisting organizations in the funding management of R&D projects by taking it to the knowledge level. The platform keeps track of the semantic-based description of R&D projects, proposals, ideas and worker resumes. Natural language processing tools are used to facilitate the generation of such semantic profiles, annotations and indexes from natural language texts. Once the system has been fed up with the semantic content, the semantic inference engine module leverages such formal content to perform general queries from the available information, producing precise and accurate results that can help managers in the decision-making process.

Several issues remain open for future work. Currently, the input to the semantic search engine is a keyword-based query. This fact limits the expressiveness of the query and, hence, the accuracy of the results. It is planned to provide either natural language interfaces or controlled vocabulary-based interfaces that assist layman end users to compose complex queries in those more expressive semantic languages. Then, the semantic search engine would use the natural language description provided by users, which is actually mapped to semantic languages, to achieve more precise searches. The approach presented here also has a number of drawbacks that should be dealt with. Since the proposed method provides support for more than one domain ontology and natural language is inherently ambiguous, inconsistencies in the semantic annotations may arise. The use of disambiguation techniques could help in providing a solid and workable solution for overcoming these inconsistency and ambiguity issues. A further problem is that of performance. Creating all the semantic annotations and the semantic indexes is very time-consuming and requires a lot of computational resources. It is left for the future to design a parallel algorithm to improve the efficiency of the proposed methods.

We are also currently working on upgrading this system and converting it into a recommendation system in which users could set their preferences and the system would return only those results that are relevant in a particular domain. The ontology evolution approach can be subject for future improvements too. First, it would be very interesting to extend the evolution module so as it has access to

further information repositories beyond Wikipedia. Providing access to more data sources increases the probability of finding the appropriate information thus boosting the evolution process. Second, ontology quality metrics to measure the quality of the evolved ontologies can be applied in order to proof that the resulting ontologies can be shared and reused in others applications or technological domains.

Capítulo IX. CONCLUSIONS AND FUTURE WORK

IX.1. CONCLUSIONS

The work presented in this thesis consists of an innovative methodology for the extraction and retrieval of information based on semantic technologies. This methodology makes use of ontologies as knowledge base. The methodology promotes the generation of metadata by analysing natural language textual resources and makes use of this metadata to add semantic annotations to these information resources. Semantic annotations help bridge the ambiguity of natural language and its computational representation in a formal language through ontologies. The process basically consists in inserting tags in a document. These tags represent links between text fragments and ontological elements (attributes, concepts, relationships and instances). As a result of this process, documents are created that can be processed not only by humans but also by automated agents. The addition of metadata enables an easier and faster information retrieval process minimizing the response time and improving its precision. Moreover, given the characteristics of the proposed semantic annotation methodology, it can give rise to many applications in different domains. Among the applications developed by leveraging this methodology, it is possible to highlight the application devoted to calculate the semantic similarity between various types of entities stored in a data repository.

Although several systems for ontology-based annotation have been proposed over the last decade, there is no standard approach for semantic annotation. Most of the current approaches have some disadvantages that make difficult its establishment as a standard solution. The main disadvantages affecting these kind of systems are as follows: (i) the management of the document and annotation consistency, and (ii) the domain evolution. Domain evolution entails ontology evolution, which can be defined as the timely adaptation of an ontology and consistent propagation of changes to dependent artefacts. More concretely, it refers to the process of changing the ontologies over time by, for example, adding

new classes or instances, modifying classes and instances or removing knowledge. Besides, the semantic annotation approaches that support ontology evolution must ensure the consistency of the annotations against the ontologies that are being modified.

In this thesis, a comparative study between the state-of-the-art semantic annotation approaches and the one proposed here is carried out. To this end, seven different features are considered, namely: semantic format support, multiple ontology support, document format support, document evolution, annotation storage, automation, and ontology evolution support (Rodríguez-García et al., 2014). The majority of the approaches included in our study are fully automatic but do not provide support for either multiple ontologies or ontology evolution. Like other approaches such as Armadillo, CREAM, KIM, the method proposed here is able to manage multiple ontologies. However, it is the only one that is able to cope with both features, multiple ontologies and ontology evolution.

The main advantage of the methodology proposed here over other related work lies in having put together a number of solutions and features that were not previously supported. In addition, the system described here resolves a more general problem, that is, the generation of semantic annotations from unstructured documents. The conceived approach can cope with multiple documents, multiple ontologies, ontology evolution, and the possibility of manual or automatic annotation. Besides, the results obtained in the experiments, which have been compared against state-of-the-art tools with a satisfactory outcome, are very promising.

By taking into account the aforementioned criteria and the analysed state-of-the-art tools, the main contributions of this thesis are the following:

- **Development of a methodology for semantic annotation and information retrieval.** The proposed methodology complies with all the seven features mentioned before. It makes use of the OWL format to represent the knowledge, provides support for both non-structured and semi structured information, enables manual and automatic annotation mechanisms, incorporates techniques to avoid incoherencies and inconsistencies, and provides support for the evolution of both, the ontology and the annotated resources. For

performance and flexibility reasons, it does also provide support for managing multiple ontologies.

- **Development of a formal ontology model.** This ontological model is devoted to semantically represent concepts. In order to define and design the ontology model, some standard properties such as attributes and annotations have been used. The use of these properties facilitates the adaptation of the ontology to any application domain and simplifies the ontology evolution process. Moreover, the simple ontology model designed improves the performance of the annotation system by accelerating the semantic annotation process, which is capable of dealing with a huge amount of data in a very short time.
- **Validation of the semantic annotation system.** The semantic annotation system validation process was carried out across two well-differentiated domains, namely, cloud computing and R&D. In order to accomplish the evaluation process in these domains, it is necessary to make use of domain ontology models that assist in the creation of the metadata required to semantically annotate the informational resources. The lack of ontologies for these domains led to the need for defining new domain models from scratch, which is an effort-intensive, time-consuming task. Also for the purposes of the validation process, domain experts were required to participate. They were in charge of validating the results generated by the proposed annotation tool. Moreover, this validation process required of some experts on both domains in order to validate the results obtained by the developed annotation system. The results obtained in the experiments, which have been compared against state-of-the-art tools with a satisfactory outcome, are very promising and are aligned with the objectives of this thesis.
- **Application of the semantic annotation methodology to calculate the similarity between ontological instances.** The encouraging results obtained during the evaluation of the semantic annotation methodology and, specifically, the semantic search engine module fostered its application in a number of different scenarios. A particularly notorious application is the use of the semantic application methodology in order to calculate the similarity between ontological instances. This measurement represents the similarity between two instances, regardless of their domain or range. The similarity methodology is

based on identifying how similar two instances are by comparing their constituent attributes. A configuration file is used to establish how the instances are to be compared, that is, which attributes are going to be used in the comparison process. A similarity matrix is generated for each comparison. Each numerical element in the matrix represents the similarity score between two instances of each compared entity. Once the matrix has been generated, it can be used to identify matching instances from each entity according to their similarity.

- **Validation of the similarity model application in a real environment.** In order to evaluate the effectiveness of the proposed application, the application domain chosen was related to R&D projects related to the ICT domain. In particular, the proposed methodology was used to calculate the similarity scores concerning ICT-related projects and the staff members that are best suit to get involved in them. For validation purposes, both domain experts and the human resources department of an organization were required. In order to carry out the experiment, a number of R&D project descriptions in natural language were introduced into the system along with the CV of a group of people having the appropriate background to carry out such projects. Once everything is in place, when the semantic inference engine module of the proposed system receives a new project description, it shall find a list of similar projects stored in the knowledge base and the staff members that are best suit to get involved in them given their background. Again, the results obtained in the experiments, which have been compared against state-of-the-art tools with a satisfactory outcome, are very promising and are aligned with the objectives of this thesis.

Once the contributions of this thesis have been enumerated, it is worth pointing out some of the limitations of the proposed annotation system. Some of these limitations are, in fact, elements that have been schedule for future work and will be further discussed at the end of this chapter.

- **Keywords-based query language**. The language used for querying in the semantic search engine is keyword-based. The expressivity of the queries is, thus, very limited and the precision of the results is directly hampered by this

lack of expressivity. Actually, according to the experimental results, the more expressive and specific the queries are the better and more precise the results get.

- **Ambiguity issue.** Providing support for multiples ontologies in the semantic annotation platform brings a significant amount of advantages over traditional alternative systems not including this feature. However, some disadvantages are also present when dealing with more than one ontology. Particularly, the most serious drawback is that of ambiguity. When using two or more ontologies, a concept could be defined differently in two independent ontologies. Under these circumstances, it is likely that the proposed semantic annotation methodology creates some inconsistencies during the annotation process. Moreover, this inconsistency could also affect the semantic indexing process, which selects an ontology in order to calculate the semantic index using the concepts previously annotated.
- **Performance of the semantic indexing algorithm.** In order to enhance the search process, the taxonomic structure of the system ontologies is leveraged to calculate the semantic index through an extended version of TF-IDF algorithm. However, given the dependencies during the generation of the semantic index, this process become very time-consuming. Redesigning and modelling this algorithm to comply with parallel programming models and paradigms could help boost the system performance.
- **Wikipedia dependency.** During the ontology evolution process, Wikipedia is used to compile new knowledge to be included in the ontology. Using a single information repository could be, sometimes, insufficient when search for a specific concept to evolve the ontology. Therefore, adding new information repositories would increase chance of finding these concepts and, consequently, succeed in the ontology enrichment process.

IX.2. FUTURE WORK

Several issues remain open for future work. In particular, some of the aforementioned drawbacks have been further analysed and planned as future work. The main future research lines are described next:

- *Update the semantic search engine module by adding new query languages.*

Currently, the input to the semantic search engine is a keyword-based query. This fact limits the expressiveness of the query and, hence, the accuracy of the results. In contrast, some current semantic search approaches (Bast et al., 2007);(Hogan et al., 2011);(Patel et al., 2003) make use of more expressive languages such as SPARQL (Prud'Hommeaux & Seaborne, 2008), RDQL (Seaborne, 2004) and OWLQL (Fikes et al., 2004). By allowing users to introduce more precise queries, the whole process is improved and the results are more accurate. However, end users are not usually familiar with this kind of languages and only Semantic Web experts can effectively leverage the advantages of these approaches. The aim of this future research line is to provide either natural language interfaces or controlled vocabulary-based interfaces that assist layman end users to compose complex queries in those more expressive semantic languages. Then, the semantic search engine would use the natural language description provided by users, which is actually mapped to semantic languages, to achieve more precise searches.

- *Deal with the ambiguities caused by supporting multiple ontologies.*

The benefits of providing support for multiple ontologies in semantic annotation systems are plentiful. It enables the system to cope with different domains at the same time. Besides, supporting multiple ontologies has a positive impact on the system performance, since handling very large unique ontologies is a very time-consuming task. However, there are also some difficulties associated to providing support for multiple ontologies in this context. In particular, dealing with a number of different ontologies representing various domains can lead to semantic inconsistencies and ambiguities in the annotations. The use of disambiguation

techniques could help in providing a solid and workable solution for overcoming these inconsistency and ambiguity issues.

- *Improve the semantic indexing algorithm performance.*

In most cases, the semantic indexing algorithm has a negative impact on the system performance. This algorithm requires vast amounts of computational resources and is inherently slow. This performance issue can be partially overcome by taking into account the inherent parallelism in the semantic indexing processes. Therefore, it is left for the future to redesign the semantic indexing algorithm to make it amenable for parallel computation.

- *Extend the semantic annotation methodology by providing support for further data formats.*

So far, the work presented here has been evaluated by using non-structured and semi-structured information. As future work, it is planned to improve the annotation and similarity calculation methodologies by providing support for structured information such as relational databases, spreadsheet, invoicing, bills, etc. Support for the so-called "Big Data" will be also analysed. The use of the proposed methodologies on this kind of information pursues a twofold objective. Firstly, semantically annotating structured data can enable the use of richer, more precise query languages than the ones traditionally associated with relational databases. Secondly, the formal underpinnings of these semantic annotations also allows the leverage of reasoning methods and so new knowledge not explicitly declared can be inferred.

- *Support additional information repositories in the ontology evolution module.*

The ontology evolution methodology presented in this thesis makes use of Wikipedia as the main and sole information provider. The use of a single information provider limits the odds to find some of the terms retrieved by the system, and therefore the possibility to enrich the ontology. Conversely, providing access to more data sources increases the probability of finding the appropriate information thus boosting the evolution process. Consequently, adding support for

other information repositories in the ontology evolution module would be a valuable contribution and is left for future work. In case of dealing with very specific application domain (e.g., medicine, biology, botany) and assuming the existence of particular data sources in such domain, it would be even possible to disregard Wikipedia as information provider.

- *Utilization of ontology quality metrics to measure the quality of the evolved ontologies.*

Ontology evaluation metrics provide a set of tools that build trust to share and reuse ontologies (Brank et al., 2005). Nowadays, there are a number of different kinds of metrics that can be employed to validate some of the properties associated with the knowledge representation model defined in the semantic annotation approach proposed here. It is worth pointing out metrics such as (Duque-Ramos et al., 2011): number of relationships per class, number of annotations per class, number of class descendants or ascendants per class, depth of taxonomy.

As part of the future work, a comprehensive and complete study of the state-of-the-art on ontology evaluation metrics will be carried out. From this study, the most appropriate metrics applicable to the proposed ontology formal model will be selected. The ultimate goal is to use these metrics in order to evaluate the quality of the ontologies resulting from the ontology evolution process. Consequently, the outcome of this enrichment process would be ontologies that can be shared and reused in others applications or technological domains.

- *Integrate an opinion mining-based recommendation system.*

The application of the annotation methodology for similarity calculation can be seen as a first proof-of-concept demonstrating the potential applicability of the proposed methodology and other information processing-based technologies in a number of different contexts. A particularly interesting research line is that of the integration of opinion mining techniques with the semantically-enhanced annotation methodology proposed in this thesis. The integrated use of opinion mining technologies could facilitate the development of recommendation systems

guided by the users' preferences. By making use of these technologies, it would be possible to define user profiles representing the users' preferences. These profiles would provide some restrictions to be taken into consideration during the search processes in order to obtain results that are more personalized and closely related to the users' preferences.

To sum up, the semantic annotation and the similarity calculation methodologies presented in this thesis can be classified into three research fields: Ontology Engineering, Retrieval Information Systems and Semantic Annotation Systems. On the one hand, the proposed annotation methodology is based on an ontology to represent the domain. The metadata produced to semantically annotate the textual information is generated from this domain ontology. On the other hand, a fully-automatic ontology evolution methodology is also proposed in this thesis. Finally, a semantic search engine is provided to retrieve the semantically-annotated information.

Even though defining the semantic annotation methodology could seem as a not so complex task, it did not only consist of conceiving a mechanism to enable the addition of metadata to make information resources more understandable for machines, but it also required the implementation of an exhaustive analysis on the numerous challenges hampering the wider use of the currently available semantic annotation approaches. Among the most important limitations, the solutions analysed in the presented state-of-the-art are affected by, for example, (i) the issues related to the resources actualization, modification or elimination, (ii) the evolution of the ontology model, or (iii) the management of the semantic annotations life cycle, among others. The semantic annotation methodology proposed in this thesis successfully faces these challenges through the development of a set of reusable and shareable modules comprising the key functionalities of the annotation system.

REFERENCIAS

Aguado de Cea, G., Álvarez de Mon Rego, I., & Pareja-Lora, A. (2003). Primeras aproximaciones a la anotación lingüístico-ontológica de documentos de Web Semántica: OntoTag. Revista Iberoamericana de Inteligencia Artificial, 18, 37-49.

Alazraqui, M., Mota, E. & Spinelli, H. (2006). Health Information Systems: from closed systems to social citizenship. A challenge for the reduction of inequalities in local management. Cadernos de Saúde Pública. 22 (12). p.pp. 2693-2702.

Alexopoulou, D., Wächter, T., Pickersgill, L., Eyre, C. & Schroeder, M. (2008). Terminologies for text-mining; an experiment in the lipoprotein metabolism domain. BMC Bioinformatics. 9 (Suppl 4). p.pp. 1-12.

Alfonseca, E. & Manandhar, S. (2002). An Unsupervised Method for General Named Entity Recognition And Automated Concept Discovery. In: In: Proceedings of the 1 st International Conference on General WordNet. 2002.

Almuhareb, A. & Poesio, M. (2004). M.: Attribute-based and value-based clustering: An evaluation. In: In: EMNLP '04, ACL. 2004, pp. 158-165.

Álvarez Silva, M., Ramírez Cruz, Y. & Anaya Sánchez, H. (2009). RECONOCEDOR DE NOMBRES DE ENTIDADES PARA EL ESPAÑOL BASADO EN EXPRESIONES REGULARES Y APRENDIZAJE AUTOMÁTICO. Innovación Tecnológica. 14 (4).

Alves, J., Marques, M.J., Saur, I. & Marques, P. (2005). Building creative ideas for successful new product development. In: 9th European Conference on Creativity and Innovation (ECCI-9):'Transformations. 2005.

Ananiadou, S. (1994). A Methodology for Automatic Term Recognition. In: Proceedings of the 15th Conference on Computational Linguistics - Volume 2. COLING '94. [Online]. 1994, Stroudsburg, PA, USA: Association for Computational Linguistics, pp. 1034-1038. Available from: http://dx.doi.org/10.3115/991250.991317. [Accessed: 9 May 2014].

Antoniou, G. & vanHarmelen, F. (2004). A Semantic Web Primer. Cambridge, MA, USA: MIT Press.

Asahara, M. & Matsumoto, Y. (2003). Japanese Named Entity Extraction with Redundant Morphological Analysis. In: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1. NAACL '03. [Online]. 2003, Stroudsburg, PA, USA: Association for Computational Linguistics, pp. 8-15. Available from: http://dx.doi.org/10.3115/1073445.1073447. [Accessed: 1 June 2014].

Bahl, L.R. & Mercer, R.L. (1976). Part of speech assignment by a statistical decision algorithm. In: IEEE International Symposium on Information Theory. 1976, pp. 88-89.

Baker, J.K. (1979). Trainable grammars for speech recognition. The Journal of the Acoustical Society of America. 65 (S1). p.pp. S132-S132.

Banerjee, J., Kim, W., Kim, H.-J. & Korth, H.F. (1987). Semantics and Implementation of Schema Evolution in Object-oriented Databases. In: Proceedings of the 1987 ACM SIGMOD International Conference on Management of Data. SIGMOD '87. [Online]. 1987, New York, NY, USA: ACM, pp. 311-322. Available from: http://doi.acm.org/10.1145/38713.38748. [Accessed: 30 May 2014].

Barrón, A., Sierra, G. & Villaseñor, E. (2006). C-value aplicado a la extracción de términos multipalabra en documentos técnicos y científicos en español. In: 7th Mexican International Conference on Computer Science (ENC 2006). IEEE Computer Press, Los Alamitos. 2006.

Bast, H., Chitea, A., Suchanek, F. & Weber, I. (2007). ESTER: Efficient Search on Text, Entities, and Relations. In: Proceedings of the 30th Annual International ACM SIGIR

Conference on Research and Development in Information Retrieval. SIGIR '07. [Online]. 2007, New York, NY, USA: ACM, pp. 671-678. Available from: http://doi.acm.org/10.1145/1277741.1277856. [Accessed: 25 July 2014].

Bast, H., Chitea, A., Suchanek, F., & Weber, I. (2007). ESTER: Efficient Search on Text, Entities, and Relations. In Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 671-678). New York, NY, USA: ACM. doi:10.1145/1277741.1277856

Bechhofer, S., Carr, L., Goble, C. & Hall, W. (2001). Conceptual Open Hypermedia = The Semantic Web? In: In Proceedings of the WWW2001, Semantic Web Workshop, Hongkong. 2001, pp. 44-50.

Belkin, N.J. & Croft, W.B. (1987). Annual Review of Information Science and Technology, Vol. 22. In: M. E. Williams (ed.). [Online]. New York, NY, USA: Elsevier Science Inc., pp. 109-145. Available from: http://dl.acm.org/citation.cfm?id=42502.42506. [Accessed: 4 June 2014].

Berners-Lee, T., Hendler, J. & Lassila, O. (2001). The semantic web. Scientific american. 284 (5). p.pp. 28-37.

Beynon-Davies, P. (2000). Database systems. Macmillan.

Bikakis, N., Giannopoulos, G., Dalamagas, T. & Sellis, T. (2010). Integrating Keywords and Semantics on Document Annotation and Search. In: R. Meersman, T. Dillon, & P. Herrero (eds.). On the Move to Meaningful Internet Systems, OTM 2010. Lecture Notes in Computer Science. [Online]. Springer Berlin Heidelberg, pp. 921-938. Available from: http://link.springer.com/chapter/10.1007/978-3-642-16949-6_19. [Accessed: 2 June 2014].

Bikel, D.M., Miller, S., Schwartz, R. & Weischedel, R. (1997). Nymble: A High-performance Learning Name-finder. In: Proceedings of the Fifth Conference on Applied Natural Language Processing. ANLC '97. [Online]. 1997, Stroudsburg, PA, USA: Association for Computational Linguistics, pp. 194-201. Available from: http://dx.doi.org/10.3115/974557.974586. [Accessed: 1 June 2014].

Bledsoe, W.W. & Browning, I. (1959). Pattern Recognition and Reading by Machine. In: Papers Presented at the December 1-3, 1959, Eastern Joint IRE-AIEE-ACM Computer Conference. IRE-AIEE-ACM '59 (Eastern). [Online]. 1959, New York, NY, USA: ACM, pp. 225-232. Available from: http://doi.acm.org/10.1145/1460299.1460326. [Accessed: 31 May 2014].

Bojārs, U. & Breslin, J.G. (2007). ResumeRDF: Expressing skill information on the Semantic Web. In: 1 st International ExpertFinder Workshop. 2007.

Bookstein, A. & Swanson, D.R. (1974). Probabilistic models for automatic indexing. Journal of the American Society for Information science. 25 (5). p.pp. 312-316.

Boole, G. (1854). An investigation of the laws of thought: on which are founded the mathematical theories of logic and probabilities. Walton and Maberly.

Borst, W.N. (1997). Construction of Engineering Ontologies for Knowledge Sharing and Reuse. Centre for Telematics and Information Technology.

Borthwick, A., Sterling, J., Agichtein, E. & Grishman, R. (1998). NYU: Description of the MENE Named Entity System as Used in MUC-7. In: In Proceedings of the Seventh Message Understanding Conference (MUC-7. 1998.

Bozsak, E., Ehrig, M., Handschuh, S., Hotho, A., Maedche, A., Motik, B., Oberle, D., Schmitz, C., Staab, S., Stojanovic, L., Stojanovic, N., Studer, R., Stumme, G., Sure, Y., Tane, J., Volz, R. & Zacharias, V. (2002). KAON - Towards a Large Scale Semantic Web. In: Proceedings of the Third International Conference on E-Commerce and Web Technologies. EC-WEB '02. [Online]. 2002, London, UK, UK: Springer-Verlag, pp. 304-313. Available from: http://dl.acm.org/citation.cfm?id=646162.680500. [Accessed: 28 July 2014].

Brank, J., Grobelnik, M. & Mladeni?, D. (2005). A survey of ontology evaluation techniques. In: In In Proceedings of the Conference on Data Mining and Data Warehouses (SiKDD 2005. 2005.

Brank, J., Grobelnik, M., & Mladeni?, D. (2005). A survey of ontology evaluation techniques. In In In Proceedings of the Conference on Data Mining and Data Warehouses (SiKDD 2005.

Brickley, D. & Miller, L. (2012). FOAF vocabulary specification 0.98. Namespace Document. 9.

Brin, S. (1999). Extracting Patterns and Relations from the World Wide Web. [Online]. 11 November 1999. WebDB Workshop at EDBT'98. Available from: http://ilpubs.stanford.edu:8090/421/. [Accessed: 1 June 2014].

Buitelaar, P. & Magnini, B. (2005). Ontology Learning from Text: An Overview. In: In Paul Buitelaar, P., Cimiano, P., Magnini B. (Eds.), Ontology Learning from Text: Methods, Applications and Evaluation. 2005, IOS Press, pp. 3-12.

Buitelaar, P., Olejnik, D. & Sintek, M. (2004). A Protégé Plug-In for Ontology Extraction from Text Based on Linguistic Analysis. In: C. J. Bussler, J. Davies, D. Fensel, & R. Studer (eds.). The Semantic Web: Research and Applications. Lecture Notes in Computer Science. [Online]. Springer Berlin Heidelberg, pp. 31-44. Available from: http://link.springer.com/chapter/10.1007/978-3-540-25956-5_3. [Accessed: 12 May 2014].

Buneman, P. (1997). Semistructured Data. In: Proceedings of the Sixteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems. PODS '97. [Online]. 1997, New York, NY, USA: ACM, pp. 117-121. Available from: http://doi.acm.org/10.1145/263661.263675. [Accessed: 31 May 2014].

Burgun, A. & Bodenreider, O. (2001). Aspects of the taxonomic relation in the biomedical domain. In: Proceedings of the international conference on Formal Ontology in Information Systems-Volume 2001. 2001, ACM, pp. 222-233.

Califf, M.E. & Mooney, R.J. (1999). Relational Learning of Pattern-match Rules for Information Extraction. In: Proceedings of the Sixteenth National Conference on Artificial Intelligence and the Eleventh Innovative Applications of Artificial Intelligence Conference Innovative Applications of Artificial Intelligence. AAAI '99/IAAI '99. [Online]. 1999, Menlo Park, CA, USA: American Association for Artificial Intelligence, pp. 328-334. Available from: http://dl.acm.org/citation.cfm?id=315149.315318. [Accessed: 31 May 2014].

Castano, S., Ferrara, A., Montanelli, S. & Lorusso, D. (2008). Instance Matching for Ontology Population. In: SEBD. 2008, pp. 121-132.

Castells, P., Fernandez, M. & Vallet, D. (2007). An Adaptation of the Vector-Space Model for Ontology-Based Information Retrieval. IEEE Transactions on Knowledge and Data Engineering. 19 (2). p.pp. 261-272.

Chapman, S., Norton, B. & Ciravegna, F. (2005). Armadillo: Integrating knowledge for the semantic web. In: Proceedings of the Dagstuhl Seminar in Machine Learning for the Semantic Web. 2005.

Chen, H., Chiang, R.H. & Storey, V.C. (2012). Business Intelligence and Analytics: From Big Data to Big Impact. MIS Quarterly. 36 (4).

Chesbrough, H., Vanhaverbeke, W. & West, J. (2006). Open innovation: Researching a new paradigm. Oxford university press.

Chesbrough, H.W. (2003). Open innovation: The new imperative for creating and profiting from technology. Harvard Business Press.

Chomsky, N. (1956). Three models for the description of language. IRE Transactions on Information Theory. 2 (3). p.pp. 113-124.

Chowdhury, G.G. (2003). Natural language processing. Annual Review of Information Science and Technology. 37 (1). p.pp. 51-89.

Church, K.W. & Church, K.W. (1980). On Memory Limitations In Natural Language Processing.

Cimiano, P. & Völker, J. (2005). Towards large-scale, open-domain and ontology-based named entity classification. In: Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP). 2005.

Cimiano, P., Hotho, A. & Staab, S. (2005). Learning Concept Hierarchies from Text Corpora Using Formal Concept Analysis. Journal of Artificial Intelligence research. 24. p.pp. 305-339.

Cimiano, P., Pivk, A., Schmidt-Thieme, L., Staab, S. & Staab, E.P.S.S. (2003). Learning Taxonomic Relations from Heterogeneous Sources of Evidence.

Cimiano, P., Völker, J. & Studer, R. (2006). Ontologies on Demand? A Description of the State-of-the-Art, Applications, Challenges and Trends for Ontology Learning from Text.

Ciravegna, F. (2001). Adaptive Information Extraction from Text by Rule Induction and Generalisation. In: Proceedings of the 17th International Joint Conference on Artificial Intelligence - Volume 2. IJCAI'01. [Online]. 2001, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., pp. 1251-1256. Available from: http://dl.acm.org/citation.cfm?id=1642194.1642261. [Accessed: 1 June 2014].

Ciravegna, F., Chapman, S., Dingli, A. & Wilks, Y. (2004). Learning to Harvest Information for the Semantic Web. In: C. J. Bussler, J. Davies, D. Fensel, & R. Studer (eds.). The Semantic Web: Research and Applications. Lecture Notes in Computer Science. [Online]. Springer Berlin Heidelberg, pp. 312-326. Available from: http://link.springer.com/chapter/10.1007/978-3-540-25956-5_22. [Accessed: 7 April 2014].

Ciravegna, F., Dingli, A., Petrelli, D. & Wilks, Y. (2002). User-system cooperation in document annotation based on information extraction. In: In Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management, EKAW02. 2002, Springer Verlag, pp. 122-137.

Ciravegna, F., Dingli, A., Wilks, Y. & Petrelli, D. (2002b). Adaptive Information Extraction for Document Annotation in Amilcare. In: Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. SIGIR '02. [Online]. 2002, New York, NY, USA: ACM, pp. 451-451. Available from: http://doi.acm.org/10.1145/564376.564492. [Accessed: 1 June 2014].

Clerc, J.L. (1692). Ontologia; sive de ente in genere. Impesis Awnsham & Johan. Churchill.

Collins, M. & Singer, Y. (1999). Unsupervised Models for Named Entity Classification. In: In Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora. 1999, pp. 100-110.

Colmerauer, A. (1975). Les grammaires de métamorphose GIA.

Cordy, J.R., Carmichael, I.H. & Halliday, R. (2000). The TXL Programming Language-Version 10. Kingston: Queen's University at Kingston and Legasys Corporation.

Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C. & others (2001). Introduction to algorithms. Second Edition. MIT Press and McGraw-Hill.

COTEC (2004). White paper: The Innovation Spanish System.

Cowie, J. & Lehnert, W. (1996). Information Extraction. Commun. ACM. 39 (1). p.pp. 80-91.

Cucchiarelli, A. & Velardi, P. (2001). Unsupervised Named Entity Recognition Using Syntactic and Semantic Contextual Evidence. Comput. Linguist. 27 (1). p.pp. 123-131.

Cunningham, H. (2005). Information extraction, automatic. Encyclopedia of language and linguistics,. p.pp. 665-677.

Cunningham, H., Maynard, D., Bontcheva, K. & Tablan, V. (2002). GATE: an Architecture for Development of Robust HLT Applications. In: In Recent Advanced in Language Processing. 2002, pp. 168-175.

David Beckett & Tim Berners-Lee (2008). Turtle - Terse RDF Triple Language. [Online]. Available from: http://www.w3.org/TeamSubmission/2011/SUBM-turtle-20110328/. [Accessed: 30 May 2014].

Davis, B. P. (2013). On Applying Controlled Natural Languages for Ontology Authoring and Semantic Annotation. National University of Ireland Galway.

Davis, R., Shrobe, H. & Szolovits, P. (1993). What Is a Knowledge Representation? AI Magazine. 14 (1). p.p. 17.

Dean, T.R., Cordy, J.R., Schneider, K.A. & Malton, A.J. (2001). Using Design Recovery Techniques to Transform Legacy Systems. In: ICSM. 2001, pp. 622-631.

Dempster, A.P., Laird, N.M., Rubin, D.B. & others (1977). Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal statistical Society. 39 (1). p.pp. 1-38.

Dijkstra, E.W. (1959). A note on two problems in connexion with graphs. Numerische Mathematik. 1 (1). p.pp. 269-271.

Dill, S., Eiron, N., Gibson, D., Gruhl, D., Guha, R., Jhingran, A., Kanungo, T., Rajagopalan, S., Tomkins, A., Tomlin, J.A. & Zien, J.Y. (2003). SemTag and Seeker: Bootstrapping the Semantic Web via Automated Semantic Annotation. In: Proceedings of the 12th International Conference on World Wide Web. WWW '03. [Online]. 2003, New York, NY, USA: ACM, pp. 178-186. Available from: http://doi.acm.org/10.1145/775152.775178. [Accessed: 1 June 2014].

Drumond, L. & Girardi, R. (2008). A Survey of Ontology Learning Procedures. WONTO. 427.

Du, T.C., Li, F. & King, I. (2009). Managing Knowledge on the Web - Extracting Ontology from HTML Web. Decis. Support Syst. 47 (4). p.pp. 319-331.

Dubuc, R. (1997). Terminology: A practical approach. Brossard, Qué.: Linguatech.

Dumbill, E. (2012). DOAP: Description of a Project. Technical report, 2012. http://trac. usefulinc. com/doap.

Duque-Ramos, A., Fernández-Breis, J. T., Stevens, R., & Aussenac-Gilles, N. (2011). OQuaRE: A SQuaRE-based approach for evaluating the quality of ontologies. Journal of Research and Practice in Information Technology, 43(2), 159.

Etzioni, O., Cafarella, M., Downey, D., Popescu, A.-M., Shaked, T., Soderland, S., Weld, D.S. & Yates, A. (2005). Unsupervised Named-Entity Extraction from the Web: An Experimental Study. ARTIFICIAL INTELLIGENCE. 165. p.pp. 91-134.

Evans, R. (2003). A framework for named entity recognition in the open domain. In: In Proceedings of the Recent Advances in Natural Language Processing (RANLP. 2003, pp. 137-144.

Feldman, S. (1999). NLP Meets the Jabberwocky: Natural Language Processing in Information Retrieval. ONLINE-WESTON THEN WILTON-. 23. p.pp. 62-73.

Fellbaum, C. (2005). WordNet and Wordnets. In: Encyclopedia of Language and Linguistics. Elsevier, pp. 2-665.

Fikes, R., Hayes, P. & Horrocks, I. (2004). OWL-QL-a language for deductive query answering on the Semantic Web. Web Semantics: Science, Services and Agents on the World Wide Web. 2 (1). p.pp. 19-29.

Fleischman, M. & Hovy, E. (2002). Fine grained classification of named entities. In: In Proc. of the 19th International Conference on Computational Linguistics. 2002, pp. 1-7.

Flouris, G. & Plexousakis, D. (2006). Bridging Ontology Evolution and Belief Change. In: G. Antoniou, G. Potamias, C. Spyropoulos, & D. Plexousakis (eds.). Advances in Artificial Intelligence. Lecture Notes in Computer Science. [Online]. Springer Berlin Heidelberg, pp. 486-489. Available from: http://link.springer.com/chapter/10.1007/11752912_51. [Accessed: 30 May 2014].

Flouris, G., Manakanatas, D., Kondylakis, H., Plexousakis, D. & Antoniou, G. (2007). Ontology change: classification and survey.

Fox, E.A. & Koll, M.B. (1988). Practical Enhanced Boolean Retrieval: Experiences with SMART and SIRE Systems. Inf. Process. Manage. 24 (3). p.pp. 257-267.

García-Moreno, C., Hernández-González, Y., Rodríguez-García, M.Á., Miñarro-Giménez, J.A., Valencia-García, R. & Almela, A. (2013). A Semantic based Platform for Research and Development Projects Management in the ICT Domain. Journal of Universal Computer Science. 19 (13). p.pp. 1914-1939.

Gianluca, R.B., Rossi, G.D. & Pazienza, M.T. (1997). Inducing Terminology for Lexical Acquisition. In: In: Preoceeding of EMNLP 97 Conference. 1997.

Giannopoulos, G., Bikakis, N., Dalamagas, T. & Sellis, T. (2010). GoNTogle: A Tool for Semantic Annotation and Search. In: Proceedings of the 7th International Conference on The Semantic Web: Research and Applications - Volume Part II. ESWC'10. [Online].

2010, Berlin, Heidelberg: Springer-Verlag, pp. 376-380. Available from: http://dx.doi.org/10.1007/978-3-642-13489-0_27. [Accessed: 1 June 2014].

GottFriedl Wilhelm Leibniz (1683). Introductio ad Encyclopaediam arcanam.

Grau, B.C., Horrocks, I., Kazakov, Y. & Sattler, U. (2009). Extracting modules from ontologies: A logic-based approach. In: Modular Ontologies. Springer, pp. 159-186.

Grau, B.C., Horrocks, I., Motik, B., Parsia, B., Patel-Schneider, P. & Sattler, U. (2008). OWL 2: The next step for OWL. Web Semantics: Science, Services and Agents on the World Wide Web. 6 (4). p.pp. 309-322.

Grosz, B.J. & Sidner, C.L. (1986). Attention, Intentions, and the Structure of Discourse. Comput. Linguist. 12 (3). p.pp. 175-204.

Gruber, T. (2004). Interview Tom Gruber. SIGSEMIS Bulletin. 1 (3). p.pp. 4-9.

Gruber, T.R. (1993). A translation approach to portable ontology specifications. Knowledge Acquisition. 5 (2). p.pp. 199-220.

Grune, D. & Jacobs, C.J.H. (1990). Parsing Techniques: A Practical Guide. Upper Saddle River, NJ, USA: Ellis Horwood.

Guarino, N. (1995). Formal Ontology, Conceptual Analysis and Knowledge Representation. Int. J. Hum.-Comput. Stud. 43 (5-6). p.pp. 625-640.

Guarino, N. (1998). Formal Ontology and Information Systems. In: 1998, IOS Press, pp. 3-15.

Gulla, J.A., Borch, H.O. & Ingvaldsen, J.E. (2007). Ontology Learning for Search Applications. In: R. Meersman & Z. Tari (eds.). On the Move to Meaningful Internet Systems 2007: CoopIS, DOA, ODBASE, GADA, and IS. Lecture Notes in Computer Science. [Online]. Springer Berlin Heidelberg, pp. 1050-1062. Available from: http://link.springer.com/chapter/10.1007/978-3-540-76848-7_69. [Accessed: 15 May 2014].

Haase, K. (2000). Interlingual BRICO. IBM Syst. J. 39 (3-4). p.pp. 589-596.

Haase, P. & Stojanovic, L. (2005). Consistent Evolution of OWL Ontologies. In: A. Gómez-Pérez & J. Euzenat (eds.). The Semantic Web: Research and Applications. Lecture Notes in Computer Science. [Online]. Springer Berlin Heidelberg, pp. 182-197. Available from: http://link.springer.com/chapter/10.1007/11431053_13. [Accessed: 30 May 2014].

Handschuh, S. & Staab, S. (2003). CREAM: CREAting Metadata for the Semantic Web. Computer Networks. 42 (5). p.pp. 579-598.

Handschuh, S., Staab, S. & Ciravegna, F. (2002). S-CREAM - Semi-automatic CREAtion of Metadata. In: Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web. EKAW '02. [Online]. 2002, London, UK, UK: Springer-Verlag, pp. 358-372. Available from: http://dl.acm.org/citation.cfm?id=645362.650866. [Accessed: 1 June 2014].

Happel, H.-J., Korthaus, A., Seedorf, S. & Tomczyk, P. (2006). KOntoR: An Ontology-enabled Approach to Software Reuse. In: IN: PROC. OF THE 18TH INT. CONF. ON SOFTWARE ENGINEERING AND KNOWLEDGE ENGINEERING. 2006.

Harris, Z. (1954). Distributional structure. Word. 10 (23). p.pp. 146-162.

Harter, S.P. (1975). A probabilistic approach to automatic keyword indexing. Part II. An algorithm for probabilistic indexing. Journal of the American Society for Information Science. 26 (5). p.pp. 280-289.

Hartig, O., Kost, M. & Freytag, J.C. (2008). Designing Component-Based Semantic Web Applications with DESWAP. In: International Semantic Web Conference (Posters & Demos). 2008.

Hearst, M.A. (1992). Automatic Acquisition of Hyponyms from Large Text Corpora. In: In Proceedings of the 14th International Conference on Computational Linguistics. 1992, pp. 539-545.

Hearst, M.A. (1998). Automated discovery of WordNet relations. WordNet: an electronic lexical database. p.pp. 131-151.

Heflin, J. & Pan, Z. (2004). A model theoretic semantics for ontology versioning. In: In Third International Semantic Web Conference. 2004, Springer, pp. 62-76.

Heflin, J., Hendler, J. & Luke, S. (1998). Reading between the lines: Using SHOE to discover implicit knowledge from the Web. In: AAAI-98 Workshop on AI and Information Integration. 1998.

Heflin, J., Hendler, J. & Luke, S. (1999). Coping with changing ontologies in a distributed environment. In: AAAI-99 Workshop on Ontology Management. 1999.

Hiemstra, D. (2009). Information retrieval models. Information Retrieval: searching in the 21st Century. p.pp. 1-19.

Hogan, A., Harth, A., Umbrich, J., Kinsella, S., Polleres, A. & Decker, S. (2011). Searching and Browsing Linked Data with SWSE: The Semantic Web Search Engine. Web Semant. 9 (4). p.pp. 365-401.

Horrocks, I., Patel-Schneider, P.F. & Harmelen, F.V. (2003). From SHIQ and RDF to OWL: The Making of a Web Ontology Language. Journal of Web Semantics. 1. p.p. 2003.

Hürsch, W.L., Landau, E., Maclane, S., Lieberherr, K., Zachos, E., Haskel, F., Silva-lepe, I., Keszenheimer, L., Videira, C., Hrsch, W. & Bchi, R. (1995). Maintaining Consistency and Behavior of Object-Oriented Systems during Evolution.

Husserl, E. (1970). Logical Investigations. Routledge and K. Paul.

Jaakkola, J. & Kilpeläinen, P. (1999). Nested Text-Region Algebra.

James Hendler & Deborah L. McGuinness (2000). The DARPA Agent Markup Language. 16 (6). p.pp. 67-73.

Jelinek, F., Bahl, L.R. & Mercer, R.L. (1975). Design of a linguistic statistical decoder for the recognition of continuous speech. IEEE Transactions on Information Theory. 21 (3). p.pp. 250-256.

Kageura, K. & Umino, B. (1996). Methods of automatic term recognition: A review. Terminology. 3 (2). p.pp. 259-289.

Kaplan, R.M. & Kay, M. (1981). Phonological rules and finite-state transducers. In: Linguistic Society of America Meeting Handbook, Fifty-Sixth Annual Meeting. 1981, pp. 27-30.

Kiyavitskaya, N., Zeni, N., Cordy, J.R., Mich, L. & Mylopoulos, J. (2005). Semi-Automatic Semantic Annotations for Web Documents. In: Proc. SWAP 2005, 2nd Italian Semantic Web Workshop. 2005, pp. 14-15.

Kiyavitskaya, N., Zeni, N., Cordy, J.R., Mich, L. & Mylopoulos, J. (2009). Cerno: Light-weight tool support for semantic annotation of textual documents. Data & Knowledge Engineering. 68 (12). p.pp. 1470-1492.

Kleene, S.C. (1951). Representation of events in nerve nets and finite automata. DTIC Document.

Klein, M. & Fensel, D. (2001). Ontology versioning on the Semantic Web. In: Stanford University. 2001, pp. 75-91.

Klyne, G. & Carroll, J. (2004). Resource Description Framework (RDF): Concepts and Abstract Syntax. [Online]. Available from: http://www.w3.org/TR/rdf-concepts/.

Krauthammer, M. & Nenadic, G. (2004). Term identification in the biomedical literature. Journal of biomedical informatics. 37 (6). p.pp. 512-526.

Kullback, S. & Leibler, R.A. (1951). On Information and Sufficiency. The Annals of Mathematical Statistics. 22 (1). p.pp. 79-86.

Lawson, T. (2004). A conception of ontology. Mimeograph, University of Cambridge.

Leech, G. (1997). Introducing corpus annotation. In: R. Garside, G. Leech, & A. Mcenery (eds.). Corpus Annotation: Linguistic Information from Computer Text Corpora. Addison Wesley Longman, pp. 1-18.

Levenshtein, V.I. (1966). Binary codes capable of correcting deletions, insertions and reversals. In: Soviet physics doklady. 1966, p. 707.

Liddy, E. (2001). Natural Language Processing. iSchool Faculty Scholarship. [Online]. Available from: http://surface.syr.edu/istpub/63.

Lin, D. (1998). Automatic Retrieval and Clustering of Similar Words. In: Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 2. ACL '98. [Online]. 1998, Stroudsburg, PA, USA: Association for Computational Linguistics, pp. 768-774. Available from: http://dx.doi.org/10.3115/980691.980696. [Accessed: 31 May 2014].

Locke, W.N. & Booth, A.D. (1956). Machine translation of languages. American Documentation. 7 (2). p.pp. 135-136.

Lorhard, J. (1606). Ogdoas scholastica. Straub.

Luhn, H.P. (1957). A statistical approach to mechanized encoding and searching of literary information. IBM Journal of research and development. 1 (4). p.pp. 309-317.

Luke, S., Spector, L., Rager, D. & Hendler, J. (1997). Ontology-based Web Agents. In: Proceedings of the First International Conference on Autonomous Agents. AGENTS '97. [Online]. 1997, New York, NY, USA: ACM, pp. 59-66. Available from: http://doi.acm.org/10.1145/267658.267668. [Accessed: 31 March 2014].

Maedche, A. & Staab, S. (2001). Ontology learning for the semantic web. IEEE Intelligent systems. 16 (2). p.pp. 72-79.

Maedche, A. & Staab, S. (2004). Ontology Learning. In: HANDBOOK ON ONTOLOGIES. 2004, Springer, pp. 173-189.

Malik, S. K., Prakash, N., & Rizvi, S. (2010). Semantic annotation framework for intelligent information retrieval using KIM architecture. International Journal of Web & Semantic Technology (IJWest), 1(4), 12-26.

Manning, C.D. & Schütze, H. (1999). Foundations of Statistical Natural Language Processing. Cambridge, MA, USA: MIT Press.

Manning, C.D., Raghavan, P. & Schütze, H. (2008). Introduction to Information Retrieval. New York, NY, USA: Cambridge University Press.

Margulis, E.L. (1993). Modelling Documents with Multiple Poisson Distributions. Inf. Process. Manage. 29 (2). p.pp. 215-227.

McCallum, A. & Li, W. (2003). Early Results for Named Entity Recognition with Conditional Random Fields, Feature Induction and Web-enhanced Lexicons. In: Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4. CONLL '03. [Online]. 2003, Stroudsburg, PA, USA: Association for Computational Linguistics, pp. 188-191. Available from: http://dx.doi.org/10.3115/1119176.1119206. [Accessed: 1 June 2014].

McCarthy, J., Minsky, M.L., Rochester, N. & Shannon, C.E. (1955). A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence. [Online]. Available from: http://www-formal.stanford.edu/jmc/history/dartmouth/dartmouth.html. [Accessed: 31 May 2014].

McCulloch, W.S. & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. The bulletin of mathematical biophysics. 5 (4). p.pp. 115-133.

McGuinness, D.L. & Van Harmelen, F. (2004). OWL web ontology language overview. W3C recommendation. 10 (10). p.p. 2004.

Mell, P. & Grance, T. (2011). The NIST definition of cloud computing.

Milios, E., Zhang, Y., He, B. & Dong, L. (2003). Automatic Term Extraction And Document Similarity In Special Text Corpora. 6th Conference of the Pacific Association for Computational Linguistics. p.pp. 275-284.

Miller, G.A. (1995). WordNet: a lexical database for English. Communications of the ACM. 38 (11). p.pp. 39-41.

Mitchell, T.M. (1997). Machine Learning. 1st Ed. New York, NY, USA: McGraw-Hill, Inc.

Mizoguchi, R., Van Welkenhuysen, J. & Ikeda, M. (1995). Task ontology for reuse of problem solving knowledge. In: Towards Very Large Knowledge Bases. IOS Press, pp. 46-57.

Moore, E.F. (1956). Gedanken-experiments on sequential machines. Automata studies. 34. p.pp. 129-153.

Mosteller, F. & Wallace, D.L. (1964). Inference and disputed authorship: The Federalist. Addison-Wesley.

Nadeau, D. & Sekine, S. (2007). A survey of named entity recognition and classification. Lingvisticae Investigationes. 30 (1). p.pp. 3-26.

Nagarajan, M. (2006). Semantic annotations in web services. In: Semantic Web Services, Processes and Applications. Springer, pp. 35-61.

Neches, R., Fikes, R., Finin, T., Gruber, T., Patil, R., Senator, T. & Swartout, W.R. (1991). Enabling Technology for Knowledge Sharing. AI Mag. 12 (3). p.pp. 36-56.

Nevill-Manning, C.G., Witten, I.H., Olsen, D.R. & Jr (1996). Compressing Semi-Structured Text Using Hierarchical Phrase Identification. In: PROC. DATA COMPRESSION CONFERENCE, J.A. STORER AND M. COHN (EDS.), LOS ALAMITOS. 1996, IEEE Press, pp. 53-72.

Newell, A. & Simon, H.A. (1956). The logic theory machine-A complex information processing system. Information Theory, IRE Transactions on. 2 (3). p.pp. 61-79.

Nobelius, D. (2004). Towards the sixth generation of R&D management. International Journal of Project Management. 22 (5). p.pp. 369-375.

Noy, N.F. & Klein, M. (2004). Ontology Evolution: Not the Same as Schema Evolution. Knowledge and Information Systems. 6 (4). p.pp. 428-440.

Noy, N.F. & Musen, M.A. (2003). The PROMPT Suite: Interactive Tools for Ontology Merging and Mapping. Int. J. Hum.-Comput. Stud. 59 (6). p.pp. 983-1024.

Ochoa, J.L., Almela, Á., Hernández-Alcaraz, M.L. & Valencia-García, R. (2011). Learning morphosyntactic patterns for multiword term extraction. Scientific Research and Essays. 6 (26). p.pp. 5563-5578.

Ogden, C.K., Richards, I.A., Malinowski, B. & Crookshank, F.G. (1946). The meaning of meaning. Harcourt, Brace & World New York.

Oren, E., Möller, K.H., Scerri, S., Handschuh, S. & Sintek, M. (2006). What are Semantic Annotations?

Paşca, M. (2007). Organizing and Searching the World Wide Web of Facts - Step Two: Harnessing the Wisdom of the Crowds. In: Proceedings of the 16th International Conference on World Wide Web. WWW '07. [Online]. 2007, New York, NY, USA: ACM, pp. 101-110. Available from: http://doi.acm.org/10.1145/1242572.1242587. [Accessed: 1 June 2014].

Park, S.-B., Kim, S.-S., Oh, S., Zeong, Z., Lee, H. & Park, S.R. (2008). Target concept selection by property overlap in ontology population. International Journal of Computer Science. 3 (1). p.pp. 14-18.

Patel, C., Supekar, K., Lee, Y. & Park, E.K. (2003). OntoKhoj: A Semantic Web Portal for Ontology Searching, Ranking and Classification. In: Proceedings of the 5th ACM International Workshop on Web Information and Data Management. WIDM '03. [Online]. 2003, New York, NY, USA: ACM, pp. 58-61. Available from: http://doi.acm.org/10.1145/956699.956712. [Accessed: 25 July 2014].

Pazienza, M.T., Pennacchiotti, M. & Zanzotto, F.M. (2005). Terminology Extraction: An Analysis of Linguistic and Statistical Approaches. In: D. S. Sirmakessis (ed.). Knowledge Mining. Studies in Fuzziness and Soft Computing. [Online]. Springer Berlin Heidelberg, pp. 255-279. Available from: http://link.springer.com/chapter/10.1007/3-540-32394-5_20. [Accessed: 10 May 2014].

Petasis, G., Karkaletsis, V., Paliouras, G., Krithara, A. & Zavitsanos, E. (2011). Ontology Population and Enrichment: State of the Art. In: G. Paliouras, C. D. Spyropoulos, & G. Tsatsaronis (eds.). Knowledge-Driven Multimedia Information Extraction and Ontology Evolution. Lecture Notes in Computer Science. [Online]. Springer Berlin Heidelberg, pp. 134-166. Available from: http://link.springer.com/chapter/10.1007/978-3-642-20795-2_6. [Accessed: 10 May 2014].

Plessers, P. & Troyer, O.D. (2005). Ontology Change Detection Using a Version Log. In: In Proceeding of the 4th International Semantic Web Conference. 2005, Springer, pp. 578-592.

Poli, R. (2001). ALWIS: Ontology for Knowledge Engineers. Zeno, The Leiden-Utrecht Research Institute of Philosophy.

Poli, R. (2003). Descriptive, Formal and Formalized Ontologies. In: Husserl's Logical Investigations Reconsidered. Contributions to Phenomenology. [Online]. Springer Netherlands, pp. 183-210. Available from: http://link.springer.com/chapter/10.1007/978-94-017-0207-2_12. [Accessed: 29 May 2014].

Popov, B., Kiryakov, A., Kirilov, A., Manov, D., Ognyanoff, D. & Goranov, M. (2003). KIM - Semantic Annotation Platform. In: D. Fensel, K. Sycara, & J. Mylopoulos (eds.). The Semantic Web - ISWC 2003. Lecture Notes in Computer Science. [Online]. Springer Berlin Heidelberg, pp. 834-849. Available from: http://link.springer.com/chapter/10.1007/978-3-540-39718-2_53. [Accessed: 1 June 2014].

Prud'Hommeaux, E. & Seaborne, A. (2008). SPARQL query language for RDF. W3C recommendation. 15.

Quélin, B. (2000). Core competencies, R&D management and partnerships. European Management Journal. 18 (5). p.pp. 476-487.

Quine, W.V.O. (1961). From a Logical Point of View: 9 Logico-philosophical Essays. Harvard University Press.

Rada, R., Mili, H., Bicknell, E. & Blettner, M. (1989). Development and application of a metric on semantic nets. IEEE Transactions on Systems, Man and Cybernetics. 19 (1). p.pp. 17-30.

Rada, R., Mili, H., Bicknell, E. & Blettner, M. (1989). Development and application of a metric on semantic nets. IEEE Transactions on Systems, Man and Cybernetics. 19 (1). p.pp. 17-30.

Rau, L.F. (1991). Extracting company names from text. In: , Seventh IEEE Conference on Artificial Intelligence Applications, 1991. Proceedings. February 1991, pp. 29-32.

Reiterer, E., Dreher, H. & Guetl, C. (2010). Automatic Concept Retrieval with Rubrico. [Online]. Available from: http://espace.library.curtin.edu.au/R/?func=dbin-jump-full&object_id=154452&local_base=GEN01-ERA02.

Rhoton, J. (2010). Cloud Computing Explained. 2nd edition. Recursive Press, Tunbridge Wells.

Richard Cyganiak, David Wood & Markus Lanthaler (2004). RDF Vocabulary Description Language 1.0: RDF Schema. Changes. [Online]. Available from: http://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/. [Accessed: 30 May 2014].

Riloff, E. & Jones, R. (1999). Learning Dictionaries for Information Extraction by Multi-level Bootstrapping. In: Proceedings of the Sixteenth National Conference on Artificial Intelligence and the Eleventh Innovative Applications of Artificial Intelligence Conference Innovative Applications of Artificial Intelligence. AAAI '99/IAAI '99. [Online]. 1999, Menlo Park, CA, USA: American Association for Artificial Intelligence, pp. 474-479. Available from: http://dl.acm.org/citation.cfm?id=315149.315364. [Accessed: 1 June 2014].

Robertson, S.E. & Walker, S. (1999). Okapi/Keenbow at TREC-8. In: TREC. 1999, pp. 151-162.

Robertson, S.E., Walker, S., Beaulieu, M. & Willett, P. (1999). Okapi at TREC-7: automatic ad hoc, filtering, VLC and interactive track. Nist Special Publication SP. p.pp. 253-264.

Rodríguez-García, M.Á., Valencia-García, R. & García-Sánchez, F. (2012). An Ontology Evolution-Based Framework for Semantic Information Retrieval. In: P. Herrero, H. Panetto, R. Meersman, & T. Dillon (eds.). On the Move to Meaningful Internet Systems: OTM 2012 Workshops. Lecture Notes in Computer Science. [Online]. Springer Berlin

Heidelberg, pp. 163-172. Available from: http://link.springer.com/chapter/10.1007/978-3-642-33618-8_25. [Accessed: 20 May 2014].

Rodríguez-García, M.Á., Valencia-García, R., García-Sánchez, F. & Samper-Zapater, J.J. (2014a). Creating a semantically-enhanced cloud services environment through ontology evolution. Future Generation Computer Systems. 32. p.pp. 295-306.

Rodríguez-García, M.Á., Valencia-García, R., García-Sánchez, F. & Samper-Zapater, J.J. (2014b). Ontology-based annotation and retrieval of services in the cloud. Knowledge-Based Systems. 56. p.pp. 15-25.

Rudolph Gockel (1613). Lexicon Philosophicum.

Ruiz-Martínez, J.M., Valencia-García, R., Martínez-Béjar, R. & Hoffmann, A. (2012). BioOntoVerb: A top level ontology based framework to populate biomedical ontologies from texts. Knowledge-Based Systems. 36. p.pp. 68-80.

Ryu, P.-M. & Choi, K.-S. (2006). Taxonomy learning using term specificity and similarity. In: Proceedings of the 2nd Workshop on Ontology Learning and Population: Bridging the Gap between Text and Knowledge. 2006, pp. 41-48.

Salton, G. & McGill, M.J. (1983). Introduction to modern information retrieval.

Salton, G. & Yang, C.-S. (1973). On the Specification of Term Values in Automatic Indexing. Journal of Documentation. 29. p.pp. 351-372.

Salton, G., Fox, E.A. & Wu, H. (1983). Extended Boolean information retrieval. Communications of the ACM. 26 (11). p.pp. 1022-1036.

Sanderson, M. & Croft, B. (1999). Deriving Concept Hierarchies from Text. In: Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. SIGIR '99. [Online]. 1999, New York, NY, USA: ACM, pp. 206-213. Available from: http://doi.acm.org/10.1145/312624.312679. [Accessed: 12 May 2014].

Santos, D., Seco, N., Cardoso, N. & Vilela, R. (2006). HAREM: An Advanced NER Evaluation Contest for Portuguese. In: Proceedings of the 5th International Conference on Language Resources and Evaluation, LREC'2006. 2006, pp. 1986-1991.

Schlobach, S., Schlobach, and S., Olsthoorn, M. & Rijke, M.D. (2004). Type Checking in Open-Domain Question Answering. In: In Proceedings of European Conference on Artificial Intelligence. 2004, IOS Press, pp. 398-402.

Schubert, L., Jeffery, K. & Neidecker-Lutz, B. (2012). Advanced Cloud Technologies under H2020.

Seaborne, A. (2004). RDQL-a query language for RDF. W3C Member submission. 9 (29-21). p.p. 33.

Sekine, S. (1998). NYU: Description of the Japanese NE system used for MET-2. In: Proc. of the Seventh Message Understanding Conference (MUC-7. 1998.

Shamsfard, M. & Barforoush, A.A. (2004). Learning Ontologies from Natural Language Texts. Int. J. Hum.-Comput. Stud. 60 (1). p.pp. 17-63.

Shannon, C.E. (1948). A Mathematical Theory of Communication. Bell System Technical Journal. 27 (3). p.pp. 379-423.

Shannon, C.E., McCarthy, J. & Ashby, W.R. (1956). Automata studies. Princeton University Press Princeton, NJ.

Shinyama, Y. & Sekine, S. (2004). Named Entity Discovery Using Comparable News Articles. In: Proceedings of the 20th International Conference on Computational Linguistics. COLING '04. [Online]. 2004, Stroudsburg, PA, USA: Association for Computational Linguistics. Available from: http://dx.doi.org/10.3115/1220355.1220477. [Accessed: 1 June 2014].

Simon, H.A., Shaw, John Clifford & Newell, A. (1959). Report on a General Problem-solving Program. Rand Corporation.

Singhal, A., Choi, J., Hindle, D., Lewis, D.D. & Pereira, F. (1999). AT&T at TREC-7. In: PROCEEDINGS OF THE SEVENTH TEXT RETRIEVAL CONFERENCE (TREC-7. 1999, pp. 239-252.

Sirin, E. & Parsia, B. (2004). Pellet: An owl dl reasoner. In: Proc. of the 2004 Description Logic Workshop (DL 2004). 2004, pp. 212-213.

Solorio, T. (2005). Improvement of Named Entity Tagging by Machine Learning. Optics and Electronics. Puebla, Mexico: National Institute of Astrophysics.

Sowa, J.F. (2000). Ontology, Metadata, and Semiotics. In: B. Ganter & G. W. Mineau (eds.). Conceptual Structures: Logical, Linguistic, and Computational Issues. Lecture Notes in Computer Science. [Online]. Springer Berlin Heidelberg, pp. 55-81. Available from: http://link.springer.com/chapter/10.1007/10722280_5. [Accessed: 11 May 2014].

Staab, S., Studer, R., Schnurr, H.-P. & Sure, Y. (2001). Knowledge Processes and Ontologies. IEEE Intelligent Systems. 16 (1). p.pp. 26-34.

Stankovic, M. (2010). Open innovation and semantic web: Problem solver search on linked data. In: Proceedings of International Semantic Web Conference (ISWC) 7th-11th Novebmer, Shanghai, China. 2010, Citeseer.

Steels, L. (1992). Corporate Knowledge Management. In: AIFIPP. 1992, pp. 91-116.

Steve, G., Gangemi, A. & Pisanelli, D.M. (1997). Integrating medical terminologies with ONIONS methodology. In: Information Modelling and Knowledge Bases VIII (IOS. 1997, Press.

Stojanovic, L. (2004). Methods and Tools for Ontology Evolution. University of Karlsruhe, Germany.

Stojanovic, L., Maedche, A., Motik, B. & Stojanovic, N. (2002). User-Driven Ontology Evolution Management. In: Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web. EKAW '02. [Online]. 2002, London, UK, UK: Springer-Verlag, pp. 285-300. Available from: http://dl.acm.org/citation.cfm?id=645362.650868. [Accessed: 30 May 2014].

Stojanovic, L., Maedche, A., Stojanovic, N. & Studer, R. (2003). Ontology Evolution As Reconfiguration-design Problem Solving. In: Proceedings of the 2Nd International Conference on Knowledge Capture. K-CAP '03. [Online]. 2003, New York, NY, USA: ACM, pp. 162-171. Available from: http://doi.acm.org/10.1145/945645.945669. [Accessed: 30 May 2014].

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

Tanev, H. & Magnini, B. (2008). Weakly Supervised Approaches for Ontology Population. In: Proceedings of the 2008 Conference on Ontology Learning and Population: Bridging the Gap Between Text and Knowledge. [Online]. 2008, Amsterdam, The Netherlands, The Netherlands: IOS Press, pp. 129-143. Available from: http://dl.acm.org/citation.cfm?id=1563823.1563835. [Accessed: 13 May 2014].

Tissaoui, A., Aussenac-Gilles, N., Hernandez, N. & Laublet, P. (2011). EvOnto - Joint Evolution of Ontologies and Semantic Annotations. In: KEOD. 2011, pp. 226-231.

Tjong Kim Sang, E.F. (2002). Introduction to the CoNLL-2002 Shared Task: Language-independent Named Entity Recognition. In: Proceedings of the 6th Conference on Natural Language Learning - Volume 20. COLING-02. [Online]. 2002, Stroudsburg, PA, USA: Association for Computational Linguistics, pp. 1-4. Available from: http://dx.doi.org/10.3115/1118853.1118877. [Accessed: 1 June 2014].

Turney, P. (2001). Mining the Web for Synonyms: PMI-IR versus LSA on TOEFL. In: L. De Raedt & P. Flach (eds.). [Online]. 2001, Springer-Verlag, pp. 491-502. Available from: http://cogprints.org/1796/. [Accessed: 11 May 2014].

Uren, V., Cimiano, P., Iria, J., Handschuh, S., Vargas-Vera, M., Motta, E. & Ciravegna, F. (2006). Semantic annotation for knowledge management: Requirements and a survey of the state of the art. Web Semantics: Science, Services and Agents on the World Wide Web. 4 (1). p.pp. 14-28.

V. Devedzic & D. Gasevic (eds.) (2009). Web 2.0 & Semantic Web. Annals of Information Systems. Springer.

Valencia-García, R., Fernández-Breis, J.T., Ruiz-Martínez, J.M., García-Sánchez, F. & Martínez-Béjar, R. (2008). A knowledge acquisition methodology to ontology construction for information retrieval from medical documents. Expert Systems. 25 (3). p.pp. 314-334.

Valencia-García, R., García-Sánchez, F., Castellanos-Nieves, D. & Fernández-Breis, J.T. (2011). OWLPath: An OWL ontology-guided query editor. Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on. 41 (1). p.pp. 121-136.

Van Heijst, G., Schreiber, A.T. & Wielinga, B.J. (1997). Using explicit ontologies in KBS development. International Journal of Human-Computer Studies. 46 (2-3). p.pp. 183-292.

Vargas-Vera, M., Motta, E., Domingue, J., Lanzoni, M., Stutt, A. & Ciravegna, F. (2002). MnM: Ontology Driven Semi-automatic and Automatic Support for Semantic Markup. In: Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web. EKAW '02. [Online]. 2002, London, UK, UK: Springer-Verlag, pp. 379-391. Available from: http://dl.acm.org/citation.cfm?id=645362.650860. [Accessed: 1 June 2014].

Velardi, P., Navigli, R., Cuchiarelli, A. & Neri, R. (2005). Evaluation of OntoLearn, a methodology for automatic learning of domain ontologies. Ontology Learning from Text: Methods, evaluation and applications. p.pp. 92-106.

Velasco, J.L., Valencia-García, R., Fernández-Breis, J.T. & Toval, A. (2009). Modelling reusable security requirements based on an ontology framework. Journal of Research and Practice in Information Technology. 41 (2). p.p. 119.

Virtuoso, O. (2009). Universal server platform for the real-time enterprise. [Online]. Available from: http://virtuoso.openlinksw.com. [Accessed: 10 June 2014].

Vivaldi, J., Màrquez, L. & Rodríguez, H. (2001). Improving Term Extraction by System Combination using Boosting.

Vossen, P. (1998). EuroWordNet: building a multilingual database with wordnets for European languages. The ELRA Newsletter. 3 (1). p.pp. 7-10.

W3C (2012a). Owl 2 functional syntax. [Online]. Available from: http://www.w3.org/TR/2012/REC-owl2-syntax-20121211/. [Accessed: 30 May 2014].

W3C (2012b). Owl 2 xml serialization. [Online]. Available from: http://www.w3.org/TR/owl2-xml-serialization/. [Accessed: 30 May 2014].

W3C (2014). Owl 2 rdf/xml syntax specification. [Online]. Available from: http://www.w3.org/TR/2004/REC-rdf-syntax-grammar-20040210/. [Accessed: 30 May 2014].

Wahl, T. & Sindre, G. (2009). A survey of development methods for semantic web service systems. International Journal of Information Systems in the Service Sector (IJISSS). 1 (2). p.pp. 1-16.

Weglarz, G. (2004). Two Worlds Data-Unstructured and Structured. DM REVIEW. 14. p.pp. 19-23.

Winograd, T. (1972). Understanding natural language. Cognitive Psychology. 3 (1). p.pp. 1-191.

Wong, W., Liu, W. & Bennamoun, M. (2012). Ontology Learning from Text: A Look Back and into the Future. ACM Comput. Surv. 44 (4). p.pp. 20:1-20:36.

Woods, W.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. AFIPS '73. [Online]. 1973, New York, NY, USA: ACM, pp. 441-450. Available from: http://doi.acm.org/10.1145/1499586.1499695. [Accessed: 31 May 2014].

Xu, Z., Luo, X., Yu, J. & Xu, W. (2011). Measuring Semantic Similarity Between Words by Removing Noise and Redundancy in Web Snippets. Concurr. Comput.?: Pract. Exper. 23 (18). p.pp. 2496-2510.

Yang, H. & Callan, J. (2008). Human-guided ontology learning. Proceedings of Human-Computer Interaction and Information Retrieval (HCIR). p.pp. 26-29.

Yang, H.-C. (2009). Automatic Generation of Semantically Enriched Web Pages by a Text Mining Approach. Expert Syst. Appl. 36 (6). p.pp. 9709-9718.

Yang, Y. & Liu, X. (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. SIGIR '99. [Online]. 1999, New York, NY, USA: ACM, pp. 42-49. Available from: http://doi.acm.org/10.1145/312624.312647. [Accessed: 27 May 2014].

Zeni, N., Kiyavitskaya, N., Mich, L., Mylopoulos, J. & Cordy, J.R. (2007). A Lightweight Approach to Semantic Annotation of Research Papers. In: Z. Kedad, N. Lammari, E. Métais, F. Meziane, & Y. Rezgui (eds.). Natural Language Processing and Information Systems. Lecture Notes in Computer Science. [Online]. Springer Berlin Heidelberg, pp. 61-72. Available from: http://link.springer.com/chapter/10.1007/978-3-540-73351-5_6. [Accessed: 2 June 2014].

Zhang, L. (2013). Content-based filtering for semi-structured documents. University of California at Santa Cruz, Santa Cruz, CA, USA.

Zhou, G. & Su, J. (2002). Named Entity Recognition Using an HMM-based Chunk Tagger. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics. ACL '02. [Online]. 2002, Stroudsburg, PA, USA: Association for Computational Linguistics, pp. 473-480. Available from: http://dx.doi.org/10.3115/1073083.1073163. [Accessed: 1 June 2014].

Zhu, J., Huang, X., Song, D. & Rüger, S. (2010). Integrating multiple document features in language models for expert finding. Knowledge and Information Systems. 23 (1). p.pp. 29-54.