Universidad de Alcalá

DEPARTAMENTO DE AUTOMÁTICA
Edificio Politécnico
28871 Alcalá de Henares (Madrid)
Teléfono: 91 885 65 94
Fax: 91 885 69 23
secre@aut.uah.es

El Dr. D. Agustín Martínez Hellín, Profesor Titular de la Universidad de Alcalá y Director del Departamento de Automática

HACE CONSTAR:

Que la Tesis Doctoral titulada, **"Contribución a la Alineación de Ontologías utilizando Lógica Difusa"**, presentada por Dª. Susel Fernández Melián, y dirigida por el Dr. D. Juan Ramón Velasco Pérez y el Dr. D. Iván Marsá Maestre, cumple con todos los requisitos científicos y metodológicos para ser defendida ante un tribunal.

Alcalá de Henares, octubre de 2013.

Fdo: Dr. D. Agustín Martínez Hellín
Director del Departamento de Automática

Universidad de Alcalá

DEPARTAMENTO DE AUTOMÁTICA
Edificio Politécnico
28871 Alcalá de Henares (Madrid)
Teléfono: 91 885 65 94
Fax: 91 885 69 23
secre@aut.uah.es

El Dr. D. Juan Ramón Velasco Pérez, Catedrático de Universidad del departamento de Automática de la Universidad de Alcalá y el Dr. D. Iván Marsá Maestre, Profesor Ayudante Doctor del departamento de Automática de la Universidad de Alcalá

HACEN CONSTAR:

Que la Tesis Doctoral titulada, **"Contribución a la Alineación de Ontologías utilizando Lógica Difusa"**, presentada por Dª. Susel Fernández Melián, y realizada bajo la dirección del Dr. D. Juan Ramón Velasco Pérez, y el Dr. D. Iván Marsá Maestre, reúne méritos suficientes para optar al grado de Doctor, por lo que puede procederse a su depósito y lectura.

Alcalá de Henares, octubre de 2013.

Fdo: Dr. D. Juan Ramón Velasco Pérez

Fdo: Dr. D. Iván Marsá Maestre

ESCUELA POLITÉCNICA SUPERIOR

Departamento de Automática

CONTRIBUCIÓN A LA ALINEACIÓN DE ONTOLOGÍAS UTILIZANDO LÓGICA DIFUSA

TESIS DOCTORAL

Susel Fernández Melián

Alcalá de Henares, 2013

Universidad de Alcalá

ESCUELA POLITÉCNICA SUPERIOR

Departamento de Automática

TESIS DOCTORAL

CONTRIBUCIÓN A LA ALINEACIÓN DE ONTOLOGÍAS UTILIZANDO LÓGICA DIFUSA

Autor:

Susel Fernández Melián

Ingeniera en Informática

Directores:

Juan Ramón Velasco Pérez

Dr. Ingeniero de Telecomunicación

Iván Marsá Maestre

Dr. Ingeniero de Telecomunicación

Alcalá de Henares, 2013

El único autógrafo digno de un hombre es el que deja escrito con sus obras

José Martí

Dedicado a mi familia

Resumen

En la actualidad, con el aumento de la cantidad de información disponible en Internet, se hace cada vez más necesario crear mecanismos para facilitar la organización el intercambio de información y conocimiento entre las aplicaciones. La Web Semántica está destinada a resolver una de las carencias fundamentales de la Web actual: la falta de capacidad de las representaciones para expresar significados. Esta tarea se puede simplificar enormemente añadiendo información semántica y de contexto a las formas actuales de representación del conocimiento utilizadas en la Web, de modo que los equipos puedan procesar, interpretar y conectar la información presentada en la WWW.

Las ontologías se han convertido en un componente crucial dentro de la Web semántica, ya que permiten el diseño de exhaustivos y rigurosos esquemas conceptuales para facilitar la comunicación y el intercambio de información entre diferentes sistemas y entidades. Sin embargo, la heterogeneidad en la representación del conocimiento en las ontologías dificulta la interacción entre las aplicaciones que utilizan este conocimiento. Por ello, para compartir información cuando se utilizan vocabularios heterogéneos se debe poder traducir los datos de un marco ontológico a otro. El proceso de encontrar correspondencias entre ontologías diferentes se conoce como *alineación de ontologías*.

En esta tesis doctoral se propone un método de alineación de ontologías utilizando técnicas de lógica difusa para combinar diversas medidas de similitud entre entidades de ontologías diferentes. Las medidas de similitud propuestas se basan en dos elementos fundamentales de las ontologías: la terminología y la estructura. En cuanto a la terminología se propone una medida de similitud lingüística utilizando varias relaciones léxicas entre los nombres de las entidades, combinada con una medida de similitud semántica que tiene en cuenta la información del contexto de las entidades en las ontologías. En cuanto a la estructura se proponen medidas de similitud que utilizan tanto la estructura relacional como la estructura interna de los conceptos en las ontologías.

Abstract

With the increasing amount of information available on the Internet today, it is becoming critical to create mechanisms to facilitate the organization and to enable the exchange of information and knowledge between applications. The Semantic Web is intended to solve one of the fundamental limitations of the current Web: the lack of ability for representations to express meanings. This task can be simplified greatly by adding semantic information and context to current forms of knowledge representation used in the Web, so that computers can process, interpret and connect the information in the WWW.

Ontologies have become a crucial component in the Semantic Web, allowing the design of exhaustive and rigorous conceptual schemas to facilitate communication and information exchange between different systems and institutions. However, the heterogeneity in knowledge representation in ontologies hampers the interaction between the applications that make use of this knowledge. Therefore, to share information between applications using heterogeneous vocabularies, they must be able to translate data from one ontological framework to another. The process of finding correspondences between different ontologies is called *ontology alignment*.

This Ph.D. thesis proposes an ontology alignment method using fuzzy logic techniques to combine a set of similarity measures between different ontologies entities. The proposed similarity measures are based on two fundamental elements of ontologies: the terminology and structure. Regarding terminology we propose a linguistic similarity measure using a set of lexical relations between the names of the entities combined with a semantic similarity measure that takes into account the context information in ontology entities. In terms of structure, we propose similarity measures which use both relational structure and the internal structure of concepts within ontologies.

Agradecimientos

Son muchas las personas e instituciones que a lo largo de este tiempo han contribuido de una forma u otra a la realización de este trabajo y para las cuales me gustaría tener unas palabras de agradecimiento. En primer lugar quiero dar las gracias a mi familia, por su apoyo incondicional desde la distancia, por animarme a seguir cuando flaqueaban las fuerzas y por ser el motor impulsor de todo lo que hago.

Agradezco de manera especial a mis directores de tesis Juan Ramón Velasco e Iván Marsá, por sus consejos y su apoyo no sólo en las cuestiones técnicas, sino también en el ámbito personal. También al grupo de Investigación de Ingeniería de Servicios Telemáticos del departamento de automática de la universidad de Alcalá, especialmente a los profesores Miguel Angel López Carmona y Enrique de la Hoz.

No puedo dejar de mencionar a los compañeros que he tenido en la sala e24 durante estos años. Algunos han estado poco tiempo y otros más, pero todos han contribuido de alguna manera a mi crecimiento como persona y han hecho que el trabajo haya sido divertido. Entre ellos quisiera agradecer especialmente la ayuda brindada por Diego Rivera, Germán López, Mario Vega y Elisa Rojas. También quiero agradecer a mis amigos de electrónica Fernando Valdés, Katia Cela, Pedro Fernández, Carlos Luna y Jose Luis Lázaro por sus consejos y sus palabras de ánimo, sobre todo en la recta final.

A Cuba, el país donde nací y donde me he formado, en especial a la Universidad de Oriente, donde cursé mis estudios universitarios. A España y a la Universidad de Alcalá por haber financiado mi investigación.

Finalmente quisiera agradecer a todas las personas que han formado parte de mi vida en este tiempo. A los que se han ido, a los que se han quedado, a los que han compartido al menos un café, a esos amigos con los que no he podido compartir todo lo que me hubiese gustado, por estar trabajando en la tesis y a todos aquellos que de una forma u otra han contribuido a que me sienta como en casa, a pesar de estar tan lejos. Muchas gracias de corazón.

Índice

Índice de figuras

Índice de tablas

CAPÍTULO 1. INTRODUCCIÓN

Internet es la base del desarrollo futuro, como hace un siglo lo fue la electricidad

Manuel Castells

En este capítulo introducimos brevemente el ámbito de desarrollo en el que se enmarca esta tesis doctoral, comenzando con una explicación de la evolución de la Web desde sus inicios hasta la actualidad. Mencionamos las carencias de la Web actual y la necesidad de añadir significado a la misma para llegar a una Web superior: la Web Semántica. Presentamos las ontologías como formas de representación del conocimiento con un papel crucial dentro de la Web Semántica e introducimos el problema de la alineación de ontologías para compartir información en este entorno. Finalmente describimos los objetivos que se articulan alrededor de dicho problema y la estructura de la tesis doctoral.

1.1 Antecedentes

La World Wide Web (WWW) es una enorme biblioteca de documentos relacionados entre sí que son transferidos por los ordenadores y presentados a los usuarios. Fue creada alrededor de 1989 por Tim Berners-Lee [Berners-Lee, 1989] con ayuda de Robert Cailliau en el Consejo Europeo para la Investigación Nuclear (*CERN*, Ginebra, Suiza) y surge de los sistemas de hipertexto [Nelson, 1965], pero la diferencia es que cualquiera puede acceder y contribuir a ella, lo que la ha convertido en un medio extraordinariamente flexible y económico para la comunicación, el comercio, el entretenimiento, el acceso a información y servicios, la difusión de la cultura, etc.

El éxito de la Web se basa principalmente en su simplicidad, proporcionando a los desarrolladores de software, proveedores de información y usuarios un fácil acceso a nuevos contenidos [Fensel y Musen, 2001]. Sin embargo, la misma sencillez que hizo posible la impresionante expansión de la Web ha traído importantes (y en algunos casos críticos) inconvenientes que impiden un mayor desarrollo de esta tecnología. La WWW actual contiene una gran cantidad de información y conocimiento, pero las máquinas por lo general sólo sirven para entregar y presentar el contenido de los documentos que describen el conocimiento, además de que sus capacidades son limitadas para realizar algunas tareas, que

suelen tardar un tiempo excesivo en ejecutarse [Obitko, 2007]. Actualmente los usuarios tienen que conectar e interpretar por sí mismos la mayoría de las fuentes de información relevante, teniendo en cuenta que la calidad de la información o incluso la persistencia de los documentos generalmente no puede ser garantizada.

Una de las carencias fundamentales de la Web actual es la falta de capacidad de las representaciones para expresar significados. Por ejemplo, si hacemos una búsqueda de la palabra "*sun*" en la Web nos aparecerán todos los documentos relacionados con la empresa *Sun Microsystems* y sus productos, como por ejemplo *Java*, pero también información sobre el sol, como estrella del sistema solar, entre otras cosas. Dentro de todo este conjunto de documentos recuperados, probablemente encontremos lo que buscamos, pero también habrá una gran cantidad de información irrelevante que tendremos que descartar nosotros mismos analizando los resultados. Esta laboriosa tarea se puede simplificar enormemente añadiendo información semántica y de contexto a las formas actuales de representación del conocimiento utilizadas en la Web.

La Web Semántica [Berners-Lee *et al*., 2001] es un esfuerzo para mejorar la Web actual para que los equipos puedan procesar la información presentada en la WWW, interpretarla, conectarla y así ayudar a las personas a encontrar el conocimiento requerido, mediante la incorporación de contenido semántico. Hereda el poder de representación de conceptos existentes como las redes semánticas [Sowa, 1991] y realza la interoperabilidad tanto en el nivel sintáctico como en el semántico. La Web Semántica está destinada a formar un enorme sistema distribuido basado en el conocimiento con el objetivo de proporcionar un marco común que permita que los datos sean compartidos y reutilizados a través de aplicaciones, empresas, y comunidades. Es considerada como un integrador de contenidos a través de diferentes aplicaciones y sistemas de información y puede ser muy útil en la industria editorial, los *blogs* y muchas otras áreas relacionadas con Internet. Berners-Lee [Berners-Lee y Fischetti, 1999] originalmente expresó su visión de la Web Semántica como sigue:

> *Yo tengo un sueño para la Web en el que los ordenadores llegan a ser capaces de analizar todos los datos -el contenido, enlaces y transacciones entre personas y ordenadores. Una Web Semántica, que haga esto posible, todavía no ha surgido, pero cuando lo haga, los mecanismos del día a día del comercio, la burocracia y nuestra vida cotidiana serán manejados por las máquinas.*

Uno de los componentes cruciales de la Web semántica son las ontologías, que permiten el diseño de exhaustivos y rigurosos esquemas conceptuales para facilitar la comunicación y el intercambio de información entre diferentes sistemas y entidades. Aunque el término ontología proviene de la filosofía, hace algún tiempo se ha extendido su uso al campo de la inteligencia artificial, donde se ha definido como la descripción formal y explícita de los elementos pertenecientes a un dominio, como son los conceptos, propiedades, relaciones, funciones, axiomas y reglas de inferencia de dicho dominio [Studer *et al*., 1998].

La investigación sobre el uso de ontologías en la Web Semántica promete una mayor interoperabilidad entre agentes software y servicios Web, permitiendo el descubrimiento automatizado de servicios basado en contenidos. Sin embargo, la heterogeneidad en la representación del conocimiento en las ontologías dificulta la interacción entre las aplicaciones que utilizan este conocimiento. Los servicios producidos y descritos por diferentes desarrolladores pueden utilizar conjuntos de ontologías distintos o quizás parcialmente solapados. Los usuarios pueden almacenar sus datos en diversas estructuras y utilizar diferentes términos para representar el mismo concepto, por lo que, para compartir información, cuando se utilizan vocabularios heterogéneos se debe poder traducir los datos de un marco ontológico a otro. El proceso de encontrar correspondencias entre ontologías diferentes se conoce como *alineación de ontologías* y es crucial para el éxito de la Web Semántica.

El proceso de alineación de ontologías se ha llevado a cabo empleando técnicas muy diversas, como por ejemplo la lingüística [Noy y Musen, 1999], las probabilidades [Noessner *et al*., 2010], el aprendizaje automático [Doan *et al*., 2004], entre otras. El objetivo general de esta tesis es la automatización del proceso de alineación de ontologías combinando diversas medidas de similitud utilizando técnicas de lógica difusa.

La lógica difusa, o lógica borrosa, es en la actualidad un campo de investigación muy importante tanto por sus aplicaciones matemáticas y teóricas como por sus aplicaciones prácticas. Es una extensión de la lógica tradicional que se asemeja más al razonamiento humano, ya que utiliza etiquetas lingüísticas asociadas a conjuntos borrosos. A diferencia de la lógica clásica, donde las proposiciones son verdaderas o falsas, en la lógica difusa se establecen áreas complejas que se solapan unas con otras (mucho más ajustadas a la realidad) y se evalúa el grado de pertenencia de las variables a estas áreas. Por ejemplo si consideramos como variable difusa la estatura, una persona no será simplemente *alta* o *baja*, sino que

participará de ambas características parcialmente, de tal forma que sólo por encima y por debajo de determinadas alturas se clasificaría estrictamente como *alta* o *baja*, mientras que en la zona intermedia de ambas alturas existirá una graduación por la que va dejando de ser *alta* o *baja*, dependiendo del grado de pertenencia de su estatura a cada conjunto difuso.

Los sistemas basados en reglas difusas constituyen una extensión de los sistemas basados en reglas clásicos debido a que se componen de reglas *IF-THEN* cuyos antecedentes y consecuentes están compuestos por instrucciones de lógica difusa en lugar de instrucciones de lógica clásica. Estos sistemas se usan como herramientas para representar diferentes formas de conocimiento de un problema, así como para modelar las interacciones y relaciones entre sus variables. Debido a esta propiedad, han sido exitosamente aplicados a un amplio rango de problemas en diferentes dominios con incertidumbre y conocimiento incompleto [Cordón *et al*., 2001]. En esta tesis se pretende hacer uso de los sistemas basados en reglas difusas para combinar diferentes medidas de similitud entre entidades en el proceso de alineación de ontologías.

1.2 El problema de la alineación de ontologías

A medida que crece la información disponible en la Web, la heterogeneidad en la representación del conocimiento en diferentes ontologías dificulta la interoperabilidad entre las aplicaciones que utilizan este conocimiento. La interoperabilidad semántica puede ser recuperada encontrando relaciones entre entidades pertenecientes a diferentes ontologías y utilizando esas relaciones. Por ejemplo, si tenemos dos bibliotecas electrónicas, cada una con su propia ontología, y necesitamos consultar información sobre alguna publicación o usuario, es necesario que exista un mecanismo que permita identificar las correspondencias entre las entidades de una ontología y las de la otra.

En la Figura 1.1 se muestran con líneas discontinuas las posibles correspondencias identificadas para dos fragmentos de ontologías del dominio de las bibliotecas. Como se puede observar, se identifican las equivalencias simples, que son aquellos casos en que los conceptos tienen los mismos nombres y se encuentran en la misma posición dentro de la estructura de la ontología, como por ejemplo: *Publication-Publication* y *Book-Book*, pero también se pueden identificar conceptos equivalentes a pesar de no tener el mismo nombre o la misma ubicación como por ejemplo: *User-Student*, *Magazine-Journal* y *Article-Paper*.

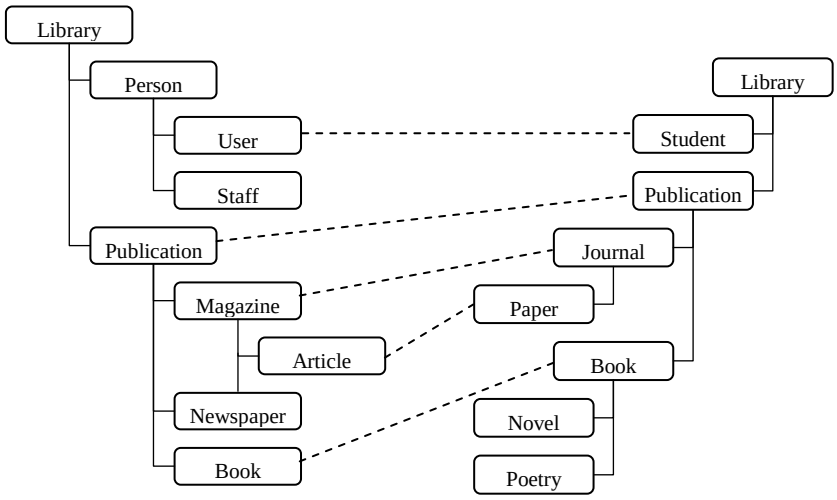


Figura 1.1. Correspondencias entre conceptos de dos ontologías.

El problema de la alineación de ontologías puede ser descrito en una frase: dadas dos ontologías que describen un conjunto de entidades discretas (que pueden ser clases, propiedades, reglas, predicados, o incluso fórmulas) es el proceso de encontrar las correspondencias entre ellas, por ejemplo, equivalencia o pertenencia [Euzenat y Shvaiko, 2007].

La alineación de ontologías es un proceso muy útil dentro de la Web Semántica. Se puede aplicar en la tecnología de agentes inteligentes con el fin de capacitarlos para la tarea de recopilar e integrar información procedente de fuentes distintas, así como para buscar servicios en la Web y utilizarlos automáticamente sin intervención del usuario [Kalfoglou y Schorlemmer, 2003]. De cara al comercio electrónico, donde las ontologías juegan un papel importante, se puede integrar y catalogar la información procedente de empresas distintas y heterogéneas, unificando descripciones de productos, procesos, transacciones, etc. A todo lo anterior hay que sumarle la gran aportación que supone la posibilidad de automatizar muchos procesos, al dotar a los datos de una semántica formal, comprensible por las máquinas. Por ejemplo, las aplicaciones de diversas empresas podrían trabajar automáticamente con fuentes de datos distintas, incluso comparar entre distintas ontologías, para transferir o intercambiar información entre ellas sin tener que ponerse de acuerdo previamente sobre el significado de los datos.

1.3 Objetivos de la tesis

La búsqueda de similitudes entre conceptos en las ontologías es un problema que involucra incertidumbre, puesto que los datos que se manejan constituyen interpretaciones de las personas sobre el dominio al que pertenecen y la información con que se cuenta puede ser ambigua, imprecisa o incompleta.

Uno de los mecanismos de inferencia más utilizados para el tratamiento de la incertidumbre es la inferencia difusa. Esto se debe a que la principal característica de la lógica difusa es que pretende producir resultados exactos a partir de datos imprecisos. Los valores de verdad utilizados en la lógica difusa generalmente tienen una connotación de incertidumbre, por lo que los sistemas basados en reglas difusas pueden ser una herramienta válida para combinar datos imprecisos, como lo son las diversas medidas de similitud entre conceptos para la alineación de ontologías. El objetivo fundamental de este trabajo es:

> *Proponer un método automático de alineación de ontologías aplicando técnicas de lógica difusa para combinar las similitudes entre entidades de ontologías diferentes teniendo en cuenta su terminología y su estructura.*

Para lograr este objetivo es necesario establecer otros más específicos, que a continuación detallamos, enmarcados en las dos temáticas fundamentales que abordamos en la tesis: la similitud terminológica y la similitud estructural.

1.3.1 Similitud Terminológica

Las ontologías sirven como vocabulario y como forma de representar y organizar el conocimiento en un determinado dominio. Estas son desarrolladas por expertos humanos, lo que hace que la terminología de los nombres de los conceptos y propiedades aporte una información muy valiosa sobre las entidades que modelan, y por lo tanto sea el primer elemento a tener en cuenta cuando se desea buscar correspondencias entre ellas.

La mayoría de las investigaciones existentes en la literatura en el campo de la alineación de ontologías se han centrado en obtener la similitud entre las entidades basándose en la terminología empleada. Por lo general, procesan los términos como elementos independientes utilizando técnicas basadas en cadenas, como las funciones de distancia [Noy y Musen, 2001].

También se han utilizado otras técnicas basadas en recursos lingüísticos externos como los tesauros y directorios especializados en diferentes dominios [Madhavan *et al*., 2001, Jean-Mary *et al*., 2009]. En estos dos ámbitos se han hecho numerosas aportaciones por separado, pero existen muy pocas medidas de similitud que combinen técnicas basadas en cadenas con las técnicas basadas en los recursos lingüísticos externos. En la línea de la similitud lingüística, el objetivo específico de esta tesis es:

> *Proponer medidas de similitud lingüísticas que utilicen tanto técnicas basadas en cadenas, como las relaciones entre las palabras dentro de los directorios léxicos.*

Otro aspecto importante en cuanto a la similitud terminológica es que existen muy pocos trabajos que utilicen información del contexto para el cálculo de las similitudes terminológicas entre las entidades de las ontologías. Una aproximación contextual podría mejorar estos valores de similitud, ya que no sólo se tendría en cuenta la construcción de los términos desde el punto de vista lingüístico, sino también la semántica del contexto en el que los términos son utilizados dentro de las ontologías. Por ejemplo el concepto *sun* no significa lo mismo en el contexto de una ontología sobre el espacio, que en el contexto de una ontología sobre informática. Debido a que los valores de la similitud lingüística y semántica suelen ser imprecisos, los sistemas basados en reglas difusas pueden constituir una herramienta idónea para combinarlos y así obtener un índice de similitud terminológica más efectivo. El objetivo de esta tesis en el ámbito de la similitud terminológica utilizando información contextual es:

> *Combinar la similitud lingüística con la información semántica del contexto de las entidades en las ontologías para obtener una similitud terminológica, utilizando técnicas de lógica difusa.*

La consecución de este objetivo permitirá diferenciar los casos en los que se utilizan los mismos términos o términos muy similares lingüísticamente en contextos diferentes, de aquellos casos en los que se emplean los mismos términos en el mismo contexto, reforzando el valor de la similitud en estos últimos casos. Este factor sería de gran utilidad por ejemplo en los foros de Internet, redes sociales y sistemas de recomendación ya que serviría de ayuda en el proceso de localización de información semejante para sugerir temas de interés común, facilitar las búsquedas y evitar la redundancia.

1.3.2 Similitud Estructural

La organización de los elementos dentro de las ontologías es otro factor importante a la hora de buscar correspondencias entre ellas. Los métodos de similitud estructural, a diferencia de los terminológicos, no tratan las entidades como elementos independientes sino que utilizan fundamentalmente técnicas basadas en las relaciones (estructura relacional) y en la composición (estructura interna) de las entidades en las ontologías.

Los enfoques estructurales relacionales, por lo general, utilizan técnicas basadas en grafos y en la organización jerárquica de los conceptos (taxonomía), que aprovechan la información de similitud que aporta la cercanía entre elementos dentro de dichos grafos. Estas medidas se basan por lo general en las distancias entre las entidades dentro de un mismo grafo conceptual, por lo que no pueden ser aplicadas si las ontologías no comparten taxonomía.

Por su parte, los enfoques que utilizan la estructura interna de los conceptos, aplican técnicas basadas en restricciones, como por ejemplo el tipo y la cardinalidad de los atributos. Debido a que el número de entidades que comparten estas restricciones puede ser muy grande, los métodos basados en la estructura interna se han utilizado comúnmente para crear grupos de alineaciones en lugar de para encontrar correspondencias exactas entre entidades, reduciendo así el número de comparaciones con la eliminación de candidatos incompatibles.

Teniendo esto en cuenta, sería deseable trabajar en métodos de similitud estructurales jerárquicos que utilicen información de diferentes taxonomías y combinarlos con métodos que utilicen la estructura interna para obtener valores de similitud más precisos. Como hemos comentado en el objetivo general, para combinar estas medidas de similitud imprecisas hemos elegido los sistemas basados en reglas difusas. Nuestros objetivos en cuanto a la similitud estructural son:

Proponer medidas de similitud estructurales, que utilicen la jerarquía taxonómica y la estructura interna en ontologías que no compartan la taxonomía y combinarlas utilizando técnicas de lógica difusa.

Utilizar la lógica difusa para el cálculo de la similitud entre las propiedades de los conceptos teniendo en cuenta su información terminológica, sus clases y sus tipos.

Con la incorporación de las medidas de similitud estructurales se puede contribuir a mejorar los valores de similitud entre conceptos, al tener en cuenta la influencia de las similitudes entre otros conceptos relacionados con ellos, como sus descendientes, padres y hermanos en las taxonomías, así como los factores estructurales internos como las similitudes entre sus propiedades, los tipos y la cardinalidad. Por ejemplo, en las ontologías de la Figura 1.1, para calcular la similitud entre los conceptos *Journal* y *Magazine*, además de su similitud terminológica sería útil tener en cuenta la similitud entre sus padres (e.g. *Publication* - *Publication*), sus descendientes (e.g. *Article* - *Paper*) y sus hermanos (e.g. *Book* - *Book*), así como las similitudes entre sus propiedades.

Con la consecución del segundo objetivo se permitirá utilizar la lógica difusa para calcular la correspondencia directa entre las propiedades como entidades independientes dentro de las ontologías, que es un aspecto de gran utilidad en el proceso de traducción. Por ejemplo, en los sistemas multiagente, cuando se necesita realizar peticiones relacionadas con varias ontologías se requiere una traducción exacta de cada propiedad involucrada en la consulta. Finalmente, la consecución de este objetivo permitirá disponer de un sistema basado en reglas difusas que sea capaz de combinar datos imprecisos para mejorar los valores de similitud entre los conceptos teniendo en cuenta la influencia de factores internos.

1.4 Breve descripción del contenido de la tesis

La memoria de la tesis doctoral está estructurada en 6 capítulos. En el Capítulo 2 se introduce la Web Semántica y se presenta el concepto de ontología, su estructura y los lenguajes utilizados para su construcción. Se realiza una breve revisión del estado de la investigación en el campo de la alineación de ontologías, clasificando las principales aportaciones encontradas en la literatura de acuerdo con las diferentes técnicas utilizadas. Finalmente se plantean los principales desafíos existentes en la actualidad en el campo de la alineación de ontologías.

Debido a que uno de los desafíos en este campo es encontrar medidas de similitud que permitan obtener correspondencias más precisas entre las entidades de ontologías diferentes, el Capítulo 3 comienza con un estudio sobre las principales medidas de similitud y distancia existentes en la literatura. Hacemos especial hincapié en las técnicas que utilizan herramientas léxicas externas para el cálculo de correspondencias entre entidades de diferentes ontologías. Por último se presentan las medidas de similitud terminológicas y estructurales propuestas en

la tesis, que son combinadas en un sistema basado en reglas difusas de varias capas para la alineación de ontologías.

El Capítulo 4 explica el sistema multicapa basado en reglas difusas, propuesto para resolver el problema de la alineación de ontologías utilizando las medidas de similitud terminológicas y estructurales explicadas en el capítulo anterior. Este capítulo comienza con una introducción a la lógica difusa, donde se definen los conceptos fundamentales y las características de los sistemas basados en reglas difusas, para después abordar la descripción de la arquitectura propuesta.

En el Capítulo 5 se detallan los experimentos realizados para la validación de la propuesta. La fase experimental se divide en tres etapas. En la primera, con el objetivo de evaluar la similitud terminológica, utilizamos pequeños fragmentos de taxonomías disponibles en la Web. La segunda tiene el objetivo de evaluar el sistema en ontologías reales, para lo cual utilizamos los tests propuestos en la plataforma de la *OAEI* (*Ontology Alignment Evaluation Initiative*)[1] para la evaluación de los métodos de alineación de ontologías. Los test de la *OAEI* incluyen ontologías generales y ontologías del dominio de la medicina. La tercera etapa consiste en aplicar el método a ontologías reales de otros dominios diferentes de la medicina, para lo cual hemos elegido el dominio de las redes de sensores.

En el Capítulo 6 se presentan las conclusiones, las principales contribuciones de la tesis, los diferentes pasos en la difusión de la investigación y las líneas de trabajo futuro que pueden desprenderse de su realización.

[1] *http://oaei.ontologymatching.org/*

CAPÍTULO 2. ALINEACIÓN DE ONTOLOGÍAS EN LA WEB SEMÁNTICA

No hemos visto la Web como la visualicé. El futuro es aún mucho más grande que el pasado

Tim Berners-Lee

En este capítulo tratamos en detalle el tema de la alineación de ontologías como elemento fundamental en el desarrollo de la Web Semántica. Comenzamos explicando la arquitectura de la Web Semántica, centrándonos después en el papel que juegan en ella las ontologías. Mencionamos algunas de las definiciones más conocidas que se han dado a las ontologías y nos centramos fundamentalmente en su definición para el campo de la inteligencia artificial. También se detallan los elementos que componen su estructura, los lenguajes más utilizados para su construcción y manejo, y se explican las tres estrategias fundamentales utilizadas en el proceso de integración de ontologías: la fusión, el mapeo y la alineación. Finalmente, se mencionan los trabajos más relevantes sobre el tema y se clasifican según las técnicas empleadas.

2.1 Arquitectura de la Web Semántica

La Web Semántica es una Web extendida y dotada de mayor significado que la Web actual, apoyada en lenguajes universales que en el futuro van a permitir que los usuarios puedan encontrar respuestas a sus preguntas de forma más rápida y sencilla gracias a una información mejor definida. Con esta Web se podrán delegar tareas en el software que será capaz de procesar, razonar, combinar y realizar deducciones lógicas con el contenido de la información para resolver automáticamente problemas cotidianos [Berners-Lee, 2000].

El término Web Semántica se utiliza generalmente para referirse a los formatos y tecnologías involucrados en su estructura. La recopilación, estructuración y recuperación de datos vinculados puede efectuarse a través de las tecnologías que proporcionan una descripción formal de los conceptos, términos y relaciones dentro de un ámbito de conocimiento determinado.

Con el objetivo de guiar la Web hacia su máximo potencial, el 1 de octubre de 1994, Tim Berners-Lee creó el *W3C*[2], una comunidad internacional que desarrolla estándares para asegurar el crecimiento de la Web a largo plazo y que tiene entre sus principales desafíos estandarizar las tecnologías de la Web Semántica. El consorcio se dedica además a la educación, divulgación y desarrollo de software y sirve como un foro abierto de discusión sobre los temas relacionados con la WWW. La Figura 2.1 muestra la arquitectura de la Web semántica propuesta por Tim Berners-Lee [Berners-Lee, 2000], que tiene forma de pila y que no es entendida como una nueva Web sino como una extensión de la Web actual que hará posible no sólo almacenar los datos, sino entender e interpretar el sentido de la información representada por esos datos. Las capas definidas por Berners-Lee para la Web Semántica son las siguientes:

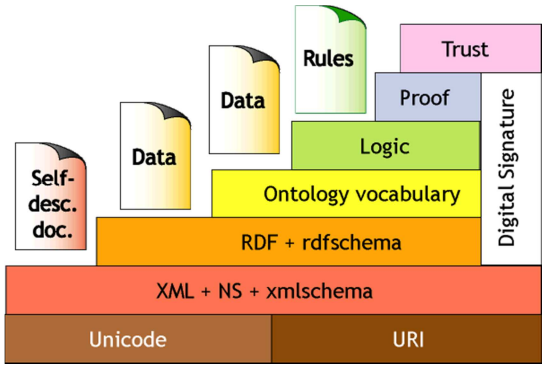


Figura 2.1. Arquitectura de la pila de la Web Semántica [Berners-Lee, 2000].

Alfabeto (Unicode): Se trata de una codificación del texto que permite utilizar los símbolos de diferentes idiomas sin que aparezcan caracteres extraños [Becker, 1988]. De esta forma, se puede expresar información en cualquier idioma en la Web Semántica.

Referencias (URI): *URI* (*Uniform Resource Identifier*) [URI, 2009] es un identificador único que permite la localización de un recurso al que se puede acceder vía Internet. Se compone del *URL* (*Uniform Resource Locator*), que es una secuencia de caracteres que describe la ubicación del recurso y el *URN* (*Uniform Resource Name*), que es una secuencia de caracteres que indica el nombre del recurso.

Sintaxis (XML + NS + XML-Schema): En esta capa se proporciona la sintaxis para la estructura del contenido. *XML* [XML, 1998] ofrece un formato común para intercambio de

[2] World Wide Web Consortium: *http://www.w3.org/*

documentos, *NS* (*namespaces*) [Bray *et al*., 2009] permite asociar los elementos y atributos de nombres usados en *XML* con los espacios de nombre identificados por referencias *URI*. Por su parte *XML-Schema* [XMLS, 2004] ofrece una plantilla estándar para elaborar documentos. Esto permite la creación de documentos uniformes en un formato común y no propietario aunque se utilicen diferentes fuentes.

Lenguaje (RDF + RDF Schema): Esta capa define el lenguaje universal con el cual expresar diferentes ideas en la Web Semántica. *RDF* (*Resource Description Framework*) [Lassila y Webick, 1999] es un lenguaje simple mediante el cual se definen sentencias en el formato de un triplete *sujeto-predicado-objeto*. Por su parte *RDF Schema* [RDFS, 1998] provee un vocabulario definido sobre *RDF* que permite el modelado de objetos con una semántica claramente definida. Esta capa no sólo ofrece descripción de los datos, sino también cierta información semántica.

Ontologías (Ontology vocabulary): Esta capa ofrece un criterio para catalogar y clasificar la información [Berners-Lee *et al*., 2001]. El uso de ontologías permite describir objetos y sus relaciones con otros objetos. Esta capa permite extender la funcionalidad de la Web Semántica, agregando nuevas clases y propiedades para describir los recursos.

Unificación Lógica (Logic): Esta capa posibilita la incorporación de reglas de inferencia, permitiendo a los programas realizar deducciones lógicas a partir de la información que expresan las ontologías. Gracias a este proceso los ordenadores pueden manipular los términos de un modo mucho más eficiente, beneficiando la inteligibilidad humana [Berners-Lee *et al*., 2001]. Por ejemplo, si existe una regla que asocia los códigos de las ciudades a los códigos de los países, entonces a partir de una dirección postal se puede deducir el país, y así utilizar un formato acorde con los estándares del país.

Pruebas (Proof): En esta capa se prueban los resultados de las deducciones. Las pruebas son escritas en el lenguaje unificador, que hace posible las inferencias lógicas a través del uso de las reglas de inferencia especificadas en las ontologías [Berners-Lee *et al*., 2001].

Confianza (Trust): En la capa de confianza los agentes comprueban de forma exhaustiva las fuentes de información para garantizar la fiabilidad en las operaciones de la Web Semántica. En este paso se utilizan las ontologías *Web Of Trust* [WOT, 2005] y *FOAF* [FOAF, 2010] fundamentalmente.

Firma digital (Digital Signature): Es el bloque encriptado de datos que serán utilizados por los ordenadores y los agentes para verificar que la información adjunta provenga de una fuente específica fiable [XML Signature WG, 2008].

Hasta la fecha se ha conseguido estandarizar las tecnologías correspondientes a las capas inferiores llegando hasta la capa ontológica, aunque todavía no se ha conseguido estandarizar del todo la parte del formato de intercambio de reglas en las ontologías, pues existen todavía muchos lenguajes y dialectos para la producción de reglas y se necesita la interconexión entre ellos [Kifer, 2008]. Otro aspecto a tener en cuenta es que los sistemas automatizados de razonamiento aún tienen que lidiar con algunas cuestiones como la vaguedad, imprecisión, incertidumbre, incoherencia y engaño, por lo que se necesita seguir trabajando en esa dirección. Las capas superiores de unificación lógica y de pruebas también constituyen un desafío en la investigación sobre las tecnologías de la Web Semántica.

2.2 Ontologías

La palabra ontología se deriva de los vocablos griegos *ontos* (ser) y *logos* (ciencia, estudio) por lo que literalmente significa "estudio del ser". En el diccionario de la Real Academia Española (*RAE*[3]) aparece la siguiente definición de ontología:

"Parte de la metafísica que trata del ser en general y de sus propiedades trascendentales"

Esta definición se centra únicamente en el sentido filosófico. En el tercer nuevo diccionario internacional de *Webster*[4], aparece una definición de ontología un poco más completa:

a) "Ciencia o estudio del ser: específicamente una rama de la metafísica relativa a la naturaleza y relaciones del ser; un sistema particular según el cuál se investigan los problemas de la naturaleza del ser; filosofía esencial"

b) "Teoría relativa a los tipos de entidades y específicamente los tipos de entidades abstractas admitidas en el lenguaje de un sistema"

[3] Real Academia Española: *http://www.rae.es/rae.html*
[4] Webster's Dictionary: *http://www.1828-dictionary.com/*

El primer sentido se refiere a una definición desde el punto de vista filosófico y el segundo es utilizado en el campo de la inteligencia artificial y la representación del conocimiento.

En el área de la inteligencia artificial una de las definiciones más conocidas y aceptadas es la de Gruber [Gruber, 1993] que luego fue extendida por Studer [Studer *et al*., 1998] y define la ontología como:

"Una especificación explicita y formal de una conceptualización compartida"

Es decir, una ontología consiste en la descripción de los elementos pertenecientes a un dominio que como veremos más adelante, pueden ser conceptos, propiedades, relaciones, funciones, axiomas y reglas de inferencia de dicho dominio. Esta especificación debe ser formal y explícita para que las máquinas sean capaces de procesarla y a la vez compartirla.

Las ontologías juegan un papel muy importante en el diseño de sistemas de información, ya que permiten una especificación más rica y estructurada del dominio de conocimiento respecto a otros enfoques [Chandresekaran *et al*., 1999]. En el marco de la Web Semántica las ontologías pueden emplearse, por ejemplo, para mejorar la efectividad de las búsquedas en la Web, porque pueden dotar al motor de búsqueda de la capacidad de buscar sólo páginas referidas a conceptos específicos y enlazar las páginas a ontologías que definen información sobre el dominio.

La definición de ontologías de dominio compartidas y comunes puede ayudar a personas y máquinas a comunicarse de forma concisa, facilitando el intercambio semántico, lo que constituye un elemento crucial para el éxito y la popularidad de la Web Semántica [Maedche y Staab, 2001]. A continuación explicamos los tipos de ontologías, sus componentes principales y las distintas formas de representación de las ontologías en la Web Semántica.

2.2.1 Tipos de ontologías

Las ontologías pueden clasificarse de diferentes maneras teniendo en cuenta diversos criterios. Entre las clasificaciones más utilizadas están las basadas en dos criterios fundamentales: 1) el contenido y 2) el área o dominio de la conceptualización.

2.2.1.1 Clasificación de las ontologías basada el contenido de la conceptualización.

Existen varias clasificaciones de ontologías atendiendo a su contenido. Entre ellas, una de las más globalmente aceptadas es la propuesta por Van Heist [Van Heijst *et al*., 1997], que plantea que las ontologías pueden clasificarse de acuerdo a la cantidad y tipo de estructura de la conceptualización en tres tipos:

- *Ontologías terminológicas o lingüísticas*: Especifican los términos usados para representar el conocimiento en el dominio. Se suelen usar para unificar el vocabulario en un campo determinado.

- *Ontologías de información*: Especifican la estructura de almacenamiento de las bases de datos. Ofrecen un marco estandarizado para el almacenamiento de la información.

- *Ontologías de modelado de conocimiento*: Especifican conceptualizaciones del conocimiento. Estas ontologías contienen una rica estructura interna y suelen estar ajustadas al uso particular del conocimiento que describen.

2.2.1.2 Clasificación de las ontologías basada en el área de la conceptualización.

Según el área o dominio de la conceptualización, la clasificación más utilizada es la propuesta por Guarino [Guarino, 1998], que plantea que las ontologías pueden clasificarse en:

- *Ontologías de alto nivel o genéricas*: Describen conceptos generales que son independientes de cualquier dominio o problema en particular. Se pueden aplicar en varios dominios e incluyen vocabulario general, como por ejemplo el relativo a eventos, tiempo, espacio, etc.

- *Ontologías de dominio*: Proporcionan vocabularios que describen los conceptos y relaciones de un dominio específico, como por ejemplo la medicina, los bosques, la pesca, etc.

- *Ontologías de aplicación o de tareas*: Describen las entidades del dominio dependiendo de una aplicación o tarea en particular. Estas ontologías están relacionadas directamente con la resolución de problemas.

2.2.2 Elementos de una ontología

Las ontologías, al ser formas de representación del conocimiento muy expresivas, proporcionan un vocabulario común en un dominio determinado y permiten definir, con diferentes niveles de formalismo el significado de los términos y las relaciones entre ellos. El conocimiento en las ontologías se formaliza usando cinco tipos de componentes: clases, relaciones, funciones, axiomas e instancias [Gruber, 1993].

Clases: Las clases representan los conceptos de la ontología. Un concepto suele ser algo sobre lo que se tiene información, como por ejemplo la descripción de una tarea, función, acción, estrategia, proceso de razonamiento, etc. Las clases suelen estar organizadas de manera jerárquica [Studer *et al*., 1998].

Funciones: Son un caso particular de relaciones entre clases donde los elementos se identifican mediante el cálculo de una función. Por ejemplo *madre_de*: *Persona→mujer*.

Axiomas: Los axiomas sirven para modelar sentencias que siempre son ciertas. Normalmente se usan para representar conocimiento que no puede ser formalmente definido por los componentes descritos anteriormente. Además también se usan para verificar la consistencia de la propia ontología. Un ejemplo de axioma puede ser: "*Una persona no puede ser madre y padre de otra a la vez*". A las ontologías que contienen axiomas se les denomina pesadas, mientras que aquellas que no los tienen se consideran ligeras.

Instancias: Las instancias se utilizan para representar objetos o individuos concretos de una clase en la ontología. Por ejemplo: *Juan García* es una instancia de la clase *Hombre*.

Relaciones: Las relaciones representan interacciones entre clases. Las más habituales son las relaciones binarias, como por ejemplo la herencia [Gomez-Perez *et al*., 2004]. Entre las relaciones más importantes en las ontologías se encuentran la taxonomía y la mereología, que explicamos a continuación.

2.2.2.1 Principales relaciones en ontologías

Existen muchas posibilidades de relacionar entidades en ontologías en dominios reales. En [Gomez-Perez *et al*, 2000] se plantea que estas relaciones pueden llegar a ser complejas y de naturaleza muy diversa. Sin embargo, no todas tienen la misma relevancia ni imponen el mismo tipo de propiedades jerárquicas a la ontología. Entre este conjunto de relaciones podemos destacar la *taxonomía* y la *mereología*, que son los dos tipos de relaciones interconceptuales más importantes.

Taxonomía

Una de las relaciones más comunes en ontologías en dominios reales es la taxonomía. Taxonomía es el estudio de la división en grupos ordenados o categorías. Esta palabra proviene de la filosofía y tiene su origen en dos términos griegos: *taxis* (orden) y *nomos* (tratado). Desde un punto de vista ontológico, una taxonomía es una organización ontológica basada en una relación de herencia, también llamada *generalización-especialización* o *es-un*, a través de la cual se agrupan las entidades y son generalizadas por clases de más alto nivel (superclases). Por ejemplo la clase *figura geométrica* y la clase *triángulo* tienen una relación taxonómica porque una es especialización de la otra. En general, las taxonomías han sido importantes para modelar esquemas de bases de datos, sistemas basados en conocimiento y vocabularios semánticos [Guarino y Welty, 2001].

Una definición más formal de la relación *es-un* [Guarino y Welty, 2001] sería la siguiente: Dadas las clases θ y δ, se dice que θ *es-un* δ si se cumple que cada individuo x perteneciente a la clase θ también pertenece a la clase δ, es decir, para todo x, $\delta(x) \rightarrow \theta(x)$, lo que significa que cada predicado satisfecho por la clase δ tiene que ser necesariamente satisfecho por la clase θ para que se cumpla la relación *es-un*. Las propiedades principales de la relación taxonómica son:

- *Asimetría*: Esta propiedad significa que la inclusión de una clase de individuos, *X* en una clase *Y* implica la no inclusión de *Y* en *X*. Formalmente, esto quiere decir que: (*X es-un Y*) → *not* (*Y es-un X*).

- *Transitividad*: Esta propiedad significa que si *X* está incluido en una clase *Y*, que a su vez está incluido en una clase *Z*, entonces, *X* está incluido en *Z*. Formalmente: (*X es-un Y*) *and* (*Y es-un Z*) → (*X es-un Z*).

- *Irreflexividad*: Las relaciones taxonómicas se consideran no reflexivas, es decir: *not* (*X es-un X*).

Una forma común de referirse a las clases relacionadas taxonómicamente en estructuras jerárquicas es llamarles *hijos* y *padres*, es decir, si *X es-un Y*, entonces se dice que *X* es *hijo* de *Y* (o *Y* es *padre* de *X*). A las clases que tienen el mismo *padre* se les llama *hermanos*. En el momento de la construcción de taxonomías hay que tener en cuenta que se debe clasificar la totalidad de los objetos del dominio y que un concepto (clase) puede tener varios padres diferentes, por lo que hereda todas las propiedades de sus padres taxonómicos. En una taxonomía los nombres deben ser simples y es deseable que objetos con clasificaciones similares tengan propiedades similares. Una buena taxonomía debe poseer un umbral de discriminación lo suficientemente alto para no agrupar diferentes objetos en la misma categoría y lo suficientemente bajo para permitir la existencia de un número adecuado de categorías a modelar de forma manejable [Fernández-Breis y Martínez-Béjar, 2002].

Mereología

Otra de las relaciones comunes en las ontologías es la mereología. La palabra mereología proviene de los términos griegos: *meros* (parte), y *logía* (estudio), por lo que literalmente significa estudio de las partes. En la filosofía y la lógica matemática es la relación que trata a las partes y sus relaciones con otras partes y las totalidades que forman [Simons, 1987]. Mientras que la teoría de conjuntos se basa en la relación de pertenencia entre un conjunto y sus elementos, la mereología hace hincapié en las relaciones *parte-de* entre las entidades. La noción de conjunto no es vista como una colección de elementos sino como una agregación de las partes que lo forman, por ejemplo una pared sería vista como una fusión de los ladrillos que fueron utilizados para su construcción.

Entre las propiedades de la relación universal *parte-de* tenemos la asimetría, la transitividad y la irreflexividad, pero existen otros conceptos que tienen que ver con la relación *parte-de*, como por ejemplo la *superposición*, que significa que dos elementos tienen

partes comunes, y la *disyunción*, que no es más que la negación de la *superposición* [Fernández-Breis y Martínez-Béjar, 2002].

En general, las relaciones mereológicas son muy variadas y complejas, y no existe una teoría única al respecto, por lo que la construcción y manejo de estas relaciones depende del tipo de las entidades definidas y el grado de expresividad de las ontologías según el dominio de aplicación.

2.2.3 Representación de las Ontologías

Durante los últimos años se han desarrollado varios lenguajes de representación de ontologías que pueden ser aplicables en el contexto de la Web Semántica. Los primeros lenguajes se basaban en la sintaxis *XML*, mientras que otros como *RDF* y *RDFS* han sido creados por los grupos de trabajo del *W3C* con fines puramente semánticos. Los lenguajes de ontologías más recientes como *OIL*, *DAML+OIL* y *OWL* constituyen una capa por encima de *RDF* y *RDFS* con el objetivo de incrementar su expresividad. A continuación explicamos brevemente cada uno de estos lenguajes.

2.2.3.1 RDF y RDFS

El Marco de Descripción de Recursos, *RDF* (*Resource Description Framework*) [Lassila y Webick, 1999] fue diseñado originalmente como un modelo de metadatos y se ha convertido en un método general muy utilizado para la descripción conceptual o modelado de información implementado en los recursos Web, utilizando varios formatos de sintaxis. El modelo de datos *RDF* es similar a los métodos clásicos de modelado conceptual como el esquema entidad-relación o los diagramas de clases, ya que se basa en la idea de formar enunciados sobre recursos (en particular, los recursos Web) utilizando expresiones en forma de *sujeto-predicado-objeto*. Estas expresiones son conocidas como tripletes *RDF*. El sujeto denota el recurso, el objeto denota rasgos o aspectos del recurso y el predicado expresa una relación entre el sujeto y el objeto. El sujeto pertenece a un conjunto llamado `Resource`, el predicado pertenece a un conjunto llamado `Properties` y el objeto puede pertenecer al conjunto `Resource` o al conjunto `Literal`. Por ejemplo, si queremos representar en *RDF* la expresión: "*La pelota es de color rojo*", el sujeto o recurso sería "*la pelota*", el predicado sería "*es de color*" y el objeto sería "*rojo*".

RDFS o *RDF Schema* [RDFS, 1998] es una extensión semántica del modelo *RDF* que proporciona un mejor soporte para la definición y clasificación, bajo la influencia de los sistemas de *frames* y los modelos orientados a objetos. Los sistemas de *frames* organizan el conocimiento de una manera centrada en el concepto con construcciones descriptivas de las ontologías y construyen axiomas de herencia. Estos modelos permiten a los usuarios representar el mundo en diferentes niveles de abstracción haciendo énfasis en las entidades, a diferencia del modelo de grafo plano ofrecido por la mayoría de las redes semánticas. Además de heredar las características básicas de los sistemas de *frames*, *RDFS* proporciona construcciones ontológicas que hacen que las relaciones sean menos dependientes de los conceptos. Por ejemplo los usuarios pueden definir relaciones como instancias de `rdf:Property`, describir relaciones de herencia entre las relaciones utilizando `rdfs:subPropertyOf`, y asociar relaciones definidas con clases utilizando `rdfs:domain` o `rdfs:range`.

2.2.3.2 OIL

OIL (Ontology Inference Layer) [Fensel *et al*., 2000] es el primer lenguaje de representación de ontologías basado en estándares del *W3C* y su sintaxis está definida como una extensión de *RDFS*. El modelo de representación del conocimiento lo ha heredado, por una parte, de la Lógica Descriptiva (declaración de axiomas o reglas) y, por otra, de los sistemas basados en *frames* (taxonomía de clases y atributos). *OIL* se encuentra estructurado en varias capas de sublenguajes. La capa base de *OIL* coincide plenamente con *RDFS*, y cada una de las capas superiores añade funcionalidad y complejidad a su capa subyacente. Esto posibilita la reutilización de agentes diseñados para *RDFS* para procesar ontologías *OIL* [Fensel, 2002].

Además de las ventajas enunciadas, *OIL* ofrece un completo conjunto de utilidades software como editores de ontologías y herramientas de razonamiento avanzado sobre las ontologías. Entre las limitaciones de *OIL* [Horrocks *et al*., 2001] podemos destacar la falta de expresividad en la declaración de axiomas (reglas) y que no soporta dominios concretos (e.g. números enteros, cadenas de caracteres, etc.).

2.2.3.3 DAML+OIL

DAML+OIL [Horrocks y van Harmelen, 2001] surge de la cooperación entre los grupos de trabajo de *OIL* y *DARPA* (*US Defense Advanced Research Projects Agency*), quienes anteriormente habían desarrollado el lenguaje *DAML* (*DARPA's Agent Markup Language*) con la finalidad de extender el nivel de expresividad de *RDFS*. Por tanto *DAML+OIL* unifica estos dos lenguajes, con el objetivo de la construcción de un lenguaje estándar para la definición de ontologías. Este lenguaje hereda muchas características de *OIL*, pero se aleja un poco del modelo basado en *frames* aprovechando más los elementos de la lógica descriptiva.

DAML+OIL divide el dominio en dos partes disjuntas: una parte se llama dominio de tipos de datos y se refiere a los valores que pertenecen a los tipos de datos del esquema *XML*; y la otra parte se llama dominio de objetos y trata los objetos que son considerados miembros de clases descritas en *DAML+OIL* (o *RDF*). En *DAML+OIL* las clases se llaman clases de objetos y son elementos de `daml:Class` (una subclase de `rdfs:Class`). Los tipos de datos son elementos implícitos de `daml:Datatype` y se usan en *DAML+OIL* incluyendo sus *URLs* en la ontología. Los elementos principales del lenguaje son:

- *Elementos de clase*: Un elemento de clase `daml:Class` contiene la definición de una clase de objetos.

- *Enumeraciones*: Es un elemento `daml:oneOf`, que contiene una lista de objetos, permitiendo enumerar los elementos concretos de una clase.

- *Propiedades*: Un elemento `rdf:Property` se refiere al nombre de una propiedad. Las propiedades que se usan en restricciones de propiedades son las que relacionan objetos (instancias de `ObjectProperty`) o las que relacionan objetos a valores de tipos de datos (instancias de `DatatypeProperty`).

- *Restricciones de propiedad*: Una restricción de propiedad es un tipo especial de expresión de clase. Se define implícitamente una clase anónima, llamada clase de objetos que satisface la restricción.

- *Instancias*: Para describir las instancias de clases y propiedades se utiliza *RDF*.

2.2.3.4 OWL

OWL (*Web Ontology Language*) [Dean y Schreiber, 2004], inspirado en *DAML+OIL* también constituye una extensión de *RDFS* para permitir una inferencia lógica más rica. Actualmente existen tres variantes de *OWL*, que incorporan diferentes funcionalidades:

- *OWL-Lite*: Es la variante más sencilla para la creación de un sistema de *frames* (o una base de datos orientada a objetos) en términos de clases, propiedades, relaciones de subclases y restricciones. En *OWL-Lite* no se utiliza el vocabulario completo de *OWL* y algunos de los términos se utilizan bajo ciertas restricciones.

- *OWL-DL*: Se basa en la lógica descriptiva, y se centra en la semántica formal común y la decidibilidad de la inferencia. La lógica descriptiva ofrece construcciones adicionales de ontología (como conjunción, disyunción y negación), además de las clases y relaciones, y tiene dos mecanismos de inferencia importantes: la subsunción, que consiste en inferir cuando un elemento es abarcado por otro; y la consistencia, que consiste en verificar la imposibilidad de contradicción dentro del sistema. La fortaleza de la teoría de conjuntos hace que la lógica descriptiva sea adecuada para capturar el conocimiento sobre un dominio en el que las instancias pueden ser agrupadas en clases y las relaciones entre clases son binarias. *OWL-DL* utiliza todas las construcciones ontológicas de *OWL* con algunas restricciones.

- *OWL-Full*: Es la versión más expresiva de *OWL*. La mayor diferencia entre *OWL-DL* y *OWL-Full* es que el espacio de clase y el espacio de instancia son disjuntos en *OWL-DL*, pero no en *OWL-Full*. Es decir, en *OWL DL* una clase no puede ser también un individuo o una propiedad, y una propiedad no puede ser tampoco un individuo o una clase. En cambio en *OWL-Full* una clase puede interpretarse simultáneamente como un conjunto de individuos y como un individuo que pertenece a otra clase. En *OWL-Full* se puede utilizar el vocabulario completo de *OWL* sin ningún tipo de restricciones.

2.3 Mediación de ontologías

La mediación de ontologías permite la reutilización de datos entre diferentes aplicaciones de la Web Semántica y, en general, la cooperación entre diferentes instituciones o comunidades. En el contexto de la gestión del conocimiento semántico, la mediación de ontologías es especialmente importante para permitir el intercambio de datos entre bases de conocimiento heterogéneas y facilita a las aplicaciones reutilizar los datos de diferentes bases de conocimiento y compartir servicios en la Web Semántica.

La mayoría de los trabajos sobre la mediación de ontologías han hecho uso de dos estrategias. Una de ellas es traducir un conjunto de datos de cualquier ontología fuente a un conjunto de datos en una ontología global centralizada, como por ejemplo ONTOLINGUA [Gruber, 1993], que sirve como estructura intermedia que puede ser traducida en un conjunto de datos de cualquier ontología destino. Esta estrategia por lo general no es factible, a menos que exista una ontología global que pueda cubrir todas las ontologías, y se pueda llegar a un acuerdo entre todos los expertos para escribir traductores entre sus propias ontologías y la global [Dou *et al*., 2004]. Aun cuando en principio tal compromiso se puede lograr, en la práctica el mantenimiento de todas las ontologías de manera coherente con una única estructura es muy difícil. En vez de crear una ontología global que cubra todos los dominios, pero siguiendo con la idea de generar una tercera ontología, muchos han optado por la fusión. El proceso de *fusión de ontologías* consiste en obtener una nueva ontología coherente a partir de dos o más ontologías distintas relacionadas con el mismo tema. Una de las principales desventajas de la fusión de ontologías es el hecho de que las aplicaciones que utilizan las ontologías originales tienen que ser adaptadas para utilizar la ontología resultante de la fusión.

La otra estrategia es hacer una correspondencia de ontologías directamente de un conjunto de datos en una ontología (fuente) a datos en otra ontología (destino), sin necesidad de utilizar ningún tipo de ontología intermedia. Esta estrategia es la que más se ha utilizado en la práctica porque su implementación y mantenimiento son más sencillos [Dou *et al*., 2004]. Dentro de esta estrategia se encuentra el *mapeo de ontologías*, que consiste en especificar relaciones entre diferentes elementos de las ontologías a través de una serie de reglas. El proceso de hallar correspondencias entre ontologías completas se denomina *alineación de ontologías*. La alineación es la tarea de crear vínculos entre las dos ontologías originales eligiendo las mejores correspondencias según los criterios establecidos por los expertos en el dominio de la

aplicación. La alineación se realiza en ontologías independientes pero coherentes entre sí, es decir, que comparten elementos estructurales y pertenecen al mismo dominio.

Las técnicas de alineación de ontologías tienen particular importancia debido a que la creación manual de correspondencias entre los conceptos no es factible, ya que consume demasiado tiempo excepto para ontologías muy pequeñas. Tanto los métodos de alineación como los de fusión permiten la interoperabilidad entre diferentes ontologías. Sin embargo, la alineación es mucho menos compleja que la fusión por el hecho de que la creación y el mantenimiento de vínculos entre los conceptos es más fácil y menos costosa que producir una ontología completamente nueva que sea coherente con las originales. Aunque la alineación completamente automática de ontologías se perfila como la solución ideal para la interoperabilidad semántica, los resultados actuales proporcionados por métodos automáticos todavía no tienen la calidad suficiente. Los desafíos que enfrentan los métodos completamente automáticos son múltiples, incluidas las diferencias de vocabulario (por ejemplo, debido a la sinonimia y homonimia), las diferencias de modelado (por ejemplo, debido a modelos o formatos de atributos diferentes) y los diferentes puntos de vista sobre la realidad modelada.

2.3.1 Técnicas de alineación de ontologías

Dadas dos ontologías, O_1 y O_2, la alineación se define como el proceso de creación de correspondencias en la forma (c_1, c_2, s), donde $c_1 \in O_1$ y $c_2 \in O_2$ son los conceptos de las dos ontologías y $s \in [0,1]$ es la similitud estimada entre los dos conceptos (también llamada confianza de correspondencia). Una alineación entre dos ontologías O_1 y O_2 es un conjunto de correspondencias, donde cada correspondencia se define como: $A(O_1, O_2) = \{(c_1, c_2, s) \mid c_1 \in O_1, c_2 \in O_2, s \in [0,1]\}$. Las correspondencias también pueden tener la forma extendida (c_1, c_2, s, r), donde r es el tipo de relación (por ejemplo equivalencia o generalización), o una forma reducida (c_1, c_2), donde el coeficiente de correspondencia no se especifica. La Figura 2.1 muestra gráficamente el proceso de alineación de ontologías.

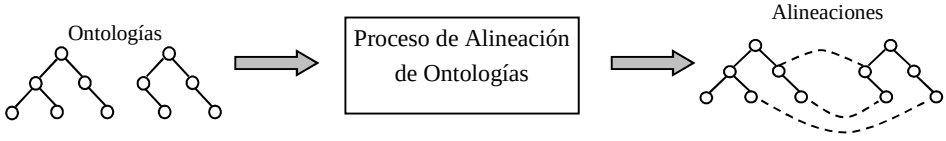


Figura 2.1. Alineación de ontologías

Se han desarrollado muchas técnicas diferentes para la implementación del proceso de alineación de ontologías, que pueden clasificarse en función de las muchas características que se pueden encontrar en las ontologías (e.g. etiquetas, estructuras, instancias, semántica…) [Shvaico y Euzenat, 2004], o respecto al tipo de disciplinas que se utilizan (e.g. estadística, combinatoria, semántica, lingüística, aprendizaje automático…) [Rahm y Bernstein, 2001; Kalfoglou y Schorlemmer, 2003; Euzenat, 2004].

En la clasificación de los algoritmos de alineación de ontologías dada en [Shvaico y Euzenat, 2004] se establece una diferencia entre los algoritmos de correspondencia elementales y los combinados. Los métodos elementales se dividen en dos niveles: el nivel de elemento y el nivel de estructura, mientras que los combinados son aquellos que utilizan combinaciones de técnicas de diferentes categorías para obtener mejores resultados. En la Figura 2.2 se muestra un esquema de dicha clasificación. En la práctica es muy difícil encontrar algoritmos de alineación de ontologías que utilicen una técnica elemental concreta, sino que por lo general se utilizan combinaciones de estas de manera secuencial, en paralelo, o combinadas con otras tecnologías de inteligencia artificial.

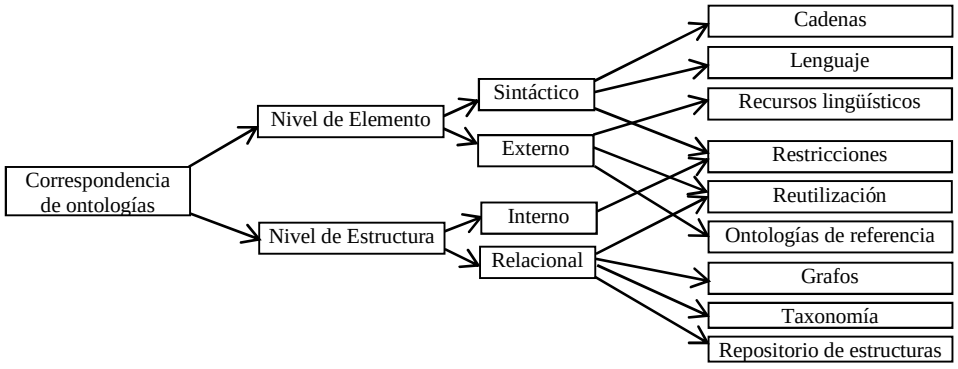


Figura 2.2. Clasificación de los métodos de alineación de ontologías [Shvaiko y Euzenat, 2004]

2.3.1.1 Técnicas de Nivel de Elemento

Las técnicas de correspondencia de nivel de elemento funcionan analizando las entidades de las ontologías en solitario, ignorando sus relaciones con otras entidades.

Técnicas basadas en cadenas: son aquellas que se basan fundamentalmente en las similitudes de las cadenas de caracteres utilizando diferentes funciones de distancia entre cadenas. Estos métodos utilizan solamente los nombres de los conceptos para calcular su similitud. Como ejemplos de métodos de alineación de ontologías que utilizan técnicas basadas en cadenas tenemos a Anchor-PROMPT [Noy y Musen, 2001] y el método de Fernández-Breis [Fernández-Breis y Martínez-Béjar, 2002].

Técnicas basadas en lenguaje: son métodos que utilizan técnicas lingüísticas adicionales para calcular las similitudes entre cadenas de caracteres. Algunas de estas técnicas son: la *tokenización* (dividir las cadenas en componentes léxicos o símbolos, por ejemplo: eliminar signos de puntuación, dígitos), eliminación de *stop words* (palabras sin significado como artículos, preposiciones, conjunciones), etc. Como ejemplos de métodos que utilizan técnicas basadas en lenguajes tenemos a CUPID [Madhavan *et al*., 2001], ASMOV [Jean-Mary *et al*., 2009] y Eff2Match [Watson *et al*., 2010].

Técnicas basadas en restricciones: son métodos que utilizan las restricciones internas de las entidades para evaluar la similitud. Por ejemplo, la mayoría de estas técnicas parten de la idea de que para que dos conceptos sean correspondientes es necesario que tengan la misma cantidad de atributos y que sus atributos sean del mismo tipo. Entre estos métodos podemos mencionar a COMA [Do y Rahm, 2002], CODI [Noessner *et al*., 2010] y SOBOM [Xu *et al*., 2010].

Técnicas basadas en recursos lingüísticos: son métodos que hacen uso de recursos lingüísticos para buscar relaciones entre los términos en el proceso de correspondencia. Ejemplos de estos recursos son los tesauros especializados, y bases de datos léxicas como *WordNet*, que permite identificar una amplia gama de relaciones lingüísticas (e.g. sinonimia). Los métodos que utilizan *WordNet* se basan en el cálculo de las distancias entre las palabras dentro del grafo conceptual de *WordNet*. La desventaja de estos sistemas es que son de uso exclusivo para aplicaciones de dominio general y que emplean terminología en inglés, debido a la carencia de bases de datos en otros idiomas. Entre los métodos que utilizan técnicas basadas en recursos lingüísticos tenemos a CUPID [Madhavan *et al*., 2001], COMA [Do y Rahm, 2002], ASMOV [Jean-Mary *et al*., 2009] y Eff2Match [Watson *et al*., 2010].

Técnicas que reutilizan alineaciones: son métodos que emplean alineaciones obtenidas previamente con ontologías dentro del mismo dominio de aplicación. Estas técnicas son útiles

sobre todo cuando se trata de ontologías muy grandes, puesto que evitan la necesidad de procesarlas en su totalidad si se tienen alineados con anterioridad algunos fragmentos de las mismas. Entre los métodos que utilizan estas técnicas podemos mencionar a COMA [Do y Rahm, 2002] y OLA [Euzenat y Valtchev, 2004].

Técnicas que utilizan ontologías globales o de referencia: son métodos que utilizan conocimiento externo, como ontologías globales, que proponen una terminología de referencia en un contexto semántico compartido. Estas ontologías definen conceptos generales que pueden ser utilizados en diferentes dominios. SUMO [Niles y Pease, 2001] y DOLCE [Gangemi *et al*., 2003] son ejemplos de esta clase de ontologías diseñadas especialmente con el propósito de la integración. Sin embargo, hasta el momento este tipo de técnica no ha resultado viable debido a la dificultad que implica el mantenimiento de todas las ontologías de manera coherente con una única estructura.

2.3.1.2 Técnicas de Nivel de Estructura

Las técnicas de correspondencia de nivel de estructura funcionan analizando las relaciones estructurales de las entidades en las ontologías.

Técnicas basadas en grafos: son métodos que tratan las ontologías de entrada como grafos etiquetados. En estos sistemas la similitud entre dos nodos se basa en el análisis de sus posiciones dentro del grafo. Se basan fundamentalmente en la idea de que si dos nodos son relativamente similares, entonces es muy probable que sus vecinos dentro del grafo también lo sean. Como ejemplos de aplicaciones de alineación de ontologías que utilizan técnicas basadas en grafos podemos mencionar a Anchor-PROMPT [Noy y Musen, 2001], AgreementMaker [Cruz *et al*., 2009] y SOBOM [Xu *et al*., 2010].

Técnicas basadas en la taxonomía: Son también métodos basados en grafos pero que sólo consideran relaciones taxonómicas (generalización-especialización) entre los nodos. Aprovechan el hecho de que los nodos conectados mediante relaciones de especialización tienen ya cierta similitud por definición y entonces existe cierta probabilidad de que los vecinos también sean similares. La mayoría de las aplicaciones que tienen en cuenta la estructura de las ontologías se basan en la taxonomía, y como ejemplos podemos mencionar a ASMOV [Jean-Mary *et al*., 2009], CODI [Noessner *et al*., 2010], SOBOM [Xu *et al*., 2010], Eff2Match [Watson *et al*., 2010] entre otros.

Repositorio de estructuras: son métodos que utilizan un repositorio para almacenar ontologías y sus fragmentos, junto con su coeficiente de similitud. Es decir, en caso de que se quiera realizar el emparejamiento de nuevas estructuras, primero se verificará si existe un índice de similitud para ellas en el repositorio. El objetivo es identificar si merece la pena realizar el proceso de correspondencia en ciertas estructuras o reutilizar las alineaciones almacenadas con anterioridad [Rahm *et al*., 2004].

2.3.2 Formas de especificar las reglas de alineación de ontologías

En los trabajos propuestos en el campo de la alineación de ontologías se han utilizado varios lenguajes para representar las reglas de correspondencia. A continuación mencionamos algunos de los más conocidos, así como sus ventajas y limitaciones.

2.3.2.1 OWL

Es posible expresar relaciones de equivalencia entre partes de las ontologías directamente en *OWL* a través de las primitivas `equivalentClass` y `equivalentProperty`. Por ejemplo, si en una ontología existe el concepto `Product` y en la otra existe un concepto equivalente `Article` la representación de esta equivalencia en *OWL* sería:

```
<owl:Class rdf:ID=”Product”>
 <owl:equivalentClass rdf:resource=”&ontology2;Article”/>
</owl:Class>
```

Sin embargo existen primitivas más específicas para expresar relaciones entre entidades como `subClassOf` y `subPropertyOf`. En el ejemplo que mostramos a continuación se puede observar la representación de que la clase `City` en la primera ontología es abarcada por la clase `City` en la segunda ontología, así como la propiedad `cv` es abarcada por `Resume`.

```
<owl:Class rdf:about="http://.../onto1#City">
 <owl:subClass ="http://.../onto2#City"/>
</owl:Class>

<owl:Property rdf:about="http://.../onto1#cv">
 <owl:subProperty rdf:resource="http://.../onto2#Resume"/>
</owl:Property>
```

Los principales inconvenientes de utilizar este lenguaje para las alineaciones son que obliga a utilizar el lenguaje de ontologías *OWL*, además de que se mezcla la alineación con las definiciones, lo que dificulta la claridad de las alineaciones.

2.3.2.2 OWL Contextualizado (*C-OWL*)

C-OWL [Bouquet *et al*., 2003] es una extensión del lenguaje *OWL* para expresar correspondencias entre ontologías heterogéneas. Las construcciones en *C-OWL* se denominan reglas puente y permiten expresar una familia de relaciones semánticas entre conceptos, relaciones e individuos interpretados en dominios heterogéneos. Dadas dos ontologías O_1 y O_2 una regla puente de O_1 a O_2 expresa una relación semántica entre un concepto, relación o individuo de O_1 y un concepto, relación o individuo de O_2. Las reglas puente de O_1 a O_2 no son el inverso de las reglas puente de O_2 a O_1. En *C-OWL* se definen 5 tipos de reglas puente, que se explican en la Tabla 2.1.

Tabla 2.1. Reglas puente de C-OWL

Tipo de regla	**Representación**	**Significado**
`Into`	$i:A \xrightarrow{\subseteq} j:B$	El concepto *A* de la ontología *i* es *más específico* que el concepto *B* de la ontología *j*
`Onto`	$i:A \xrightarrow{\supseteq} j:B$	El concepto *A* de la ontología *i* es *más general* que el concepto *B* de la ontología *j*
`Equivalent`	$i:A \xrightarrow{\equiv} j:B$	El concepto *A* de la ontología *i* es *equivalente* al concepto *B* de la ontología *j*
`Disjoint`	$i:A \xrightarrow{\perp} j:B$	El concepto *A* de la ontología *i* es *disjunto* del concepto *B* de la ontología *j*
`Overlap`	$i:A \xrightarrow{*} j:B$	El concepto *A* de la ontología *i se solapa* con el concepto *B* de la ontología *j*

En *C-OWL* una representación de correspondencia se compone de la siguiente información:

- Identificador único para referirse a la correspondencia
- Referencia a la ontología fuente
- Referencia a la ontología de destino
- Conjunto de reglas puente relativas a los conceptos de las dos ontologías, cada una descrita por:
    1. Referencia al concepto de origen

2. Referencia al concepto de destino

3. El tipo de la regla puente (*Into*, *Onto*, *Equivalent*, *Disjoint*, *Overlap*)

La propuesta *C-OWL* puede expresar alineaciones relativamente simples. La parte más expresiva se establece en las relaciones utilizadas por la correspondencia. Estas alineaciones tienen una semántica clara. Sin embargo, están dadas desde un punto de vista particular: el de la ontología de destino.

A continuación se presenta un ejemplo sencillo donde se vinculan las propiedades *amount* y *quantity* a través de la regla puente *Into* entre las ontologías *StandardOrder* y *order*.

```
<cowl:Mapping>
  <cowl:sourceOntology>
    <owl:Ontology rdf:about="&StandardOrder;"/>
  </cowl:sourceOntology>
  <cowl:targetOntology>
    <owl:Ontology rdf:about="&order;"/>
  </cowl:targetOntology>
  <cowl:bridgeRule>
    <cowl:Into>
      <cowl:source>
        <owl:Class rdf:about=="&StandardOrder;#amount"/>
      </cowl:source>
      <cowl:target>
        <owl:Class rdf:about=="&order;#quantity"/>
      </cowl:target>
    </cowl:Into>
  </cowl:bridgeRule>
</cowl:Mapping>
```

2.3.2.3 SWLR

SWRL (*Semantic Web Rule Language*) [Horrocks *et al*., 2004] es un lenguaje de reglas para la Web Semántica. Consiste en una extensión de *OWL* con una noción explícita de las reglas (de *RuleML* [Grosof, 2001]) que son interpretadas como *cláusulas de Horn de primer orden* [Horn, 1951] (disyunción de literales con un literal positivo como máximo). Estas reglas pueden ser entendidas como correspondencias entre ontologías.

SWRL mezcla el vocabulario de *RuleML* para el intercambio de reglas con el vocabulario *OWL* para expresar el conocimiento. Define una regla (`RuleML:imp`) con un cuerpo (`RuleML:body`) y una cabecera (`RuleML:head`).

En el ejemplo que mostramos a continuación la última regla expresa que una persona (*person*) de la ontología `http://.../ontology2` con *M* como el valor de su atributo género (*gender*) es una mujer (*woman*) en la ontología `http://.../ontology1`. La ventaja de usar *SWRL* sobre *OWL* es que las reglas son identificadas como tal y son más fáciles de manipular.

```
<ruleml:imp>
  <ruleml:_body>
    <swrlx:classAtom>
      <owlx:Class owlx:name="http://.../ontology2#Person" />
      <ruleml:var>p</ruleml:var>
    </swrlx:classAtom>
    <swrlx:individualPropertyAtom
       swrlx:property="http://.../ontology2#gender">
      <ruleml:var>p</ruleml:var>
      <owlx:Individual owlx:name="M" />
    </swrlx:individualPropertyAtom>
 </ruleml:_body>
 <ruleml:_head>
   <swrlx:classAtom
      swrlx:property="http://.../ontology1#woman">
     <ruleml:var>p</ruleml:var>
   </swrlx:classAtom>
 </ruleml:_head>
</ruleml:imp>
```

2.3.2.4 OML

El lenguaje de mapeo de ontologías *OML* (*Ontology Mapping Language*) [de Bruijn *et al*., 2004] surge como parte del proyecto *SEKT*, que tuvo como objetivo desarrollar una plataforma completa para la gestión de ontologías. Este lenguaje proporciona un formato completo como base para representar las correspondencias entre ontologías por medio de las construcciones *ClassMapping*, *AttributeMapping*, *RelationMapping*, *ClassAttributeMapping*, *ClassRelationMapping*, *ClassInstanceMapping* e *IndividualMapping*, un conjunto de construcciones lógicas (*AND*, *OR*, *NOT*, *JOIN*) y la posibilidad de especificar las condiciones sobre los tipos o los valores de las clases y atributos. Tiene la ventaja de ser independiente del lenguaje de ontologías, proporcionando una base común para la investigación sobre técnicas de correspondencia de esquemas. En [Scharffe y

de Bruijn, 2005] se presenta una descripción detallada del lenguaje y se muestran varios ejemplos, como el que presentamos a continuación, donde se realiza una correspondencia entre las clases `Creature` y `LivingThing` de dos ontologías diferentes.

```
classMapping(
annotation(<"rdfs:label"> 'Creature-LivingThing')
annotation(<"http://..../elements/1.1/description">
 'Mapea los conceptos: Creature y LivingThing')
bidirectional
<"http://...../creature#Creature">
<"http://...../LivingThing#LivingThing">)
```

El lenguaje es un formato de alineación expresivo y ofrece a los usuarios muchas clases de relaciones y constructores de entidades. Una de sus principales ventajas es la independencia del lenguaje de ontologías utilizado, lo que proporciona un formato común para expresar correspondencias entre ontologías escritas en diferentes lenguajes.

2.3.2.5 SKOS

El vocabulario básico de *SKOS* (*Simple Knowledge Organisation System*) [Miles y Brickley, 2005a], [Miles y Brickley, 2005b] se basa en *RDF* y fue diseñado con el objetivo de expresar las relaciones entre ontologías ligeras (e.g. los tesauros). En *SKOS* los conceptos se identifican con referencias *URI* y pueden etiquetarse en cadenas de texto en uno o varios idiomas, documentarse y estructurarse a través de relaciones semánticas de diversa tipología. Este modelo permite mapear conceptos de diferentes esquemas, así como definir colecciones ordenadas y agrupaciones de conceptos. También permite establecer relaciones entre las etiquetas asociadas a los conceptos. *SKOS* actualmente se encuentra en fase de desarrollo y funciona como una capa en la parte superior de otros formalismos constituyendo un puente entre el formalismo lógico riguroso de los lenguajes de ontologías como *OWL* y el conocimiento caótico y débilmente estructurado de las herramientas colaborativas basadas en Web, como por ejemplo las aplicaciones de etiquetado social. *SKOS* permite la representación de las siguientes relaciones entre los esquemas de conceptos:

- *Exacta*: correspondencia donde el significado del concepto de origen es idéntico al significado del concepto de destino.

- *Inexacta*: correspondencia donde existe cierto solapamiento en el significado de los conceptos de origen y de destino. La correspondencia inexacta puede ser:

    1. *Mayor*: expresa un solapamiento significativo.

    2. *Menor*: expresa un solapamiento pequeño.

- *Parcial*: correspondencia donde el significado del concepto de origen abarca o es abarcado por el significado del concepto de destino. La correspondencia parcial puede ser:

    1. *Amplia*: para expresar que el concepto de origen se encuentra abarcado por el concepto de destino.

    2. *Reducida*: para expresar que el concepto de origen abarca al concepto de destino.

Además de las correspondencias `1:1`, *SKOS* soporta correspondencias `1:n`. Para este propósito se usan los operadores *`AND`*, *`OR`* y *`NOT`* y se permite definir el objetivo de una correspondencia como una combinación de conceptos. El ejemplo del siguiente cuadro muestra que la clase *`address`* es la unión de la clase *`Shipping address`* y la clase *`Address for dispatch`*, utilizando el operador *`AND`*.

```
<hpm:Concept>
 <prefLabel>Address</prefLabel>
 <map:exactMatch>
  <map:AND>
   <map:memberList rdf:parseType="Collection">
    <gcl:Concept><prefLabel>Shipping address</prefLabel>
    </gcl:Concept>
    <gcl:Concept>
      <prefLabel>Address for dispatch</prefLabel>
    </gcl:Concept>
  </map:memberList>
   </map:AND>
 </map:exactMatch>
</hpm:Concept>
</rdf:RDF>
```

SKOS tiene la ventaja de ser un vocabulario ligero definido desde la base de una rica colección de relaciones entre las entidades. Dado que se usan *URIs* para referirse a los objetos, *SKOS* está completamente integrado en la arquitectura de la Web Semántica y no está

comprometido con un lenguaje en particular. Otra de las ventajas de *SKOS* es que convierte cualquier tipo de descripción organizada a un conjunto fácilmente utilizable de clases. Sin embargo, tiene la desventaja de carecer de una semántica formal como por ejemplo la de *OWL*. Al igual que otros formatos que no separan las ontologías de las correspondencias, *SKOS* mezcla el alto potencial de *RDFS* con la expresión de las alineaciones.

2.3.2.6 Alignment Format

Alignment Format [Euzenat y Shvaiko, 2007] es utilizado en las tareas de la Iniciativa para la Evaluación de la Alineación de Ontologías *OAEI* (*Ontology Alignment Evaluation Initiative*). En esta iniciativa se ha desarrollado un *API* de código abierto implementada en *Java* basada en este formato de alineación que permite la generación y el almacenamiento de alineaciones, y facilita la comparación entre distintos métodos de alineación de ontologías. En este formato, la descripción de las alineaciones contiene la siguiente información:

1. *Referencias*: representa las referencias a las ontologías alineadas.

2. *Nivel*: representa el tipo de alineación. Los posibles tipos son:

    - *Nivel 0*: Las alineaciones de nivel 0 representan correspondencias entre dos entidades (clases, propiedades, individuos) representadas por *URIs*.

    - *Nivel 1*: En las alineaciones de nivel 1 es posible establecer correspondencias entre grupos o listas de entidades.

    - *Nivel 2*: En las alineaciones de nivel 2 es posible establecer otro tipo de relaciones entre entidades como por ejemplo fórmulas y consultas.

3. *Conjunto de correspondencias*: representa la relación existente entre entidades de la primera ontología con entidades de la segunda.

4. *Aridad*: representa la multiplicidad de la correspondencia, es decir, si es *inyectiva, sobreyectiva, total* o *parcial* en ambos lados. Se utiliza la siguiente notación: *'1'* para *inyectiva* y *total*, *'?'* para *inyectiva*, '+' para *total*, y '*' para indicar que no se precisa multiplicidad en la correspondencia. La multiplicidad por defecto es *1:1*.

5. *Entidad 1*: representa la primera entidad alineada.

6. *Entidad 2*: representa la segunda entidad alineada.

7. *Relación*: representa el tipo de relación entre las dos entidades. La relación por defecto es la equivalencia (=) pero se pueden definir otras relaciones.

8. *Fuerza*: representa el valor de confianza de la alineación. Esta medida puede ser por ejemplo un valor real entre 0 y 1, donde el 0 denota la mínima confianza y 1 la máxima confianza.

9. *Id*: es un identificador para la alineación.

A continuación mostramos un ejemplo de alineación de Nivel 0 entre dos ontologías de referencias bibliográficas. En el ejemplo se establece la correspondencia entre los conceptos: `article` y `paper`.

```
<Alignment>
  <xml>yes</xml>
  <level>0</level>
  <type>**</type>
  <onto1>http://…/ontology1</onto1>
  <onto2>http://…/ontology2</onto2>
  <map>
    <Cell>
      <entity1 rdf:resource='http://.../ontology1#article'/>
      <entity2 rdf:resource='http://.../ontology2#paper'/>
      <measure rdf:datatype='&xsd;float'>1.0</measure>
      <relation>=</relation>
    </Cell>
  </map>
</Alignment>
```

2.3.3 Trabajos relacionados con la alineación de ontologías

El proceso de alineación de ontologías se ha llevado a cabo empleando técnicas muy diversas, como por ejemplo la lingüística, las probabilidades, el aprendizaje automático, entre otras. A continuación hacemos un breve repaso de los principales enfoques encontrados en la literatura en el campo de la alineación de ontologías.

2.3.3.1 Método de Fernández-Breis y Martínez-Béjar

Fernández-Breis y Martínez-Béjar [Fernández-Breis y Martínez-Béjar, 2002] proponen un marco de cooperación para la integración de ontologías. En particular, presentan un sistema para la construcción cooperativa, integración y derivación de ontologías que distingue entre dos tipos de usuarios: los de un nivel de conocimiento normal y los expertos. Los usuarios normales introducen en el sistema información relativa a los conceptos, atributos, relación taxonómica y términos asociados, mientras que los usuarios expertos procesan esta información y el sistema les ayuda a obtener la ontología integrada. El algoritmo que apoya esta integración se basa en las características taxonómicas y en la detección de sinónimos en las dos ontologías. También se tienen en cuenta los atributos de conceptos y se define una tipología de los criterios de igualdad de conceptos.

El punto de partida del entorno de integración es un conjunto de ontologías que pertenecen a diferentes usuarios. Una operación fundamental del proceso de integración es la comparación del conocimiento contenido en ontologías diferentes. Por lo tanto, se necesitan diferentes funciones para comprobar la equivalencia entre conceptos y/o ontologías, así como funciones para encontrar inconsistencias entre ontologías. Se consideran las equivalencias y las inconsistencias desde dos puntos de vista: (1) atributos y (2) estructura organizativa de la ontología. Otra función permite decidir cuando dos conceptos son sinónimos, lo cual favorece que los usuarios puedan utilizar su propia terminología. A continuación se explican los pasos para el proceso de integración.

1. *Verificación de dominio*: Todas las ontologías a integrar deben cubrir el mismo dominio. Esta es la primera consideración a tener en cuenta para integrar varias ontologías usando este entorno.

2. *Selección de ontologías compatibles*: En esta propuesta no pueden integrarse dos ontologías si son inconsistentes o equivalentes.

3. *Selección de ontologías a integrar*: Como se pueden integrar varios conjuntos de ontologías, el criterio adoptado a la hora de elegir las ontologías a integrar es que el conjunto con mayor número de ontologías es el mejor.

4. *Generación de la ontología de integración*: Se genera una nueva ontología, reutilizando las ontologías fuentes. Primero se coloca el conjunto previamente

seleccionado de ontologías como parte de la nueva ontología. Esto se hace en la práctica insertando cada ontología fuente en la ontología derivada de la integración como hijo del concepto raíz, cuyo nombre es el dominio que está siendo integrado, a través de la relación *parte-de*.

5. *Generación de la ontología integrada o instanciada (Unificación)*: La siguiente fase es transformar la terminología de cada ontología fuente de forma que se comparta la terminología. La ontología obtenida de este proceso de unificación terminológica se llama ontología integrada e instanciada. La unificación de vocabulario consiste en seleccionar una ontología de referencia (por lo general la ontología con el mayor número de conceptos) y luego unificar los conceptos sustituyendo cada concepto por su sinónimo (si existe) en la ontología de referencia.

6. *Generación de la ontología final:* En este paso el sistema genera la ontología final y transformada a partir de la ontología integrada e instanciada. Para ello, se toma una de las sub-ontologías hijas de la raíz de la ontología integrada e instanciada como esqueleto para realizar la transformación y se agregan nuevos conceptos, atributos y relaciones a ésta para obtener la ontología final mostrada al usuario. Para agregar estos nuevos conceptos, cada ontología se procesa concepto a concepto, verificando la existencia de algún concepto equivalente o sinónimo en la ontología transformada. Si no hay ningún concepto equivalente o sinónimo, el concepto se agrega a la ontología transformada en su lugar correspondiente, que vendrá dictaminado por sus relaciones.

En este trabajo se define y formaliza un modelo de conocimiento propio para representar las ontologías. También se define un lenguaje propio para construir ontologías conforme a dicho modelo, lo que hace que su uso esté limitado a sus propias ontologías y no pueda aplicarse a ontologías escritas en lenguajes estándares como *XML*, *RDF*, *OWL*. En el modelo de integración sólo se tienen en cuenta los nombres de los atributos de los conceptos, obviando elementos de su estructura interna como los tipos y la cardinalidad, lo que provoca que las correspondencias finales no sean muy precisas. Realizan la integración de ontologías completas generando una tercera, para lo cuál las ontologías deben cumplir ciertos criterios de compatibilidad, por lo que se hace imposible una integración parcial, desechando las inconsistencias.

2.3.3.2 SMART, PROMPT, ANCHOR-PROMPT, PROMPTDIFF

Noy y Musen han desarrollado una serie de herramientas para la realización del mapeo, la alineación y el control de versiones de ontologías. Estas herramientas son SMART [Noy y Musen, 1999], ANCHOR-PROMPT [Noy y Musen, 2001], PROMPTDIFF [Noy y Musen, 2002], y PROMPT [Noy y Musen, 2003]. Todas están disponibles como *plugins* para el editor de ontologías de código abierto *Protégé*[5] [Grosso *et al*., 1999]. Estas herramientas utilizan los emparejadores de similitud lingüística entre los conceptos para el inicio de la fusión o el proceso de alineación y luego usan las estructuras ontológicas subyacentes del entorno *Protégé* (clases, ranuras, facetas) para publicar un conjunto de heurísticos para la identificación de las mejores correspondencias entre las ontologías.

PROMPT [Noy y Musen, 2003], inicialmente llamado SMART [Noy y Musen, 1999], está pensado para acoplarse en el entorno de edición de las ontologías. Es un algoritmo que proporciona un enfoque semi-automático de la fusión y la alineación de ontologías. Ayuda al desarrollador de ontologías en la ejecución de ciertas tareas de forma automática y sirve de guía en otras tareas donde se requiera su intervención. También determina las posibles inconsistencias resultantes de las acciones del usuario en el estado de la ontología y sugiere formas de solucionarlas. Se define el conjunto de operaciones básicas que se realizan durante la fusión y la alineación de ontologías y se determinan los efectos que la invocación de cada una de estas operaciones tiene en el proceso. El algoritmo está basado en un modelo de conocimiento muy general compatible con el protocolo *OKBC* (*Open Knowledge Base Connectivity*) [Chaudhri *et al*., 1998]. Cuando una decisión automática no es posible, el algoritmo guía al usuario a los lugares de la ontología donde su intervención es necesaria, sugiere posibles acciones y determina y propone soluciones para los conflictos en las ontologías.

ANCHOR-PROMPT (una extensión de PROMPT) [Noy y Musen, 2001] es una herramienta de alineación y fusión de ontologías con un sofisticado mecanismo del sistema para posibles términos coincidentes. Es un algoritmo de alineación híbrido que toma como entrada dos ontologías (representadas internamente en forma de grafos) y un conjunto de anclajes de los pares de términos relacionados que se identifican con la ayuda de técnicas basadas en cadenas, pruebas definidas por el usuario o algún otro emparejador de cálculo de

[5] *http://protege.stanford.edu/*

similitud lingüística. A continuación, el algoritmo los refina mediante el análisis de los caminos en las ontologías de entrada, limitadas por los anclajes, a fin de determinar los términos que aparecen con frecuencia en posiciones similares en caminos similares. Finalmente, sobre la base de las frecuencias y una retroalimentación del usuario, el algoritmo determina candidatos coincidentes.

La herramienta PROMPTDIFF [Noy y Musen, 2002] integra diferentes emparejadores heurísticos para comparar versiones de ontologías. Los autores combinan estos emparejadores en forma de punto fijo, utilizando los resultados de uno como entrada para los otros hasta que el algoritmo no produce más cambios. PROMPTDIFF aborda la comparación basada en la estructura de las ontologías. Su algoritmo trabaja en dos versiones de la misma ontología y se basa en la evidencia empírica de que una gran parte de las entidades no ha cambiado y que, si dos entidades tienen el mismo tipo y el mismo nombre o nombres muy similares es altamente probable que una sea una imagen de la otra.

Estas herramientas no tienen en cuenta las restricciones de las propiedades, tales como: máximo, mínimo, cardinalidad y tipos. Asumen que las ontologías cumplen con el modelo de conocimiento *OKBC*. Más específicamente, se basan en contar con un modelo de *Protégé* (basado en *frames* con clases, ranuras y facetas), lo que hace que el sistema no sea aplicable a otros modelos de conocimiento.

2.3.3.3 GLUE

Algunos trabajos como GLUE [Doan, *et al*., 2004], utilizan técnicas de aprendizaje automático y probabilidades para realizar el mapeo. La arquitectura de GLUE se compone de tres módulos fundamentales: el *módulo de estimación de la distribución*, el *módulo de estimación de la similitud*, y el *módulo de relajación del etiquetado*.

Módulo de estimación de la distribución: Este enfoque se basa en la idea de que muchas de las medidas prácticas de similitud se pueden definir basándose únicamente en la distribución de probabilidad conjunta de los conceptos involucrados. Por lo tanto, en lugar de comprometerse con una determinada definición de similitud, GLUE calcula la distribución de probabilidad conjunta de los conceptos y utiliza esta distribución para obtener una medida de similitud adecuada. Específicamente, para cualquier par de conceptos *A* y *B*, la distribución de probabilidad conjunta consta de $P(A, B)$, $P(A, B')$, $P(A', B)$, y $P(A', B')$, donde $P(A, B')$ es la

probabilidad de que una entidad en el dominio sea instancia del concepto (clase) *A* y no lo sea del concepto (clase) *B*.

Se calcula la probabilidad conjunta de los conceptos *A* y *B*, asumiendo que bajo ciertas circunstancias será un término *P*(*A*, *B*), que pueda ser aproximado como la fracción de instancias que pertenecen a *A* y a *B*. Por lo tanto, el problema se reduce a decidir para cada caso si pertenece a la intersección de *A* y *B*. Sin embargo, como la entrada en este caso incluye instancias de *A* y *B* aisladas, GLUE soluciona el problema mediante técnicas de aprendizaje automático utilizando un enfoque de aprendizaje multi-estrategia. La implementación actual de GLUE tiene dos módulos de aprendizaje de base: el de contenido y el de nombre, y un módulo de meta-aprendizaje que es una combinación lineal de los dos de base. A continuación se describen estos módulos brevemente:

- *Aprendizaje de contenido*: Este módulo aprovecha las frecuencias de las palabras en el contenido textual de una instancia para hacer predicciones, recordando que una instancia típicamente tiene un nombre y un conjunto de atributos, junto con sus valores. En la versión actual de GLUE, no se manejan los atributos directamente, sino que junto con sus valores son tratados como el contenido textual de la instancia. El módulo de aprendizaje de contenido emplea la técnica de aprendizaje de *Naive-Bayes* [Mitchell, 1997]. En general funciona bien con grandes elementos textuales como elementos con valores muy distintos y descriptivos, pero es menos efectivo con elementos numéricos pequeños.

- *Aprendizaje de nombre*: Es similar al módulo de aprendizaje de contenido, pero hace predicciones usando el nombre completo de la instancia de entrada, en vez de su contenido. El nombre completo de una instancia es la concatenación de nombres de conceptos que van desde la raíz de la taxonomía hasta esa instancia. Funciona mejor en nombres específicos y descriptivos, no siendo así con los nombres que son demasiado vagos.

- *Meta-Aprendizaje*: Combina las predicciones de los módulos de aprendizaje de base. El módulo de meta-aprendizaje asigna a cada módulo de aprendizaje un peso que indica en qué medida son ciertas sus predicciones. Luego se combinan las predicciones de los módulos de base a través de una suma ponderada.

Módulo de Estimación de la similitud: Aplica una función de similitud suministrada por el usuario para calcular el valor de similitud para cada par de conceptos, teniendo en cuenta las probabilidades conjuntas obtenidas en el módulo anterior. La salida de este módulo es una matriz de similitud entre los conceptos de las dos ontologías.

Módulo de Relajación de etiquetado: Por último, en el módulo de relajación de etiquetado, GLUE intenta aprovechar las restricciones del dominio e introduce heurísticos generales con el fin de mejorar la precisión del mapeo. El módulo de relajación de etiquetado utiliza la matriz de similitud obtenida en el módulo anterior. Un ejemplo de heurístico es la observación de que dos nodos probablemente coincidan si los nodos de su vecindad también coinciden. Un ejemplo de una restricción de dominio es: si el nodo *X* coincide con *profesor* y el nodo *Y* es un ancestro de *X* en la taxonomía, entonces es poco probable que *Y* coincida con *profesor asistente*. La relajación de etiquetado es una técnica de gran alcance usada ampliamente en el campo de la visión y el procesamiento de imágenes, y adaptado con éxito para solucionar los problemas de correspondencia y clasificación en el procesamiento del lenguaje natural y clasificación de hipertexto.

El uso de técnicas de aprendizaje automático hace que GLUE necesite muchos datos de entrenamiento, lo que provoca que en ocasiones algunos nodos no puedan ser mapeados si esto no ocurre. Además, los módulos de aprendizaje utilizados en este enfoque son clasificadores de texto de propósito general relativamente simples, lo que hace que no se obtengan buenos resultados en esquemas de conocimiento más específicos. Por otro lado la técnica de relajación de etiquetado desarrolla optimizaciones que a veces convergen a un solo máximo local, provocando que no se encuentren resultados correctos para todos los nodos, que nodos ambiguos no puedan ser mapeados automáticamente y que con frecuencia se obtengan correspondencias incorrectas. Otro factor que limita la funcionalidad de GLUE es que no se tienen en cuenta los atributos de los conceptos para el mapeo.

2.3.3.4 CODI

CODI (*Combinatorial Optimization for Data Integration*) [Noessner *et al*., 2010] es un sistema lógico-probabilístico, que obtiene alineaciones para individuos, conceptos y propiedades de dos ontologías heterogéneas. CODI combina los esquemas de información lógicos y las medidas de similitud léxica con una semántica definida para emparejamiento de *A-Box* y *T-Box*, utilizando la sintaxis y semántica de la *Lógica de Markov* [Domingos *et al*.,

2008]. Las alineaciones se calculan a través de la resolución de los correspondientes problemas de optimización combinatoria. El sistema mapea las entidades con mayor similitud léxica y refuerza la coherencia del resultado utilizando restricciones de cardinalidad, coherencia, y estabilidad. CODI implementa un método de agregación de múltiples medidas de similitud léxica y también permite el reconocimiento de pares que constituyen diferentes versiones de la misma ontología.

Correspondencia de T-Box: En lógica descriptiva los *T-Box* se utilizan para describir entidades jerárquicas y sus relaciones, por tanto la correspondencia de *T-Box* abarca los conceptos y sus propiedades. Dadas dos ontologías y una primera medida de similitud a priori, introducen predicados observables para modelar la estructura con respecto a conceptos y propiedades. En particular, se añaden átomos de base para los predicados observables de acuerdo a ciertas reglas definidas. Los átomos de base de predicados observables se añaden al conjunto de restricciones duras, obligándolas a mantener las alineaciones calculadas. Los predicados ocultos, por otro lado, modelan las correspondencias de los conceptos buscados y de las propiedades respectivamente. Dado el estado de los predicados observables, se necesita determinar el estado de los predicados ocultos que maximicen la probabilidad a posteriori de la posible palabra correspondiente. A los átomos de base de estos predicados ocultos se asignan los pesos especificados por la similitud a priori. Cuanto mayor sea este valor para una correspondencia es más probable que la correspondencia *a-priori* sea correcta.

Correspondencia de A-Box: En lógica descriptiva los *A-Box* indican las relaciones entre los individuos y las clases o conceptos a los cuáles pertenecen en la jerarquía. En otras palabras, la correspondencia de *A-Box* se refiere a las instancias. CODI utiliza métodos de similitud léxica para la obtener la similitud entre las instancias, pero considera que este proceso puede ser muy costoso en ontologías con muchas instancias, por lo que no aplica el método a todas las parejas de individuos sino que utiliza las propiedades objeto para reducir el número de comparaciones.

Restricciones de cardinalidad: Un método frecuentemente aplicado en situaciones del mundo real es la selección de una correspondencia funcional *uno-a-uno*. En el marco de la *Lógica de Markov* se puede incluir un conjunto de restricciones de cardinalidad fuertes, restringiendo la correspondencia para que sea funcional y *uno-a-uno*.

Restricciones de coherencia: La incoherencia ocurre cuando los axiomas conducen a contradicciones lógicas en las ontologías, por lo que claramente es deseable evitar la incoherencia durante el proceso de alineación. Todos los enfoques existentes para la reparación de alineaciones eliminan correspondencias incoherentes después del cálculo de la alineación. En el marco de la *Lógica de Markov* se pueden incorporar restricciones reductoras de incoherencia durante el proceso de alineación.

Restricciones de estabilidad: Algunos enfoques de correspondencia de ontologías propagan la evidencia de la alineación derivada de las relaciones estructurales entre conceptos y propiedades. Estos métodos aprovechan el hecho de que las evidencias existentes para la equivalencia de los conceptos *A* y *B* también hacen más probable que, por ejemplo, conceptos hijos de *A* y de *B* sean equivalentes. Uno de estos enfoques de propagación de evidencia es *SF* (*Similarity Flooding*) [Melnik *et al*., 2002].

La versión actual de CODI utiliza una medida de similitud léxica muy simple basada únicamente en la distancia *Levenshtein* [Levenshtein, 1966] pero su punto fuerte es la modularidad, permitiendo la incorporación de diferentes medidas de similitud de manera sencilla a través de las funciones de agregación.

2.3.3.5 AgreementMaker

AgreementMaker [Cruz *et al*., 2009] se enfoca en aplicaciones del mundo real y en particular aplicaciones geoespaciales. Comprende varios algoritmos de correspondencia que se pueden utilizar para mapear (o alinear) ontologías. La arquitectura de AgreementMaker se basa en una pila de tres capas cuyos componentes son: los *algoritmos de correspondencia*, los *módulos de combinación y evaluación*, y el *módulo de alineación final*. El proceso de correspondencia de un algoritmo genérico se puede dividir en dos módulos principales: (1) el *cálculo de similitud*, en el que cada concepto de la ontología fuente es comparado con todos los conceptos de la ontología destino, produciendo así dos matrices de similitud (una para las clases y otra para las propiedades) y (2) la *selección de correspondencias* en la que se seleccionan sólo los mejores valores de similitud de la matriz de acuerdo a un determinado umbral y a la cardinalidad de las correspondencias (e.g. *1-1*, *1-N*, *N-1*, *M-N*).

El primer paso es aplicar los algoritmos de correspondencia. Se utilizan *algoritmos de correspondencia basados en conceptos* que se apoyan en la comparación de cadenas, y *algoritmos de correspondencia estructurales* que consideran un subgrafo de la ontología.

Entre los *algoritmos de correspondencia basados en conceptos* que utiliza están: el Emparejador Básico de Similitud *BSM* (*Base Similarity Matcher*) que calcula la similitud entre los conceptos mediante la comparación de todas las cadenas asociadas a ellos; el Emparejador Paramétrico Basado en Cadenas *PSM* (*Parametric String-based Matcher*), configurado para utilizar medidas de subcadenas y medidas de distancia de edición; y el Emparejador Multi-Palabra basado en vector *VMM* (*Vector-based Multi-word Matcher*) que transforma las cadenas resultantes en vectores y luego calcula su similitud con la medida de *similitud del coseno* [Duda *et al*., 2001]. Estos algoritmos de correspondencia se amplían añadiendo un conjunto de herramientas léxicas que permiten el tratamiento de sinónimos.

Los *algoritmos de correspondencia estructurales* incluyen: el algoritmo de Herencia de Similitud de los Descendientes *DSI* (*Descendant's Similarity Inheritance*) como algoritmo de entrada; el Emparejador Buscador de Grupo *GFM* (*Group Finder Matcher*), en el que se identifican grupos de conceptos y propiedades de las ontologías y se supone que dos conceptos (o propiedades) que pertenecen a dos grupos que no fueron mapeados por el algoritmo de entrada probablemente tengan significados diferentes y no se deban mapear; y finalmente el algoritmo Estructural Iterativo de Instancia *IISM* (*Iterative Instance Structural Matcher*), que tiene en cuenta si las instancias de las ontologías y las clases cuyos individuos han sido mapeados pueden ser alineadas. También en este algoritmo se consideran los valores de las propiedades, subpropiedades, superclases, subclases y cardinalidades, así como el rango y dominio de las propiedades.

Posteriormente los módulos de combinación y evaluación se utilizan juntos, de la siguiente manera. La combinación lineal ponderada *LWC* (*Linear Weighted Combination*) combina sus entradas (por ejemplo, a partir de distintos algoritmos de correspondencia de cadenas), utilizando una medida de calidad de confianza local proporcionada por el módulo de evaluación, a fin de asignar automáticamente el peso de cada resultado calculado por los algoritmos de correspondencia de entrada. Después de este paso, se obtiene un único conjunto combinado de alineaciones que incluye los mejores resultados de cada uno de los algoritmos de entrada. El módulo de alineación final recibe como entrada la cardinalidad de las

alineaciones (e.g. *1:1*) y un umbral, para retornar finalmente el mejor conjunto de alineaciones.

En AgreementMaker existe la limitación de que la similitud estructural depende absolutamente de los valores de la similitud lingüística, lo que hace que no se obtengan resultados correctos en ontologías cuyos conceptos no sean similares lingüísticamente, aunque las ontologías sean muy parecidas en su estructura.

2.3.3.6 ASMOV

ASMOV (*Automated Semantic Mapping of Ontologies*) [Jean-Mary *et al*., 2009] calcula iterativamente la similitud entre las entidades de dos ontologías mediante el análisis de cuatro aspectos: los elementos léxicos (identificadores, etiquetas y comentarios), la estructura relacional (jerarquía ancestro-descendiente), la estructura interna (restricciones de propiedades de los conceptos, tipos, dominios y rangos de las propiedades, valores de datos para los individuos) y la extensión (instancias de las clases y valores de las propiedades).

Similitud léxica: En ASMOV, por defecto, se utiliza *WordNet* para validar los *tokens* y recuperar la semántica de las palabras. Específicamente se utilizan las relaciones de sinónimos, antónimos e hiperónimos. Si uno de los sinónimos de la palabra de origen es igual a uno de los sinónimos de la palabra objetivo, las palabras se consideran semánticamente iguales. Por otro lado, si el significado de la palabra de origen contiene un antónimo a cualquiera de los sinónimos de la palabra objetivo, las palabras se consideran semánticamente ortogonales. Cuando las palabras no son sinónimos ni antónimos, se recorre el hiperónimo de cada una de las definiciones de forma recursiva para encontrar el camino más corto hacia una definición común. ASMOV utiliza la ecuación de distancia semántica propuesta por Lin [Lin, 1998] para calcular una medida de similitud textual. Para cada hiperónimo encontrado, también se busca su antónimo; el algoritmo infiere que si los padres de dos palabras son antónimos, entonces dichas palabras también deben ser antónimos. Si el proceso de *tokenización* no produce ninguna palabra significativa, el texto se utiliza en su forma original y se utiliza la *distancia de Levenshtein* [Levenshtein, 1965] para calcular la similitud. Finalmente la similitud léxica es calculada combinando en una suma ponderada los pesos de las similitudes de los identificadores, etiquetas y comentarios.

Similitud Externa: La similitud externa se calcula mediante la combinación de las similitudes entre los padres y los hijos de las entidades que se comparan. Como las entidades pueden contener múltiples padres e hijos, el cálculo de similitud se normaliza con el fin de restringir los resultados entre 0 y 1. Por ejemplo, cada padre de la entidad de origen se empareja con el padre más cercano de la entidad de destino, todas las medidas de similitud de los pares de padres se suman y se normalizan dividiendo por el número total de entidades padres de la fuente y destino. Para combinar las dos medidas de similitud se utiliza la suma ponderada.

Similitud Interna: Debido a que las construcciones internas de un concepto son diferentes a las construcciones internas de una propiedad, su medida de similitud interna se calcula de manera diferente.

- *Similitud interna de conceptos*: Esta medida de similitud es una combinación de la similitud entre las propiedades y las restricciones locales (restricciones de cardinalidad y valor). Como un concepto puede tener varias propiedades, la medida de similitud se normaliza en una forma similar a la similitud externa.

- *Similitud interna de las propiedades*: El cálculo de la similitud interna de las propiedades supone comparar el tipo, el dominio y rango. En ASMOV se considera que la similitud entre dos tipos sólo puede ser 1 si los tipos son los mismos y 0 en caso contrario. En el caso de la similitud de dominio, se utiliza la mejor similitud entre los mejores pares de dominio (conceptos de origen y destino), teniendo en cuenta que el dominio de una propiedad puede estar constituido por varias clases. Este mismo método se utiliza para calcular la medida de similitud del rango. Una vez comparados los rangos de dos propiedades dato, se utiliza la medida de similitud semántica propuesta por Wu y Palmer [Wu y Palmer, 1994]. La medida de similitud interna final se calcula combinado las similitudes de tipo, dominio y rango.

Similitud de los individuos: ASMOV no depende de los individuos para encontrar la similitud entre conceptos y propiedades. Sin embargo, si los individuos están presentes, la similitud entre ellos puede ser explotada con el fin de perfeccionar el mapeo. ASMOV considera que dos individuos son similares si sus estructuras internas son las mismas y coinciden los correspondientes valores literales.

Como principal limitación de ASMOV podemos mencionar que se utilizan pesos fijos para cada característica en el cálculo de las sumas ponderadas para combinar las similitudes, lo que trae como consecuencia que a veces no se obtengan correspondencias correctas en ontologías de determinados dominios. Las necesidades de convergencia que se observan en la solución actual de ASMOV pueden hacer que el proceso iterativo termine antes de tiempo y por tanto no se produzca un resultado óptimo.

2.3.3.7 SOBOM

SOBOM (*Sub-Ontology based Ontology Matching*) [Xu *et al*., 2010] fue desarrollado para la alineación de ontologías de propósito general. Sobre la base de dos puntos de vista diferentes, se proponen tres algoritmos en la versión actual. El primero es un *generador de anclaje*, el segundo es un algoritmo de correspondencia estructural *SISF* (*Semantic Inductive Similarity Flooding*) inspirado en los algoritmos Anchor-Prompt [Noy y Musen, 2001] y *SF* [Melnik *et al*., 2002]. El último es un algoritmo de correspondencia relacional que utiliza los resultados de *SISF* para obtener las alineaciones de las relaciones. Además, se integra un extractor de sub-ontologías *SoE* (*Sub-ontology Extractor*) para extraer sub-ontologías de acuerdo a los anclajes obtenidos por el algoritmo de correspondencia lingüístico y clasificarlas por su profundidad en orden descendente. SOBOM en general es un método secuencial, por lo que no considera la forma de combinar los resultados de los diferentes algoritmos.

El generador de anclajes se basa en la idea de que el contexto local de una entidad puede expresar el significado de la misma. En detalle, el contexto local de una entidad, incluye los siguientes aspectos: la información textual (etiqueta, identificador, comentarios, etc), la información estructural (el número de súper o sub conceptos, el número de restricciones) y la información individual (el número de individuos si existen). Teniendo esto en cuenta, definen tres matrices de similitud, respectivamente y eligen los valores máximos como resultado final.

SISF utiliza *RDF* para representar las ontologías, y utiliza los puntos de anclaje para inducir la construcción del grafo de propagación de similitud para las sub-ontologías. *SISF* sólo genera correspondencias *concepto-a-concepto*.

Lo más importante es la integración de *SoE* en SOBOM que extrae sub-ontologías de acuerdo con los anclajes. *SoE* clasifica las sub-ontologías extraídas de acuerdo a sus profundidades. Las reglas para extraer sub-ontologías son las siguientes. Por un lado, sólo los

sub-conceptos de anclaje están incluidos en la ontología. En otras palabras, una sub-ontología es una taxonomía que tiene como raíz el anclaje. Después de extraer las sub-ontologías, SOBOM las mapea de acuerdo a su profundidad en la ontología original. Primero mapea las sub-ontologías con mayor valor de profundidad. Usando *SoE*, SOBOM puede reducir la escala de la ontología y hacer más fácil de operar las sub-ontologías en *SISF*.

Como punto débil de SOBOM podemos mencionar que en los casos en que el algoritmo de correspondencia lingüístico no produce resultados, el sistema devuelve resultados nulos. Este método necesita anclajes para extraer las sub-ontologías, por lo que depende completamente de la precisión del algoritmo de anclaje, que al ser puramente lingüístico ofrece malos resultados en caso de que no existan literales para los conceptos.

2.3.3.8 Eff2Match

En Eff2Match (*Effective and Efficient ontology matching tool*) [Watson *et al*., 2010] el proceso de alineación se divide en cuatro pasos: 1) *Generación de anclaje*, 2) *Generación de candidatos*, 3) *Expansión de anclaje* y 4) *Impulso de puntuación iterativo*. A continuación se explican brevemente cada uno de estos pasos:

- *Generación de anclaje*: En este paso las entidades correspondientes se identifican utilizando una técnica de emparejamiento exacto de cadenas. Primero se realiza la normalización y la eliminación de los delimitadores y luego se utiliza una tabla *hash* para mapear los nombres locales preprocesados y etiquetas para sus entidades correspondientes. Si se encuentra una coincidencia de nombre local o etiqueta en la tabla, se considera que la entidad correspondiente en la ontología de destino es equivalente a la entidad de origen.

- *Generación de candidatos*: En la etapa de generación de candidatos, se enumeran los candidatos para las entidades en la ontología de origen que no fueron mapeadas en la etapa anterior mediante un modelo de espacio vectorial *VSM* (*Vector Space Model*) [Salton *et al*., 1975]. Para cada concepto, se generan tres vectores a partir de las anotaciones (nombre local, etiquetas y comentarios) en los ancestros, descendientes y el concepto en sí. Para cada propiedad, los vectores generados consisten en sus anotaciones, el dominio y el rango del concepto al que pertenece. La similitud *VSM* entre dos conceptos $C1_i$ y $C2_j$ es una agregación de la similitud

del coseno [Duda *et al*., 2001] entre los vectores de los conceptos, ancestros y descendientes usando una media ponderada. Los valores de similitud para las propiedades se obtienen de manera similar y se utilizan dos matrices para almacenar los valores de las similitudes entre conceptos y propiedades respectivamente. Las similitudes *VSM* se normalizan a [0,1] dividiendo cada entrada en la matriz por su mayor valor. La selección de candidatos se realiza para cada entidad de la ontología fuente tomando las primeras *k* entidades en la ontología destino de acuerdo a sus similitudes *VSM*.

- *Expansión de anclaje*: En esta fase se identifican las parejas de entidades más parecidas mediante la comparación de las entidades de origen con sus entidades candidatas utilizando métodos terminológicos. En Eff2Match, se utiliza un algoritmo de eliminación de términos *TRA* (*Term-Removing Algorithm*) con fines de eficiencia.

- *Impulso iterativo*: En la etapa final del proceso de correspondencia, se utiliza un proceso de impulso iterativo para identificar más pares de conceptos equivalentes utilizando el conjunto ampliado de anclaje. En esta etapa, el algoritmo intenta mapear los conceptos de origen que no han sido emparejados con sus candidatos iterativamente. En cada iteración, los conceptos de origen se clasifican en base a la suma de los ancestros y descendientes que se encuentren en el conjunto de anclaje. Se seleccionan los primeros *k* conceptos de origen y se utiliza una fórmula basada en la superposición estructural para aumentar la puntuación de sus candidatos en función del número de ancestros y descendientes comunes.

Eff2match se basa en las similitudes lingüísticas y estructurales, por lo que en ontologías con diferente lingüística y estructura se obtienen malos resultados y le resulta difícil obtener resultados correctos en ontologías heterogéneas. La versión actual del sistema solo mapea conceptos y propiedades con relaciones de equivalencia.

2.3.3.9 GeRMeSMB

Es la integración de dos herramientas dirigidas al mapeo de diferentes esquemas: Por un lado GeRoMeSuite ofrece una variedad de emparejadores, mientras que SMB proporciona información de cómo combinarlos para mejorar sus resultados.

GeRoMeSuite

El sistema se basa en *GeRoMe* [Kensche *et al*., 2007] que representa modelos de diferentes lenguajes de modelado (por ejemplo, *XML, OWL, SQL*) de una manera genérica. Emplea el enfoque de modelado basado en roles. Por lo tanto, los operadores pueden ser utilizados polimórficamente independientes de los metamodelos concretos y pueden centrarse sólo en los roles para su ejecución. Además de proporcionar un marco para la gestión de modelos, *GeRoMeSuite* implementa varios operadores fundamentales como: correspondencia, fusión, y composición de ontologías.

- *Correspondencia de esquemas*: La implementación de la correspondencia, que integra *GeRoMeSuite* ofrece una gran flexibilidad para la combinación de distintos algoritmos de nivel de elemento, de nivel estructural, y las estrategias de agregación para combinar los resultados de los métodos de correspondencia para ser utilizados en los pasos posteriores.

- *Mezcla o fusión de* esquemas: La implementación del operador de fusión se basa en un fundamento teórico que define los mapas intensionales (mapas que describen las relaciones del mundo real representados por elementos del modelo) como conjuntos de relaciones entre semánticas del mundo real *RWS* (*Real World Semantics*) de los elementos del modelo correspondientes. El resultado de la fusión es un modelo *GeRoMe* válido, que sin embargo, no tiene por qué ser un modelo válido para algún metamodelo nativo determinado. Por lo tanto, el resultado de la fusión tiene que ser transformado en el metamodelo de destino mediante un operador, antes de que pueda ser exportado.

- *Composición de mapas*: El esquema de fusión requiere mapas intensionales, pero estas correspondencias son menos útiles para el mapeo de ejecución. Por ese motivo para esta tarea se necesitan mapas extensionales (mapas que describan las relaciones entre dos modelos a nivel de instancias, incluidas las funciones de conversión). En *GeRoMeSuite* los mapas extensionales se basan en las dependencias de segundo orden, y permiten la agrupación de elementos. El mapeo permite cuantificadores universales sobre las variables, cuantificadores existenciales sobre los símbolos de funciones, e igualdades.

SMB

SMB (*Schema matching boosting*) es un servicio que consiste en un conjunto de herramientas para mejorar el rendimiento de los algoritmos de correspondencia de esquemas. SMB funciona en 3 modos: mejora, aprendizaje y recomendación.

En el modo de mejora, SMB recibe las matrices de correspondencia producidas por GeRoMeSuite (con valores de similitud entre atributos en el rango de [0,1]) y realiza un análisis de los resultados por fila y columna. Posteriormente, utiliza algoritmos para mejorar los resultados de las filas y columnas "prometedoras" y debilitar los resultados de las filas y columnas que no lo son. El modo de aprendizaje se utiliza para realizar un entrenamiento *off-line* de SMB en el comportamiento del rendimiento de los algoritmos de correspondencia, ejecutando diversas tareas que se clasifican en clases de acuerdo a sus características a priori, tales como el tamaño del esquema. La recomendación clasifica una tarea de correspondencia determinada en tiempo de ejecución, proporcionando el conjunto de pesos recomendado para los diferentes componentes de los sistemas de correspondencia. Se proporcionan interfaces genéricas para permitir el uso de SMB por cualquier sistema de correspondencia a través de su invocación en la línea de comandos.

Como limitación de GeRMeSMB podemos mencionar que el sistema no da buenos resultados en ontologías con información incompleta (si faltan etiquetas, jerarquía o comentarios) y en los casos en que los espacios de nombres no se definen de manera estándar. Por ser un sistema muy genérico no funciona del todo bien en ontologías reales en dominios específicos. También se presentan problemas de escalabilidad en ontologías grandes.

2.3.3.10 Framework de Rong Pan

La propuesta de Pan [Pan *et al*., 2005] es un enfoque basado en *Redes Bayesianas* (*RBs*) para el mapeo automático de ontologías. Los autores definen un *framework* probabilístico para modelar la incertidumbre en la Web Semántica que han denominado *BayesOWL*. En este enfoque, las ontologías de origen y destino son primero traducidas en *RBs*; el mapeo de conceptos entre las dos ontologías es tratado como razonamiento evidencial entre las dos *RBs* traducidas. Las probabilidades necesarias para la construcción de tablas de probabilidad condicional (*TPCs*) durante la traducción y para la medición de similitud semántica durante el mapeo se aprenden utilizando técnicas de clasificación de textos donde cada concepto en una

ontología se asocia con un conjunto de documentos de texto semánticamente relevantes, que se obtienen por minería de Web guiada por ontologías.

El *framework* consta de tres componentes: 1) un módulo *BayesOWL* para traducir a *RBs* las ontologías dadas, 2) un módulo de mapeo de conceptos que tiene como entrada un conjunto aprendido de similitudes y encuentra correspondencias entre conceptos de dos ontologías diferentes basado en las pruebas de razonamiento a través de las dos *RBs* y 3) un módulo de clasificación de texto basado en aprendizaje de datos e información probabilística ontológica de la Web dentro de ontologías individuales y entre conceptos en dos ontologías diferentes.

- *BayesOWL* es un marco que proporciona un conjunto de reglas y procedimientos de traducción directa de una ontología *OWL* en una estructura de *RBs* (un grafo acíclico dirigido) y un método que utiliza restricciones de probabilidad disponibles acerca de las relaciones de clases en la construcción de las tablas de probabilidad condicional (*TPCs*). La *RB* traducida que conserva la semántica de la ontología original, puede soportar el razonamiento ontológico y las inferencias bayesianas.

- *Mapeo de conceptos entre ontologías utilizando mapeo de RBs*: El proceso de mapeo de conceptos se divide en dos categorías: 1) el mapeo simple ("*1-1*"), que se traduce en una noción de enlace semántico probabilístico entre un par de conceptos/variables, 2) y el mapeo múltiple ("*m-n*"), que abarca los casos en que cada variable en una *RB* es mapeada a varias similares en la otra.

    1. *Enlace semántico probabilístico simple (1-1)*: En este caso se asume que la información de similitud entre la variable *A* de RB_1 y *B* de RB_2 es capturada por la distribución de probabilidad conjunta *P* (*A*, *B*). Esta distribución está en un espacio de probabilidad, indicado como $PS^{1,2}$, que, a pesar de estar relacionado, es diferente de los espacios para *A* y *B*, denotados como PS^1 y PS^2, respectivamente. La función de probabilidad para el enlace semántico probabilístico simple se calcula mediante dos aplicaciones de la *Regla de Jeffrey* [Pan *et al*., 2005] a través de los tres espacios: primero de PS^1 a $PS^{1,2}$ y finalmente de $PS^{1,2}$ a PS^2.

2. *Enlace semántico probabilístico múltiple (m-n)*: En este caso el mapeo de cada variable A^i perteneciente a RB_1 con respecto a RB_2 abarca el mapeo de todos los vínculos semánticos que se inician desde el extremo A^i de RB_1 y terminan en cada concepto similar B^J en RB_2. Esta correspondencia "*m-n*" puede ser llevada a cabo por un proceso que combina tanto *la regla de Jeffrey* como el *Procedimiento de Ajuste Proporcional Iterativo* (*IPFP*) [Pan *et al*., 2005]. Este proceso es iterativo sobre los enlaces en un ciclo hasta la convergencia.

- *Módulo de aprendizaje de probabilidades de la Web*: En este trabajo, se utilizan las distribuciones de *probabilidad a priori P* (*C*) para capturar la incertidumbre acerca de conceptos (es decir, la probabilidad de que un individuo arbitrario pertenezca a la clase *C*), las distribuciones de *probabilidad condicional* P_{cond} (*C*|*D*) para las relaciones entre *C* y *D* en la misma ontología, y las distribuciones de *probabilidad conjunta P* (*C, D*) para la similitud semántica entre conceptos *C* y *D* de diferentes ontologías. Estas probabilidades se aprenden utilizando técnicas de clasificación de textos [McCallum y Nigam, 1998; Craven *et al*., 2000], mediante la asociación de un concepto con un grupo de muestras de documentos de texto denominado *ejemplares*.

El principal inconveniente de este *framework* es que la similitud entre los conceptos se basa únicamente en los valores de las funciones de probabilidad, que se calculan teniendo en cuenta los ejemplares relevantes a cada concepto recuperados de la Web. Esto provoca que se requiera una función de relevancia muy precisa o algoritmos de minería de datos de post-procesamiento de los documentos para garantizar que se minimice la cantidad de documentos no relevantes a considerar. Otro factor que afecta el resultado final de las funciones de probabilidad es que no todos los conceptos tienen el mismo índice de popularidad en la Web, por lo que muchas veces no se obtienen probabilidades correctas. En este trabajo no se hace un procesamiento léxico de la terminología de los conceptos ni se tienen en cuenta sus propiedades para obtener la similitud.

2.3.4 Clasificación de los métodos de Alineación de Ontologías consultados

La Tabla 2.2 muestra una clasificación de los métodos estudiados según las técnicas básicas utilizadas. Se puede observar que la mayoría son métodos híbridos, es decir, que hacen uso de combinaciones de varias técnicas para obtener mejores resultados en la alineación. La mayoría de los trabajos que hemos estudiado utilizan técnicas de nivel de elemento de cadenas, incorporando en algunos casos otras técnicas como las basadas en lenguaje y el uso de recursos lingüísticos externos. En cuanto al nivel de estructura gran parte de los trabajos utilizan la información de la jerarquía taxonómica. El aprendizaje automático y las técnicas heurísticas constituyen los enfoques más utilizados, y se aplican en los casos en que o bien no existe la ontología compartida, o no se puede garantizar que una de las dos va a ser utilizada.

Tabla 2.2. Clasificación de los trabajos analizados.

Métodos	**Nivel de Elementos**				**Nivel Estructural**		**Otras técnicas**
	Cadenas	Lenguaje	Recursos lingüísticos	Restricciones	Grafos	Taxonomía	
Fdez-Breis	X					X	
Anchor-PROMPT	X				X		Heurísticos, retroalimentac. del usuario
GLUE	X					X	Aprendizaje automático, probabilidades
CODI	X			X		X	Probabilidades (Lógica de Markov)
AgrMaker	X			X	X	X	
ASMOV	X	X	X	X		X	
SOBOM	X			X	X	X	
Eff2Match	X	X	X	X		X	
GeRMeSMB	X			X		X	Aprendizaje automático
Pan					X		Redes Bayesianas

2.4 Resumen y consideraciones finales

En este capítulo hemos tratado el tema de la alineación de ontologías como elemento fundamental para mejorar la interoperabilidad en la Web Semántica. Comenzamos explicando la arquitectura de la Web Semántica y las diferentes tecnologías involucradas en cada capa, para a continuación centrarnos en la capa ontológica, donde detallamos los tipos de ontologías, los elementos fundamentales que componen la estructura de las ontologías y los lenguajes más utilizados actualmente para su construcción. Finalmente hemos hecho un estudio de algunas de las investigaciones más significativas en el campo de la alineación de ontologías. Estas investigaciones por lo general proponen métodos híbridos porque hacen uso de técnicas elementales y estructurales, combinadas con otras técnicas clásicas como el aprendizaje automático, las probabilidades y los métodos heurísticos para conseguir mejores resultados.

A pesar de las numerosas contribuciones en el marco de la alineación de ontologías que se han desarrollado hasta el momento, no existe una solución integrada que haya demostrado ser lo suficientemente efectiva y robusta como para ser tomada como base para el desarrollo futuro y que pueda ser usada por usuarios no expertos. Uno de los desafíos actuales en el campo de la alineación de ontologías es el tratamiento de la incertidumbre. Este problema ha sido abordado de diversas maneras, como por ejemplo utilizando matrices de similitud para modelar el mapeo de ontologías como un proceso incierto, donde un algoritmo de alineación se mide por el ajuste de su estimación de la certeza de una correspondencia con el mundo real. La incertidumbre también ha sido reducida de manera iterativa, fortaleciendo o descartando las hipótesis iniciales para perfeccionar así las medidas iniciales de correspondencia. Algunos trabajos han utilizado esquemas probabilísticos, adjuntando a las correspondencias un valor de probabilidad y otros han optado por técnicas de aprendizaje automático. Sin embargo, ninguno de estos enfoques ha conseguido resolver completamente de manera automática el problema de la incertidumbre en la alineación de ontologías. Otro de los frentes abiertos en este campo es la mejora de las medidas de similitud con el fin de obtener valores más precisos para minimizar o eliminar la intervención humana en el proceso. Para afrontar estas limitaciones surge *FuzzyAlign*, que es la contribución fundamental de esta tesis doctoral.

CAPÍTULO 3. MEDIDAS DE SIMILITUD

El ingenio consiste en conocer la semejanza de las cosas que son diferentes, y la diferencia de las cosas que son iguales.

Anne-Louise

En este capítulo abordamos las medidas de similitud terminológicas y estructurales empleadas en nuestra propuesta para la alineación de ontologías. Comenzamos con una revisión de las principales medidas de similitud encontradas en la literatura y finalmente explicamos las medidas propuestas en este trabajo. En el caso de la similitud terminológica utilizamos dos medidas: la lingüística, que tiene en cuenta las relaciones de sinonimia, derivación de las palabras que componen los términos y un factor léxico basado en las distancias entre las palabras; y la semántica, que se calcula a partir de búsquedas contextualizadas de los términos en la Web. En la similitud estructural proponemos medidas basándonos en la estructura interna, teniendo en cuenta las propiedades de los conceptos y la estructura relacional utilizando la jerarquía taxonómica de las ontologías.

3.1 Similitud Terminológica

En esta tesis utilizamos ontologías terminológicas de propósito general en inglés, por ser el idioma más extendido para describir la terminología de la mayoría de las ontologías y por existir una serie de herramientas léxicas que facilitan el procesamiento lingüístico en este idioma. También se utilizan algunas ontologías de dominio para las cuáles existen herramientas léxicas que permiten establecer similitudes entre sus términos, como es el caso de las ontologías médicas, y se han hecho pruebas para evaluar el comportamiento del sistema en ontologías de dominio para las cuales no existen herramientas léxicas especializadas como es el caso de las ontologías de redes de sensores.

Para calcular la similitud entre las terminologías de las entidades aplicamos técnicas semánticas y lingüísticas a nivel de los elementos de las ontologías. Estas técnicas se basan en el análisis de las entidades de las ontologías por separado, ignorando sus relaciones con otras entidades. En este apartado hacemos un estudio de las diversas técnicas de similitud semántica y lingüística existentes en la literatura, centrándonos en las medidas utilizadas en esta Tesis.

En el caso de la similitud semántica mencionamos algunos de los índices binarios más utilizados, haciendo hincapié en el *coeficiente de Jaccard*, y explicamos su aplicación en el contexto de nuestro trabajo. En el caso de la similitud lingüística comenzamos con un análisis de las técnicas para el cálculo de la distancia entre cadenas de caracteres. Después mencionamos las características fundamentales de las herramientas léxicas *WordNet* y *UMLS*, así como las diferentes medidas de similitud existentes que hacen uso de estas herramientas. Finalmente, describimos nuestras medidas propuestas para el cálculo de la similitud lingüística.

3.1.1 Medidas de Similitud Semántica.

3.1.1.1 Medidas binarias de Similitud y Distancia

Las medidas binarias de similitud y distancia juegan un papel primordial en problemas de análisis de patrones, clasificación y agrupamiento, y han sido aplicadas en muy diversos campos de la ciencia como la biología, la antropología, la química, las ciencias de la computación, la teoría de la información, la geología, la física y la estadística, entre otros. Desde el punto de vista matemático, la distancia se define como un grado cuantitativo del alejamiento de dos objetos, mientras que por el contrario, la similitud denota la proximidad [Cha, 2007]. La elección de una medida de distancia o similitud depende del tipo de aplicación y de las características de los objetos.

Para explicar los índices binarios de similitud que son objeto de esta sección, usaremos los parámetros que se muestran en la Tabla 3.1. Supondremos que queremos evaluar la similitud entre dos características, *a* y *b*, dentro de un vector binario de elementos de dimensión *n*. El valor *x* representará el número de elementos en los que están presentes ambas características, el valor *y* representará el numero de elementos con la presencia de la característica *a* y la ausencia de la característica *b*, mientras que *z* será el número de elementos con la presencia de la característica *b* y la ausencia de la característica *a*. Finalmente, *w* será el número de elementos con ausencia de ambas características. La suma de $x+y$ representará, por tanto, el número de total de elementos con la característica *a*, mientras que la suma de $x+z$ representará el numero total de elementos con la característica *b*.

Tabla 3.1. Tabla de parámetros para el cálculo de índices binarios de similitud.

a/b	***1 (presencia)***	***0 (ausencia)***	***Total***
1 (presencia)	x	y	x+y
0 (ausencia)	z	w	z+w
Total	x+z	y+w	n= x+y+z+w

Las medidas binarias de similitud se pueden clasificar atendiendo a sus principales características en tres grupos fundamentales: las basadas en distancia, las basadas en correlación y las basadas en no correlación. A continuación hacemos un breve repaso de las principales medidas encontradas en la literatura en cada categoría.

Medidas basadas en distancia

Podemos definir la distancia entre las entidades *a* y *b* como una función *d* (*a*, *b*) que mide la disimilitud entre ellas. Es decir, cuanto menor sea el valor de la distancia, mayor será la similitud entre las entidades. Las funciones basadas en distancia deben cumplir las siguientes propiedades:

1. Positividad: $d(a, b) >= 0$
2. Simetría: $d(a, b) = d(b, a)$
3. Desigualdad triangular: $d(a, c) <= d(a, b) + d(b, c)$

No podemos hablar de las medidas basadas en distancia sin mencionar la *distancia euclídea*, también conocida como distancia pitagórica porque se deriva del Teorema de Pitágoras. Esta métrica ha sido ampliamente utilizada por una gran variedad de comunidades para la resolución de multitud de problemas, pero cuando se trata de variables binarias, se utiliza una modalidad de la *distancia euclídea* denominada *distancia euclídea binaria*, que consiste en obtener la distancia teniendo en cuenta únicamente las ausencias de las características. Partiendo de la Tabla 3.1, la *distancia euclídea binaria* [Anderberg, 1973] se define como:

$$d_{B-EUCLID}(a,b) = \sqrt{y+z} \qquad 3.1$$

Otra medida es la *distancia de Hamming*, también llamada *City Block* o *Manhattan*, que se utiliza en el campo de la teoría de la información, para calcular la efectividad de los códigos de bloque y depende de la diferencia entre una palabra de código y otra [Krause, 1986]. La *distancia de Hamming* se define como:

$$d_{HAMMING}(a,b) = y + z \qquad 3.2$$

La medida de *Sorensen* [Sorensen, 1948], también conocida como distancia de *Bray y Curtis* [Bray y Curtis, 1957] es muy utilizada en estadística aplicada a la ecología y las ciencias ambientales para cuantificar las diferencias en la composición de dos lugares distintos. Aunque habitualmente aparece en la literatura como una medida de distancia, este término no es correcto debido a que no cumple con la ley de la desigualdad triangular, por lo que es más adecuado llamarle *disimilitud*, y se define como:

$$d_{BRAY-CURTIS}(a,b) = \frac{y + z}{2x + y + z} \qquad 3.3$$

La justificación de su uso es más empírica que teórica, aunque puede estar justificada teóricamente como la intersección de dos conjuntos borrosos [Roberts, 1986]. En comparación con la *distancia euclídea*, la distancia de *Bray* y *Curtis* conserva la sensibilidad ante conjuntos de datos más heterogéneos dando menos peso a los valores atípicos.

Medidas basadas en correlación

Las medidas basadas en correlación son aquellas que se aplican a casos en que las variables estadísticas están correlacionadas, es decir, cuando los valores de una de ellas varían sistemáticamente con respecto a los valores de la otra. Un grupo de medidas binarias basadas en correlación son las de la familia χ^2, dentro de las cuáles podemos citar el *coeficiente de correlación de Pearson* [Pearson, 1900]:

$$S_{PEARSON}(a,b) = \frac{n(xw - yz)^2}{(x + y)(x + z)(z + w)(y + w)} \qquad 3.4$$

El *coeficiente de correlación de Pearson* se utiliza generalmente para medir la fuerza y dirección de la dependencia lineal entre dos variables. Su valor se encuentra en el rango de -1 a 1. El valor 1 significa que la relación entre las dos variables puede ser descrita mediante una

ecuación lineal, donde el aumento de una variable implica el aumento de la otra; el valor -1 significa que el aumento de una variable implica la disminución de la otra; y el valor 0 significa que las variables no están correlacionadas.

Otra familia de métricas son las que se basan en el producto escalar, que se puede interpretar como el número de correspondencias o solapamiento entre dos vectores binarios. Entre ellas tenemos el *coeficiente del coseno* [Duda *et al*., 2001] que mide el ángulo entre dos vectores, y las métricas *Ochiai I*, y *Ochiai II* que son variantes del coeficiente del coseno [Deza y Deza, 2006].

$$S_{COS}(a,b) = \frac{x}{\sqrt{(x+y)(x+z)^2}} \quad 3.5$$

$$S_{OCHIAI-I}(a,b) = \frac{x}{\sqrt{(x+y)(x+z)}} \quad 3.6$$

$$S_{OCHIAI-II}(a,b) = \frac{xw}{\sqrt{(x+y)(x+z)(y+w)(z+w)}} \quad 3.7$$

Medidas basadas en no correlación

Las medidas basadas en no correlación son aquellas que se utilizan cuando las variables no están correlacionadas. Dentro de este grupo uno de los primeros y más utilizados índices binarios de similitud es el *coeficiente de Jaccard* [Jaccard, 1901]. Este coeficiente ha sido utilizado fundamentalmente en la biología y la ecología para agrupar diferentes especies, entre otras muchas aplicaciones. El *coeficiente de Jaccard* se define como la razón entre la intersección y la unión de los dos conjuntos de datos, como se muestra en la Ecuación 3.8.

$$S_{JACCARD}(a,b) = \frac{x}{x+y+z} \quad 3.8$$

Otra de las métricas basadas en no correlación es la medida de distancia propuesta por Tanimoto [Tanimoto, 1957] que es una extensión del *coeficiente de Jaccard* y el *coeficiente del coseno*. La métrica de Tanimoto ha sido muy utilizada en la *quimio-informática* para comparar moléculas con atributos binarios y también en los sistemas de recomendación para calcular la frecuencia con que los usuarios siguen determinados temas. La distancia de Tanimoto ha sido definida como:

$$d_{TANIMOTO}(a,b) = \frac{x}{y^2 + z^2 - x} \qquad 3.9$$

Otra medida de similitud que ha sido ampliamente utilizada en diversas áreas como por ejemplo la ecología es el *coeficiente de Dice* [Dice, 1945]. Esta medida, al igual que la distancia de *Bray y Curtis* conserva la sensibilidad en los conjuntos de datos más heterogéneos y otorga menos peso a los valores atípicos. El *coeficiente de Dice* se define como:

$$S_{Dice}(a,b) = \frac{2x}{2x + y + z} \qquad 3.10$$

Una generalización de las métricas de Dice y Tanimoto es la *similitud de Tversky* [Tversky, 1977] que es conocida como una medida de similitud asimétrica porque permite establecer distinto peso a las características no comunes en ambos conjuntos. La medida de *similitud de Tversky* se define como:

$$S_{Tversky}(a,b) = \frac{x}{x + \alpha y + \beta z} \qquad 3.11$$

Analizando esta ecuación podemos observar que en caso de que $\alpha=\beta=1$, esta medida se convierte en el *coeficiente de Jaccard* (Ecuación 3.8), mientras que si $\alpha=\beta=1/2$, estaríamos en presencia del *coeficiente de Dice* (Ecuación 3.10).

Otra métrica, propuesta por Faith [Faith *et al*., 1987], calcula la similitud considerando tanto las correspondencias positivas como las negativas. Es decir, tiene en cuenta las presencias y ausencias mutuas de las características, aunque de manera asimétrica, dándole menos peso a la cantidad de elementos que no contienen ninguna de las características:

$$S_{FAITH}(a,b) = \frac{x + 0.5w}{x + y + z + w} \qquad 3.12$$

3.1.1.2 Contribuciones en el cálculo de la similitud semántica

La inclusión o la exclusión de las correspondencias negativas (w) en las medidas de similitud binarias ha sido discutida en varios trabajos [Goodman y Kruskal, 1963; Sneath y Sokal, 1973; Dunn y Everitt, 1982]. Entre los métodos que incluyen las correspondencias

negativas podemos mencionar a *Faith*, *Ochiai II* y *Pearson*, mientras que entre los que excluyen dichas correspondencias se encuentran *Jaccard*, *Tanimoto* y *Ochiai I*, entre otros. En [Sokal y Sneath, 1963] se plantea que las correspondencias negativas no implican necesariamente ningún tipo de similitud entre dos objetos, debido a que existe un infinito número de atributos que pueden estar ausentes en ellos. Sin embargo en los casos en que los dos estados binarios no sean igualmente importantes, como por ejemplo en los tipos de datos binarios asimétricos, existe la variante de darle distinto peso a las características [Sneath y Sokal, 1973]. Como por lo general las correspondencias positivas son más significativas que las negativas, *Faith* [Faith *et al*., 1987] incluye las correspondencias negativas pero opta por darle la mitad del crédito, otorgando el crédito completo a las correspondencias positivas.

En esta Tesis Doctoral el proceso de cálculo de la similitud semántica se realiza teniendo en cuenta la presencia de los conceptos en documentos recuperados en búsquedas sucesivas en la Web (específicamente en la *Wikipedia*[6]). Como la cantidad total de documentos indexados en la *Wikipedia* es inmensa, una medida de similitud que tuviese en cuenta las correspondencias negativas (la ausencia de ambos conceptos en los documentos) no era una opción. Por este motivo decidimos utilizar una medida basada en no correlación que no tuviese en cuenta este factor y optamos por utilizar el *coeficiente de Jaccard*.

De manera similar a la propuesta en [Pan *et al.*, 2005], para garantizar que la búsqueda en la Web sólo retornara aquellos documentos que fueran relevantes a los conceptos, las consultas se realizan de manera contextualizada. Cada consulta se forma por la combinación de los nombres de todos los términos en el camino de la raíz al nodo actual en la ontología tal y como se describe a continuación.

Sea A el nombre de una entidad de una ontología, llamemos A^+ al conjunto que contiene todos los documentos con correspondencia positiva de A, y A^- al conjunto que contiene todos los documentos con correspondencia negativa de A. Los elementos del conjunto A^+ se obtienen realizando una búsqueda de los documentos que contienen al concepto A y todos sus ancestros en la taxonomía, mientras que los elementos de A^- serán aquellos documentos que contienen los ancestros de A, pero no contienen el concepto A. Para cada pareja de entidades A y B de las dos ontologías se calculan los valores x, y y z, representados en la Tabla 3.1, donde:

[6] Wikipedia: *http://www.wikipedia.org*

x: es el número de elementos de $A^+ \cap B^+$, es decir, el número de documentos que contienen a los conceptos *A* y *B*.

y: es el número de elementos de $A^+ \cap B^-$, es decir, el número de documentos que contienen al concepto *A* pero no contienen al concepto *B*.

z: es el número de elementos de $A^- \cap B^+$, es decir, el número de documentos que contienen al concepto *B* pero no contienen al concepto *A*.

Una vez obtenidos estos valores, se calcula el índice de similitud semántica entre cada pareja de conceptos *A* y *B* a través del *coeficiente de Jaccard*, según la Ecuación 3.8. Por ejemplo, si necesitamos calcular la similitud semántica de los conceptos *Sensing device* y *Sensor* en los fragmentos de ontologías presentados en la Figura 3.1, debemos formar las siguientes consultas de búsqueda:

Consulta *x* ($A^+ \cap B^+$): "*Process*"+"*Observation*"+"*Sensing device*"+"*Capability*"+"*Sensor*".
Consulta *y* ($B^+ \cap A^-$): "*Process*"+"*Observation*"+"*Capability*"+"*Sensor*" -"*Sensing device*".
Consulta *z* ($A^+ \cap B^-$): "*Process*"+"*Observation*"+"*Sensing device*"+"*Capability*"-"*Sensor*".

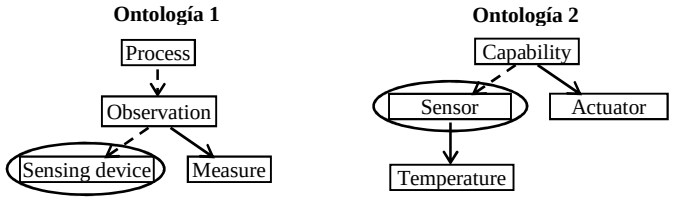


Figura 3.1. Fragmentos de dos ontologías para la similitud semántica

3.1.2 Medidas de Similitud Lingüística

La similitud lingüística constituye el indicador más fuerte del parecido entre dos conceptos, debido a que por lo general los desarrolladores de ontologías dentro de un mismo dominio emplean términos relacionados lingüísticamente para expresar conceptos equivalentes. La tarea de la correspondencia lingüística entre entidades ha sido investigada desde diferentes disciplinas (e.g. estadística, bases de datos, inteligencia artificial…) y cada una de ellas ha formulado el problema de manera diferente proponiendo técnicas muy diversas. Las principales medidas de similitud y distancia encontradas en la literatura se basan

en la correspondencia entre cadenas [Bunke y Sanfeliu, 1990]. Esta sección la hemos dividido en cuatro apartados; en el primero hacemos un estudio de las principales medidas de distancia y similitud entre cadenas encontradas en la literatura; en el segundo se resumen las herramientas léxicas que utilizamos como apoyo para el procesamiento lingüístico (*WordNet* y *UMLS*); en el tercero se explican brevemente las medidas de similitud basadas en *WordNet* encontradas en la literatura; finalmente en el cuarto apartado se explican en detalle nuestras contribuciones en cuanto a las medidas de similitud lingüísticas.

3.1.2.1 Medidas de distancia/similitud de cadenas

Una de las clases más importantes de funciones de distancia son las medidas de edición de cadenas que como esquema de correspondencia general fue propuesto por Monge y Elkan en 1996 [Monge y Elkan, 1996]. Estas medidas han sido aplicadas en la corrección de errores en oraciones con ruido [Oommen, 1987], en tareas de reconocimiento [Marzal y Vidal, 1993], entre otras. En el campo de la inteligencia artificial se han utilizado técnicas de aprendizaje supervisado para obtener los parámetros de las métricas de distancia de edición de cadenas y se han combinado los resultados de diferentes funciones de distancia [Bilenko y Mooney, 2002]. La mayoría de las funciones de distancia de edición de cadenas encontradas actualmente en la literatura se basan en la conocida *Distancia de Levenshtein* [Levenshtein, 1966], que no es más que el coste de la mínima secuencia de operaciones de edición para convertir una cadena en la otra, asignando un único coste a cada una de las operaciones de edición.

Otra métrica de distancia de cadenas conocida es la de *Jaro* [Jaro, 1995], aunque no se basa en el modelo de distancia de edición sino en el número y orden de los caracteres comunes entre las dos cadenas. Dadas dos cadenas *a* y *b*, sean *A* los caracteres de *a* en común con *b* y *B* los caracteres de *b* en común con *a* (en común significa que coinciden los caracteres en las dos cadenas en la misma posición), y sea $TR_{A,B}$ la medida del número de transposiciones de caracteres en *A* en relación a *B*, la similitud de *Jaro* para las cadenas *a* y *b* se define como:

$$Jaro(a,b) = \frac{1}{3}\left(\frac{|A|}{|a|} + \frac{|B|}{|b|} + \frac{|A| - TR_{A,B}}{2|A|}\right) \qquad 3.13$$

Una variante de la métrica de *Jaro* fue propuesta por Winkler [Winkler, 1999], que consiste en una pequeña mejora en el peso de los pares de cadenas con un índice de similitud

bajo que comparten un prefijo común. Si se define el parámetro *P* como la longitud del mayor prefijo común entre *a* y *b* (hasta un máximo de 4 caracteres) y *l* un factor de escala constante cuyo valor en esta aproximación es 0.1, la métrica de *Jaro-Winkler* se define como:

$$JaroWinkler(a,b) = Jaro(a,b) + P \cdot l\left(1 - Jaro(a,b)\right) \qquad 3.14$$

Otro tipo de esquema de correspondencia es el basado en *tokens.* Estos esquemas parten de la descomposición de las cadenas de caracteres en palabras, frases, símbolos u otros elementos léxicos que tienen un significado coherente en algún lenguaje o vocabulario. Estos elementos se denominan *tokens* o componentes léxicos y sirven como entrada para cualquier tipo de procesamiento.

Siguiendo este esquema, una métrica que ha sido ampliamente utilizada en el campo de la recuperación de información es la medida de *similitud del coseno* [Duda *et al*., 2001], que se basa en los términos que tienen las cadenas en común. Sean $A = \{a_0,.., a_n\}$ y $B = \{b_0,.., b_n\}$ los vectores de frecuencia de las palabras que componen las cadenas *a* y *b* respectivamente, el producto escalar (x) entre los vectores *A* y *B* se define como:

$$A \times B = \|A\| \cdot \|B\| \cos\theta \qquad 3.15$$

La métrica de similitud del coseno equivale al coseno del ángulo entre los dos vectores. Si ||*A*|| y ||*B*|| son las normas de los vectores de frecuencia, partiendo de la ecuación 3.15, la similitud del coseno entre las cadenas *a* y *b* se define como:

$$Sim(a,b) = \cos\theta = \frac{A \times B}{\|A\| \cdot \|B\|} = \frac{\sum_{i=1}^{n} (A_i \cdot B_i)}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \cdot \sqrt{\sum_{i=1}^{n} (B_i)^2}} \qquad 3.16$$

También existen las medidas de distancia que combinan esquemas basados en cadenas y esquemas basados en *tokens*, como la propuesta de Monge y Elkan [Monge y Elkan, 1996], que plantean un esquema de correspondencia recursivo para comparar dos cadenas largas *a* y *b*. En primer lugar, las cadenas *a* y *b* se dividen en subcadenas $a = a_1 \ldots a_K$ y $b = b_1 \ldots b_L$ y finalmente la medida de similitud se define como:

$$sim(a,b) = \frac{1}{K}\sum_{i=1}^{K}\max_{j=1}^{L}(sim'(a_i,b_j)), \qquad 3.17$$

donde *sim'* es cualquier otra función de distancia secundaria, como por ejemplo alguna de las mencionadas anteriormente.

3.1.2.2 Herramientas léxicas

En la actualidad existen muchas herramientas léxicas para establecer relaciones lingüísticas entre palabras. Estas herramientas pueden considerarse como especies de diccionarios tanto de carácter general como de carácter específico en determinados dominios. Entre los diccionarios electrónicos más populares se encuentran el meta-tesauro de propósito general *WordNet* y el conjunto de herramientas léxicas *UMLS* del dominio de la medicina. Este último se ha elegido por el elevado número de ontologías médicas disponibles y con las que hemos podido experimentar en este trabajo. Evidentemente para otros dominios sería muy conveniente contar con herramientas especializadas similares. A continuación explicamos brevemente ambas herramientas.

El meta-tesauro WordNet

WordNet [Fellbaum, 1998] es un meta-tesauro electrónico ampliamente utilizado en muchas aplicaciones en diversas áreas, como la recuperación de información y el procesamiento de lenguaje natural. Se basa en teorías psico-lingüísticas para definir el significado de las palabras y modela no sólo asociaciones de significados con palabras, sino también de significados con significados [Ferrer, 2005]. Se centra en el significado de las palabras en lugar de en su construcción, a pesar de considerar también la inflexión morfológica. *WordNet* utiliza tres bases de datos: una para los sustantivos, una para los verbos y la tercera para los adjetivos y adverbios. Los elementos básicos de *WordNet* son los conjuntos de sinónimos, llamados *synsets*. Un *synset* denota un concepto o un significado (sentido) de un grupo de términos y proporciona diferentes relaciones semánticas que difieren dependiendo de la categoría gramatical (sustantivos, verbos, adjetivos o adverbios). Estas relaciones pueden ser de diferentes tipos: sinónimos (igualdad), antónimos (oposición), hiperónimos (superconceptos), hipónimos (subconceptos), merónimos (parte-de) y halónimos (tiene-un). En la Figura 3.2 se muestra un pequeño fragmento de la taxonomía de sustantivos de *WordNet*.

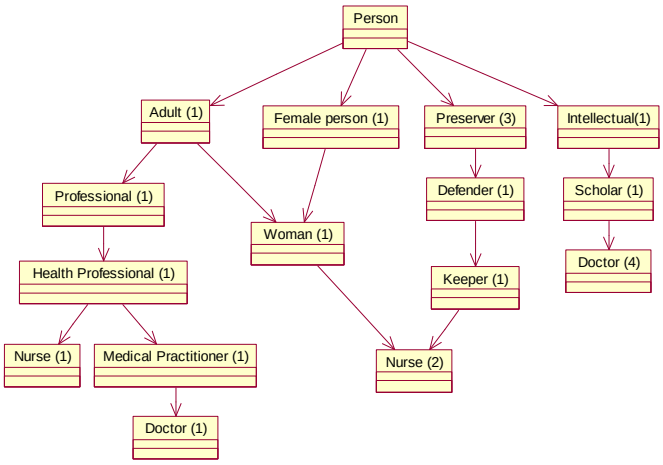


Figura 3.2. Fragmento de la taxonomía de sustantivos de WordNet

Como cada sustantivo puede tener varios sentidos, al lado de cada palabra se muestra entre paréntesis el número del sentido que le corresponde a la definición del sustantivo. Por ejemplo, considerando *Person* como el sustantivo raíz de la taxonomía, si buscamos la relación de hiperónimos del sustantivo *Doctor* nos aparecen 4 sentidos (*senses*), de los cuáles mostramos el 1 y el 4 en el siguiente cuadro.

Sense 1

Doctor, doc, physician, MD, Dr., medico: A licensed medical practitioner; "I felt so bad I went to see my doctor"

=> Medical practitioner, medical man: Someone who practices medicine
=> Health professional, health care provider, caregiver: A person who helps in identifying or preventing or treating illness or disability
=> Professional, professional person: A person engaged in one of the learned professions
=> Adult, grownup: A fully developed person from maturity onward
=> Person, individual, someone, somebody, mortal, soul: A human being; "there was too much for one person to do"

Sense 4

Doctor, Dr.: A person who holds Ph.D. degree (or the equivalent) from an academic institution; "she is a doctor of philosophy in physics"

=> Scholar, scholarly person, bookman, student: A learned person (especially in the humanities); someone who by long study has gained mastery in one or more disciplines
=> Intellectual, intellect: A person who uses the mind creatively
=> Person, individual, someone, somebody, mortal, soul: A human being; "there was too much for one person to do"

En el caso del sustantivo *Nurse* los dos sentidos que aparecen y su relación de hiperónimos en *WordNet* es la siguiente:

Sense 1

Nurse: One skilled in caring for young children or the sick (usually under the supervision of a physician)

=> Health professional, health care provider, caregiver: A person who helps in identifying or preventing or treating illness or disability
=> Professional, professional person: A person engaged in one of the learned professions
=> Adult, grownup: A fully developed person from maturity onward
=> Person, individual, someone, somebody, mortal, soul: A human being; "there was too much for one person to do"

Sense 2

Nanny, nursemaid, nurse: A woman who is the custodian of children

=> Woman, adult female: An adult female person (as opposed to a man); "the woman kept house while the man hunted"
=> Female, female person: A person who belongs to the sex that can have babies
=> Person, individual, someone, somebody, mortal, soul: A human being; "there was too much for one person to do"
=> Adult, grownup: A fully developed person from maturity onward
=> Person, individual, someone, somebody, mortal, soul: A human being; "there was too much for one person to do"
=> Keeper: Someone in charge of other people; "am I my brother's keeper?"
=> Defender, guardian, protector, shielder: A person who cares for persons or property
=> Preserver: Someone who keeps safe from harm or danger
=> Person, individual, someone, somebody, mortal, soul: A human being; "there was too much for one person to do"

El sistema médico UMLS

UMLS[7] (*Unified Medical Language System*) de la librería nacional de medicina de los Estados Unidos (*NLM*[8]) es un conjunto de ficheros y programas que engloban muchos vocabularios relacionados con la biomedicina y la salud para permitir la interoperabilidad entre sistemas de computación específicos de este dominio. Los desarrolladores utilizan el conjunto de bases de conocimientos de *UMLS* y sus herramientas software asociadas para construir y mejorar los sistemas que crean, procesan, recuperan e integran información relacionada con la medicina. Las fuentes de conocimiento son multi-propósito y se utilizan en sistemas que realizan funciones que implican diferentes tipos de información, como registros

[7] UMLS (*Unified Medical Language System*): *http://www.nlm.nih.gov/research/umls/*
[8] NLM (*National Library of Medicine*): *http://www.nlm.nih.gov/*

de pacientes, literatura científica, guías y datos de salud pública. Las herramientas software asociadas ayudan a los desarrolladores en la personalización o el uso de las fuentes de conocimiento de *UMLS* para fines particulares. Las herramientas léxicas trabajan más eficazmente en combinación con las fuentes de conocimiento de *UMLS*, pero también se pueden utilizar de forma independiente. En *UMLS* existen tres fuentes de conocimiento: el *Meta-tesauro*, la *Red Semántica*, y el *Léxico Especialista*. A continuación describimos las tres fuentes de información.

a) *Meta-tesauro*: El meta-tesauro es un vocabulario amplio, polivalente y multilingüe que contiene información sobre conceptos biomédicos y sanitarios, sus diversos nombres y las relaciones entre ellos. Se construye a partir de versiones electrónicas de numerosos tesauros, clasificaciones, conjuntos de códigos y listas de términos utilizados en la atención al paciente, facturación de servicios de salud, estadísticas de salud pública, indexación de literatura biomédica, servicios clínicos, etc. [NLM, 2009]. En el meta-tesauro, todos los vocabularios de origen están disponibles en un formato común, en bases de datos completamente especificadas. Entre los más de 100 vocabularios de origen del meta-tesauro se encuentran: *MeSH* [Lipscomb, 2000], *NCI* [Golbeck *et al*., 2003], *RxNorm* [NLM, 2009] y *SNOMED CT* [Truran *et al*., 2010].

El meta-tesauro está organizado por conceptos o significados. En esencia, se vinculan nombres alternativos y puntos de vista de un mismo concepto y se identifican relaciones útiles entre diferentes conceptos. Cada concepto o significado en el meta-tesauro tiene asociado un identificador de concepto único y permanente (*CUI*) que no tiene ningún significado en sí mismo. En otras palabras, no se puede inferir nada acerca de un concepto con sólo mirar su *CUI*. Sin embargo, el identificador de un concepto nunca cambia, independientemente de los cambios en el tiempo en los nombres que se le atribuyan en el meta-tesauro o en los vocabularios de origen. Todas las entidades de diferentes vocabularios con el mismo *CUI* se consideran sinónimos.

b) *Red Semántica*: La red semántica ofrece una clasificación coherente de todos los conceptos representados en el meta-tesauro y proporciona un conjunto de relaciones útiles entre estos conceptos. Toda la información sobre conceptos específicos se encuentra en el meta-tesauro; la red proporciona información sobre el conjunto de

tipos semánticos básicos, o categorías, que pueden ser asignados a estos conceptos, y define el conjunto de relaciones que pueden existir entre los tipos semánticos. La red semántica que contiene 133 tipos semánticos y 54 relaciones [NLM, 2009] sirve como una autoridad para los tipos semánticos que se asignan a los conceptos del meta-tesauro y define estos tipos, tanto con descripciones textuales como por medio de la información inherente en sus jerarquías.

Los tipos semánticos son los nodos de la red y las relaciones semánticas entre ellos son los enlaces. Hay grandes grupos de tipos semánticos para los organismos, las estructuras anatómicas, funciones biológicas, productos químicos, eventos, objetos físicos y los conceptos o ideas. El alcance actual de los tipos semánticos de *UMLS* es bastante amplio, lo que permite la categorización semántica de una amplia gama de terminología en varios dominios. A todos los conceptos en el meta-tesauro se les asigna al menos un tipo semántico de la red semántica para proporcionar una categorización consistente en un nivel relativamente general representado en la red semántica. En la Figura 3.3 se presenta un fragmento de la red semántica de *UMLS*, específicamente la jerarquía *Biologic function*.

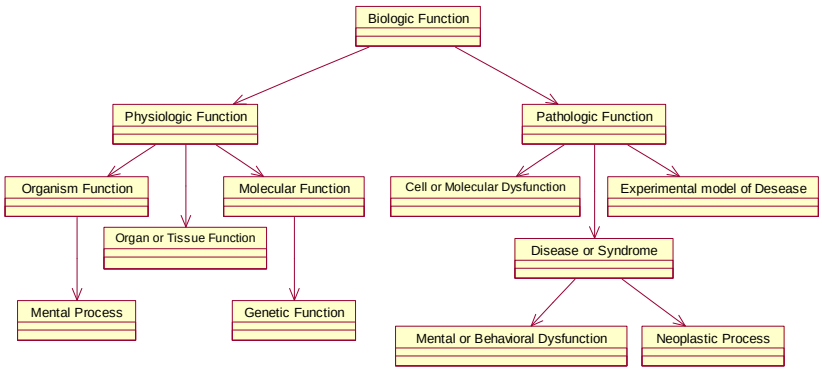


Figura 3.3. Fragmento de la jerarquía Biologic Function de la red semántica de UMLS

c) *Léxico especialista*: El léxico especialista pretende ser un diccionario general en inglés que incluye muchos términos biomédicos. La cobertura incluye tanto palabras que comúnmente se utilizan en inglés como en el vocabulario biomédico [Browne *et al.*, 2000]. La entrada léxica para cada palabra o término registra la información sintáctica, morfológica y ortográfica que necesita el sistema de procesamiento de lenguaje natural especialista.

3.1.2.3 Medidas que utilizan *WordNet* para el mapeo de ontologías

El cálculo de la similitud entre cadenas empleando *WordNet* ha sido ampliamente utilizado en el campo del procesamiento del lenguaje natural y la recuperación de información, y muchos de estos métodos se han aplicado eficazmente en el área del mapeo de ontologías. Los métodos existentes en esta rama enfocados en el mapeo de ontologías se pueden clasificar atendiendo a sus principales características en tres grupos: los métodos basados en arcos, los métodos estadísticos basados en información, y por último los métodos híbridos. A continuación explicamos los principales trabajos encontrados en la literatura en las diferentes categorías.

Métodos basados en arcos

Estos métodos se basan en medir la distancia de la ruta de enlace de los conceptos y la posición de los mismos en la taxonomía de *WordNet*. En otras palabras, estos métodos consisten en calcular el camino más corto de un nodo a otro dentro del grafo de *WordNet*.

Algunos de los métodos basados en arcos encontrados en la literatura sugieren utilizar las profundidades de los conceptos en la taxonomía de *WordNet*. Leacock y Chodorow [Leacock y Chodorow, 1998] sugieren que la relación semántica entre dos conceptos *a* y *b*, puede ser estimada utilizando el número de arcos en el camino más corto entre *a* y *b* y su profundidad en la taxonomía de *WordNet*. En la propuesta de Sussna [Sussna, 1993] se añade a la profundidad un factor específico para computar la relación semántica entre ellos. Este factor se define teniendo en cuenta el tipo de los arcos que conectan a los conceptos y el número de arcos en el camino entre uno y otro en la taxonomía de *WordNet*.

Wu y Palmer proponen un método para calcular la similitud que se basa en los conceptos comunes utilizando el camino dentro de la taxonomía de *WordNet* [Wu y Palmer, 1994]. En este método para calcular la similitud entre los conceptos *a* y *b*, se introduce el concepto *c*, que es el primer superconcepto común entre *a* y *b*. Sean N_{ac} el número de nodos en el camino de *a* a *c*, N_{bc} el número de nodos en el camino de *b* a *c*, y N_{cr} el número de nodos en el camino de *c* a la raíz, la similitud entre *a* y *b* se calcula por:

$$sim(a,b) = \frac{2N_{cr}}{N_{ac} + N_{bc} + 2N_{cr}} \qquad 3.18$$

Métodos estadísticos basados en información

Estos métodos parten de la idea básica de que cuanta más información tengan los conceptos en común, más alta será su similitud. Resnik propuso un método estadístico basado en información [Resnik, 1999] motivado por el problema de encontrar una distancia de enlace uniforme en los métodos basados en arcos. El primer paso es asociar probabilidades a los conceptos de la taxonomía, luego se cuantifica el contenido de la información de cada concepto. Según [Ross, 1976] el contenido de la información de *c* puede ser cuantificado como el negativo del logaritmo de verosimilitud, – *log p* (*c*). Sea *C* el conjunto de conceptos de la taxonomía, se extiende la taxonomía con una función *p*: *C*→[0,1] tal que para cualquier $c \in C$, *p* (*c*) es la probabilidad de encontrar una instancia del concepto *c* en la taxonomía. Esto implica que *p* es monotónicamente no decreciente a medida que se asciende en la jerarquía taxonómica, es decir, si c_1 *es-un* c_2 entonces *p* (c_1) < = *p* (c_2). Si la taxonomía tiene un único nodo superior, entonces esta probabilidad es 1. Finalmente la similitud entre dos conceptos *a* y *b* se define como:

$$sim(a,b) = \max_{c \in S(a,b)} \left[-\log p(c) \right], \qquad 3.19$$

donde *S* (*a*,*b*) es el conjunto de conceptos que contiene a *a* y *b*.

Posteriormente Lin [Lin, 1998] propuso una modificación al método de Reisnik que ha sido utilizada en varios trabajos encaminados al mapeo de ontologías. En este método se define la similitud de dos conceptos como la razón entre la cantidad de información necesaria para alcanzar la igualdad entre ellos y la información necesaria para describirlos completamente. Sea *c* la clase más específica que contiene a *a* y *b*, la similitud de Lin se define como:

$$sim(a,b) = \frac{2 \log p(c)}{\log p(a) + \log p(b)} \qquad 3.20$$

Métodos híbridos

Los métodos híbridos constituyen combinaciones de las técnicas de las categorías anteriores. Uno de los más conocidos es el propuesto por Jiang y Conrath [Jiang y Conrath, 1997], que proponen un método combinado que se deriva del principio de los métodos

basados en arcos añadiendo el contenido de información como un factor de decisión. Al igual que Reisnik se basan en que el contenido de información de un concepto *c* puede ser cuantificado como – *log p* (*c*), donde *p* es la probabilidad de encontrar una instancia del concepto *c* dentro de la jerarquía de *WordNet*. La medida de distancia de Jiang y Conrad entre las entidades *a* y *b* se define como:

$$d(a,b) = 2\log p(c) - (\log p(a) + \log p(b)) \qquad 3.21$$

En [Rodriguez y Egenhofer, 2003] se presenta otra métrica para determinar la similitud entre entidades utilizando *WordNet*, que considera las relaciones de hiperónimos/hipónimos y halónimos/merónimos. Para las entidades *a* y *b*, la similitud de Rodríguez se calcula a través de la suma ponderada de un conjunto de funciones de similitud *S* que tienen en cuenta medidas de intersección y diferencia entre conjuntos. La ecuación 3.22 muestra esta suma ponderada, donde las funciones S_w, S_u y S_n representan las funciones de similitud entre conjuntos de sinónimos, características y vecinos semánticos respectivamente.

$$S(a,b) = w_w \cdot S_w(a,b) + w_u \cdot S_u(a,b) + w_n \cdot S_n(a,b) \qquad 3.22$$

El cálculo de las funciones *S* se basa en el modelo de normalización propuesto por Tversky y se definen como:

$$S(a,b) = \frac{|A \cap B|}{|A \cap B| + \alpha(a,b)|A - B| + (1 - \alpha(a,b))|B - A|}, \qquad 3.23$$

donde *A* y *B* son los conjuntos de descripción de *a* y *b* respectivamente, que incluyen los conjuntos de sinónimos y el resto de las relaciones lingüísticas. La función α define la importancia relativa de las características no comunes, por ejemplo, para las relaciones de especialización, α se expresa en términos de la profundidad de las clases de las entidades. Cuando ambas clases pertenecen a la misma ontología la profundidad equivale a la distancia desde cada clase hasta la superclase más inmediata que las contenga a ambas. En cambio si las entidades pertenecen a ontologías diferentes esta profundidad se calcula de manera distinta pues no existe una superclase común. En este caso se considera que ambas ontologías se enlazan a través de una clase raíz imaginaria, y la profundidad sería la distancia de cada clase a esta raíz. El término α se define de la manera siguiente:

$$\alpha(a,b) = \begin{cases} \dfrac{depth(a)}{depth(a)+depth(b)} & , si\ depth\ (a)\ \leq\ depth\ (b) \\ 1 - \dfrac{depth(a)}{depth(a)+depth(b)} & , si\ depth\ (a)\ >\ depth\ (b) \end{cases} \quad 3.24$$

Una adaptación del método de Rodríguez es la métrica *X-Similarity* [Petrakis et al., 2006], donde se sustituye la ecuación 3.23 por:

$$S(a,b) = \frac{|A \cap B|}{|A \cup B|} \quad 3.25$$

Siguiendo la ecuación 3.25 se calcula un valor de similitud S_s para el caso en que A y B sean *synsets* y otro valor de S_d para el caso en que A y B sean conjuntos de descripción de términos. Sea i el tipo de relación (por ejemplo: *es-un*, *parte-de*), la similitud entre los términos de la vecindad S_n se define para todos los tipos de relaciones como:

$$S_n(a,b) = \max \frac{|A_i \cap B_i|}{|A_i \cup B_i|} \quad 3.26$$

Finalmente, la similitud entre los conceptos *a* y *b* queda definida como:

$$Sim(a,b) = \begin{cases} 1 & , si\ S_s\ (a,b) > 0 \\ \max(S_n(a,b), S_d(a,b)) & , si\ S_s\ (a,b) = 0 \end{cases} \quad 3.27$$

En la métrica de Rodríguez se tiene en cuenta la profundidad de los conceptos en las ontologías. Sin embargo el proceso de mapeo de ontologías no depende de este factor. Una de las principales ventajas de la medida *X-Similarity* de Petrakis sobre la propuesta de Rodríguez es que en *X-Similartity* esto no se tiene en cuenta. En la métrica de Rodríguez se penaliza la existencia de palabras no comunes en la definición de los términos. Sin embargo el número de palabras contenidas en las descripciones de términos puede variar significativamente de una ontología a otra. En *X-Similarity* sólo se tienen en cuenta las uniones e intersecciones, por lo que cuantas más palabras tengan en común las dos definiciones, mayor será la similitud entre los términos y viceversa.

3.1.2.4 Contribuciones en el cálculo de la similitud lingüística

En este trabajo se proponen tres medidas de similitud lingüística: una basada en la relación de sinonimia, otra basada en las palabras derivadas y una tercera medida que hemos llamado factor léxico y se basa en las distancias entre las palabras. Antes de aplicar las medidas para el cálculo de la similitud lingüística realizamos el proceso de normalización o pre-procesamiento de las cadenas, que consiste en estandarizar los términos aplicando técnicas lingüísticas para su mejor reconocimiento. En esta sección explicamos las principales técnicas de normalización de cadenas y las medidas de similitud lingüística propuestas en la tesis.

Normalización o pre-procesamiento

La normalización o pre-procesamiento engloba un conjunto de técnicas procedentes del campo de la minería de datos aplicadas al análisis de textos. Se aplican fundamentalmente a los documentos de texto con el objetivo de trasformar el texto original en una estructura más apropiada para su posterior análisis, identificando rasgos significativos dentro de ellos. Entre las técnicas de pre-procesamiento más utilizadas se encuentran el análisis léxico o *tokenización*, la eliminación de *stop words* y la lematización o *stemming*:

- El análisis léxico o *tokenización* del texto no es más que el proceso de convertir las cadenas en secuencias de componentes léxicos o *tokens* [Yates y Neto, 1999]. Una parte de este proceso es hacer que el sistema no distinga entre mayúsculas y minúsculas, por lo que se convierte toda la cadena a minúscula. En la *tokenización*, por lo general, se diferencian tres tipos de caracteres: los caracteres de palabra, que suelen ser las letras y los números; los caracteres inter-palabra, que suelen ser los separadores como por ejemplo los espacios en blanco, las comas, los puntos; y finalmente el grupo de caracteres especiales, que son aquellos que tienen un significado determinado dependiendo del contexto de las palabras y por lo tanto requieren un tratamiento especial.

- La eliminación de *stop words* consiste en eliminar las palabras vacías, que son aquellas palabras muy frecuentes en los textos y que carecen de significado como por ejemplo las preposiciones, artículos y conjunciones, así como algunas formas verbales, adverbios y adjetivos en determinados idiomas [Yates y Neto, 1999]. Dada su poca utilidad, estas palabras son desechadas, lo que permite un

considerable ahorro de recursos, ya que si bien estas palabras representan una parte ínfima del vocabulario, suponen, en cambio, una reducción muy importante del número de términos a procesar.

- La lematización o *stemming* es una técnica muy utilizada en el área de la recuperación de información y consiste en la reducción de una palabra a su raíz morfológica [Yates y Neto, 1999]. Frecuentemente, la palabra especificada por el usuario en la consulta no aparece exactamente en un documento pero sí alguna variante gramatical de la misma como plural, gerundios, conjugaciones verbales, etc. Este problema puede resolverse con la sustitución de las palabras por su raíz (*stem*). Un *stem* es la porción de una palabra que resulta de la eliminación de sus afijos (prefijos y sufijos). Un ejemplo podría ser la palabra "*connect*" que es el *stem* de "*connected*", "*connection*", "*connections*". Los *stems* permiten reducir variantes de la misma raíz gramatical a un concepto común y ampliar la definición de la consulta con las variantes morfológicas de los términos usados, mejorando así los resultados de las búsquedas. El algoritmo clásico de *stemming* por excelencia para el idioma inglés es el algoritmo de Porter [Porter, 1980].

Medidas de similitud lingüística por sinonimia y derivación propuestas

En esta tesis proponemos medidas de similitud lingüísticas que hacen uso de dos herramientas léxicas externas. Estas son *WordNet* para el caso de ontologías generales, y *UMLS* para ontologías especializadas en el dominio de la medicina. Las primeras medidas de similitud lingüística propuestas tienen en cuenta las relaciones de sinonimia y derivación entre las palabras.

Nuestra propuesta para el cálculo de la similitud tiene algunos elementos en común con la métrica de Rodríguez [Rodriguez y Egenhofer, 2003] y con X-Similarity [Petrakis *et al.*, 2006], como por ejemplo la elección del módulo de la intersección entre los *synsets* de los conceptos en el numerador, pero a diferencia de ellos, no se tienen en cuenta los elementos no comunes entre los conjuntos.

Cuando se trata de alinear ontologías generales partimos de los *synsets* de *WordNet*, y utilizamos las relaciones de sinonimia y derivación de los términos. Hemos elegido la sinonimia y la derivación por ser las relaciones lingüísticas más frecuentes en los nombres de

las entidades de las ontologías diferentes dentro de un mismo dominio. En el caso de ontologías de dominio muy específico, como por ejemplo las ontologías médicas, *WordNet* no constituye una buena elección como herramienta léxica debido a que es un tesauro muy general y muchos de los términos especializados no aparecen en sus bases de datos. Para obtener mejores resultados en la similitud lingüística en ontologías médicas utilizamos las herramientas de *UMLS* en lugar de *WordNet*. En el meta-tesauro de *UMLS* se considera que dos términos de diferentes vocabularios son sinónimos si tienen el mismo identificador de concepto (*CUI*).

Dados los conceptos *a* y *b* de dos ontologías diferentes, la similitud lingüística por sinonimia y la similitud lingüística por derivación se calculan de manera similar siguiendo los pasos que se explican a continuación:

1. *Normalización*: El paso inicial consiste en pre-procesar las cadenas que forman los conceptos para obtener mejores resultados. Las técnicas de normalización que utilizamos en este paso son la *tokenización* y la eliminación de *stop words*.

2. *Relación*: El segundo paso consiste en obtener las palabras relacionadas con los conceptos. Como hemos elegido las relaciones de sinonimia y derivación, se buscan los conjuntos de sinónimos y de palabras derivadas de las palabras que forman los conceptos, utilizando los *synsets* de *WordNet* y los términos de *UMLS*.

3. *Lematización*: El tercer paso es la lematización y consiste en obtener las raíces morfológicas de las palabras, por lo que a cada una de las palabras de los conjuntos de sinónimos y palabras derivadas obtenidos en el paso anterior se le aplica el algoritmo de *Stemming de Porter* [Porter, 1980].

4. *Similitud*: El paso final es el cálculo de la similitud. Si llamamos *A* y *B* a los conjuntos de raíces de los sinónimos o de las palabras derivadas de cada concepto obtenidos en el paso anterior, la similitud lingüística por sinonimia y por derivación entre los conceptos se calcula según la siguiente ecuación:

$$S(a,b) = \min\left[\frac{|A \cap B|}{|A|}, \frac{|A \cap B|}{|B|}\right], \qquad 3.28$$

donde $|A \cap B|$ es la cardinalidad de la intersección entre los conjuntos *A* y *B*, $|A|$ es la cardinalidad del conjunto *A*, y $|B|$ es la cardinalidad del conjunto *B*. En otras palabras, la primera razón nos indica la fracción de solapamiento del conjunto *A* respecto al conjunto *B*, y la segunda razón nos indica la fracción de solapamiento del conjunto *B* respecto al conjunto *A*. Como ambos conjuntos no suelen tener necesariamente la misma cantidad de elementos, hemos elegido el valor mínimo entre sus grados de solapamiento como indicador de su similitud. En cada conjunto se incluye el propio concepto para asegurar que no quede vacío.

El Factor Léxico propuesto

El *Factor léxico* propuesto en este trabajo consiste en una media ponderada de dos medidas de similitud. La primera de las medidas de similitud se basa en la distancia de Levenshtein entre las palabras que forman los conceptos y la segunda se basa en la distancia entre estos dentro de la taxonomía de *WordNet*, o *UMLS*. El factor léxico de similitud se define de la siguiente manera:

$$Lex\_factor(a,b) = \alpha \cdot Sim_{Lev}(a,b) + \beta \cdot Sim_{WordNet}(a,b), \qquad 3.29$$

donde α y β son los factores de ponderación establecidos para la similitud basada en la distancia de Levenshtein y la similitud basada en la distancia de *WordNet* respectivamente. Los factores de ponderación α y β se han ajustado de manera heurística a través de experimentación.

Como se explica anteriormente la distancia de Levenshtein no es más que el número de operaciones de edición que se necesitan para convertir una cadena en la otra, por lo que su resultado es un número entero cuyo valor máximo es igual a la longitud de la cadena más larga. Teniendo en cuenta esto, la distancia de Levenshtein calculada es normalizada dividiendo su valor entre la longitud de la cadena más larga para convertir este valor entero en un número real entre 0 y 1. Sea $dist_{Lev}(a,b)$ la distancia de Levenshtein normalizada, la similitud basada en la distancia de Levenshtein, $Sim_{Lev}(a,b)$ se calcula a través de la siguiente ecuación:

$$Sim_{Lev}(a,b) = 1 - dist_{Lev}(a,b) \qquad 3.30$$

La distancia de Levenshtein a veces resulta engañosa, porque en inglés existen algunas palabras que aunque requieren pocos pasos de edición para transformarse la una en la otra, sus significados son completamente distintos. Como ejemplo podemos mencionar el caso de *fact* y *fat*, *beard* y *bear*, *beer* y *beef*, *belt* y *bel*, entre otras. Por este motivo hemos decidido darle menor peso a la similitud basada en la distancia de Levenshtein.

El hecho de que los conceptos no se encuentren aislados, sino que formen parte de una ontología con su propia taxonomía, puede sugerir que se otorgue más peso a las relaciones de los conceptos dentro de la estructura de la propia ontología que al lugar que ocupan dentro de la taxonomía del meta-tesauro léxico (*WordNet* o *UMLS*). Sin embargo, la explotación de la jerarquía del meta-tesauro léxico aporta un valor añadido relacionando lingüísticamente los conceptos, independientemente del lugar que ocupen dentro de las ontologías. Como las palabras en *WordNet* pueden tener más de un sentido, sean $v = \{v_1,..,v_n\}$ y $w = \{w_1,..,w_m\}$ los conjuntos de sentidos de los conceptos *a* y *b* dentro de *WordNet*, la similitud entre *a* y *b* sería la similitud máxima entre todos los sentidos de *a* y todos los sentidos de *b*:

$$Sim(a,b) = \max_{\substack{i=1,..,n \\ j=1,..,m}} (S(v_i, w_j)), \qquad 3.31$$

donde $S(v_i,w_j)$ se calcula por el conocido método de Wu y Palmer [Wu y Palmer, 1994], que se basa en las distancias de los conceptos hasta el primer superconcepto común y entre el superconcepto común y la raíz:

$$Sim(v,w) = \frac{2N_{cr}}{N_{vc} + N_{wc} + 2N_{cr}}, \qquad 3.32$$

donde *c* es el primer superconcepto en común entre *v* y *w*, N_{vc} el número de nodos en el camino de *v* a *c*, N_{wc} el número de nodos en el camino de *w* a *c*, y N_{cr} el número de nodos en el camino de *c* a la raíz de *WordNet*.

La similitud basada en *WordNet* es un indicador más realista de la correspondencia entre conceptos que la distancia de Levenshtein, por lo que hemos decidido otorgar un factor de ponderación más elevado. Después de realizar experimentos con varias ontologías establecemos de manera heurística los factores de ponderación $\alpha = 0.25$ y $\beta = 0.75$.

Ejemplo del cálculo de las diferentes medidas de similitud lingüística propuestas

En la Tabla 3.2 se muestran los resultados de la similitud por sinonimia para los conceptos: *a* = "*Sensing device*" y *b* = "*Sensor*", de los fragmentos de ontologías de la Figura 3.1. Primero calculamos los conjuntos de sinónimos de ambos conceptos, que son: *A* = {*sensing*, *detection*, *perception*, *sense*, *device*, *artifact*} y *B* = {*sensor*, *detector*, *sensing element*}. Después de aplicar el algoritmo de *stemming* los conjuntos quedan de la siguiente manera: *A* = {*sens*, *detect*, *percept*, *devic*, *artifact*} y *B* = {*sens*, *detect*, *element*}. Finalmente según la ecuación 3.28 la similitud lingüística por sinonimia sería 0.4.

Tabla 3.2. Similitud por sinonimia entre los conceptos "Sensing device" y "Sensor".

		Conjuntos de sinónimos	**Después del stemming..**	**Cardinalidad**
a	*Sensing device*	A = {*sensing, detection, perception, sense, device, artifact*}	A = {*sens, detect, percept, devic, artifact*}	|A|=5
b	*Sensor*	B = {*sensor, detector, sensing element*}	B = {*sens, detect, element*}	|B|=3
			A∩B={*sens, detect*}	|A∩B|=2
Sim(a,b)	0.4			

De manera similar se calcula la similitud por derivación, cuyos resultados se muestran en la Tabla 3.3. En este caso los conjuntos de palabras derivadas de los conceptos serían: *A* = {*sensing device*, *sense*} y *B* = {*sensor*, *sense*}. Después de aplicar el *stemming*, los conjuntos quedan de la siguiente manera: *A* = {*sens*, *devic*} y *B* = {*sens*}. Finalmente la similitud lingüística por derivación sería 0.5.

Tabla 3.3. Similitud por derivación entre los conceptos "Sensing device" y "Sensor".

		Conjuntos de palabras derivadas	**Después del stemming..**	**Cardinalidad**
a	*Sensing device*	A = {*sensing device, sense*}	A = {*sens, devic*}	|A|=2
b	*Sensor*	B = {*sensor, sense*}	B = {*sens*}	|B|=1
			A∩B={*sens*}	|A∩B|=1
Sim(a,b)	0.5			

Para obtener el factor léxico entre los conceptos "*Sensing device*" y "*Sensor*", el primer paso es calcular la distancia de Levenshtein, que es 10 (0.71 si normalizamos). Según la Ecuación 3.30, la similitud basada en la distancia de Levenshtein es 0.29.

Para explicar el cálculo de la similitud basada en *WordNet* para estos dos conceptos, nos apoyaremos en la Figura 3.4, donde se muestra un fragmento de la taxonomía de *WordNet* para los diferentes sentidos de estos dos conceptos. En el caso del primer concepto (*Sensing device*), este no aparece íntegro como *synset* en *WordNet*, por lo que vamos a dividir el cálculo de la similitud en dos casos, considerando cada palabra como un concepto independiente.

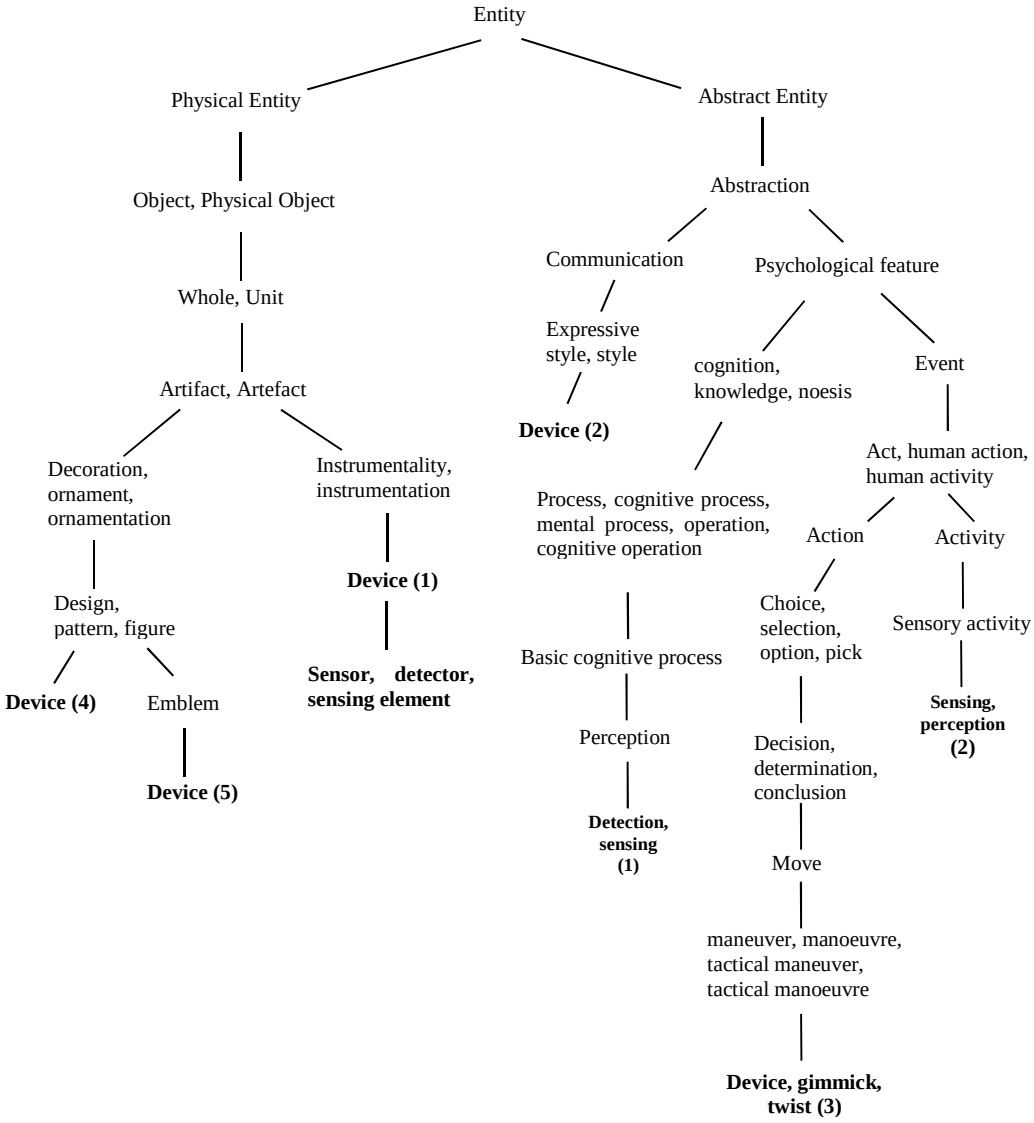


Figura 3.4. Fragmento de la taxonomía de WordNet

En el primer caso calculamos la similitud entre *sensing* y *sensor*, en el segundo calculamos la similitud entre *device* y *sensor*, y finalmente nos quedaremos con el promedio entre los dos valores. En el cálculo de la similitud del primer caso, la palabra *sensing* tiene dos sentidos, (como se observa en la Figura 3.4) mientras que la palabra *sensor* tiene uno solo. En este caso calculamos las similitudes entre *sensor* y los dos sentidos de *sensing* y nos quedamos con el valor máximo (Ecuación 3.31). Como se puede observar en la figura y siguiendo la ecuación 3.32, los resultados para ambas similitudes son los mismos:

$$Sim_{WordNet}(sensing_1, sensor) = \frac{2}{7+8+2} = 0.12$$

$$Sim_{WordNet}(sensing_2, sensor) = \frac{2}{7+8+2} = 0.12$$

Por lo tanto, según la Ecuación 3.31, la similitud de *WordNet* entre *sensing* y *sensor* es 0.12.

Para el segundo caso procedemos de la misma manera, considerando que la palabra *device* tiene 5 sentidos. Los valores de similitud entre la palabra *sensor* y los 5 sentidos de *device* son:

$$Sim_{WordNet}(device_1, sensor) = \frac{14}{0+1+14} = 0.93$$

$$Sim_{WordNet}(device_2, sensor) = \frac{2}{5+7+2} = 0.14$$

$$Sim_{WordNet}(device_3, sensor) = \frac{2}{10+7+2} = 0.11$$

$$Sim_{WordNet}(device_4, sensor) = \frac{10}{3+3+10} = 0.625$$

$$Sim_{WordNet}(device_5, sensor) = \frac{10}{4+3+10} = 0.59$$

Siguiendo la Ecuación 3.31, la similitud de *WordNet* entre *device* y *sensor* es 0.93. Promediando las similitudes basadas en *WordNet* de los dos casos (*sensing-sensor*), y (*device-sensor*), la similitud final entre los conceptos *sensing device* y *sensor* sería 0.525.

Finalmente el factor léxico entre los dos conceptos combinando la similitud basada en la distancia de Levenshtein y la similitud basada en la distancia dentro de la taxonomía de *WordNet* a través de la ecuación 3.29 es:

$$Lex\_Factor(Sensing\ device, Sensor) = 0.47$$

Una vez obtenidos los valores de las tres similitudes lingüísticas (sinonimia, derivación y factor léxico) para cada pareja de conceptos, calculamos su similitud lingüística final a través de un sistema basado en reglas difusas, que detallaremos en el siguiente capítulo.

3.2 Contribuciones en las Medidas de Similitud Estructural

Las técnicas de correspondencia de nivel estructural son aquellas que funcionan teniendo en cuenta las relaciones de los elementos dentro de las ontologías. En este trabajo proponemos dos tipos de similitud desde el punto de vista estructural: la primera utilizando la estructura relacional de los conceptos en las ontologías, concretamente la jerarquía taxonómica, y la segunda utilizando la información de la estructura interna de los conceptos, que incluye sus propiedades, tipos y cardinalidad.

3.2.1 Similitud Estructural Relacional

Para calcular la similitud estructural relacional nos basamos en la jerarquía taxonómica de los conceptos en las ontologías, donde tenemos en cuenta las similitudes de los hijos, padres y hermanos de los conceptos en las taxonomías, además de la similitud basada en sus nombres. Para ello partimos de la idea de que si dos conceptos son similares, y sus hijos, padres y hermanos también lo son, entonces es muy probable que estemos en presencia del mismo concepto o conceptos equivalentes. La similitud final entre las clases se define como una combinación entre la similitud de sus nombres y el valor de similitud que aportan sus hijos, padres y hermanos en la taxonomía. Esta combinación se realiza a través del sistema basado en reglas difusas que explicaremos en el siguiente capítulo.

Dados los conceptos *A* y *B* de dos ontologías diferentes, sean *n* el número de padres del concepto *A* y *m* el número de padres del concepto *B*, y sean A_i y B_j el *i-esimo* y *j-esimo* padre de los conceptos *A* y *B* respectivamente, llamamos similitud jerárquica asociada a los padres al valor de similitud que aportan los padres a la similitud total de los conceptos y se calcula

promediando los máximos de los valores de similitud entre los padres de *A* respecto a los de *B*. Esto se muestra en la Ecuación 3.33. Las similitudes jerárquicas asociadas a los hijos y a los hermanos se calculan de forma análoga. En la Figura 3.5 se muestra como calcular los tres tipos de similitud jerárquica para los conceptos *A* y *B*.

$$Sim(A,B) = \frac{1}{n}\sum_{i=1}^{n} \max\left\{Sim(A_i, B_j)\right\}_{j=1}^{m} \qquad 3.33$$

Por ejemplo, si queremos obtener la similitud estructural relacional entre los conceptos "*Sensing device*" y "*Sensor*" de los fragmentos de ontologías mostrados en la Figura 3.6, los valores de similitud jerárquica se calcularían utilizando la Ecuación 3.33 de la manera siguiente:

$$S_{Padres}(Sensing\ device, Sensor) = S(Observation, Capability)$$

$$S_{Hermanos}(Sensing\ device, Sensor) = \frac{\left[S(Measure, Actuator) + S(Value, Actuator)\right]}{2}$$

$$S_{Hijos}(Sensing\ device, Sensor) = S(Thermometer, Temperature)$$

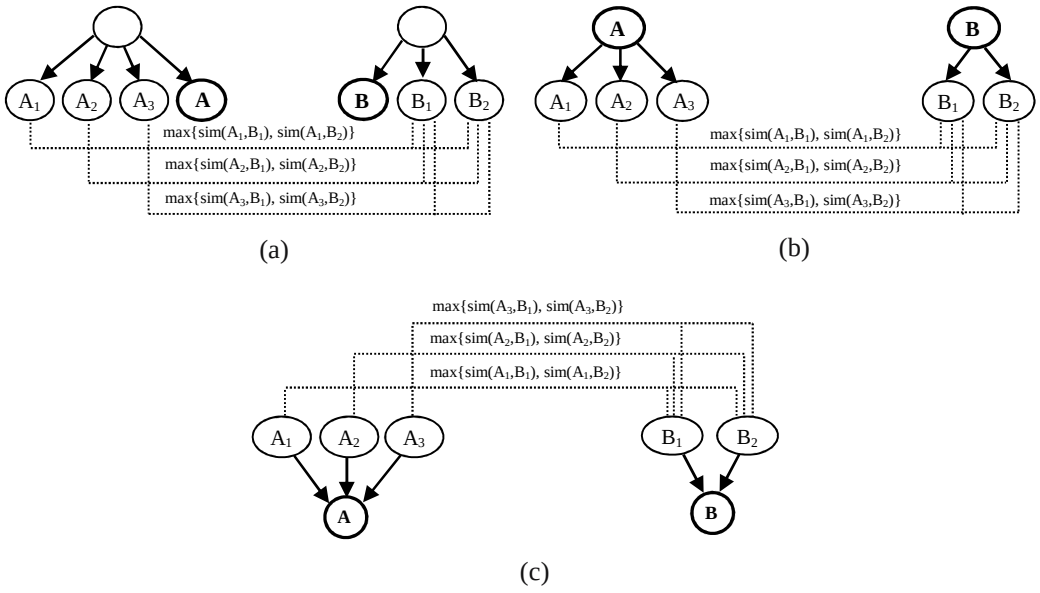


Figura 3.5. Similitud jerárquica (a) *Similitud jerárquica de los hermanos,* (b) *Similitud jerárquica de los hijos y* (c) *Similitud jerárquica de los padres.*

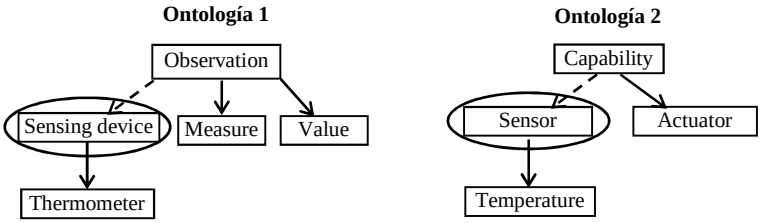


Figura 3.6. Fragmentos de dos ontologías

3.2.2 Similitud Estructural Interna

La similitud estructural interna se basa en la estructura interna de los conceptos, donde influyen, además de sus nombres y relaciones, otros factores, como la similitud de sus propiedades y la cardinalidad. De manera similar al cálculo de la similitud estructural relacional aquí partimos de la idea de que si dos conceptos son similares, tienen el mismo número de propiedades y a su vez estas propiedades tienen un alto grado de similitud entre sí, entonces es muy probable que nos encontremos ante conceptos equivalentes y reforzamos el valor de su similitud. Por el contrario, si los conceptos tienen un alto grado de similitud, pero no tienen el mismo número de propiedades o sus propiedades no son similares, entonces puede que no sean equivalentes y por lo tanto debilitamos el valor de su similitud.

Dados los conceptos *A* y *B* de dos ontologías diferentes, sean *n* el número de propiedades del concepto *A* y *m* el número de propiedades del concepto *B* y sean P_i y Q_j la *i-esima* y *j-esima* propiedad de los conceptos *A* y *B* respectivamente, llamamos similitud interna de las propiedades al valor de similitud que aportan las propiedades a la similitud final entre los conceptos y lo definimos de la manera siguiente:

$$Sim(A,B) = \frac{1}{n}\sum_{i=1}^{n} \max\left\{Sim(P_i,Q_j)\right\}_{j=1}^{m} \qquad 3.34$$

Por ejemplo, si queremos calcular la similitud interna que aportan las propiedades de los conceptos "*Sensing device*" y "*Sensor*" de los fragmentos de ontologías mostrados en la Figura 3.6, suponiendo que las propiedades del concepto "*Sensing device*" son: (`Type: string, serie: integer, parameter: string`) y las propiedades del concepto "*Sensor*" son: (`sensor_type: string, serie: string, measure: string`), utilizando la Ecuación 3.34, la similitud interna de las propiedades se calcularía como sigue:

$$S_{Int\_Propiedades}(Sensing\ device, Sensor) = \frac{S(Type,\ Sensor\_Type) + S(Serie,\ Serie) + S(Parameter,\ Measure)}{3}$$

3.2.2.1 Similitud entre propiedades

Para calcular la similitud entre las propiedades debemos tener en cuenta que en las ontologías existen dos tipos de propiedades:

Propiedades Objeto: Tal y como indica su nombre son propiedades que a su vez son objetos o lo que es lo mismo, propiedades de clases que son instancias de otras clases.

Propiedades Dato: Son propiedades atómicas, es decir, propiedades cuyo valor es un dato concreto y no un objeto.

Las características fundamentales de las propiedades son el nombre, el dominio, que no es más que el concepto o conjunto de conceptos al que pertenece la propiedad y el rango, que es su tipo. En el caso de las *propiedades objeto* el rango es la clase de la que dicha propiedad es una instancia, mientras que en el caso de las *propiedades dato*, el rango es directamente un tipo de dato *XML*. En nuestro trabajo la similitud entre dos propiedades consiste en la integración de tres tipos de similitud:

- *Similitud lingüística*: Es la similitud lingüística entre los nombres de las propiedades, calculada tal y como se detalla en la Sección 3.1.2.

- *Similitud de Dominio*: Es la similitud que aporta el dominio, es decir, la similitud entre las clases a las que pertenecen las propiedades. Como en una ontología una misma propiedad puede pertenecer a varias clases, estamos ante un caso similar al del cálculo de la similitud jerárquica de los padres, la similitud del dominio se calcula de forma similar a la jerárquica promediando los máximos de las similitudes entre cada clase a la que pertenece una propiedad con las clases a las que pertenece la otra.

  Para el ejemplo de la Figura 3.7, sean CP_i, la *i-ésima* clase padre de la propiedad *P*, CQ_j, la *j-ésima* clase padre de la propiedad *Q*, sustituyendo en la Ecuación 3.34, la similitud de dominio de las propiedades *P* y *Q* sería:

$$S_{DOMINIO}(P,Q) = \frac{1}{3}\sum_{i=1}^{3} \max\left\{Sim(CP_i, CQ_j)\right\}_{j=1}^{2}$$

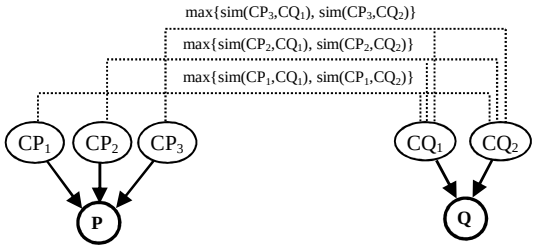


Figura 3.7. Similitud de dominio de las propiedades

- *Similitud de Rango*: Es la similitud entre los tipos de las propiedades, y su valor sólo puede ser 0 o 1. En caso de ser propiedades objeto, para esta similitud se utiliza directamente la similitud entre las clases de las que dichos objetos son instancias, dándole valor 1 si pertenecen a la misma clase y 0 si no es así. Si son propiedades dato, para calcular esta similitud hay que tener en cuenta la compatibilidad entre los tipos de datos *XML*. Si ambos tipos de datos son primitivos, la correspondencia entre ellos tiene que ser exacta para que su similitud sea 1, en caso contrario sería 0.

  En la Tabla 3.4 pueden consultarse los tipos de datos *XML* primitivos. Si al menos uno de los tipos de las propiedades es derivado, se busca su tipo primitivo raíz y el problema se reduce a comparar dos tipos primitivos. En este trabajo consideramos que si dos tipos de datos se derivan del mismo tipo primitivo, entonces son compatibles. Por ejemplo, si una propiedad es de tipo `Name` y la otra es de tipo `Token`, como ambas se derivan del tipo primitivo `String`, las consideramos compatibles. Los tipos *XML* derivados pueden consultarse en la Tabla 3.5.

  La integración de estas tres variables para calcular la similitud final entre las propiedades se realiza a través del sistema basado en reglas difusas que explicaremos en el siguiente capítulo.

Tabla 3.4. Tipos de datos XML primitivos

Tipo de dato	**Descripción**
string	Representa cadenas de caracteres.
boolean	Representa valores booleanos, que son true o false.
decimal	Representa números de precisión arbitraria.
float	Representa números de punto flotante de 32 bits de precisión simple.
doble	Representa números de punto flotante de 64 bits de doble precisión.
duration	Representa una duración de tiempo.
dateTime	Representa una instancia específica de tiempo.
time	Representa una instancia de tiempo que se repite cada día.
date	Representa una fecha de calendario.
gYearMonth	Representa un mes gregoriano específico de un año gregoriano específico. Conjunto de instancias no periódicas de un mes de duración.
gYear	Representa un año gregoriano. Conjunto de instancias no periódicas de un año de duración.
gMonthDay	Representa una fecha gregoriana determinada que se repite, específicamente un día del año.
gDay	Representa un día gregoriano que se repite, específicamente un día del mes.
gMonth	Representa un mes gregoriano que se repite cada año.
hexBinary	Representa datos binarios arbitrarios codificados en hexadecimal.
base64Binary	Representa datos binarios arbitrarios codificados en Base64.
anyURI	Representa un identificador URI como lo define RFC 2396.
QName	Representa un nombre completo, que se compone de un prefijo y un nombre local separados por un signo de dos puntos.
NOTATION	Representa un tipo de atributo NOTATION.

Tabla 3.5. Tipos de datos XML derivados

Tipo de dato	**Descripción**
normalizedString	Representa cadenas normalizadas de espacios en blanco. Se deriva de string.
token	Representa cadenas convertidas en símbolos. Se deriva de normalizedString.
language	Representa identificadores de lenguaje natural. Se deriva de token.
IDREFS	Representa el tipo de atributo IDREFS. Se deriva de IDREF.
ENTITIES	Representa el tipo de atributo ENTITIES. Se deriva de ENTITY.
NMTOKEN	Representa el tipo de atributo NMTOKEN. Se deriva de token.
NMTOKENS	Representa el tipo de atributo NMTOKENS. Se deriva de NMTOKEN.
Name	Representa nombres en XML. Se deriva de token.
NCName	Representa nombres sin el signo de dos puntos. Se deriva de Name.

ID	Representa un ID. Se deriva de NCName.
IDREF	Representa una referencia a un elemento con un atributo ID. Se deriva de NCName.
ENTITY	Representa el tipo de atributo ENTITY.
integer	Representa una secuencia de dígitos decimales. Se deriva de decimal.
nonPositiveInteger	Representa un número entero menor o igual que cero. Se deriva de integer.
negativeInteger	Representa un número entero menor que cero. Se deriva de nonPositiveInteger.
long	Se deriva de integer.
int	Se deriva de long.
short	Se deriva de int.
byte	Se deriva de short.
nonNegativeInteger	Representa un número entero mayor o igual que cero. Se deriva de integer.
unsignedLong	Se deriva de nonNegativeInteger.
unsignedInt	Se deriva de unsignedLong.
unsignedShort	Se deriva de unsignedInt.
unsignedByte	Se deriva de unsignedShort.
positiveInteger	Se deriva de nonNegativeInteger.

3.3 Resumen y consideraciones finales

En este capítulo hemos abordado el tema de las medidas de similitud entre las entidades de ontologías diferentes. Presentamos medidas de similitud teniendo en cuenta dos elementos fundamentales de las ontologías: la terminología y la estructura. La similitud terminológica se basa únicamente en la información que aportan los nombres de las entidades, que puede ser tanto lingüística como semántica.

En este trabajo proponemos medidas para el cálculo de la similitud lingüística que tienen en cuenta las relaciones de sinonimia y derivación entre los conceptos así como las distancias entre las palabras que los componen. Se procesan ontologías generales y especializadas en medicina, con ayuda de las herramientas léxicas *WordNet* y *UMLS* respectivamente. Las medidas de similitud léxicas consisten en cuantificar el grado de solapamiento existente entre las listas de sinónimos y de palabras derivadas de los conceptos a través de la identificación de términos comunes, así como las distancias entre las palabras que componen los términos dentro de los directorios léxicos.

También se propone un método para el cálculo de la similitud semántica utilizando la información del contexto de los conceptos en las ontologías, que consiste en aplicar el *coeficiente de Jaccard* a los resultados de búsquedas sucesivas de documentos en la *Wikipedia*. Para cada pareja de conceptos se realizan tres búsquedas en la *Wikipedia*: la primera para obtener la cantidad de documentos en los que se encuentran los dos conceptos y las dos búsquedas restantes para obtener la cantidad de documentos en los que se encuentra uno de los conceptos y no se encuentra el otro. Para asegurarnos de obtener solamente documentos relevantes introducimos información de contexto que consiste en incluir en la cadena de búsqueda los nombres de todos los ancestros de los conceptos en las ontologías.

Para cada pareja de conceptos de ontologías diferentes la medida de similitud terminológica final se obtiene combinando la similitud lingüística y la semántica a través de un sistema basado en reglas difusas que explicaremos en detalle en el siguiente capítulo.

En cuanto a la similitud estructural, tenemos en cuenta dos factores: la estructura relacional que se enfoca en las relaciones entre las entidades y la estructura interna que tiene que ver con la composición de las mismas. Desde el punto de vista relacional, en este trabajo proponemos medidas de similitud jerárquicas utilizando las relaciones taxonómicas entre los conceptos. Estas medidas consisten en calcular el grado de similitud entre los padres, hermanos y descendientes de los conceptos en las ontologías, para luego determinar su influencia en la similitud final entre los conceptos. Partimos de la idea de que si dos conceptos tienen un alto grado de similitud, y sus descendientes, padres o hermanos también son similares, es probable que estos conceptos sean equivalentes. En el caso de la similitud estructural interna calculamos la similitud entre las propiedades de los conceptos teniendo en cuenta su información lingüística, sus tipos y la similitud de las clases a las que pertenecen. Estas medidas se utilizan para reforzar o debilitar la similitud entre los conceptos.

Todas las medidas de similitud detalladas en este capítulo se combinan en las diferentes capas de un sistema basado en reglas difusas que se explica en el siguiente capítulo.

CAPÍTULO 4. SISTEMA BASADO EN REGLAS DIFUSAS

A medida que aumenta la complejidad, las declaraciones precisas pierden relevancia y las declaraciones relevantes pierden precisión.

Lotfi Zadeh

En esta tesis proponemos un sistema multicapa basado en reglas difusas para la alineación de ontologías, que utiliza aprendizaje genético de la base de reglas. En este capítulo explicamos en detalle el sistema, comenzando con un breve repaso de los conceptos básicos de la lógica difusa, los sistemas basados en reglas difusas y el uso de algoritmos genéticos para el aprendizaje de reglas. Posteriormente detallamos cada una de las capas del sistema basado en reglas difusas propuesto y finalmente explicamos el algoritmo genético utilizado para el aprendizaje de las bases de reglas del mismo.

4.1 Conceptos básicos de Lógica Difusa.

La lógica difusa es un campo de investigación muy importante en la actualidad tanto por sus aplicaciones matemáticas y teóricas como por sus aplicaciones prácticas. Esta disciplina fue investigada por primera vez a mediados de los años 60 en la Universidad de Berkeley (California) por Lotfi A. Zadeh, quien introdujo los conceptos de conjuntos difusos basándose en la idea de que el pensamiento humano se constituye de etiquetas lingüísticas y no de números [Zadeh, 1965].

La lógica difusa permite representar el conocimiento común en un lenguaje matemático a través de la teoría de conjuntos difusos y funciones características asociadas a ellos, permitiendo el manejo de datos numéricos y lingüísticos a la vez. Por ejemplo, si queremos clasificar a un grupo de personas según su estatura, la lógica difusa puede realizar una representación del conocimiento que maneje las etiquetas lingüísticas: "*Alta*" y "*Baja*" asociadas a conjuntos difusos con funciones de pertenencia diferentes: una persona que mida más de *1.80 m* se considera "*Alta*", y una persona que mida menos de *1.80 m* se considera "*Baja*". Evidentemente en la vida real si una persona mide *2 metros* es considerada "*Alta*", pero si una persona mide *1.79 m*, no se considera igual de baja que una que mida *1.50 m*.

En la lógica difusa, como en la vida real, las fronteras entre un conjunto y otro son imprecisas, por lo que una variable puede pertenecer a varios conjuntos, con un grado de pertenencia diferente. La imprecisión es una cualidad de la lógica difusa, en contraposición a la lógica clásica, donde los conjuntos son separados por umbrales que no dan lugar a imprecisión, sino que son nítidos. Esta cualidad hace que los sistemas basados en lógica difusa sean adecuados para resolver problemas en dominios donde existe incertidumbre, como son los problemas de la vida cotidiana. A continuación explicamos algunos de los conceptos básicos dentro del dominio de la lógica difusa.

- *Universo de discurso*: En lógica difusa se le denomina universo de discurso al rango de valores que puede tomar una variable difusa. En el ejemplo de la estatura, mencionado anteriormente, el universo de discurso para la estatura de una persona adulta podría estar entre 1.30 *m* y 2.30 *m*.

- *Variable lingüística*: Cada una de las variables de entrada o salida de un sistema difuso, que toma valores lingüísticos dentro del conjunto de etiquetas lingüísticas definidas en el mismo, y valores reales dentro del universo de discurso. En el ejemplo de la estatura, mencionado anteriormente, la variable lingüística *estatura* puede tener el valor 1.85 *m*, que estará asociado a una etiqueta perteneciente al conjunto de etiquetas lingüísticas {"*Baja*", "*Alta*"}.

- *Conjunto difuso*: Conjunto que se representa mediante una etiqueta lingüística y puede contener elementos de forma parcial. La pertenencia de un elemento x a un conjunto difuso A viene dada por un grado de verdad $\mu_A(x)$, también llamado grado de pertenencia, de tal forma que:

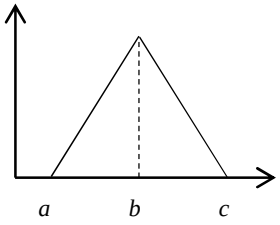
    - Si $\mu_A(x) = 1$, entonces x pertenece al conjunto A.
    - Si $\mu_A(x) = 0$, entonces x no pertenece al conjunto A.
    - Si $0 < \mu_A(x) < 1$, entonces x pertenece parcialmente al conjunto A.

  En el ejemplo de la estatura, las etiquetas lingüísticas "*Baja*" y "*Alta*" describen dos conjuntos difusos, a los cuáles el valor x =1.85 tendrá distintos grados de pertenencia.

- *Funciones de pertenencia*: Cada función de pertenencia de un conjunto nos indica el grado en que cada elemento de un universo dado, pertenece a dicho conjunto. Es decir, la función de pertenencia de un conjunto *A* sobre un universo *X* será de la forma: μ_A: $X \rightarrow [0,1]$, y $\mu_A(x) = r$, donde *r* es el grado en que *x* pertenece al conjunto *A*. Por su sencillez las funciones de pertenencia más utilizadas son las triangulares y las trapezoidales:

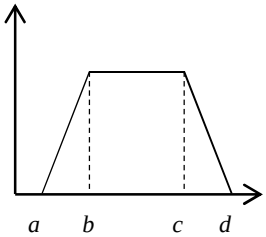
  - *Función triangular*: Se define mediante un límite inferior *a*, un límite superior *c*, y un valor modal *b*, tal que $a < b < c$. El grado de pertenencia de un valor *x* a la función triangular se calcula como:

$$\mu_A(x) = \begin{cases} 0, & \text{si } x <= a \\ \dfrac{x-a}{b-a}, & \text{si } a < x <= b \\ \dfrac{c-x}{c-b}, & \text{si } b < x < c \\ 0, & \text{si } x >= c \end{cases}$$



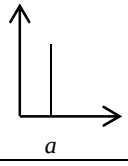
  - *Función trapezoidal*: Se define mediante un límite inferior *a*, un límite superior *d*, el límite de soporte inferior *b* y el límite de soporte superior *c*. El grado de pertenencia de un valor *x* a la función trapezoidal se calcula como:

$$\mu_A(x) = \begin{cases} 0, & \text{si } (x < a) \text{ ó } (x > d) \\ \dfrac{x-a}{b-a}, & \text{si } a <= x <= b \\ 1, & \text{si } b <= x <= c \\ \dfrac{d-x}{d-c}, & \text{si } c <= x <= d \end{cases}$$



  - Función *singleton*: Esta función tiene un valor único cuando $x = a$ y se define como:

$$\mu_A(x) = \begin{cases} 0, & \text{si } x = a \\ 1, & \text{si } x \neq a \end{cases}$$



- *t-norma*: Una función T_n: [0,1] x [0,1] → [0,1] es una *t-norma* si para todo *x*, *y*, *z* pertenecientes al conjunto [0,1] se verifican las siguientes propiedades:

    - Motonicidad: Si $x \leq y$, entonces $T_n(x, z) \leq T_n(y, z)$
    - Asociatividad: $T_n(x, T_n(y, z)) = T_n(T_n(x, y), z))$
    - Conmutatividad: $T_n(x, y) \leq T_n(y, x)$
    - Elemento neutro 1: $T_n(1, x) = x$
    - Elemento 0: $T_n(0, x) = 0$

    Como ejemplos de *t-normas* podemos mencionar la función *mínimo* y el *producto algebraico*.

- *t-conorma*: Una función T_c: [0,1] x [0,1] → [0,1] es un *t-conorma* si para todo *x*, *y*, *z* pertenecientes al conjunto [0,1] se verifican las siguientes propiedades:

    - Motonicidad: Si $x \leq y$, entonces $T_c(x, z) \leq T_c(y, z)$
    - Asociatividad: $T_c(x, T_c(y, z)) = T_c(T_c(x, y), z))$
    - Conmutatividad: $T_c(x, y) \leq T_c(y, x)$
    - Elemento 1: $T_c(1, x) = 1$
    - Elemento neutro 0: $T_c(0, x) = x$

    Como ejemplo de *t-conormas* podemos mencionar la función *máximo* y la *suma algebraica*.

4.1.1 Sistemas Basados en Reglas Difusas

En un sistema basado en reglas cada regla tiene la forma *IF <antecedente> THEN <consecuente>*, donde la parte que se encuentra a la izquierda del *THEN* se denomina antecedente o condición, y la parte de la derecha se denomina consecuente o conclusión. Los sistemas basados en reglas difusas constituyen una extensión de los sistemas basados en reglas

clásicos, pero se componen de reglas cuyos antecedentes y consecuentes se forman con instrucciones de lógica difusa en lugar de instrucciones de lógica clásica. Estos sistemas son usados como herramientas para representar diferentes formas de conocimiento de un problema, así como para modelar las interacciones y relaciones entre sus variables. Debido a esta propiedad, han sido exitosamente aplicados a un amplio rango de problemas en diferentes dominios con incertidumbre y conocimiento incompleto [Cordón *et. al*, 2001]. Los sistemas basados en reglas difusas se clasifican en dos categorías según sus esquemas de inferencia: los que usan modelos de reglas aditivos (utilizan una inferencia similar a la suma ponderada para agregar la conclusión de múltiples reglas en una conclusión final), como el modelo de *Takagi-Sugeno-Kang* [Takagi y Sugeno, 1985]; y los que usan el modelo de *Mamdani* [Mamdani, 1974] que realiza la inferencia de manera no aditiva.

4.1.1.1 Sistemas Basados en Reglas Difusas de tipo *Takagi-Sugeno-Kang*

El modelo de *Takagi-Sugeno-Kan* [Takagi y Sugeno, 1985] fue presentado por primera vez en 1985, y en los años 90 fue el más utilizado en el ámbito industrial debido fundamentalmente a que puede utilizarse para aproximar una función utilizando pocas reglas. En el modelo de *Takagi-Sugeno-Kan* (*TSK*) los consecuentes de las reglas difusas de las que hace uso se representan como una combinación lineal de los antecedentes de dichas reglas.

$$IF\ X_1\ IS\ A_1\ AND\ \ldots\ AND\ X_n\ IS\ A_n\ THEN\ Y = p_1 \,.\, X_1 + \ldots + p_n \,.\, X_n + p_0,$$

donde X_i son las variables de entrada del sistema, Y es la variable de salida y $p = (p_0, p_1, .., p_n)$ es un vector de parámetros reales. La salida de un sistema basado en reglas difusas del tipo *TSK* usando una base de conocimientos de m reglas se obtiene a través de una suma ponderada de las salidas individuales (Y_i, $i=1,.., m$) proporcionadas por cada regla como sigue:

$$Y_0 = \frac{\sum_{i=1}^{m} h_i \cdot Y_i}{\sum_{i=1}^{m} h_i}, \qquad 4.1$$

donde $h_i = T(A_{i1}(x_1), .., A_{in}(x_n))$ es el grado de correspondencia entre la parte antecedente de la *i-ésima* regla y la entrada actual del sistema $x_0 = (x_1, .., x_n)$. T es el operador de conjunción, generalmente representado por el mínimo o el producto algebraico. Una representación gráfica del modelo TSK se muestra en la Figura 4.1.

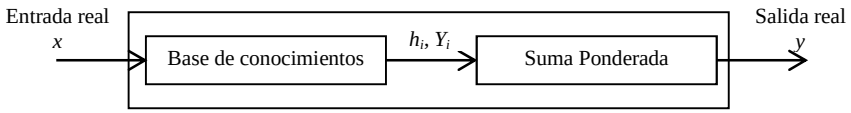


Figura 4.1. Sistema Basado en Reglas Difusas de tipo TSK [Cordón et. al, 2001]

La principal ventaja de estos sistemas es que presentan un conjunto de sistemas de ecuaciones compactas que permiten estimar los parámetros p_i por medio de métodos clásicos. Su principal limitación radica en la forma de los consecuentes de las reglas, que no proporciona un marco natural para el conocimiento de los expertos del dominio [Cordón *et. al*, 2001].

4.1.1.2 Sistemas Basados en Reglas Difusas de tipo *Mamdani*

En 1974, Mamdani propuso el primer sistema basado en reglas difusas que aplica la formulación de Zadeh a un problema de control. Este sistema fue conocido por las siglas *FLC* (*Fuzzy Logic Controller*) [Mamdani, 1974]. Los sistemas basados en reglas difusas de tipo *Mamdani* se componen de una base de conocimientos y un motor de inferencias, que está compuesto por una *interfaz de fuzzificación* en la entrada, un sistema de inferencia y una *interfaz de defuzzificación* en la salida, como se muestra en la Figura 4.2.

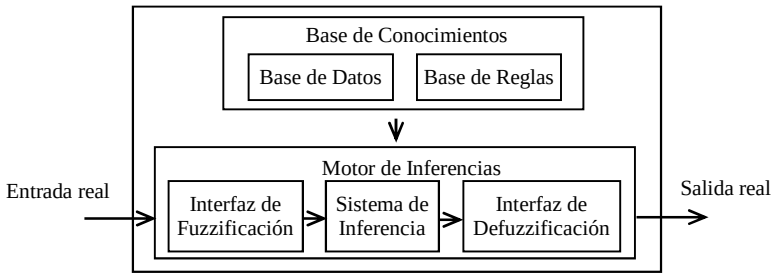


Figura 4.2. Sistema Basado en Reglas Difusas de tipo Mamdani [Cordón et. al, 2001]

Base de Conocimientos

La base de conocimientos sirve como repositorio del conocimiento disponible del problema, a través del cual tiene lugar el razonamiento del sistema de inferencia para procesar una entrada y generar una salida difusa. La base de conocimientos está constituida por una base de datos y una base de reglas.

La base de datos contiene un conjunto de variables lingüísticas de entrada y salida del sistema, así como un conjunto de etiquetas lingüísticas, cada una de las cuales define un conjunto difuso. Cada conjunto difuso consta de una función que indica el grado en que una variable lingüística pertenece a dicho conjunto, llamada función de pertenencia.

La base de reglas es una colección de reglas construidas con las variables y etiquetas lingüísticas de la base de datos, operadores lógicos y la estructura condicional *IF-THEN*. Cada regla tiene la forma `IF <antecedente> THEN <consecuente>`. La base de reglas permite determinar el conjunto difuso de la salida en función de la entrada.

Motor de Inferencias

El motor de inferencias consta de una *interfaz de fuzzificación* que transforma un valor real de la entrada en un valor difuso para el proceso de razonamiento, un *sistema de inferencia* que utiliza ese valor difuso para inferir una salida en forma de conjuntos difusos de acuerdo con la base de conocimientos y una *interfaz de defuzzificación* que convierte esta salida difusa en un valor real, que será el resultado final del sistema.

La interfaz de fuzzificación permite al sistema basado en reglas difusas traducir entradas reales a sus correspondientes valores en el universo de los conjuntos difusos, con los que opera el sistema de inferencia. Sea F el operador de *fuzzificación*, la función de pertenencia del conjunto difuso A' sobre el universo de discurso U asociada al valor real x_0 se define como [Cordón *et. al*, 2001]:

$$A' = F(x_0) \qquad 4.2$$

El sistema de inferencia genera salidas difusas a partir de las entradas difusas obtenidas de la *interfaz de fuzzificación*, de acuerdo a las relaciones definidas en la base de reglas. Este procedimiento se lleva a cabo a través de una extensión del *modus ponens* de la lógica clásica llamado *modus ponens generalizado* [Zadeh, 1973], que se define como:

```
IF X is A THEN Y is B
X is A'
-----------------------
          Y is B'
```

Una estructura difusa condicional de la forma *"IF X is A THEN Y is B"* representa una relación difusa entre *A* y *B* definida en *U* x *V*, que se expresa por un conjunto difuso *R* cuya función de pertenencia $\mu_R(x, y)$ está dada por:

$$\mu_R(x, y) = I(\mu_A(x), \mu_B(y)), \forall x \in U, y \in V, \quad 4.3$$

donde $\mu_A(x)$ y $\mu_B(y)$ son las funciones de pertenencia de *A* y *B*, e *I* es un operador de implicación. La función de pertenencia del conjunto difuso *B'* se obtiene como resultado de la regla composicional de inferencia [Zadeh, 1973], que plantea: *Si R es una relación difusa definida en U x V y A' es un conjunto difuso definido en U, entonces el conjunto difuso B', inducido por A', se obtiene de la composición de A' y R*, es decir:

$$B' = A' \circ R \quad 4.4$$

Aplicando la regla composicional de inferencia a la *i-ésima* regla de la base de reglas:

$$R_i\texttt{: IF } X_{i1} \texttt{ is } A_{i1} \texttt{ and … and } X_{in} \texttt{ is } A_{in} \texttt{ THEN Y is } B_i,$$

la regla composicional de inferencia se reduce a:

$$\mu_{B_i'}(y) = I(\mu_{A_i}(x_0), \mu_{B_i}(y)), \quad 4.5$$

donde: $\mu_{A_i}(x_0) = T(\mu_{A_{i1}}(x_1), .., \mu_{A_{in}}(x_n)), x_0 = (x_1, .., x_n)$ es la entrada actual del sistema, *T* es un operador difuso de conjunción, e *I* un operador difuso de implicación [Cordón et. al, 2001]. Si el sistema tiene *m* reglas en la base de conocimientos, la regla composicional genera *m* conjuntos difusos de salida.

La *interfaz de defuzzificación* agrega la información de los *m* conjuntos difusos y la convierte en un valor no difuso correspondiente al dominio de salida del sistema (inferencia). Teniendo en cuenta el orden de ejecución de la agregación y la inferencia, se definen dos modos distintos de defuzzificación: el modo *FATI* (*First Aggregate, Then Infer*), en el que primero se realiza la agregación y luego la inferencia, y el modo *FITA* (*First Infer, Then Aggregate*), donde primero se realiza la inferencia y después la agregación.

a) Modo FATI (*First Aggregate, Then Infer*)

- En la interfaz de defuzzificación del modo *FATI*, el primer paso consiste en agregar los conjuntos difusos individuales *Bi'* a un conjunto difuso global *B'* por medio de un operador de agregación *G*. El operador de agregación utilizado es por lo general una *t-conorma*:

$$\mu_{B'}(y) = G\left\{\mu_{B_1'}(y), \mu_{B_2'}(y), .., \mu_{B_n'}(y)\right\} \qquad 4.6$$

- El segundo paso consiste en transformar el conjunto difuso *B'* en un valor de salida real y_0 utilizando un método de defuzzificación *D* como sigue:

$$y_0 = D(\mu_{B'}(y)) \qquad 4.7$$

Existen muchos métodos de defuzzificación, pero los más conocidos son: el centro de gravedad (*CG*), la media ponderada y los que utilizan los máximos.

- El método del centro de gravedad (*CG*) se define como:

$$y_0 = \frac{\int_Y y \cdot \mu_{B'}(y) dy}{\int_Y \mu_{B'}(y) dy} \qquad 4.8$$

- El método de la media ponderada consiste en aplicar una ponderación a cada función de salida equivalente a sus respectivos grados máximos de pertenencia.

$$y_0 = \frac{\sum \mu_{B'}(y) \cdot y}{\sum \mu_{B'}(y)} \qquad 4.9$$

- Los métodos que usan los máximos son: *FOM* (*First of maxima*), que usa el menor de los máximos, *LOM* (*Last of maxima*), que usa el mayor de los máximos, y por último *MOM* (*Mean of maxima*), que consiste en la media aritmética de los máximos. Los tres métodos de máximos se definen como:

$$y_0 = y_{\inf} \qquad FOM \qquad 4.10$$

$$y_0 = y_{\sup} \qquad LOM \qquad 4.11$$

$$y_0 = \frac{y_{\inf} + y_{\sup}}{2} \qquad MOM \qquad 4.12$$

con

$$y_{\inf} = \inf\{y \mid \mu_{B'}(y) = \sup \mu_{B'}(y)\},$$
$$y_{\sup} = \sup\{y \mid \mu_{B'}(y) = \sup \mu_{B'}(y)\}.$$

b) Modo FITA (First Infer, Then Aggregate): En el modo *FITA* la contribución de cada conjunto difuso se considera por separado y el valor final real se calcula a través del promedio o la operación de selección desarrollada en el conjunto de los valores reales derivados de cada conjunto difuso individual B_i'. La opción más común para calcular la salida final es utilizar el método *CG* o el del valor máximo (*MV*) ponderado por el grado de correspondencia de la manera siguiente:

$$y_0 = \frac{\sum_{i=1}^{m} h_i \cdot y_i}{\sum_{i=1}^{m} h_i}, \qquad 4.13$$

donde y_i es el *CG* o el *MV* del conjunto difuso B_i' inferido de la regla R_i, y $h_i = \mu_A(x_0)$ es la correspondencia entre la entrada x_0 del sistema y el antecedente de la regla.

Entre las principales ventajas de los sistemas basados en reglas difusas de tipo *Mamdani* podemos mencionar que facilitan la confección, interpretación y depuración de la base de reglas por parte de los expertos, debido a que su base de conocimiento está formada por etiquetas y variables lingüísticas usadas en el lenguaje humano. También poseen un alto grado de flexibilidad en el diseño del motor de inferencias, permitiendo el uso de distintos métodos de *fuzzificación*, *defuzzificación*, agregación e inferencia, conjunción e implicación. Permiten además el diseño de funciones de pertenencia específicas para cada aplicación. Como limitación podemos mencionar que el sistema se hace más preciso al aumentar el número de variables, pero esto a la vez dificulta su comprensión por parte de los usuarios debido a que el número de reglas puede crecer considerablemente.

4.2 Aprendizaje con Algoritmos Genéticos

Una de las técnicas más utilizadas para la resolución de problemas de aprendizaje automático son los algoritmos evolutivos, dentro de los cuáles se encuentran los algoritmos genéticos. Los algoritmos genéticos surgen en los años 70 de la mano de John Henry Holland [Holland, 1975] inspirados en la evolución biológica de las especies, de acuerdo a los principios postulados por Darwin. Estos algoritmos hacen evolucionar una población de individuos, a través de acciones aleatorias semejantes a las que actúan en la evolución biológica de organismos vivos, como las mutaciones y recombinaciones genéticas, así como una selección de acuerdo con algún criterio, en función del cual se determina cuáles son los individuos más fuertes, que sobreviven y cuáles los más débiles, que son descartados.

Los algoritmos genéticos crean soluciones a problemas del mundo real, con el objetivo de hacerlas evolucionar hacia valores óptimos. El conjunto de soluciones posibles al problema son los individuos, que pueden representarse como un conjunto de parámetros (denominados genes), que se agrupan en una serie de valores (cromosoma). El conjunto de parámetros que representa un cromosoma particular se denomina fenotipo y contiene la información requerida para construir un organismo, el cual se refiere como genotipo [Domínguez-Dorado, 2004].

La adaptación de un individuo al problema depende de la evaluación del genotipo, que puede inferirse a partir del fenotipo, es decir, puede ser computada a partir del cromosoma. Los cromosomas evolucionan a través de iteraciones, llamadas generaciones, que se obtienen aplicando los operadores genéticos de cruce y mutación. En cada generación, los cromosomas son evaluados usando alguna medida de aptitud, que determina el nivel de adaptación y debe ser diseñada para cada problema de manera específica. En el diagrama de flujo de la Figura 4.3 se muestran los pasos a seguir en un algoritmo genético general.

El primer paso es la inicialización, que consiste en generar aleatoriamente la población inicial, luego se aplica la función de aptitud a cada uno de los cromosomas de esta población para saber qué tan “buena” es la solución que se está codificando. Posteriormente se verifica la condición de terminación, que suele cumplirse cuando el sistema alcanza el número máximo de iteraciones predefinido, o cuando no haya cambios en la población. Mientras no se cumpla la condición de terminación se seleccionan los individuos con mejor aptitud para ser cruzados en la siguiente generación y se aplican sobre ellos los operadores genéticos de cruce o

mutación. Una vez aplicados los operadores genéticos, se agregan los nuevos individuos a la población y se seleccionan los mejores para formar parte de la generación siguiente.

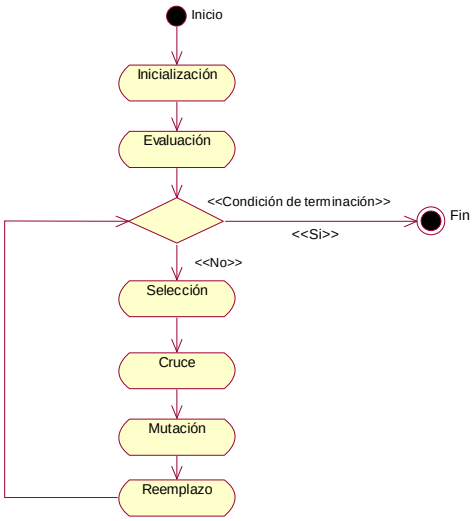


Figura 4.3. Diagrama de flujo de un algoritmo genético general

4.2.1 Población

La población inicial se escoge generando cadenas de cromosomas al azar, donde cada gen puede contener uno de los posibles valores del alfabeto con probabilidad uniforme. Una cuestión importante en esta etapa es elegir el tamaño de la población. Sobre esta cuestión se han realizado varios estudios, como el de Goldberg [Goldberg, 1989], que plantea que el tamaño óptimo de la población para cadenas de longitud *l*, con codificación binaria, crece exponencialmente con el tamaño de la cadena, lo que hace que este método sea demasiado costoso computacionalmente. En cambio, Alander [Alander, 1992], basándose en evidencia empírica sugiere que un tamaño de población comprendido entre *l* y *2l* es suficiente para afrontar con éxito la mayoría de los problemas.

4.2.2 Función de adaptación

Dado un cromosoma particular, la función de adaptación, también conocida como función objetivo o *fitness*, le asigna un número real que refleja el nivel de adaptación al problema del

individuo representado por el cromosoma. La regla general para construir una buena función objetivo es que esta debe reflejar el valor del individuo de una manera real. Sin embargo, en muchos problemas de optimización combinatoria, donde existen gran cantidad de restricciones, buena parte de los puntos del espacio de búsqueda representan individuos no válidos. Teniendo en cuenta este planteamiento se han propuesto varias soluciones. La primera no descarta a los individuos que no verifican las restricciones, y se siguen efectuando cruces y mutaciones hasta obtener individuos válidos, o bien a dichos individuos se les asigna una función objetivo igual a cero. Otra posibilidad consiste en reconstruir aquellos individuos que no verifican las restricciones por medio de un nuevo operador denominado *reparador*.

Otro enfoque está basado en la penalización de la función objetivo y consiste en dividir la función objetivo del individuo por una cantidad que guarda relación con las restricciones que dicho individuo viola. Dicha cantidad suele tener en cuenta el número de restricciones violadas o el denominado coste esperado de reconstrucción, que no es más que el coste asociado a la conversión de dicho individuo en otro que no viole ninguna restricción. Otra técnica que se ha venido utilizando es la evaluación aproximada de la función objetivo. En algunos casos la obtención de *n* funciones objetivo aproximadas puede resultar mejor que la evaluación exacta de una única función objetivo.

El método propuesto por Goldberg y Richardson [Goldberg y Richardson, 1987] utiliza una modificación de la función objetivo de cada individuo, de tal manera que individuos que estén muy cercanos entre sí devalúen su función objetivo para que la población gane en diversidad.

4.2.3 Selección

El mecanismo de selección más simple es en el que se fuerza a que el mejor individuo de la población sea seleccionado como padre, pero la mayoría de los algoritmos genéticos no utilizan mecanismos de selección basados en este sistema, debido a que se aleja un poco del comportamiento real de la evolución.

Algunas de las funciones de selección de padres más utilizadas son las del grupo de esquemas de selección proporcionales a la función objetivo, en las cuales cada individuo tiene una probabilidad de ser seleccionado como padre que es proporcional al valor de su función objetivo. Entre los métodos que utilizan la función objetivo para determinar la probabilidad de

selección se encuentran el denominado *muestreo estocástico con reemplazamiento del resto*, introducido por Brindle [Brindle, 1991], que empíricamente ha proporcionado buenos resultados. En este esquema de selección cada individuo es seleccionado un número de veces que coincide con la parte entera del número esperado de ocurrencias de esta selección, compitiendo los individuos por los restos.

En 1985, Baker introduce un método denominado *muestreo universal estocástico* [Baker, 1985] que utiliza un único giro de ruleta donde los sectores circulares son proporcionales a la función objetivo. Los individuos son seleccionados a partir de marcadores equi-espaciados y con comienzo aleatorio. Estos métodos por lo general tienen asociado el problema de la convergencia prematura provocado por la existencia de *superindividuos*.

Una manera de superar el problema de la convergencia temprana es aplicar el esquema de *selección por rango* propuesto por Baker [Baker, 1987], debido a que se produce un reparto más uniforme de la probabilidad de selección. Este método consiste en ordenar la población de menor a mayor, asignar el número de copias que cada individuo debe recibir de acuerdo a una función de asignación no creciente, y luego realizar la selección proporcional conforme a esa asignación. Si denotamos por *rango* (f (I_j)) al rango de la función objetivo del individuo I_j, cuando los individuos de la población han sido ordenados de menor a mayor rango (el peor individuo tiene rango 1 y el mejor tiene rango λ), la probabilidad de que el individuo I_j sea seleccionado como padre cuando la selección se efectúa proporcionalmente al rango del individuo se define como:

$$P_{rango}(I_j) = \frac{rango(f(I_j))}{\lambda(\lambda+1)/2}, \qquad 4.14$$

donde f es la función objetivo, y el denominador es una constante de normalización.

Otro modelo de selección es la *selección de estado estacionario* [Whitley, 1989], donde la descendencia de los individuos seleccionados en cada generación vuelve a la población genética preexistente, reemplazando algunos de los miembros menos aptos de la anterior generación. La *selección por torneo* [Brindle, 1981] constituye un procedimiento de selección de padres muy extendido que consiste en escoger al azar un número de individuos de la población (en función del tamaño del torneo), seleccionar el mejor individuo de este grupo y repetir el proceso hasta que el número de individuos seleccionados coincida con el tamaño de

la población. Ejemplos de trabajos que utilizan la selección por torneo encontramos en [Suh y Van Gucht, 1987; Goldberg *et. al*, 1989; Muhlenbein, 1990].

4.2.4 Cruce

El cruce es el principal operador genético y representa el proceso de reproducción. Opera sobre dos cromosomas a la vez para generar dos descendientes donde se combinan las características de ambos padres. El operador de cruce más sencillo es el basado en un punto, en el cual los dos individuos seleccionados para jugar el papel de padres, son recombinados por medio de la selección de un punto de corte, para posteriormente intercambiar las secciones que se encuentran a la derecha de dicho punto (Figura 4.4).

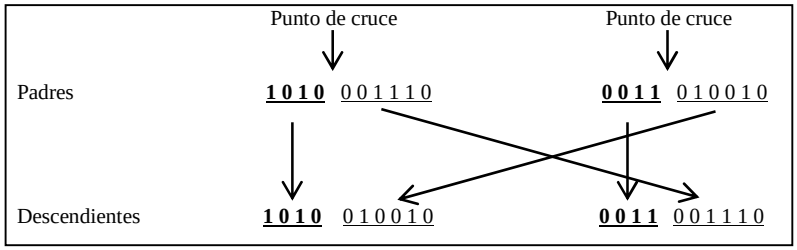


Figura 4.4. Operador de cruce basado en un punto para individuos binarios

En el operador de cruce basado en dos puntos [De Jong, 1975], los cromosomas (individuos) pueden contemplarse como un circuito en el cual se efectúa la selección aleatoria de dos puntos. El cruce basado en un punto, puede verse como un caso particular del cruce basado en dos puntos, en el cual uno de los puntos de corte se encuentra fijo al comienzo de la cadena que representa al individuo. La Figura 4.5 muestra un ejemplo del cruce basado en dos puntos para individuos binarios.

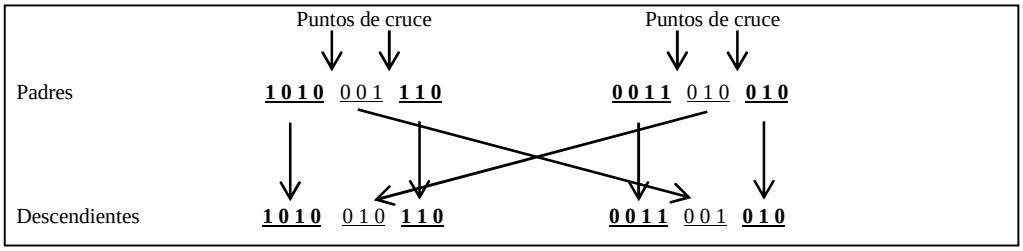


Figura 4.5. Operador de cruce basado en dos puntos para individuos binarios

En el denominado operador de cruce uniforme [Syswerda, 1991] cada gen en la descendencia se crea copiando el correspondiente gen de uno de los dos padres, escogido de acuerdo a una *máscara de cruce* generada aleatoriamente de manera tal que cualquiera de los elementos del alfabeto tenga asociada la misma probabilidad. Cuando existe un 1 en la *máscara de cruce*, el gen es copiado del primer padre, mientras que cuando existe un 0 en la máscara, el gen se copia del segundo padre, tal y como se muestra en la Figura 4.6. En 1994, Larrañaga y Poza propusieron un método de construcción de la *máscara de cruce* basado en la función objetivo [Larrañaga y Poza, 1994], de manera tal que mientras mayor sea el grado de adaptación de un individuo, mayor es la probabilidad de heredar sus características.

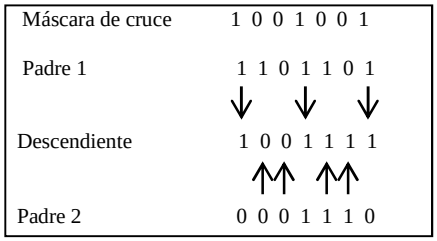


Figura 4.6. Operador de cruce uniforme para individuos binarios

Existen muchos otros operadores de cruce, como el de Sirag y Weiser [Sirag y Weiser 1987] que introducen un umbral de energía y una temperatura que influyen en la manera de escoger los bits individuales, y los que consideran la idea de que el cruce debería ser más probable en algunas posiciones de la cadena [Holland, 1975; Davis, 1985].

4.2.5 Mutación

La mutación se aplica a cada hijo de manera individual y consiste en la alteración aleatoria de una parte del cromosoma del individuo. La mutación proporciona una exploración rápida del espacio de búsqueda, asegurando que ningún punto tenga probabilidad cero de ser examinado y es de capital importancia para asegurar la convergencia de los Algoritmos Genéticos. La Figura 4.7 muestra la mutación del tercer gen de un cromosoma.

Se han desarrollado algunos trabajos sobre la búsqueda del valor óptimo para la probabilidad de mutación, como por ejemplo [De Jong, 1975] que recomienda utilizar una probabilidad de mutación de l^{-1} siendo l la longitud de la cadena. En [Schaffer *et. al*, 1989] se propone utilizar un valor proporcional al número de individuos de la población, mientras que

algunos autores consideran que la probabilidad de mutación debe ser variable a medida que aumenta el número de iteraciones [Michalewicz y Janikow, 1991].

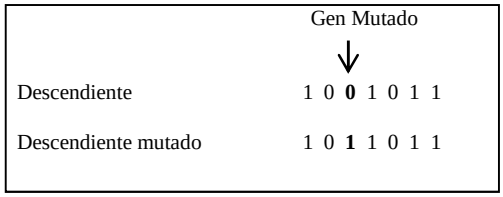


Figura 4.7. Operador de mutación para individuos binarios

4.2.6 Reemplazo

Una vez obtenidos los descendientes de una determinada población, se escogen los individuos que formarán parte de la siguiente generación. Dicho proceso se suele hacer fundamentalmente de dos formas distintas. La primera es la reducción simple y consiste en considerar que la siguiente población la formarán sólo los descendientes, y la segunda es la reducción elitista, que consiste en elegir los mejores individuos de entre los que forman la generación anterior y los descendientes.

En [De Jong, 1975] se introdujo el concepto de tasa de reemplazo generacional (*trg*), con el objetivo de efectuar un solapamiento controlado entre padres e hijos. En su trabajo, en cada paso se selecciona una proporción (*trg*) de la población para ser cruzada y los hijos resultantes pueden reemplazar a miembros de la población anterior. Este tipo de Algoritmos Genéticos se conocen bajo el nombre de *SSGA* (*Steady State Genetic Algorithm*).

En [Michalewicz, 1992] se introduce el *Algoritmo Genético Modificado*, en el cual para llevar a cabo el reemplazo generacional, teniendo en cuenta el valor de la función objetivo, se selecciona al azar un conjunto de individuos para la reproducción y un conjunto de individuos destinados a morir. De esta manera cuanto mayor sea la función objetivo, mayor será la probabilidad de que un individuo sea seleccionado para la reproducción, y menor la probabilidad de que dicho individuo fallezca.

4.2.7 Aprendizaje de reglas difusas con algoritmos genéticos

En la actualidad el empleo de algoritmos evolutivos, y específicamente algoritmos genéticos en problemas de aprendizaje está cada vez más extendido debido a que ofrecen numerosas ventajas, como por ejemplo su flexibilidad para manejar diferentes representaciones. Los procesos de aprendizaje evolutivo cubren diferentes niveles de complejidad con respecto a los cambios estructurales de un sistema difuso, desde el caso más simple de optimización de parámetros hasta el más alto nivel de complejidad como puede ser el aprendizaje de una base de reglas completa. Los algoritmos genéticos ofrecen mecanismos para codificar y evolucionar operadores de agregación para los antecedentes de las reglas, con diferentes semánticas, operadores de agregación y métodos de *defuzzificación*. Por ello, siguen siendo hoy una técnica válida para diseñar y optimizar por completo los sistemas basados en reglas difusas.

Los algoritmos genéticos utilizados en la obtención de reglas difusas son clasificados según dos grandes grupos: los de tipo *Michigan* y los de tipo *Pittsburgh*. La principal diferencia entre estas dos estrategias radica en la representación de los individuos de la población, es decir, la codificación de las reglas.

4.2.7.1 La aproximación de Michigan

En la aproximación de *Michigan* [Holland, 1975], propuesta por el profesor John Henry Holland de la Universidad de Michigan, cada individuo de la población codifica una única regla de longitud constante. Cada regla se evalúa sin tener en cuenta al resto de la población, lo que evita la existencia de una conexión entre las reglas y posibilita su evolución de manera independiente. El número reducido de individuos permite disminuir también el tiempo empleado para calcular la función de evaluación y simplificar el diseño de los operadores genéticos. Sin embargo, la evaluación de las reglas de manera independiente dificulta el análisis y evaluación de las interacciones entre las reglas. Este método se utiliza fundamentalmente en aprendizaje continuo (*online*) en problemas no inductivos.

4.2.7.2 La aproximación de Pittsburgh

En la aproximación de *Pittsburgh* [Smith, 1980], propuesta por el profesor Jeff Smith de la universidad de Pittsburgh en su tesis doctoral, cada individuo de la población codifica un

conjunto de reglas en vez de una regla individual. Aunque la longitud de las reglas es constante, cada individuo puede codificar un número distinto de reglas. Las reglas se evalúan en conjunto, siendo la calidad de una regla dependiente del resto. El utilizar individuos más complejos respecto a los métodos que siguen la filosofía de *Michigan* hace que aumente el coste computacional del cálculo de la función de evaluación en los métodos que utilizan el enfoque de *Pittsburgh*. Mientras el enfoque de *Michigan* está cercano a la idea de representar el conocimiento en una única entidad que aprende a través de su interacción con el entorno, *Pittsburgh* se acerca más a la idea de evolución a través de la competición entre individuos y la adaptación al entorno [Cordón *et. al*, 2001]. Este enfoque ha sido habitualmente aplicado a la resolución de problemas de clasificación entre otras tareas de minería de datos, siendo adaptado para el entrenamiento tanto en problemas inductivos como no inductivos. Entre los sistemas de aprendizaje que utilizan el enfoque de *Pittsburgh* podemos mencionar a *GABIL* [DeJong *et. al*, 1993] y *GIL* [Janikow, 1993] que han sido diseñados para tareas de clasificación.

4.3 *FuzzyAlign*: Sistema Difuso Multicapa para la Alineación de Ontologías.

En esta tesis hemos desarrollado *FuzzyAlign*, un sistema basado en reglas difusas de tipo *Mamdani*, que consta de 4 capas para la alineación de ontologías. En la Figura 4.8 se presenta la arquitectura del sistema difuso de 4 capas. Las capas definidas fueron las siguientes:

- *Capa léxica*: En esta capa se calcula la similitud lingüística, teniendo en cuenta los sinónimos, las palabras derivadas y el factor léxico basado en las distancias lingüísticas entre los términos dentro del directorio léxico.

- *Capa Terminológica*: En esta capa se obtiene la similitud basada en la terminología de los nombres de los conceptos. Recibe como entrada la similitud lingüística, junto con la similitud semántica, calculada en el módulo léxico-semántico.

- *Capa Estructural*: En esta capa se mejora la similitud terminológica utilizando la similitud que aportan los conceptos en la vecindad de la jerarquía taxonómica, calculada en el módulo estructural-relacional. En esta capa se calcula también la similitud entre las propiedades a través del módulo estructural interno.

- *Capa de alineación*: En esta capa se refina la similitud entre los conceptos utilizando la similitud de las propiedades a partir de los resultados de la capa estructural. Finalmente se seleccionan los mejores valores de similitud y se constituyen las alineaciones para las entidades de las ontologías en el módulo de alineación.

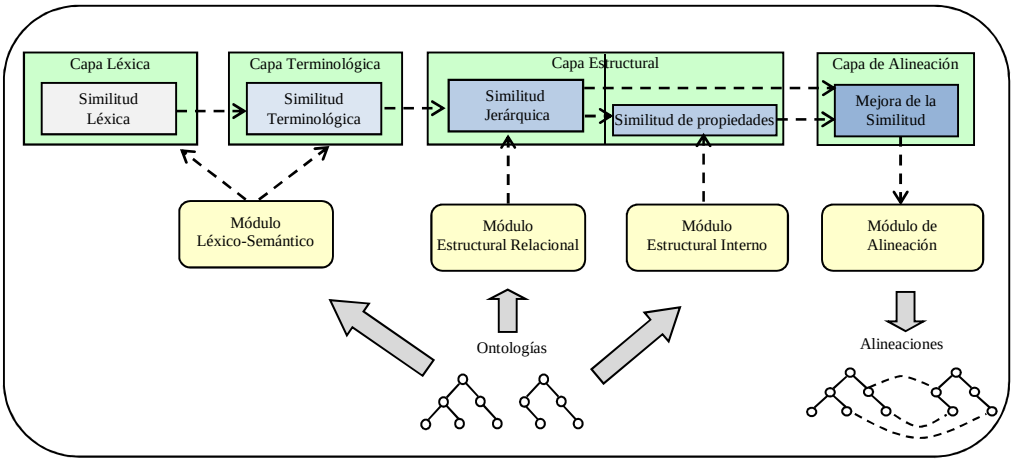


Figura 4.8. Arquitectura de FuzzyAlign

4.3.1 Capa Léxica

La capa léxica es la primera capa del sistema basado en reglas difusas. Se encarga de calcular la similitud lingüística, tomando como entrada los valores de similitud basados en la sinonimia, las palabras derivadas de los términos y el factor léxico. Estas tres medidas de similitud son calculadas en el módulo léxico-semántico utilizando la herramienta *WordNet*, en el caso de ontologías generales, y *UMLS* en el caso de ontologías médicas, como se explicó en la Sección 3.1.2 del Capítulo 3. La Figura 4.9 muestra gráficamente la estructura de la capa léxica, donde hemos definido tres variables de entrada y una de salida que se explican a continuación:

Sim_Sinónimos: Esta variable difusa de entrada representa la similitud basada en los sinónimos de los conceptos.

Sim_Derivación: Esta variable difusa de entrada representa la similitud basada en las palabras derivadas de los conceptos.

Factor_Léxico: Esta variable difusa representa el factor léxico, compuesto por la combinación entre la medida de similitud basada en la distancia de Levenshtein y la distancia entre las palabras dentro del directorio léxico (Capítulo 3. Sección 3.1.2.4).

Sim_Lingüística: Ésta es la variable de salida del sistema y representa la similitud lingüística total entre los conceptos.

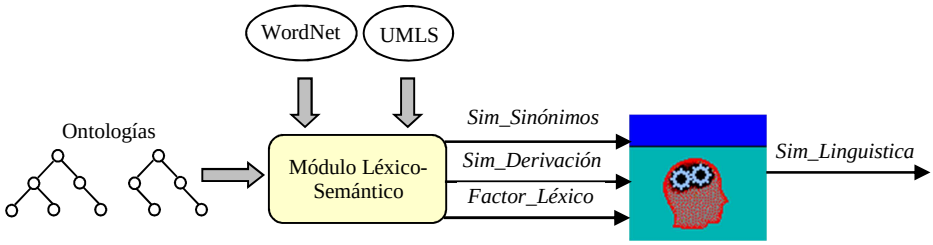


Figura 4.9. Estructura de la capa léxica del sistema basado en reglas difusas

Las cuatro variables del sistema tienen asociado el siguiente conjunto de términos lingüísticos: D = {*Baja* (*B*), *Regular* (*R*), *Media* (*M*), *Alta* (*A*), *Muy Alta* (*MA*)}, cuya semántica ha sido definida a través de funciones triangulares de pertenencia, como se muestra en la Figura 4.10.

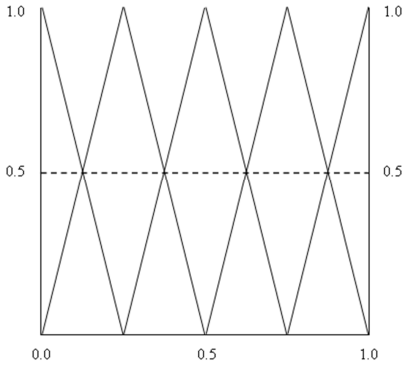


Figura 4.10. Funciones triangulares de pertenencia de las variables de la capa léxica

4.3.2 Capa Terminológica

La capa terminológica del sistema basado en reglas difusas es la encargada de realizar el proceso de cálculo de la similitud entre los nombres de las entidades de las ontologías, combinando la similitud lingüística con los elementos semánticos del contexto de las

entidades. Esta capa recibe como entrada el valor de la similitud lingüística obtenido en la capa léxica y la similitud semántica, calculada en el módulo léxico-semántico utilizando el *coeficiente de Jaccard*, como se explica en la Sección 3.1. La Figura 4.11 muestra la estructura de la capa terminológica. En esta capa hemos definido dos variables de entrada y una de salida que se explican a continuación.

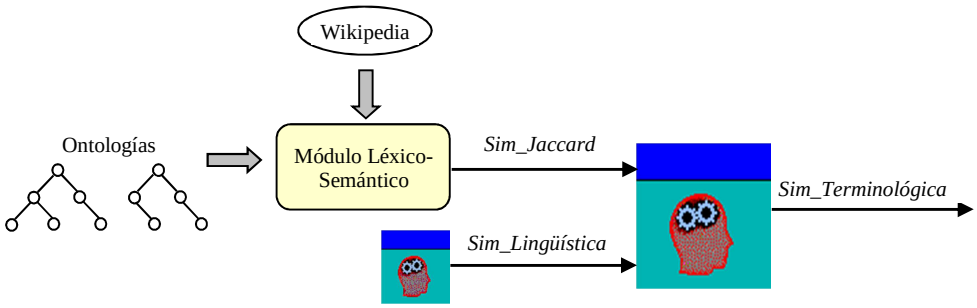


Figura 4.11. Estructura de la capa terminológica del sistema basado en reglas difusas

Sim_Jaccard: Esta variable de entrada representa la similitud semántica entre dos conceptos, obtenida en el módulo léxico-semántico. Tiene asociado el siguiente conjunto de términos lingüísticos: D = {*Baja* (*B*), *Regular* (*R*), *Media* (*M*), *Alta* (*A*), *Muy Alta* (*MA*)}. Las funciones de pertenencia definidas para esta variable son triangulares, como se muestra en la Figura 4.12 (a). Para definir las funciones de pertenencia de esta variable fue necesario dividir los datos de la similitud de *Jaccard* en grupos. Debido a la distribución de los valores obtenidos tras aplicar el *coeficiente de Jaccard* a los conceptos de varias ontologías de prueba, hemos utilizado los cuartiles del conjunto de similitudes para acotar los triángulos de pertenencia de la siguiente manera:

- *Baja*: (-0.0364448, 0.0, 0.00174448)
- *Regular*: (0.0, 0.00174448, 0.01464863)
- *Media*: (0.00174448, 0.01464863, 0.16344766)
- *Alta*: (0.01464863, 0.16344766, 1.0)
- *Muy Alta*: (0.16344766, 1.0, 1.93655234)

Sim_Lingüística: Esta variable de entrada representa la similitud lingüística obtenida en la capa léxica. Tiene asociado el siguiente conjunto de términos lingüísticos: *D* = {*Baja* (*B*), *Regular* (*R*), *Media* (*M*), *Alta* (*A*), *Muy Alta* (*MA*)}. Debido a la distribución de los valores de la similitud lingüística y del resto de las variables del sistema difuso que siguen, hemos definido conjuntos difusos equi-espaciados. Las funciones triangulares de pertenencia para esta variable se muestran en la Figura 4.12 (b).

Sim_Terminológica: Es la variable de salida y representa el valor de la similitud terminológica de los conceptos, obtenida por el sistema difuso. Tiene asociado el siguiente conjunto de términos lingüísticos: *D* = {*Muy Baja* (*MB*), *Baja* (*B*), *Medio Baja* (*MedB*), *Regular* (*R*), *Medio Alta* (*MedA*), *Alta* (*A*), *Muy Alta* (*MA*)}. Sus funciones de pertenencia se muestran en la Figura 4.12 (c).

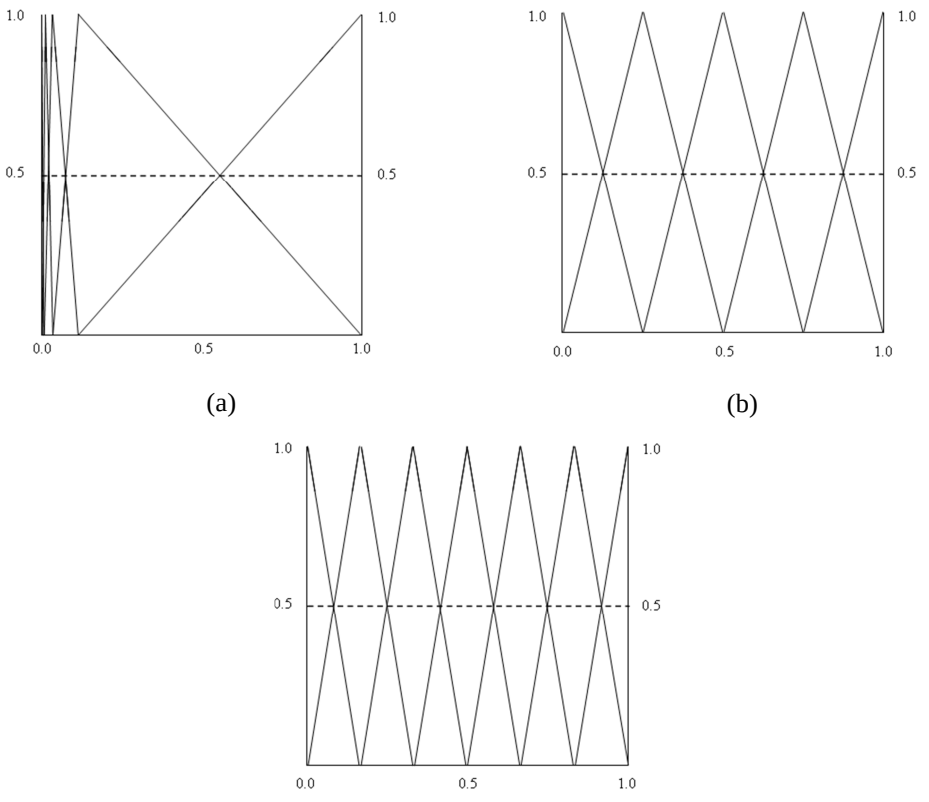


Figura 4.12. Funciones triangulares de pertenencia para las variables: (a) *Sim_Jaccard*, (b) *Sim_Lingüística* y (c) *Sim_Terminológica*

4.3.3 Capa Estructural

En la capa estructural se desarrollan dos tareas fundamentales relacionadas con la estructura de las ontologías: Una es el cálculo de la similitud entre los conceptos teniendo en cuenta la influencia de las similitudes entre los conceptos de la vecindad en la jerarquía taxonómica (*Similitud Jerárquica*), y la otra es el cálculo de la similitud entre las propiedades de los conceptos de manera independiente.

4.3.3.1 Similitud Jerárquica

La similitud jerárquica contribuye a refinar el valor de la similitud entre los conceptos, teniendo en cuenta la influencia de las similitudes de los hijos, padres y hermanos en la jerarquía taxonómica de cada ontología. Esta tarea recibe como entrada la similitud terminológica calculada en la capa anterior del sistema difuso y las similitudes jerárquicas que aportan los hijos, padres y hermanos de los conceptos, que son calculadas en el módulo estructural relacional utilizando el método explicado en la Sección 3.2.1. La salida es la similitud jerárquica de los conceptos. La estructura del sistema para el cálculo de la similitud jerárquica se muestra en la Figura 4.13.

Las 5 variables del sistema tienen asociado el siguiente conjunto de términos lingüísticos: D = {*Muy Baja* (*MB*), *Baja* (*B*), *Medio Baja* (*MedB*), *Regular* (*R*), *Medio Alta* (*MedA*), *Alta* (*A*), *Muy Alta* (*MA*)}, cuya semántica ha sido definida por medio de siete funciones triangulares de pertenencia equi-espaciadas al igual que en la Figura 4.12 (c).

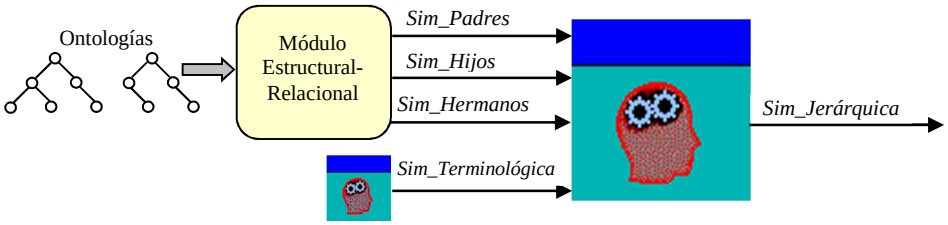


Figura 4.13. Estructura del cálculo de la similitud jerárquica del sistema basado en reglas difusas

4.3.3.2 Similitud entre Propiedades

La similitud entre las propiedades de los conceptos tiene en cuenta tres factores fundamentales: la similitud lingüística de sus nombres, la similitud entre las clases a las que pertenecen (*dominio*) y la similitud de sus tipos (*rango*). Se calcula en el módulo estructural interno, utilizando el método explicado en la Sección 3.2.2. La Figura 4.14 muestra la estructura del sistema para el cálculo de la similitud entre las propiedades. Como se puede observar en el sistema se han definido tres variables de entrada y una de salida que se explican a continuación:

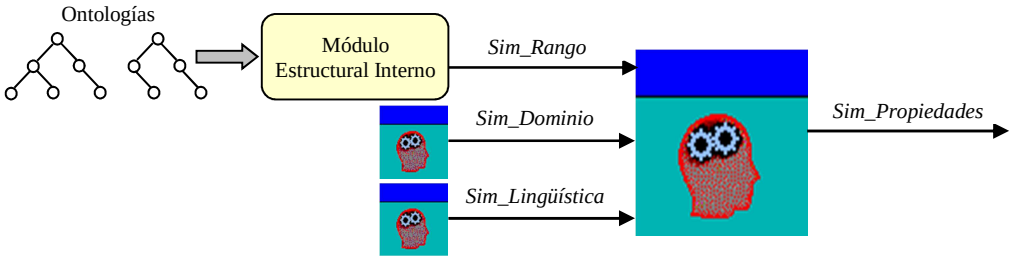


Figura 4.14. Estructura del cálculo de la similitud de propiedades del sistema basado en reglas difusas

Sim_Lingüística: Esta variable de entrada representa el valor de la similitud lingüística entre los nombres de las propiedades, obtenido en la capa léxica del sistema. Las funciones triangulares de pertenencia definidas para esta variable se muestran en la Figura 4.15 (a).

Sim_Dominio: Esta variable de entrada representa la similitud entre las clases a las que pertenecen las propiedades. El valor de la similitud entre las clases se obtiene en el módulo de similitud jerárquica del sistema. Esta variable tiene asociadas las funciones triangulares de pertenencia que se muestran en la Figura 4.15 (b) y el siguiente conjunto de términos lingüísticos: D = {*Muy Baja* (*MB*), *Baja* (*B*), *Medio Baja* (*MedB*), *Regular* (*R*), *Medio Alta* (*MedA*), *Alta* (*A*), *Muy Alta* (*MA*)}.

Sim_Rango: Esta variable de entrada representa la similitud entre los tipos de las propiedades. Es una variable lógica que indica si los tipos de las propiedades son equivalentes o no. En el caso de las propiedades objeto, como su tipo es una clase, esta variable será verdadera solamente si las clases son equivalentes. En el caso de las propiedades dato se

verificará la correspondencia a través de las tablas de equivalencia de tipos *XML* con el método explicado en la Sección 3.2.2.1.

Sim_Propiedades: Es la variable de salida del sistema y representa el valor de la similitud entre dos propiedades. Los conjuntos lingüísticos y las funciones de pertenencia que hemos definido son los mismos que para la variable *Sim_Dominio*, como se observa en la Figura 4.15 (c).

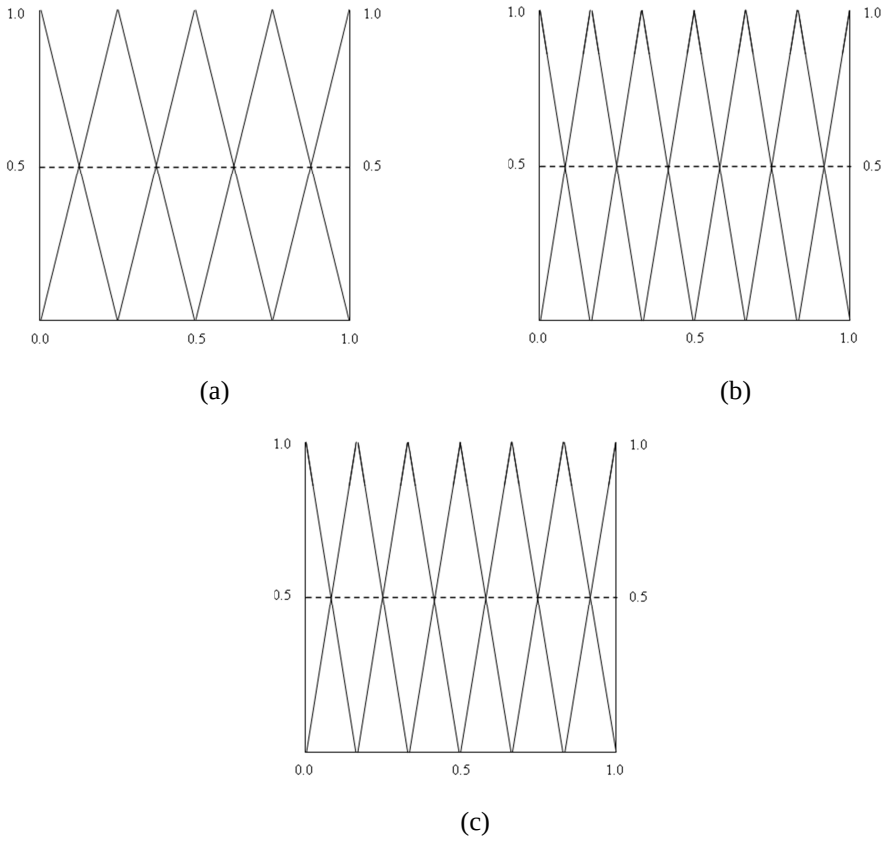


Figura 4.15. Funciones triangulares de pertenencia de las variables para el cálculo de la similitud entre las propiedades: (a) *Sim_Lingüística*, (b) *Sim_Dominio y* (c) *Sim_Propiedades*

4.3.4 Capa de Alineación

La capa de alineación es la capa final del sistema basado en reglas difusas y su objetivo es obtener un índice de similitud final entre los conceptos teniendo en cuenta la influencia que ejercen el número de propiedades y el valor que aportan las similitudes entre ellas, que se calcula en el módulo estructural interno (Sección 3.2.2). Dependiendo de estos valores, la similitud jerárquica es reforzada o debilitada, dando como resultado un indicador más exacto de la similitud entre los conceptos. En la Figura 4.16 se muestra la arquitectura de la capa de alineación del sistema basado en reglas difusas. Las variables que hemos definido en esta capa son las que se describen a continuación.

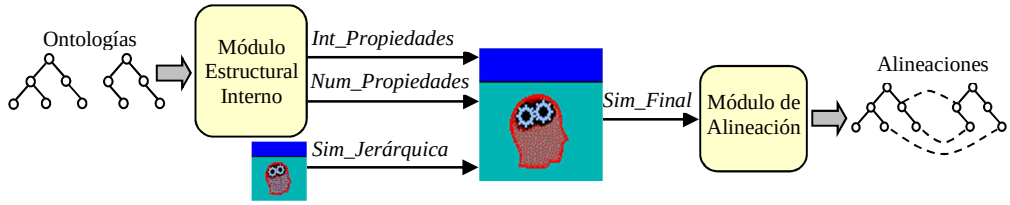


Figura 4.16. Estructura de la capa de alineación del sistema basado en reglas difusas

Sim_Jerárquica: Esta variable de entrada representa la similitud entre los conceptos después de considerar la influencia de los hijos, padres y hermanos en ambas taxonomías. Para describir esta variable fueron definidos siete conjuntos difusos y funciones triangulares de pertenencia como se explican en la capa anterior (Figura 4.12 (c)).

Num_Propiedades: Esta variable de entrada es binaria y nos indica si ambos conceptos tienen el mismo número de propiedades.

Int_Propiedades: Esta variable de entrada representa la influencia de la similitud interna de las propiedades (calculada en el módulo estructural interno) en la similitud final de los conceptos. Sus conjuntos lingüísticos son: D = {*Muy Baja*(*MB*), *Baja*(*B*), *Medio Baja*(*MedB*), *Regular*(*R*), *Medio Alta*(*MedA*), *Alta*(*A*), *Muy Alta*(*MA*)}. Para esta variable también se han definido siete funciones triangulares de pertenencia equi-espaciadas.

Sim_Final: Es la variable de salida del sistema y representa el valor de la similitud final entre los conceptos después de considerar sus elementos terminológicos y estructurales. Esta

variable, como las anteriores, tiene asociados siete conjuntos equiespaciados con funciones triangulares de pertenencia.

Una vez obtenidas las similitudes entre todos los conceptos y propiedades de las dos ontologías, el módulo de alineación se encarga de seleccionar las entidades equivalentes y construir el conjunto de alineaciones finales.

4.3.4.1 Umbral de similitud para la selección de alineaciones.

Como los resultados de aplicar el sistema basado en reglas borrosas son matrices de similitud (valores reales entre 0 y 1) fue necesario establecer un umbral de similitud a partir del cuál las entidades serían consideradas equivalentes, es decir, el valor mínimo de similitud requerido para pertenecer al conjunto de alineaciones. Este umbral de similitud puede ser ajustado de acuerdo a las características y necesidades de cada aplicación. Un umbral de similitud óptimo debe ser lo suficientemente alto como para no considerar equivalentes a las entidades que no lo son (exactitud), y lo suficientemente bajo para obtener todas las correspondencias existentes (completitud). En esta tesis el umbral de similitud se aprende mediante un algoritmo genético cuya función objetivo (*fitness*) es maximizar la medida *F1*, que es la media armónica entre la *Precisión* y el *Recall* sobre un conjunto de ontologías mapeadas, que ha sido particionado utilizando el *método de validación cruzada de 10 iteraciones* (*10-Fold Cross-Validation*) [Refaeilzadeh *et. al*, 2008]. Las medidas de evaluación (*Precisión*, *Recall* y *F1*) se detallan en el capítulo 5, donde se presentan los resultados experimentales del trabajo. El operador de cruce utilizado fue el estándar de dos puntos y el mecanismo de selección, elitista [Davis, 1991]. Finalmente el umbral con el que se obtuvo el mayor valor de *F1* para el conjunto de datos analizados fue 0.88. Los parámetros de entrada del algoritmo fueron:

- *Tamaño de la población*: Número de individuos (cromosomas) para el algoritmo genético = 100.

- *Número de evaluaciones*: Número máximo de llamadas a la función de evaluación con el objetivo de detener la búsqueda = 1000.

- *Probabilidad de cruce*: Probabilidad de aplicar el operador genético de cruce sobre una pareja de cromosomas = 0.6.

- *Probabilidad de mutación*: Probabilidad de aplicar el operador genético de mutación sobre un cromosoma = 0.01

4.3.4.2 Construcción de las alineaciones finales

Las reglas de correspondencia se construyen por medio del *API* de java *Alignment Format* [Euzenat y Shvaiko, 2007] utilizado por la iniciativa para la alineación de ontologías (*OAEI*). El *API Alignment Format* permite la comparación de varios métodos de alineación de ontologías, así como la generación de las salidas en varios formatos como por ejemplo *SWRL*, *OWL* y *RDF*, entre otros. La Figuras 4.17 y 4.18 muestran un fragmento del fichero de alineación generado en uno de los experimentos realizados en la etapa de evaluación del sistema.

```
<ruleml:imp>
    <ruleml:_body>
      <swrl:datavaluedPropertyAtom
               swrlx:property="http://cmt#email"/>
        <ruleml:var>x</ruleml:var>
        <ruleml:var>y</ruleml:var>
      <swrl:datavaluedPropertyAtom>
    </ruleml:_body>
    <ruleml:_head>
      <swrl:datavaluedPropertyAtom
             swrlx:property="http://confOf#hasEmail"/>
        <ruleml:var>x</ruleml:var>
        <ruleml:var>y</ruleml:var>
      </swrl:datavaluedPropertyAtom>
    </ruleml:_head>
  </ruleml:imp>
 <ruleml:imp>
    <ruleml:_body>
      <swrl:classAtom>
        <owllx:Class
             owllx:name="http://cmt#Conference"/>
        <ruleml:var>x</ruleml:var>
      </swrl:classAtom>
    </ruleml:_body>
    <ruleml:_head>
      <swrlx:classAtom>
        <owllx:Class
              owllx:name="http://confOf#Conference"/>
        <ruleml:var>x</ruleml:var>
     </swrl:classAtom>
    </ruleml:_head>
  </ruleml:imp>
```

Figura 4.17. Fragmento de fichero de alineación compuesto por reglas SWRL

En el ejemplo de la Figura 4.17 hemos utilizado el formato *SWRL* para la expresar las alineaciones resultantes, y en el de la Figura 4.18 hemos utilizado directamente el formato *Alignment Format*. En ambas figuras se presentan dos reglas de correspondencia entre las

ontologías del dominio de la organización de conferencias `cmt` y `confOf`: la primera es entre las propiedades `email` de `cmt` y `hasEmail` de `confOf`; y la segunda regla es entre dos clases llamadas `Conference` en las dos ontologías.

```
<Alignment>
 <xml>yes</xml>
 <level>0</level>
 <type>**</type>
 <onto1>http://…/ontology1</onto1>
 <onto2>http://…/ontology2</onto2>
  <map>
    <Cell>
     <entity1 rdf:resource='http://cmt#email'/>
     <entity2 rdf:resource='http://confOf#hasEmail'/>
     <measure rdf:datatype='&xsd;float'>0.91</measure>
     <relation>=</relation>
    </Cell>
  </map>
  <map>
    <Cell>
     <entity1 rdf:resource='http://cmt#Conference'/>
     <entity2 rdf:resource='http://confOf#Conference'/>
     <measure rdf:datatype='&xsd;float'>1.0</measure>
     <relation>=</relation>
   </Cell>
  </map>
</Alignment>
```

Figura 4.18. Fragmento de fichero de alineación utilizando Alignment Format

4.3.5 Visión global de las Capas del Sistema

Cada una de las capas del sistema basado en reglas difusas realiza una tarea dirigida a mejorar el cálculo de la similitud. A medida que ascendemos en la arquitectura de capas, los valores de similitud se van refinando, con ayuda de los diferentes módulos de apoyo hasta obtener los valores finales de similitud entre conceptos y propiedades para finalmente construir el conjunto de alineaciones entre las dos ontologías. En la Figura 4.19 se muestra una representación detallada de la arquitectura, donde se pueden observar todas las entradas y salidas de datos de las diferentes capas y su interacción con los módulos externos. De abajo hacia arriba, el proceso comienza en la capa léxica, que recibe como entrada del módulo léxico-semántico los valores de la similitud por sinonimia, derivación y factor léxico. La similitud lingüística resultante de esta capa, junto con la similitud semántica (calculada con el *coeficiente de Jaccard*) sirven como entrada a la capa terminológica, cuyo resultado final junto con las similitudes que aportan los hijos, padres y hermanos de los conceptos en la taxonomía son utilizados para calcular la similitud jerárquica. La similitud jerárquica obtenida

entre las clases es utilizada como similitud de dominio, que junto con la similitud del rango sirven para calcular las similitudes entre las propiedades. Finalmente la capa superior del sistema utiliza los valores de similitud interna de las propiedades y la cardinalidad para refinar el valor de la similitud jerárquica y obtener el índice de similitud final.

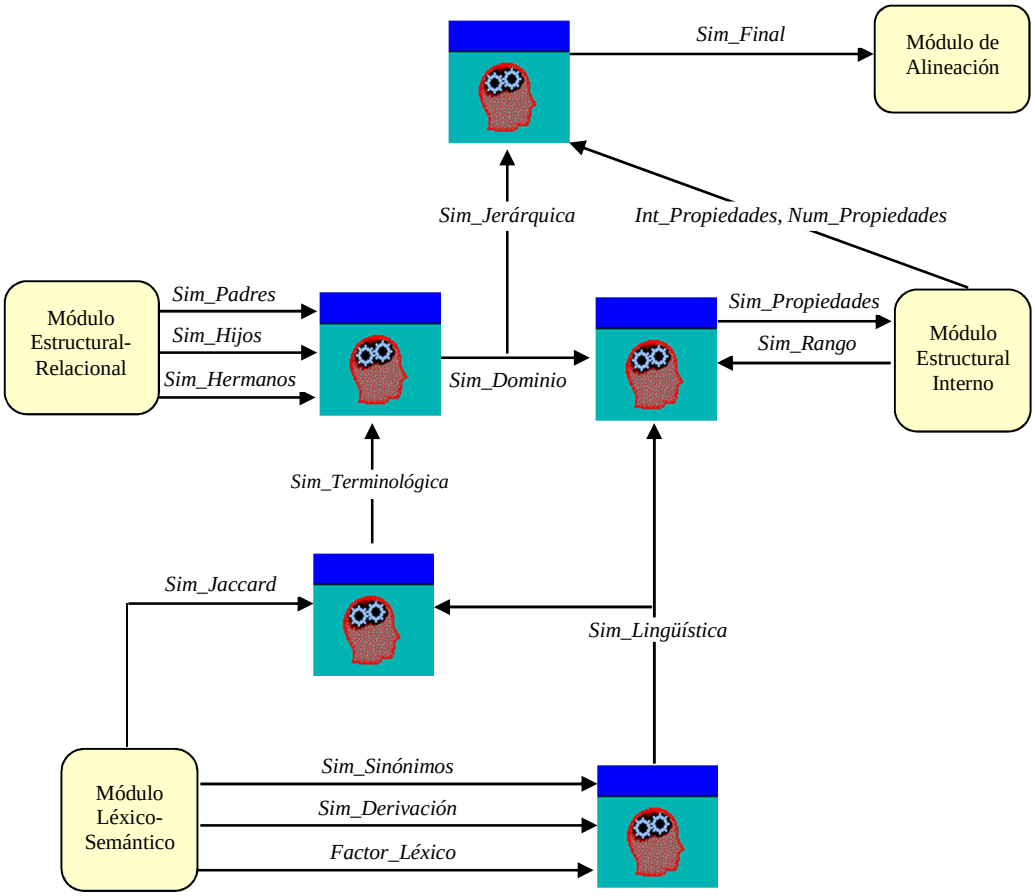


Figura 4.19. Arquitectura global detallada de FuzzyAlign

4.3.6 Diseño del Motor de Inferencias

Para diseñar el motor de inferencias el primer paso consiste en elegir el *operador de implicación* (*I*) de las reglas. Generalmente los operadores de implicación son los derivados de la conjunción lógica, entre los que se encuentran las *t-normas* y *t-conormas*. Los operadores de implicación más utilizados son la *t-norma* del mínimo, la del producto algebraico y la

función de implicación de Lukasiewicz [Mizumoto y Zimmermann, 1982]. En nuestro trabajo utilizamos como operador de implicación la *t-norma* del mínimo. El *operador de conjunción* (*T*) para los antecedentes de las reglas por lo general es una *t-norma*, por lo que al igual que en el caso del operador de implicación, en este trabajo utilizamos la *t-norma* del mínimo. Para la *interfaz de defuzzificación* hemos elegido el modo *FATI* (*First Aggregate, Then Infer*), utilizando la *t-conorma* del máximo como *operador de agregación* (*G*), y el mayor de los máximos (*LOM*) como *método de defuzzificación*.

4.3.7 Bases de Reglas

Para la deducción de las bases de reglas de cada capa del sistema difuso hemos utilizado un algoritmo genético para el aprendizaje de bases de reglas difusas llamado *THRIFT* [Thrift, 1991]. A continuación explicamos brevemente el algoritmo *THRIFT*, su funcionamiento y la configuración que hemos utilizado de sus parámetros de entrada para generar las bases de reglas del sistema.

4.3.7.1 Aplicación del algoritmo THRIFT para el aprendizaje de las bases de reglas

El algoritmo *THRIFT* [Thrift, 1991] se enmarca dentro del grupo de algoritmos que siguen el enfoque de *Pittsburgh* y está diseñado para aprender bases de reglas del tipo *Mamdani*. Trabaja sobre la base de un fenotipo constituido por una matriz de decisión completa. Una matriz de decisión difusa representa un caso especial de relación real definida sobre las colecciones de conjuntos difusos correspondientes a las variables de entrada y salida del sistema.

Las matrices de decisión se codifican en los cromosomas utilizando un código de posición y estableciendo una correspondencia entre el conjunto de etiquetas lingüísticas asociado a la variable de salida del sistema y un conjunto creciente de números enteros que representa el conjunto de genes y que comienza con el valor *0*. Los primeros *N* elementos de este conjunto ($\{0, .., N-1\}$) representan los *N* elementos del conjunto de términos de una variable de entrada, mientras que el último elemento juega el rol de elemento *nulo* e indica la ausencia de valor para la variable de salida. Cada cromosoma se constituye concatenando el entero asociado a cada una de las etiquetas lingüísticas contenidas en las celdas de la matriz de decisión. La

Tabla 4.1 es un ejemplo de matriz de decisión difusa para un sistema con dos variables de entrada (X_1 y X_2) y una de salida (Y).

Tabla 4.1. Matriz de decisión difusa para un sistema con dos variables de entrada y una de salida.

	A_{21}	A_{22}	A_{23}
A_{11}		B_1	B_2
A_{12}	B_1	B_2	B_3
A_{13}	B_1	B_3	B_4

Las etiquetas lingüísticas asociadas a la variable X_1 son: A_{11}, A_{12}, A_{13}, las etiquetas asociadas a la variable X_2 son: A_{21}, A_{22} y A_{23} y las etiquetas asociadas a la variable de salida Y son: B_1, B_2, B_3, B_4. El cromosoma resultante de codificar esta base de reglas sería: 012123134.

En la implementación del algoritmo *THRIFT* empleamos el esquema de *selección por rango* propuesto por Baker [Baker, 1987]. El operador de cruce es el estándar de dos puntos [Davis, 1991], mientras que el operador de mutación consiste en cambiar el código de la etiqueta difusa del gen mutado incrementándolo en una unidad, decrementándolo en una unidad, o sustituyéndolo por el valor nulo. Cuando el valor del gen es nulo, se selecciona un nuevo valor no nulo de forma aleatoria. La función objetivo utilizada es el *Error Cuadrático Medio* (*ECM*), por lo que los mejores individuos serán aquellas bases de reglas que minimicen esta función. El *ECM* se define como:

$$ECM(F) = \sqrt{\frac{\sum_{i=1}^{N} (F(x_i) - y_i)^2}{N}}, \qquad 4.15$$

donde F es el sistema difuso, N es el tamaño del conjunto de datos, y el par (x_i , y_i) representa la *i-ésima* entrada-salida del conjunto de datos.

Para el aprendizaje de las bases de reglas de las distintas capas del sistema difuso el algoritmo *THRIFT* fue ejecutado con distintos parámetros de configuración. De estos parámetros los más importantes son el tamaño de la población, para el cuál utilizamos el criterio de Alander [Alander, 1992] que sugiere un tamaño de población comprendido entre l y $2l$ para un cromosoma de tamaño l y la probabilidad de mutación, para la cuál utilizamos el criterio de [De Jong, 1975] que recomienda utilizar una probabilidad de mutación de l^{-1}. En el

caso de la probabilidad de cruce y el número de generaciones (o evaluaciones) hemos elegido los mismos valores para el aprendizaje de cada base de reglas. Los valores de los parámetros de entrada utilizados para la ejecución del algoritmo genético *THRIFT* para el aprendizaje de las bases de reglas de las distintas capas del sistema fueron los siguientes:

Capa Léxica:

- *Tamaño de la población*: Número de individuos (cromosomas) para el algoritmo genético = 125
- *Número de evaluaciones*: Número máximo de llamadas a la función de evaluación con el objetivo de detener la búsqueda = 1000
- *Probabilidad de cruce*: Probabilidad de aplicar el operador genético de cruce sobre una pareja de cromosomas = 0.6
- *Probabilidad de mutación*: Probabilidad de aplicar el operador genético de mutación sobre un cromosoma = 0.008

Capa terminológica:

- *Tamaño de la población*: Número de individuos (cromosomas) para el algoritmo genético = 50
- *Número de evaluaciones*: Número máximo de llamadas a la función de evaluación con el objetivo de detener la búsqueda = 1000
- *Probabilidad de cruce*: Probabilidad de aplicar el operador genético de cruce sobre una pareja de cromosomas = 0.6
- *Probabilidad de mutación*: Probabilidad de aplicar el operador genético de mutación sobre un cromosoma = 0.02

Similitud jerárquica:

- *Tamaño de la población*: Número de individuos (cromosomas) para el algoritmo genético = 2400

- *Número de evaluaciones*: Número máximo de llamadas a la función de evaluación con el objetivo de detener la búsqueda = 1000

- *Probabilidad de cruce*: Probabilidad de aplicar el operador genético de cruce sobre una pareja de cromosomas = 0.6

- *Probabilidad de mutación*: Probabilidad de aplicar el operador genético de mutación sobre un cromosoma = 4.1 e^{-4}

Similitud entre propiedades:

- *Tamaño de la población*: Número de individuos (cromosomas) para el algoritmo genético = 70

- *Número de evaluaciones*: Número máximo de llamadas a la función de evaluación con el objetivo de detener la búsqueda = 1000

- *Probabilidad de cruce*: Probabilidad de aplicar el operador genético de cruce sobre una pareja de cromosomas = 0.6

- *Probabilidad de mutación*: Probabilidad de aplicar el operador genético de mutación sobre un cromosoma = 0.01

Al igual que en el caso del aprendizaje del umbral de similitud (Sección 4.3.4.1), la base de datos utilizada para el proceso de aprendizaje contiene información de 40 ontologías mapeadas y ha sido particionada utilizando el *método de validación cruzada de 10 iteraciones* (*10-Fold Cross-Validation*) [Refaeilzadeh *et. al*, 2008], donde los datos de muestra se dividen en 10 subconjuntos. Uno de los subconjuntos se utiliza como conjunto de prueba y el resto como conjuntos de entrenamiento. El proceso se repite durante 10 iteraciones, con cada uno de los posibles subconjuntos de datos.

Como resultado del proceso de aprendizaje a través del algoritmo genético se han obtenido las bases de reglas óptimas para el sistema basado en reglas difusas. La Tabla 4.2 muestra la base de reglas de la capa terminológica. El resto de las bases de reglas no se muestran debido a que contienen más de dos variables de entrada, y esto trae como consecuencia un incremento notable del número de reglas y la imposibilidad de representarlas en forma tabular.

Tabla 4.2. Base de reglas obtenida por el algoritmo THRIFT para la capa Terminológica.

Similitud Jaccard	***Similitud Lingüística***				
	B	***R***	***M***	***A***	***MA***
B	MB	B	MedB	R	MedA
R	B	MedB	R	MedA	A
M	B	MedB	R	MedA	A
A	MedB	R	MedA	A	MA
MA	MedB	MedA	A	A	MA

4.4 Resumen y consideraciones finales

En este capítulo se explica el sistema basado en reglas difusas propuesto en la tesis para la alineación de ontologías. La arquitectura del sistema está constituida por cuatro capas, cada una de las cuáles desarrolla una medida de similitud que sirve como entrada a la capa superior. La primera capa es la capa léxica y es la encargada del calcular la similitud lingüística. Esta similitud se calcula a partir de las similitudes de los sinónimos, las palabras derivadas y las distancias léxicas de las entidades, utilizando herramientas externas como *WordNet* en caso de ontologías con terminología de dominio general y *UMLS* en el caso de ontologías específicas del dominio de la medicina. La similitud lingüística y la similitud semántica calculada a través del *coeficiente de Jaccard* sirven como variables de entrada a la capa terminológica, que es la encargada de combinar ambas similitudes para obtener la similitud entre los nombres de las entidades.

Posteriormente, esta similitud terminológica es enriquecida en la capa estructural teniendo en cuenta los valores de similitud jerárquica que aportan los hijos, padres y hermanos de los conceptos en las taxonomías. En la capa estructural también se calculan las similitudes entre las propiedades de los conceptos de manera independiente, que luego son utilizadas en la capa de alineación para obtener el índice de similitud final entre los conceptos. Las bases de reglas del sistema fueron deducidas utilizando el algoritmo genético *THRIFT* para el aprendizaje de bases de reglas difusas de tipo *Mamdani* siguiendo el enfoque de *Pittsburgh*. En el siguiente capítulo presentamos los experimentos realizados para la evaluación de los resultados obtenidos por el sistema.

CAPÍTULO 5. EXPERIMENTOS Y EVALUACIÓN

No importa lo bonita que sea tu teoría, no importa lo listo que seas. Lo que no se demuestra con experimentos, es falso.

Richard P. Feynman

En este capítulo detallamos los experimentos realizados para evaluar el método de alineación de ontologías. Comenzamos explicando brevemente las medidas utilizadas para evaluar los sistemas de alineación de ontologías que son la *Precisión* y el *Recall*. Luego detallamos la parte experimental que hemos dividido en tres fases. En la primera fase se ha evaluado el sistema en fragmentos pequeños de taxonomías reales, la segunda fase consiste en realizar las pruebas de propuestas por la Iniciativa para la Evaluación de la Alineación de Ontologías (*OAEI*) y en la tercera fase se evalúa el método en ontologías de dominio para las cuáles no existe directorio especializado, como por ejemplo las redes de sensores.

5.1 Medidas de Evaluación

Las medidas de evaluación más utilizadas para los sistemas de alineación de ontologías proceden del área de la recuperación de información y la clasificación de documentos, como la *Precision*, el *Recall* y la *Medida-F*. Estas medidas están en función de dos conjuntos de datos: el conjunto de elementos relevantes (positivos) y el conjunto total de elementos recuperados (positivos + falsos positivos). En la Figura 5.1 se representan gráficamente estos conjuntos para la alineación de ontologías y a continuación, se definen las tres medidas de evaluación.

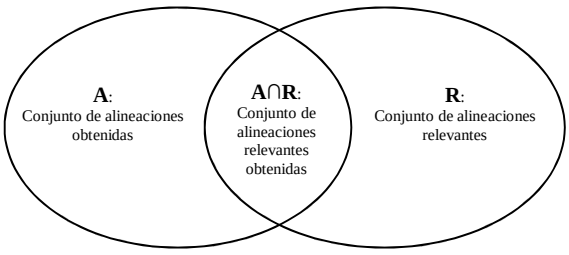


Figura 5.1. Conjuntos relevantes y recuperados por los algoritmos de alineación de ontologías

La Precisión es una medida de la exactitud de las alineaciones. Se calcula como la fracción entre el número de alineaciones relevantes y la totalidad de alineaciones obtenidas por el algoritmo [Rijsbergen, 1979]. Dado un conjunto *R* de alineaciones de referencia, la precisión de un conjunto de alineaciones *A* está dada por:

$$Precision(A,R) = \frac{|R \cap A|}{|A|} \qquad 5.1$$

El Recall es una medida de la exhaustividad de las alineaciones. Se calcula como la fracción entre el número de alineaciones relevantes obtenidas y el número de alineaciones que pertenecen realmente al conjunto de alineaciones relevantes [Rijsbergen, 1979]. Dado un conjunto de alineaciones de referencia *R*, el *Recall* del conjunto de alineaciones *A* está dado por:

$$Recall(A,R) = \frac{|R \cap A|}{|R|} \qquad 5.2$$

La Medida-F es usada para combinar los resultados de la *Precisión* y el *Recall*. Dado un conjunto *R* de alineaciones de referencia, la *Medida-F* de un conjunto de alineaciones *A* está dada por:

$$F_{\alpha}(A,R) = \frac{Precision(A,R) \cdot Recall(A,R)}{(1-\alpha) \cdot Precision(A,R) + \alpha \cdot Recall(A,R)} \qquad 5.3$$

Donde α es un coeficiente entre 0 y 1, que representa el peso que se le confiere a la *Precisión* y al *Recall*. Un valor mayor de α significa que se le da mayor importancia a la *Precisión* con respecto al *Recall*, mientras que un valor menor significa que tiene más peso el *Recall*. En la mayoría de los sistemas se requiere un equilibrio entre la *Precisión* y el *Recall*, por lo que se suele utilizar un α=0.5, que se corresponde con la media armónica entre la *Precisión* y el *Recall*. A esta medida se le llama *F1* y se define como:

$$F1(A,R) = 2\frac{Precision(A,R) \cdot Recall(A,R)}{Precision(A,R) + Recall(A,R)} \qquad 5.4$$

La media armónica es útil para evaluar el equilibrio entre la *Precision* y el *Recall* porque resulta poco influida por la existencia de determinados valores mucho mayores que los demás, siendo en cambio sensible a valores mucho más pequeños que los demás dentro del conjunto.

5.2 Experimentos

La evaluación del algoritmo de alineación de ontologías propuesto se ha realizado a través de un conjunto de experimentos organizados en tres fases. En la primera fase se ha probado el sistema de alineación en pequeños fragmentos de taxonomías reales para obtener alineaciones únicamente teniendo en cuenta la similitud terminológica, la segunda fase consiste en alinear ontologías completas teniendo en cuenta la terminología y la estructura, utilizando las pruebas de evaluación propuestos por la *OAEI*. En la tercera fase se aplica el método propuesto en ontologías del dominio de las redes de sensores. Todos los experimentos se realizaron en un PC con un procesador Intel core i3-2.0 GHz y 4 GB de RAM.

5.2.1 Primera fase: Fragmentos de taxonomías reales. ACM-DMOZ

Para evaluar los resultados de esta fase hacemos una comparativa con el trabajo de Pan [Pan *et al*., 2005], que propone un método con algunas características similares al nuestro, pero utilizando técnicas probabilísticas. El método de Pan consiste en un marco probabilístico basado en Redes Bayesianas (*RBs*), donde las ontologías son primero traducidas a *RBs* y a continuación, el mapeo de conceptos se realiza como razonamiento evidencial entre las dos *RBs*. Las probabilidades necesarias para el mapeo se obtienen mediante el uso de programas de clasificación de texto asociando distintos conceptos con documentos de texto relevantes recuperados de la Web. Este enfoque sólo tiene en cuenta la probabilidad de aparición de los conceptos en la Web y no su similitud lingüística, por lo que falla en el caso en que dos conceptos muy similares no tengan el mismo índice de popularidad en Internet.

Uno de los experimentos que hemos realizado en esta fase consiste en alinear pequeños fragmentos de las taxonomías ACM[9] y DMOZ[10]. La primera taxonomía es una jerarquía de clases desarrollada por ACM (*Association for Computing Machinery*) para el sistema de clasificación de sus librerías de documentos (CCS). El árbol contiene 4 niveles, formado por 11 nodos de primer nivel y uno o dos niveles de profundidad en cada uno de ellos. Por su parte, el *Open Directory Project* (ODP), también conocido como DMOZ, por su nombre de dominio original (*directory.mozilla.org*) es un proyecto colaborativo multilingüe, en el que

[9] ACM Topic, *http://www.acm.org/about/class/1998/*
[10] DMOZ hierarchie, *http://www.dmoz.org/*

editores voluntarios categorizan enlaces a páginas web. Está dividido en varias taxonomías organizadas jerárquicamente.

De manera similar a [Pan *et al*., 2005] en este experimento hemos seleccionado el tópico "*Artificial Intelligence*" de los directorios ACM y DMOZ, haciendo una poda de algunos conceptos en ambas taxonomías para que exista cierto grado de solapamiento entre ellas con el fin de comprobar la efectividad del método. En esta primera fase sólo se tiene en cuenta la terminología de los nombres de las entidades y no su estructura relacional. La Figura 5.2 muestra los 18 conceptos seleccionados de la taxonomía ACM y sus equivalencias con los 24 conceptos seleccionados en DMOZ. Las alineaciones de referencia que debían obtenerse a priori se muestran en la figura con líneas discontinuas.

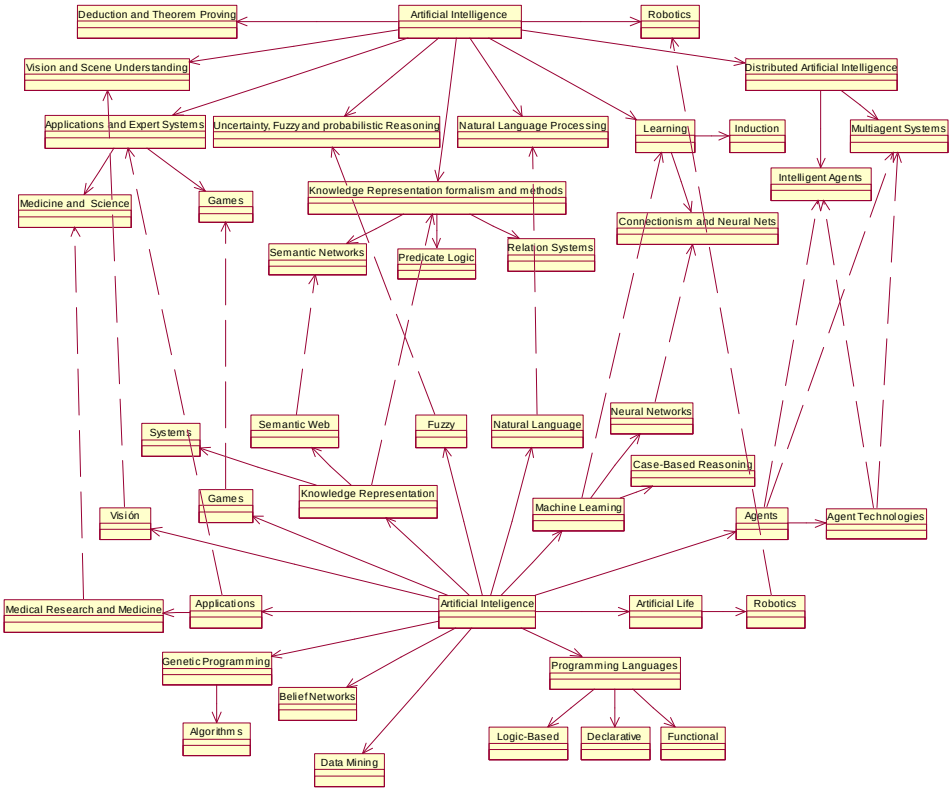


Figura 5.2. Fragmentos de las taxonomías ACM y DMOZ

En la Tabla 5.1 se muestra una comparación entre los valores de similitud obtenidos por los dos métodos para los conceptos seleccionados. En el trabajo de Pan, utilizando fragmentos similares de las taxonomías ACM y DMOZ, plantean que según su método se obtiene un índice de similitud de sólo 0.61 para los conceptos "*connectionism and neural nets*" y "*Neural Networks*" que realmente son equivalentes. Esto se debe a que sólo tienen en cuenta la probabilidad de aparición de ambos conceptos en la Web, y el término "*connectionism*" es muy poco popular. Como se puede observar los resultados de similitud obtenidos por *FuzzyAlign* para los conceptos "*connectionism and neural nets*" y "*Neural Networks*" fueron mejores que los obtenidos por Pan, debido a que además del contexto hemos tenido en cuenta su similitud lingüística.

Tabla 5.1. Comparativa de los valores de similitud más altos obtenidos por FuzzyAlign y el método de Pan aplicados a los fragmentos seleccionados de las taxonomías ACM y DMOZ

Conceptos ACM	*Conceptos DMOZ*	*Similitud-Pan*	*Similitud-FuzzyAlign*
Robotics	Robotics	1.00	1.00
Games	Games	1.00	1.00
Multiagent Systems	Agent Technologies	0.88	0.95
Natural Language Processing	Natural Language	0.90	1.00
Learning	Machine learning	0.92	1.00
Knowledge Representation Formalisms and Methods	Knowledge Representation	0.96	1.00
Intelligent Agents	Agents	0.90	0.95
Vision and Scene Understanding	Vision	0.68	0.92
Uncertainty, fuzzy and probabilistic reasoning	Fuzzy	0.75	0.90
Connectionism and neural nets	**Neural Networks**	**0.61**	**0.90**
Semantic Networks	Semantic Web	0.75	0.88
Multiagent Systems	Agents	0.89	0.90
Applications and Expert Systems	Applications	0.55	0.89
Medicine and Science	Medical Research and Medicine	0.92	0.95
Intelligent Agents	Agent Technologies	0.90	0.90

A pesar de la mejora conseguida en los valores de similitud al aplicar *FuzzyAlign* a estos fragmentos de taxonomías, en esta primera fase obtuvimos algunos valores más elevados de lo que debían en algunos conceptos que lingüísticamente tenían cierto parecido, pero que no eran equivalentes. En la Tabla 5.2 se muestran los conceptos para los cuáles el índice de similitud

obtenido por *FuzzyAlign* fue superior al esperado. Este problema ocurría debido a que en esta fase sólo se tuvo en cuenta la similitud terminológica entre los conceptos, obviando elementos importantes como sus relaciones con otros conceptos. Al incorporar al sistema otras medidas de similitud que tenían en cuenta las relaciones taxonómicas entre los conceptos, así como las similitudes entre sus propiedades eliminamos este problema obteniendo índices de similitud más precisos.

Tabla 5.2. Parejas de conceptos para las cuáles los resultados de similitud de FuzzyAlign fueron más altos de lo esperado.

Conceptos ACM	*Conceptos DMOZ*	*Similitud*
Multiagent Systems	Systems	0.77
Semantic Networks	Neural Networks	0.55
Natural Language Processing	Programming Language	0.55

Finalmente en la Tabla 5.3 se muestra la comparación entre los resultados de la evaluación del método de Pan y *FuzzyAlign* en términos de *Precisión*, *Recall* y *F1*. Como se puede observar, de manera general los valores de *Precisión*, *Recall* y *F1* obtenidos por *FuzzyAlign* fueron mejores.

Tabla 5.3. Comparativa de los resultados de aplicar FuzzyAlign y el método de Pan a los fragmentos seleccionados de las taxonomías ACM y DMOZ

Métodos	*Correspondencias esperadas*	*Correspondencias obtenidas*	*P*	*R*	*F1*
Pan	15	9	1.00	0.60	0.75
FuzzyAlign	15	15	1.00	1.00	1.00

5.2.2 Segunda fase: Pruebas de evaluación de la *OAEI*

La Iniciativa para la Evaluación de la Alineación de Ontologías (*OAEI*) es una organización internacional coordinada para la evaluación y mejora de los métodos de alineación de ontologías. El principal objetivo de la *OAEI* es poder comparar los sistemas y algoritmos en las mismas condiciones y permitir que cualquier usuario pueda sacar conclusiones acerca de las mejores estrategias de correspondencia. Se pretende que a partir de estas evaluaciones, los desarrolladores de herramientas puedan aprender y mejorar sus sistemas. La campaña *OAEI* proporciona una serie de pruebas aprobadas por consenso para la evaluación de los sistemas de alineación de ontologías.

Para evaluar los resultados de los experimentos hemos elegido tres pruebas y cuatro sistemas de alineación de ontologías que han participado en las tres pruebas, comparando sus resultados con los de *FuzzyAlign*. Las pruebas elegidas para realizar la comparativa fueron *OAEI-Benchmark*, *OAEI-Conference* y *OAEI-Anatomy*. La prueba *Benchmark* sirve para evaluar los sistemas de alineación de ontologías de forma general y su comportamiento ante la falta de información ontológica; la prueba *Conference* evalúa los sistemas ante ontologías heterogéneas de la vida real; y la prueba *Anatomy* tiene como objetivo la evaluación en ontologías completas del dominio de la medicina. Los métodos elegidos para realizar las comparaciones fueron: ASMOV [Jean-Mary *et al*., 2009], CODI [Noessner *et al*., 2010] y SOBOM [Xu *et al*., 2010], por estar disponibles para su descarga desde Internet y poder ejecutarse con diferentes configuraciones.

5.2.2.1 Prueba *OAEI-Benchmark*

El objetivo del experimento *Benchmark* [Euzenat *et al*., 2010] es proporcionar una imagen estable y detallada de cada método objeto de evaluación, identificando los puntos fuertes y débiles de cada algoritmo. La prueba se basa en una ontología particular perteneciente al dominio concreto de las referencias bibliográficas y una serie de ontologías alternativas del mismo dominio para las que se proporcionan alineaciones de referencia. La clasificación elegida para este caso es común en el ámbito de la ciencia y se basa en las categorías de publicación.

La prueba sistemática *Benchmark* se basa en una ontología de referencia y muchas variaciones de la misma. Las ontologías se describen en OWL-DL. La ontología de referencia es la de la prueba #101 y contiene 33 clases nombradas, 24 propiedades objetos, 40 propiedades datos, 56 individuos nombrados y 20 individuos anónimos. Esta prueba consiste en alinear la ontología de referencia con todas sus variaciones. Las variaciones se centran en la caracterización del comportamiento de las herramientas en lugar de enfocarse en los problemas de la vida real. El experimento se organiza en tres categorías:

- *Pruebas simples (1xx)*: Consisten en alinear la ontología de referencia con ella misma, luego con una ontología irrelevante perteneciente a otro dominio como por ejemplo la

ontología *wine*[11], y finalmente con la misma ontología en su generalización y restricción del lenguaje (*OWL-Lite*).

- *Pruebas sistemáticas (2xx)*: En estas pruebas se utiliza una ontología nueva descartando algunas características de la ontología de referencia. Su objetivo es evaluar cómo se comporta el algoritmo cuando falta algún tipo particular de información. Según las modificaciones realizadas en las ontologías esta prueba se divide en tres grupos:

    - *#200-210*: En este grupo se descartan o modifican algunas características lingüísticas (nombres y comentarios), manteniendo la estructura de las ontologías. Los nombres y comentarios pueden ser suprimidos (N), reemplazados por cadenas aleatorias (R), reemplazados por sinónimos (S), reemplazados por cadenas en otro idioma que no sea inglés (F), etc.

    - *#221-247*: En este grupo se realizan modificaciones en los elementos que componen la estructura de las ontologías. Entre estas modificaciones están: suprimir (N), expandir (E), o contraer (F) las clases de la jerarquía taxonómica, suprimir las propiedades (N) y suprimir las instancias (N).

    - *#248-266*: En este grupo se encuentran las pruebas más complejas, ya que se realizan modificaciones sustanciales tanto lingüísticas como estructurales.

- *Ontologías reales (3xx)*: En estas pruebas se utilizan íntegramente 4 ontologías reales encontradas en la Web pertenecientes al dominio de las bibliografías sin hacerles cambios. Estas ontologías son: *BibTex/MIT*[12], *BibTex/UMBC*[13], *Karlsruhe*[14] e *INRIA*[15].

En la Tabla 5.4 se muestra una comparativa de los resultados obtenidos por *FuzzyAlign* y el resto de los métodos estudiados en términos de *Precisión*, *Recall* y *F1* en la prueba *Benchmark*. En la tabla se puede observar que en las pruebas simples el rendimiento de los cuatro sistemas que analizamos fue óptimo. En el conjunto de pruebas sistemáticas *201-210*, que realiza cambios lingüísticos, *FuzzyAlign* disminuyó considerablemente el *Recall*, debido a

[11] *http://www.w3.org/TR/2003/PR-owl-guide-20031209/wine.*
[12] *http://oaei.ontologymatching.org/2011/benchmarks/#301*
[13] *http://oaei.ontologymatching.org/2011/benchmarks/#302*
[14] *http://oaei.ontologymatching.org/2011/benchmarks/#303*

que está pensado fundamentalmente para ontologías terminológicas, es decir, que se le da más peso a la lingüística que a la estructura. En este grupo de pruebas donde no existe una terminología coherente, los resultados de la similitud final se vieron afectados. Por este mismo motivo, en el conjunto de pruebas *221-247*, donde se realizaron únicamente cambios estructurales, los resultados de nuestro sistema no se vieron afectados, alcanzando valores de *Precisión* y *Recall* superiores a los demás. Los peores resultados de las pruebas sistemáticas se alcanzaron en el grupo *248-266*, debido que se realizaron cambios más drásticos tanto en la terminología como en la estructura. En las pruebas con ontologías reales es donde nuestro sistema ha mostrado un mejor comportamiento, superando al resto en términos de *Precisión*, *Recall* y *F1*. Esto nos permite llegar a la conclusión de que *FuzzyAlign* funciona mejor en ontologías terminológicas, con construcciones léxicas correctas, por lo que se ve más afectado ante cambios lingüísticos que estructurales. En la Figura 5.3 se muestran las curvas de *Precisión-Recall* de los 4 sistemas en la prueba *Benchmark*.

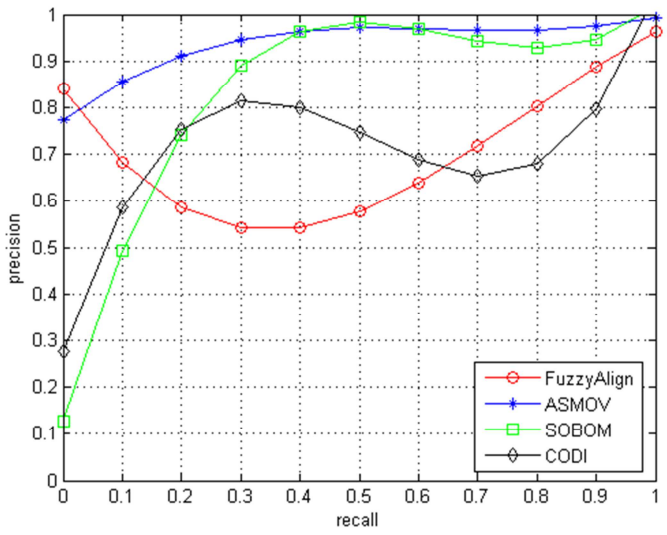


Figura 5.3. Curvas de precisión-recall de los métodos analizados

En la gráfica de las curvas de *Precision* y *Recall* se puede observar que para valores pequeños de *Recall*, con *FuzzyAlign* se obtienen valores de *Precisión* relativamente altos (Este intervalo se corresponde con los casos en que el sistema obtuvo pocas correspondencias debido fundamentalmente a la ausencia de elementos léxicos). Los valores de *Precisión* van

[15] *http://oaei.ontologymatching.org/2011/benchmarks/#304*

disminuyendo a medida que aumenta el *Recall* hasta equilibrarse en los valores medios, donde comienza a incrementar la *Precisión* a medida que aumenta el *Recall*, hasta alcanzar valores altos de ambos indicadores. En el caso de CODI y SOBOM, para valores pequeños de *Recall* se obtiene también una baja *Precisión*, que se va incrementando en ambos sistemas a medida que aumenta el *Recall*. En el caso de SOBOM, la *Precisión* continúa aumentando con el *Recall* hasta equilibrarse, mientras que en CODI la *Precisión* disminuye para valores medios de *Recall* y finalmente vuelve a crecer cuando el *Recall* alcanza sus valores más elevados. Como se observa en la gráfica, el sistema que tuvo mejor comportamiento en este test fue ASMOV, que comienza con valores altos de *Precisión* y se mantiene en valores relativamente constantes a medida que aumenta el *Recall*.

En la Tabla 5.5 se muestran en detalle todos los resultados obtenidos por *FuzzyAlign* en la prueba *OAEI-Benchmark.*

Tabla 5.4. Resultados de los métodos estudiados en la prueba Benchmark en términos de Precisión, Recall y F1

Prueba	*1xx*			*201-210*			*221-247*			*248-266*			*3xx*		
Sistema	*P*	*R*	*F1*	*P*	*R*	*F1*	*P*	*R*	*F1*	*P*	*R*	*F1*	*P*	*R*	*F1*
ASMOV	1.00	1.00	1.00	**0.99**	**0.98**	**0.99**	0.98	0.87	0.92	**0.93**	**0.47**	**0.60**	0.88	0.84	0.86
CODI	1.00	0.99	1.00	0.69	0.38	0.49	0.81	0.45	0.58	0.59	0.42	0.49	0.92	0.43	0.56
SOBOM	1.00	1.00	1.00	0.99	0.85	0.91	0.99	0.99	0.99	0.87	0.57	0.69	0.77	0.70	0.73
FuzzyAlign	1.00	1.00	1.00	0.72	0.50	0.59	**1.00**	**1.00**	**1.00**	0.68	0.38	0.49	**0.93**	**0.84**	**0.88**

Tabla 5.5. Resultados de FuzzyAlign en la prueba Benchmark en términos de Precisión, Recall y F1

test	*Nom.*	*Com.*	*Tax.*	*Inst.*	*Prop.*	*Clas*	*Comentario*	*P*	*R*	*F1*
101							Se compara la ontología ella misma.	1.00	1.00	1.00
102							Se compara la ontología con una ontología irrelevante (*wine*).	-	-	-
103							Se compara la ontología con su generalización en OWL Lite (se eliminan las relaciones owl:unionOf y owl:TransitiveProperty.	1.00	1.00	1.00
104							Se compara la ontología con su resticción en OWL Lite (las restricciones no disponibles se descartan).	1.00	1.00	1.00
201	R						Se sustituyen los nombres por cadenas aleatorias.	0.55	0.16	0.25

test	*Nom.*	*Com.*	*Tax.*	*Inst.*	*Prop.*	*Clas*	*Comentario*	*P*	*R*	*F1*
201-2	R							0.46	0.34	0.39
201-4	R							0.45	0.37	0.41
201-6	R							0.45	0.36	0.40
201-8	R							0.43	0.37	0.40
202	R	N					Se sustituyen los nombres por cadenas aleatorias y los comentarios se eliminan.	1.00	0.01	0.02
202-2								0.46	0.24	0.32
202-4								0.45	0.27	0.34
202-6								0.45	0.26	0.33
202-8		N						0.43	0.17	0.24
203	C						Sin comentarios.	1.00	1.00	1.00
204	S						Diferentes convenciones de nombres (mayúsculas, guiones y otros caracteres especiales).	0.99	0.99	0.99
205	F	F					Los nombres se reemplazan por sinónimos.	0.99	0.98	0.98
206	F						La ontología completa se traduce al idioma francés.	0.98	0.99	0.98
207	C	N					Cada etiqueta o identificador se traduce al idioma francés.	0.99	0.27	0.42
208	S	N					Se suprimen los comentarios y se utilizan caracteres especiales en las etiquetas.	0.98	0.99	0.98
209	F	N					Se suprimen los comentarios y se utilizan sinónimos en las etiquetas.	0.99	0.98	0.98
210			N				Se suprimen los comentarios y se utilizan etiquetas en francés.	0.98	0.27	0.42
221			F				Sin especialización (Se eliminan todas las subclases).	1.00	1.00	1.00
222			E				Se contrae la jerarquía eliminando clases intermedias.	0.99	1.00	0.99
223				N			Se expande la jerarquía introduciendo clases intermedias.	1.00	1.00	1.00
224					R		Se eliminan todos los individuos.	1.00	1.00	1.00
225					N		Se eliminan todas las restricciones locales en las propiedades.	1.00	1.00	1.00
228						F	Se eliminan todas las propiedades.	1.00	1.00	1.00
230						E	Las clases se contraen.	0.98	1.00	0.99
231			N	N			Las clases de expanden (Se añaden clases nuevas).	1.00	1.00	1.00
232			N		N		Se eliminan las instancias y la jerarquía taxonómica.	1.00	1.00	1.00

test	*Nom.*	*Com.*	*Tax.*	*Inst.*	*Prop.*	*Clas*	*Comentario*	*P*	*R*	*F1*
233				N	N		Se eliminan la jerarquía taxonómica y las propiedades.	1.00	1.00	1.00
236			F	N			Se eliminan las instancias y las propiedades.	1.00	1.00	1.00
237			E	N			Se contrae la jerarquía (eliminando clases) y se eliminan las instancias.	0.98	1.00	0.99
238			F		N		Se expande la jerarquía (insertando clases) y se eliminan las instancias	1.00	1.00	1.00
239			E		N		Se contrae la jerarquía (eliminando clases) y se eliminan las propiedades.	0.99	1.00	0.99
240			N	N	N		Se expande la jerarquía (insertando clases) y se eliminan las propiedades.	0.99	1.00	0.99
241			F	N	N		Se eliminan la jerarquía, las instancias y las propiedades.	1.00	1.00	1.00
246			E	N	N		Se contrae la jerarquía (eliminando clases) y se eliminan las propiedades y las instancias.	0.97	1.00	0.98
247	N	N	N				Se expande la jerarquía (insertando clases) y se eliminan las propiedades y las instancias.	0.97	1.00	0.98
248							Se eliminan las etiquetas, los comentarios y la jerarquía.	1.00	0.01	0.02
248-2								0.52	0.28	0.36
248-4								0.51	0.35	0.42
248-6	N	N		N				0.39	0.15	0.22
248-8								0.46	0.19	0.27
249							Se eliminan las etiquetas, los comentarios y las instancias.	1.00	0.01	0.02
249-2								0.84	0.44	0.58
249-4	N	N			N			0.66	0.34	0.45
249-6								0.52	0.27	0.36
249-8								0.52	0.18	0.27
250							Se eliminan las etiquetas, los comentarios y las propiedades.	1.00	0.01	0.02
250-2	N	N	F					0.78	0.78	0.78
250-4								0.63	0.61	0.62
250-6								0.63	0.42	0.50
250-8								0.58	0.21	0.31
251	N	N	E				Se eliminan las etiquetas, los comentarios y la jerarquía se contrae.	1.00	0.01	0.02
251-2								0.79	0.53	0.63
251-4								0.65	0.33	0.44

test	*Nom.*	*Com.*	*Tax.*	*Inst.*	*Prop.*	*Clas*	*Comentario*	*P*	*R*	*F1*
251-6								0.46	0.25	0.32
251-8	N	N	N	N				0.48	0.27	0.35
252							Se eliminan las etiquetas, los comentarios y la jerarquía se expande.	1.00	0.01	0.02
252-2								0.84	0.54	0.66
252-4								0.84	0.54	0.66
252-6	N	N	N		N			0.84	0.54	0.66
252-8								0.84	0.54	0.66
253							Se eliminan las etiquetas, los comentarios, la jerarquía y las instancias.	1.00	0.01	0.02
253-2								0.41	0.84	0.55
253-4	N	N		N	N			0.39	0.84	0.53
253-6								0.37	0.84	0.51
253-8								0.35	0.84	0.49
254							Se eliminan las etiquetas, los comentarios, la jerarquía y las propiedades.	1.00	0.01	0.02
254-2	N	N	F	N				0.48	0.78	0.59
254-4								0.48	0.31	0.38
254-6								0.52	0.32	0.40
254-8								0.41	0.11	0.17
257	N	N	E	N			Se eliminan las etiquetas, los comentarios, las instancias y las propiedades.	1.00	0.01	0.02
257-2								0.78	0.78	0.78
257-4								0.71	0.61	0.66
257-6								0.68	0.42	0.52
257-8	N	N	F		N			0.58	0.21	0.31
258							Se eliminan las etiquetas, los comentarios, las instancias y se contrae la jerarquía.	1.00	0.01	0.02
258-2								0.79	0.53	0.63
258-4								0.61	0.33	0.43
258-6	N	N	E		N			0.45	0.36	0.40
258-8								0.51	0.28	0.36
259							Se eliminan las etiquetas, los comentarios, las instancias y se expande la jerarquía.	1.00	0.01	0.02
259-2								0.85	0.65	0.74

test	*Nom.*	*Com.*	*Tax.*	*Inst.*	*Prop.*	*Clas*	*Comentario*	*P*	*R*	*F1*
259-4	N	N	N	N	N			0.86	0.66	0.75
259-6								0.84	0.64	0.73
259-8								0.85	0.65	0.74
260							Se eliminan las etiquetas, los comentarios, las propiedades y se contrae la jerarquía.	1.00	0.01	0.02
260-2								0.69	0.79	0.74
260-4								0.55	0.62	0.58
260-6								0.36	0.21	0.27
260-8								0.31	0.24	0.27
261							Se eliminan las etiquetas, los comentarios, las propiedades y se expande la jerarquía.	1.00	0.01	0.02
261-2								0.76	0.79	0.77
261-4								0.76	0.79	0.77
261-6								0.76	0.79	0.77
261-8								0.76	0.79	0.77
262							Se eliminan las etiquetas, los comentarios, las propiedades, las instancias y la jerarquía.	1.00	0.01	0.02
262-2								0.38	0.48	0.42
262-4								0.41	0.38	0.39
262-6								0.39	0.31	0.35
262-8								0.33	0.21	0.26
265								0.00	0.00	0.00
266								0.00	0.00	0.00
301							Real: BibTeX/MIT	0.96	0.88	0.92
302							Real: BibTeX/UMBC	0.93	0.91	0.92
303							Real: Karlsruhe	0.88	0.67	0.76
304							Real: INRIA	0.94	0.89	0.91

5.2.2.2 Prueba *OAEI-Conference*

La prueba *Conference*[16] [Euzenat *et al.*, 2010] contiene algunas ontologías reales que han sido elegidas por su origen heterogéneo. El objetivo de este experimento es encontrar todas las

[16] *http://oaei.ontologymatching.org/2012/conference/index.html*

correspondencias correctas dentro de una colección de ontologías que describen el dominio de la organización de conferencias. El experimento consta de siete ontologías: *Cmt*, *Conference*, *Confof*, *Edas*, *Ekaw*, *Iasted* y *Sigkdd.* Estas siete ontologías se combinan dos a dos, para hacer un total de 21 ejecuciones del algoritmo, por lo que contamos con 21 alineaciones de referencia. La Tabla 5.6 muestra los resultados obtenidos al aplicar *FuzzyAlign* a las ontologías de la prueba *OAEI-Conference*.

Tabla 5.6. Resultados de FuzzyAlign en la prueba Conference

	Conf.			*confof*			*edas*			*ekaw*			*iasted*			*sigkdd*		
	P	*R*	*F*	*P*	*R*	*F*	*P*	*R*	*F*	*P*	*R*	*F*	*P*	*R*	*F*	*P*	*R*	*F*
cmt	0.85	0.90	0.87	1.00	0.29	0.45	0.87	0.85	0.86	0.84	0.92	0.88	0.77	1.00	0.87	1.00	0.44	0.61
Conf.				0.75	1.00	0.86	0.81	0.97	0.88	0.78	0.99	0.87	0.84	0.95	0.89	1.00	0.44	0.80
confof							1.00	0.42	0.59	1.00	0.35	0.52	0.99	0.55	0.71	0.94	0.73	0.82
edas										1.00	0.28	0.44	0.90	0.79	0.84	0.99	0.49	0.66
ekaw													0.77	1.00	0.87	0.98	0.59	0.74
iasted																0.98	0.63	0.77

La Tabla 5.7 muestra la comparación entre los resultados obtenidos en la prueba *Conference* por los cuatro sistemas que estamos estudiando. Esta comparación se realiza teniendo en cuenta los valores obtenidos de *Precisión*, *Recall*, *F1* y tiempo de ejecución con el mismo umbral de selección para las alineaciones. En la tabla podemos observar que con un umbral de alineación de 0.88, *FuzzyAlign* alcanzó valores de *Precisión*, *Recall* y *F1* muy superiores a los demás. En cuanto al tiempo de ejecución, el sistema más rápido fue ASMOV, y el resto de los sistemas tuvieron un tiempo de ejecución comprendido entre los 2 y los 5 min.

Tabla 5.7. Resultados de los métodos estudiados en la prueba Conference en términos de Precisión, Recall, F1 y promedios de tiempos de ejecución

Sistema	*P*	*R*	*F1*	*Promedio T (min)*
ASMOV	0.60	0.41	0.48	0.5
CODI	0.86	0.48	0.62	4
SOBOM	0.63	0.36	0.46	5
FuzzyAlign	**0.90**	**0.60**	**0.72**	**2**

La Figura 5.4 muestra las curvas de *Precisión-Recall* de los cuatro métodos analizados. En la gráfica se observa claramente que en el test *Conference*, los cuatro sistemas alcanzaron valores de *Precision* que disminuyen a medida que aumenta el *Recall*. *FuzzyAlign* fue el que

tuvo mejor comportamiento en este test, ya que se mantuvo con los valores más elevados de *Precisión* y *Recall*.

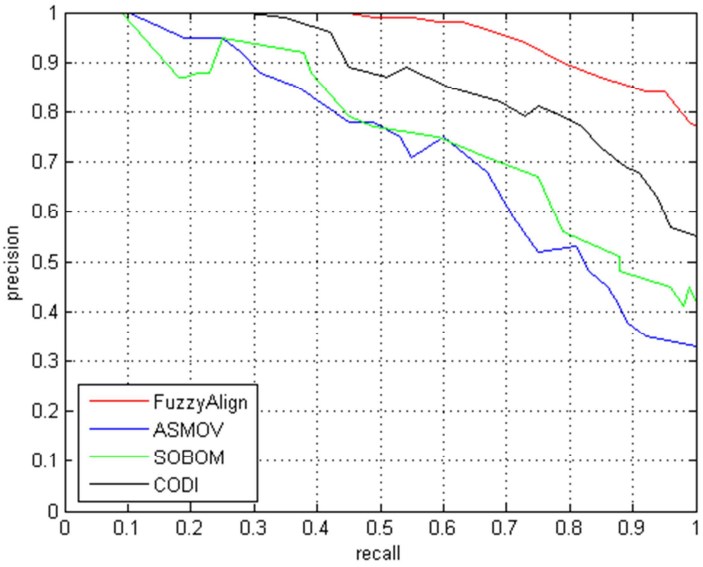


Figura 5.4. Curvas de Precisión-Recall de los métodos analizados en el test Conference

5.2.2.3 Prueba *OAEI-Anatomy*

La prueba *Anatomy*[17] consiste en buscar correspondencias entre dos ontologías reales [Euzenat *et al*., 2010]. La primera ontología tiene 2744 clases y describe la anatomía de los ratones adultos, mientras que la segunda es el tesauro NCI que describe la anatomía humana y tiene 3304 clases. En esta prueba hemos desarrollado tres tareas: la tarea #1 que consiste en alinear las ontologías maximizando la medida *F1*, la tarea #2 que prioriza la *Precisión*, y la tarea #3 que prioriza el *Recall*. La configuración de los parámetros de *FuzzyAlign* para ejecutar cada tarea fue la siguiente:

- *Tarea #1*: Se obtienen las alineaciones óptimas entre las ontologías utilizando un umbral de alineación de 0.88.

[17] *http://oaei.ontologymatching.org/2012/anatomy/index.html*

- *Tarea #2*: Se obtienen las alineaciones con una máxima *Precisión* elevando el umbral de alineación para las correspondencias válidas al valor de 0.95.

- *Tarea #3*: Se obtienen las alineaciones con un máximo *Recall*, disminuyendo el valor del umbral de alineación al valor de 0.65.

En el test *Anatomy*, los métodos con los que nos comparamos utilizan la misma configuración que para el resto de los experimentos, excepto ASMOV, que usa el directorio médico *UMLS* en lugar de *WordNet*. Considerando que las ontologías utilizadas en esta prueba son muy específicas del dominio de la medicina y muchos de los conceptos no aparecen en *WordNet*, hemos decidido probar *FuzzyAlign* con ambos tesauros, para observar el impacto en la exactitud de las alineaciones. A continuación se detallan los resultados obtenidos en el test *Anatomy* en ambos conjuntos de experimentos: el primero utilizando *WordNet* como tesauro léxico para calcular la similitud lingüística y el segundo utilizando las bases de datos de *UMLS* en lugar de *WordNet*.

Prueba OAEI-Anatomy utilizando WordNet

En esta primera prueba con las ontologías del test *Anatomy* ejecutamos *FuzzyAlign* con su configuración original (utilizando *WordNet* como directorio léxico externo). Los resultados obtenidos por *FuzzyAlign* en esta prueba no fueron los mejores. Esto se debe fundamentalmente al hecho de que el sistema está pensado para dar el mayor peso al componente léxico y *WordNet*, al ser un tesauro de ámbito general, no abarca la totalidad de los términos específicos del dominio de la medicina. El uso de *WordNet* como herramienta léxica en lugar de un tesauro médico provoca que el sistema no obtenga similitudes léxicas óptimas en los casos en que las entidades no aparezcan en *WordNet* y la falta de esta información afecta los valores de similitud finales.

En la Tabla 5.8 se muestra una comparativa de los resultados obtenidos en la prueba *Anatomy* por los sistemas que estamos evaluando (exepto ASMOV, debido a que usa *UMLS*). Como se observa en la tabla el sistema que alcanzó mejores resultados fue CODI, seguido de SOBOM, aunque este último sólo fue capaz de computar resultados en la primera tarea.

Tabla 5.8. Resultados obtenidos por los métodos estudiados en la prueba Anatomy en términos de Precisión, Recall, y F1. En el caso de FuzzyAlign se utiliza WordNet como herramienta léxica.

	Tarea #1			*Tarea #2*			*Tarea #3*			*H-Mean*		
Sistema	*P*	*R*	*F1*	*P*	*R*	*F1*	*P*	*R*	*F1*	*P*	*R*	*F1*
CODI	**0.96**	**0.65**	**0.77**	**0.96**	**0.66**	**0.78**	**0.78**	**0.69**	**0.73**	**0.89**	**0.66**	**0.76**
SOBOM	0.95	0.78	0.86	-	-	-	-	-	-	-	-	-
FuzzyAlign	0.72	0.74	0.73	0.75	0.45	0.56	0.44	0.76	0.56	0.61	0.62	0.64

Prueba OAEI-Anatomy utilizando UMLS

En esta segunda prueba, se configura *FuzzyAlign* para utilizar el metatesauro *UMLS* en lugar de *WordNet* en el cálculo de la similitud lingüística. En la Tabla 5.9 se muestran los resultados de la comparación con ASMOV en términos de *Precisión*, *Recall*, *F1* y tiempo de ejecución promedio en minutos. En esta prueba nos comparamos solamente con ASMOV, por ser el único sistema que ajusta su configuración para utilizar *UMLS* en lugar de *WordNet* en el test *Anatomy*.

Como se puede observar en la Tabla 5.9, los resultados de *FuzzyAlign* en esta etapa de la prueba mejoran considerablemente con el uso de *UMLS*. Esto se debe al hecho de que los valores de similitud lingüística obtenidos son más apropiados, lo que se traduce en un incremento de la exactitud de los valores finales de similitud entre las entidades de las ontologías. En comparación con ASMOV, los valores de *Precisión* y *Recall* obtenidos por *FuzzyAlign* fueron considerablemente mejores pero el tiempo de ejecución fue peor. *FuzzyAlign* tardó 45 minutos como promedio en procesar las ontologías completas debido a su gran tamaño, lo que nos lleva a la conclusión de que se necesita mejorar la escalabilidad del sistema para procesar ontologías grandes.

Tabla 5.9. Resultados obtenidos por FuzzyAlign y ASMOV en la prueba Anatomy en términos de Precisión, Recall, F1 y tiempo de ejecución en min., utilizando UMLS como herramienta léxica.

	Tarea #1				*Tarea #2*				*Tarea #3*				*Media*			
System	*P*	*R*	*F1*	*T*	*P*	*R*	*F1*	*T*	*P*	*R*	*F1*	*T*	*P*	*R*	*F1*	*T*
ASMOV	0.79	0.77	0.78	15	0.86	0.75	0.81	15	0.71	0.79	0.75	15	0.78	0.77	0.79	15
FuzzyAlign	**0.97**	**0.91**	**0.94**	**45**	**1.0**	**0.85**	**0.92**	**45**	**0.85**	**0.92**	**0.88**	**45**	**0.94**	**0.89**	**0.91**	**45**

5.2.3 Tercera fase: Experimentos en el dominio de las redes de sensores

Uno de los objetivos de la evaluación experimental es analizar el comportamiento del sistema ante ontologías de dominio. Con este fin, además de los experimentos en el dominio de la medicina, propuestos por la *OAEI* (*Anatomy*), hemos realizado experimentos con ontologías de redes de sensores. Por su nivel de compatibilidad, las ontologías reales del dominio de las redes de sensores que hemos seleccionado son: *MMI Device* [Underbrink *et al*., 2008], *CSIRO Sensor* [Neuhaus y Compton, 2009] y *SSN* [Compton *et al*., 2012].

La ontología *CSIRO Sensor* [Neuhaus y Compton, 2009] ha sido desarrollada por la *Commonwealth Scientific and Industrial Research Organisation* (CSIRO) para describir sensores y redes de sensores. Está destinada a ser utilizada en la integración de datos, búsqueda y clasificación. La ontología *CSIRO Sensor* cubre una gama bastante amplia de conceptos que permiten describir la mayor parte del vocabulario y relaciones en el dominio de los sensores. Por su parte la ontología *MMI Device* [Underbrink *et al*., 2008] ha sido diseñada para la interoperabilidad de datos oceanográficos y se centra en modelar sensores, dispositivos de oceanografía y medidas. La ontología *Semantic Sensor Netrwork* (SSN) [Compton *et al*., 2012] fue desarrollada por el *W3C* para modelar los dispositivos de sensores, sistemas, procesos, observaciones y conocimiento ambiental. Esta ontología ya ha comenzado a adoptarse y aplicarse como ontología de referencia dentro de la comunidad de las redes de sensores. Las principales clases de la ontología *SSN* se han mapeado a las clases de la ontología de referencia global DOLCE Ultra Lite (DUL) [Janowicz y Compton, 2010] para facilitar la reutilización y la interoperabilidad.

El primer paso del experimento consistió en alinear las ontologías *CSIRO Sensor* y *MMI Device* con la ontología *SSN*, debido a que esta última es la más general y exhaustiva que se ha desarrollado en el dominio de las redes de sensores y por ofrecer alineaciones con DOLCE, que es una de las ontologías generales de referencia más utilizadas. Posteriormente el sistema se aplicó a las ontologías *CSIRO Sensor* y *MMI Device*. En la Figura 5.5 se muestra un pequeño fragmento del conjunto de alineaciones obtenidas entre las tres ontologías. Las correspondencias obtenidas se han marcado con líneas discontinuas. La Tabla 5.10 muestra los resultados finales de aplicar los cuatro algoritmos analizados entre las tres ontologías. Como se puede observar en la tabla, los resultados de *Precisión*, *Recall* y *F1* obtenidos por *FuzzyAlign* superan claramente al resto de aproximaciones analizadas. A pesar de esto, hubo algunas correspondencias que no fueron detectadas por tratarse de términos muy específicos

de las redes de sensores que no se encontraban en las bases de datos de *WordNet*, por lo que sería deseable contar con un tesauro especializado en el dominio de los sensores, similar a *WordNet* que nos permita relacionar lingüísticamente la mayor parte de los términos de este dominio.

Tabla 5.10. Resultados de los métodos de alineación entre las tres ontologías del dominio de las redes de sensores.

	MMI Device-SSN			*CSIRO-SSN*			*MMI Device-CSIRO*			*H-Mean*		
System	*P*	*R*	*F*	*P*	*R*	*F*	*P*	*R*	*F*	*P*	*R*	*F*
ASMOV	0.84	0.65	0.73	0.72	0.78	0.75	0.79	0.72	0.75	0.78	0.71	0.74
CODI	0.78	0.83	0.80	0.81	0.78	0.79	0.86	0.71	0.78	0.82	0.77	0.79
SOBOM	0.81	0.74	0.77	0.76	0.81	0.78	0.85	0.63	0.72	0.81	0.72	0.76
FuzzyAlign	**0.92**	**0.84**	**0.88**	**0.95**	**0.82**	**0.88**	**0.90**	**0.85**	**0.87**	**0.92**	**0.84**	**0.88**

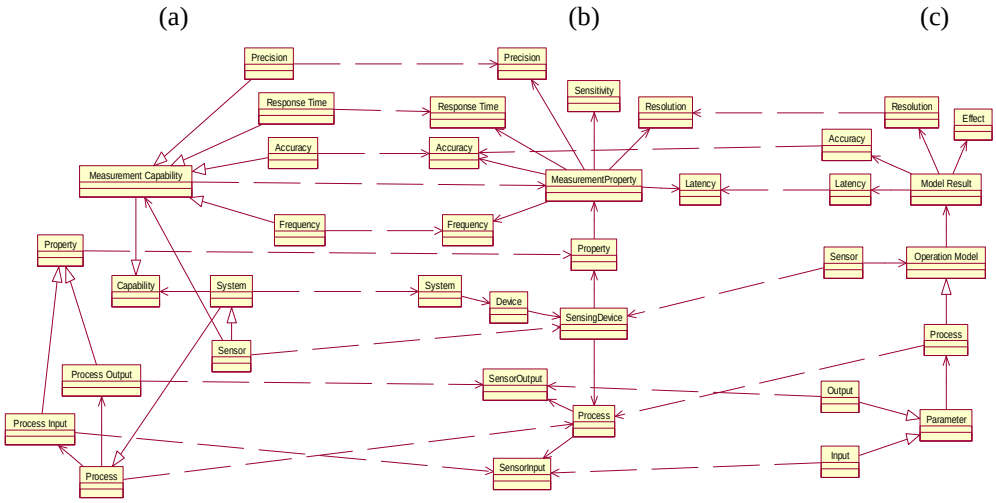


Figura 5.5. Pequeño fragmento de las alineaciones resultantes de aplicar FuzzyAlign a las tres ontologías seleccionadas. (a) *Ontología MMI Device.* (b) *Ontología SSN.* (c) *Ontología CSIRO Sensor.*

5.3 Resumen y consideraciones finales

En este capítulo explicamos en detalle los experimentos realizados para evaluar las contribuciones del método propuesto en la tesis. Los experimentos fueron divididos en tres fases, la primera encaminada a evaluar el método para el cálculo de la similitud terminológica,

para la cuál se utilizaron fragmentos de taxonomías reales con cierto grado de solapamiento, la segunda con el objetivo de evaluar el sistema completo combinando la similitud terminológica con la similitud estructural en las ontologías propuestas por la *OAEI*. La tercera fase se llevó a cabo con el objetivo de probar el comportamiento del sistema ante ontologías reales del dominio de las redes de sensores, utilizando el directorio léxico general (*WordNet*). En los resultados de la primera fase se pudo comprobar que los índices de similitud terminológica obtenidos para los conceptos equivalentes fueron razonablemente buenos. Sin embargo, se obtenían algunos valores de similitud elevados en conceptos no relacionados que contenían algunas palabras con cierta similitud lingüística en su terminología. Este problema fue resuelto con la incorporación al sistema de las medidas de similitud estructural, que tenían en cuenta otros factores como la estructura interna y relacional de los conceptos en las ontologías.

En la segunda fase se realizaron tres de los experimentos propuestos por la *OAEI* para la evaluación de los sistemas de alineación de ontologías y se compararon los resultados con 6 de los métodos existentes que habían realizado los tres experimentos. El primer experimento, llamado *Benchmark* se basa en una ontología de referencia perteneciente al dominio de las bibliografías, sobre la que se van haciendo variaciones sistemáticas. El experimento *Benchmark* está dividido en tres pruebas: la prueba simple, la prueba sistemática y la prueba real. La prueba simple consiste en alinear la ontología de referencia con ella misma y con una ontología perteneciente a otro dominio, como por ejemplo la ontología *wine*. La prueba sistemática consiste en comprobar el comportamiento del sistema cuando falta algún tipo de información (eliminando fragmentos de las ontologías). Por último, la prueba real se realiza con ontologías reales en el dominio de las referencias bibliográficas. En las pruebas simples y en las pruebas con ontologías reales de este experimento los resultados obtenidos por *FuzzyAlign* fueron muy buenos, superando al resto de los sistemas estudiados en cuanto a *Precisión* y *Recall*. En las pruebas sistemáticas, que evalúan el comportamiento del sistema ante la falta de información los resultados demostraron que el sistema funciona mejor cuando no se modifican los elementos terminológicos, siendo menos sensible ante cambios estructurales.

El segundo experimento, llamado *Conference*, tiene como objetivo encontrar todas las correspondencias correctas dentro de una colección de ontologías del dominio de la organización de conferencias. El experimento consta de siete ontologías, que se combinan dos a dos para hacer un total de 21 ejecuciones del algoritmo. Los resultados obtenidos por nuestro

sistema en este experimento superan ampliamente al resto de los métodos analizados en cuanto a *Precisión* y *Recall* con un umbral de selección de alineaciones del 88%.

El tercer experimento, llamado *Anatomy*, consiste en alinear dos ontologías del campo específico de la medicina. Los resultados obtenidos por *FuzzyAlign* en este experimento con su configuración inicial, utilizando la herramienta *WordNet* como diccionario léxico no fueron los mejores. Esto ocurre debido a que *WordNet* al ser un tesauro de ámbito general no abarca la totalidad de los términos específicos del dominio de la medicina. Los mejores resultados se obtuvieron para la herramienta ASMOV, que hace uso del tesauro médico específico *UMLS*. Teniendo esto en cuenta, hicimos una modificación del algoritmo para utilizar *UMLS* como herramienta léxica especializada en el dominio de la medicina en lugar de *WordNet*. Los resultados de *FuzzyAlign* en el experimento *Anatomy* mejoraron considerablemente con el uso de *UMLS*, superando a *ASMOV*. La principal limitación en este test ha sido el elevado tiempo de ejecución al tener que procesar ontologías muy grandes.

El último grupo de experimentos se realizó con el objetivo de evaluar el comportamiento del sistema en ontologías del dominio de las redes de sensores. Los resultados obtenidos en este experimento en cuanto a *Precisión* y *Recall* fueron muy superiores a los alcanzados por los otros sistemas, demostrando la efectividad del método en este dominio. A pesar de esto, hubo algunas correspondencias que no fueron detectadas por tratarse de términos muy específicos de las redes de sensores que no se encontraban en las bases de datos de *WordNet*, por lo que sería deseable contar con un tesauro especializado en el dominio de los sensores, similar a *WordNet* y *UMLS*, que nos permita relacionar lingüísticamente la mayor parte de los términos de este dominio.

CAPÍTULO 6. CONCLUSIONES Y TRABAJO FUTURO

Solo podemos ver poco del futuro, pero lo suficiente para darnos cuenta de que hay mucho que hacer.

Allan Turing

En este capítulo se presentan las conclusiones derivadas de la investigación realizada en esta tesis doctoral dentro del campo de la alineación de ontologías para la Web Semántica. Se resumen también las principales contribuciones, publicaciones y difusión del trabajo realizado en la tesis doctoral. Finalmente se mencionan las líneas de trabajo futuras que se desprenden de este trabajo de tesis.

6.1 Conclusiones

El trabajo presentado en esta tesis está encaminado a proporcionar un mecanismo que contribuyan a resolver el problema de la heterogeneidad semántica provocada por la coexistencia de diferentes ontologías en la Web semántica. Las ontologías son un elemento crucial para conseguir la interoperabilidad semántica, por lo que se están realizando muchos esfuerzos en la investigación en técnicas de alineación y fusión de ontologías que permitan a las aplicaciones intercambiar información en entornos donde coexistan ontologías heterogéneas. En particular nos centramos en el problema de la alineación de ontologías utilizando técnicas de lógica difusa. En esta línea nos planteamos el siguiente objetivo general:

- *Proponer un método automático de* alineación *de ontologías aplicando técnicas de lógica difusa para combinar las similitudes entre entidades de ontologías diferentes teniendo en cuenta su terminología y su estructura.*

De este objetivo general se derivaron los siguientes objetivos específicos:

- *Proponer medidas de similitud lingüísticas que utilicen tanto técnicas basadas en cadenas, como las relaciones entre las palabras dentro de los directorios léxicos.*

- *Combinar la similitud lingüística con la información semántica del contexto de las entidades en las ontologías para obtener una similitud terminológica, utilizando técnicas de lógica difusa.*

- *Proponer medidas de similitud estructurales, que utilicen la jerarquía taxonómica y la estructura interna en ontologías que no compartan la taxonomía y combinarlas utilizando técnicas de lógica difusa.*

- *Utilizar la lógica difusa para el cálculo de la similitud entre las propiedades de los conceptos teniendo en cuenta su información terminológica, sus clases y sus tipos.*

En correspondencia con los objetivos planteados, en esta tesis se ha desarrollado un sistema multicapa para la alineación de ontologías, que combina diversas medidas de similitud utilizando técnicas de lógica difusa. En esta línea se proponen varias medidas de similitud terminológicas y estructurales entre entidades de ontologías diferentes. Las similitudes entre las entidades de diferentes ontologías se calculan teniendo en cuenta elementos semánticos y léxicos de la terminología, así como la estructura interna y relacional de las ontologías. Para la similitud lingüística se proponen medidas de similitud que utilizan herramientas léxicas externas como los directorios *WordNet* y *UMLS*, y para la similitud semántica se aplica el *coeficiente de Jaccard* sobre los resultados de búsquedas contextualizadas de los términos en la *Wikipedia*. La similitud estructural se calcula teniendo en cuenta las relaciones de los conceptos en la jerarquía taxonómica, así como elementos de la estructura interna de los mismos como son las propiedades, los tipos y la cardinalidad.

El sistema ha sido probado en un amplio conjunto de ontologías terminológicas y de dominio reales, entre las cuáles se encuentran las ontologías de evaluación propuestas en los experimentos de la *OAEI*. Los resultados de estas pruebas fueron muy buenos en ontologías reales con construcciones léxicas correctas superando al resto de los métodos comparados en cuanto a *precisión* y *recall*. Sin embargo en las pruebas que evalúan el comportamiento del sistema ante la falta de información los resultados fueron variados. Debido al peso que se le concede a la terminología en *FuzzyAlign*, los resultados en los casos en que se suprime la información terminológica se vieron afectados de manera considerable, demostrando que el sistema funciona mejor cuando no se modifican los elementos terminológicos, siendo menos sensible ante cambios estructurales.

En los experimentos con las ontologías médicas, los resultados obtenidos por *FuzzyAlign* utilizando la herramienta *WordNet* no fueron buenos, debido a que *WordNet* es un tesauro de ámbito general y no abarca la totalidad de los términos específicos del dominio de la

medicina. Los resultados de *FuzzyAlign* en el experimento mejoraron considerablemente con el uso de *UMLS* como directorio léxico especializado en medicina. En el caso de las ontologías del dominio de las redes de sensores los resultados del sistema con su configuración original (utilizando *WordNet* como herramienta léxica) fueron buenos, demostrando que su utilización es viable en ontologías de dominio. A pesar de esto, hubo algunas correspondencias que no fueron detectadas por tratarse de términos muy específicos que no se encontraban en las bases de datos de *WordNet*, por lo que sería deseable contar con un tesauro especializado en el dominio de los sensores, similar a *WordNet* o *UMLS*, que nos permita relacionar lingüísticamente la mayor parte de los términos de este dominio.

6.2 Contribuciones de la tesis

Las contribuciones de esta tesis las podemos dividir en dos categorías teniendo en cuenta el ámbito de la aportación. Entre los principales desafíos que se perfilan en el campo de la alineación de ontologías están la completa automatización de las técnicas de alineación y la mejora de las medidas de similitud para obtener correspondencias más ajustadas. A continuación resumimos las principales contribuciones de la tesis en cuanto a las medidas de similitud (tanto terminológicas como estructurales) y en cuanto a las aproximaciones de alineación de ontologías en general.

6.2.1 Contribuciones en las medidas de similitud

- Se proponen medidas de similitud lingüísticas basada en las relaciones de sinonimia y derivación de las palabras que componen los conceptos utilizando los directorios léxicos externos, tanto de propósito general, como es el caso de *WordNet*, como de dominio específico, para el que se ha utilizado el directorio médico *UMLS*.

- Se propone una medida de similitud lingüística a la que hemos denominado factor léxico, que se basa en las distancias entre las palabras que componen los conceptos. Esta medida utiliza la distancia de Levensthein y la distancia entre las palabras dentro de la taxonomía del directorio léxico. Ambas distancias se combinan a través de una suma ponderada con un factor de peso diferente.

- Se proponen una medida de similitud semántica que consiste en aplicar *el coeficiente de Jaccard* sobre la totalidad de documentos recuperados en búsquedas contextualizadas de los conceptos en la *Wikipedia*.

- Se proponen medidas de similitud estructurales basadas en la estructura relacional y en la estructura interna de las ontologías. Estas medidas se aplican en el módulo de similitud jerárquica para calcular la influencia de las similitudes entre los hijos, padres y hermanos de los conceptos en las taxonomías y en la capa de alineación para calcular la influencia que tienen las similitudes entre las propiedades en la similitud final entre los conceptos.

6.2.2 Contribución a la alineación de ontologías

- Se ha desarrollado un sistema basado en reglas difusas de varias capas para realizar de manera automática el proceso de alineación de ontologías. En cada capa se combinan diversas medidas de similitud terminológicas y estructurales que se van refinando hasta llegar a la capa superior, donde se construye el fichero de alineaciones.

- Se ha evaluado el sistema propuesto tanto en ontologías escogidas ad-hoc como en los tests planteados por la *OAEI*, demostrando que la aproximación propuesta supera en la mayoría de los casos a las existentes en las diferentes métricas de evaluación consideradas (*Precisión*, *Recall* y *F1*).

- Se ha identificado como factor crucial de mejora de nuestra aproximación en las ontologías de dominio la existencia de tesauros especializados como *UMLS*, lo que proporciona una estrategia de acción clara para mejorar las alineaciones de ontologías en los dominios en los que éstas tienen mayor dificultad.

6.3 Difusión de las contribuciones de la tesis

Las contribuciones esta tesis doctoral han sido difundidas a través de una serie de publicaciones en congresos, y revistas. Los primeros resultados de la investigación realizada en la tesis tuvieron aplicación en el escenario del proyecto RESULTA, financiado por el Ministerio de Industria, Turismo y Comercio de España. A continuación se presenta un

resumen de los primeros pasos encaminados a la difusión de las contribuciones de esta tesis doctoral.

6.3.1 Publicaciones derivadas de la tesis

A continuación citamos las principales publicaciones derivadas de esta investigación:

Revistas indexadas en JCR:

- Fernandez, S.; Marsa-Maestre, I.; Velasco, J.R.; Alarcos, B. Ontology Alignment Architecture for Semantic Sensor Web Integration. *Sensors* 2013, 13, 12581-12604.

Capítulos de Libro:

- Fernández S., Marsa-Maestre I. and Velasco J. Performing Ontology Alignment via a Fuzzy-Logic Multi-Layer Architecture. Aceptado para su publicación como capítulo de libro en "Lecture Notes in Communications in Computer and Information Science (CCIS)" de Springer-Verlag.

- Fernández, Susel; Velasco, Juan R.; López-Carmona, Miguel A.: A Fuzzy Rule-Based System for Ontology Mapping. 5925. In: PRIMA: Springer, 2009 (Lecture Notes in Computer Science). - ISBN 978-3-642-11160-0, S. 500-507.

Contribuciones a Congresos:

- Fernández S., Velasco J., Marsa-Maestre I. and López-Carmona M. (2012). FuzzyAlign-A Fuzzy Method for Ontology Alignment. In Proceedings of the International Conference on Knowledge Engineering and Ontology Development, pages 98-107. DOI: 10.5220/0004139500980107. ISBN: 978-989-8565-30-3. Barcelona. Spain.

- Susel Fernández, Juan R. Velasco, Miguel A. López-Carmona. Similitud difusa basada en nombres y relaciones taxonómicas de conceptos para el matching de ontologías. Jornadas de Ingeniería telemática. JITEL 2011. ISBN: 978-84-694-5948-5 88. Cantabria, España.

- Susel Fernández, Juan R. Velasco, Miguel A. López-Carmona. Sistema Basado en Reglas Difusas para el Mapeo de Ontologías. Congreso Español Sobre Tecnologías y Lógica Fuzzy (ESTYLF 2010). Huelva, España.

- Fernández, Susel; Velasco, Juan R.; López-Carmona, Miguel A.: A Fuzzy Rule-Based System for Ontology Mapping. 5925. In: PRIMA: Springer, 2009 (Lecture Notes in Computer Science). - ISBN 978-3-642-11160-0, S. 500-507.

- Susel Fernández, Andrés Navarro, Diego Casado, Juan R. Velasco. Técnicas de traducción de ontologías para la interconexión de servicios en escenarios de catástrofe. Congreso Internacional de Telemática y Telecomunicaciones. CITTEL'08. ISBN: 978-84-95227-61-4. La Habana. Cuba.

6.3.2 Proyecto RESULTA

El proyecto RESULTA - Red de Consultoría para la gestión de procesos y relaciones (TSI-020301-2009-31) que es parte del programa AVANZA I+D financiado por el Ministerio de Industria, Turismo y Comercio de España, se alinea con los objetivos de la Internet del Conocimiento y los Contenidos, ya que se orienta a proporcionar a los usuarios mecanismos para la generación y distribución de contenidos en el ámbito de la consultoría. La difusión del conocimiento permite mantener relaciones de confianza entre empresas, clientes, proveedores y colaboradores, proporcionando además los mecanismos necesarios para una interacción eficiente entre usuarios, aprovechando su experiencia y orientando a las personas hacia la sociedad virtual del futuro. En este contexto, RESULTA construye una plataforma de consultoría basada en software libre para facilitar e investigar la mejora de la calidad de los procesos de consultoría con el aprovechamiento efectivo de las redes sociales, mejorar la participación y el aprendizaje de los empleados de las empresas de consultoría e incrementar la relación de cooperación y seguimiento con los clientes, la coordinación entre consultoras, la gestión del conocimiento en el negocio y el despliegue de servicios en las empresas de consultoría.

El principal objetivo científico del proyecto fue investigar la interrelación entre la estructura formal e informal de las organizaciones y las posibilidades que el uso de tecnologías de la Internet del Futuro para facilitar la localización de información y usuarios, así como automatizar la publicación de relaciones entre contenidos. Intrínseco a este objetivo

es el estudio sobre la mejora de los procesos de consultoría y el establecimiento de mecanismos de mejora de la calidad de dichos procesos, facilitando la aplicación de metodologías en la tarea de prestar el mejor servicio posible a los clientes que demandan sus servicios.

Los trabajos de investigación preliminares que han derivado en esta tesis, fueron aplicados en el paquete de trabajo PT5 del proyecto RESULTA, cuyo objetivo fundamental fue la aplicación de tecnologías de la Web semántica a la gestión de contenidos en la consultoría, abordado con la integración de un gestor de metadatos para los contenidos y la visualización de personas y contenidos mediante una navegación guiada por la ontología subyacente.

6.4 Líneas de Trabajo Futuro

Por su vital importancia dentro de la evolución hacia la Web semántica, la solución al problema de la heterogeneidad semántica y específicamente la alineación de ontologías es un campo de investigación que tiene mucho camino por recorrer. A pesar de que se han realizado muchas contribuciones interesantes en los últimos años, las técnicas de alineación de ontologías existentes no explotan la totalidad de la riqueza expresiva de las ontologías y todavía requieren la intervención de expertos en algunas fases del proceso. También se necesita mejorar las medidas de similitud existentes para conseguir correspondencias más exactas entre las entidades y evitar alineaciones erróneas. De las limitaciones de este trabajo se han derivado líneas futuras más concretas encaminadas a resolver los problemas actuales de nuestro sistema y a mejorar algunos aspectos para obtener mejores resultados.

La principal limitación del sistema ha sido el elevado tiempo de ejecución al tener que procesar ontologías muy grandes, debido a la gran cantidad de información y al gran número de consultas en Internet. Por ello, como primera línea de trabajo futuro nos planteamos mejorar la escalabilidad del sistema. En este sentido hemos pensado utilizar técnicas de procesamiento paralelo que permitan el uso de múltiples procesadores para analizar de manera concurrente las distintas partes de las ontologías y así optimizar el tiempo de ejecución.

La segunda de las líneas que nos hemos planteado como trabajo futuro es extender la técnica de *FuzzyAlign* para proponer un modelo de integración que permita tener en cuenta el uso de otras relaciones de correspondencia entre las entidades en dominios reales además de la equivalencia. En cuanto a la mejora de las medidas de similitud, se pretende incorporar

medidas de similitud estructurales que tengan en cuenta relaciones inter-conceptuales más complejas, como por ejemplo la mereología, así como el procesamiento axiomático de las ontologías para eliminar las inconsistencias.

Otro de los frentes abiertos en esta rama es la coexistencia de múltiples idiomas en la terminología de las ontologías. En estos casos existe una limitación importante debido a que el directorio léxico WordNet sólo contiene diccionarios en idioma inglés. En esta línea pretendemos explorar la utilización de la herramienta EuroWordNet, que ha emergido como un proyecto para interconectar varios diccionarios en distintos idiomas europeos con la misma estructura del WordNet original de Princeton.

Otra de las líneas futuras en la que hemos comenzado a trabajar es probar el sistema en ontologías especializadas en otros dominios, más allá de la medicina y las redes de sensores, para construir un repositorio de alineaciones que pueda ser accesible de manera directa desde las aplicaciones y así mejorar la interoperabilidad semántica en estos entornos.

Finalmente, estamos trabajando en la puesta a punto del sistema para presentarnos a la próxima competición anual de la *OAEI* que ofrece una plataforma de pruebas rigurosas para evaluar el rendimiento de los métodos de alineación de ontologías. De esta forma, nuestro sistema y sus resultados quedarían a disposición de la comunidad científica que investiga en esta rama y podrán ser utilizados para seguir trabajando en la mejora de las técnicas de alineación de ontologías.

Bibliografía

[Alander, 1992] J.T. Alander (1992). On optimal population size of genetic algorithms. Proceedings Com-pEuro 1992, Computer Systems and Software Engineering, 6th Annual European Computer Conference, 65-70.

[Anderberg, 1973] Anderberg, M. R. 1973. Cluster analysis for applications. New York: Academic Press.

[Baker, 1985] Baker, J. E. (1985). Adaptive selection methods for genetic algorithms. Proceedings of an International Conference on Genetic Algorithms and Their Applications, 100-111.

[Baker, 1987] J.E. Baker (1987). Reducing bias and inefficiency in the selection algorithm. Proceedings of the Second International Conference on Genetic Algorithms and Their Applications, 14-21.

[Becker, 1988] Becker, Joseph D. (August 29, 1988). Unicode 88. Xerox Corporation. Palo Alto. CA. Reprinted with permission of the author, by the Unicote Consortium, September 1998.

[Berners-Lee *et al*., 2001] T. Berners-Lee, J. Hendler, O Lassila. The Semantic Web: a new form of Web content that is meaninful to computers will unleash a revolution of new possibilities. Scientific American, May 2001.

[Berners-Lee y Fischetti, 1999] Berners-Lee, Tim; Fischetti, Mark (1999). Weaving the Web. HarperSanFrancisco. chapter 12. ISBN 978-0-06-251587-2.

[Berners-Lee, 1989] T. Berners-Lee. Information Management: A Proposal. Internal Project Proposal, CERN, 1989. Disponible en http://www.w3.org/History/1989/ proposal.html.

[Berners-Lee, 2000] Tim Berners-Lee. Semantic Web -XML2000. Architecture http://www.w3.org/2000/Talks/1206-xml2k-tbl/slide11-0.html

[Bilenko y Mooney, 2002] Bilenko, M., and Mooney, R. 2002. Learning to combine trained distance metrics for duplicate detection in databases. Technical Report Technical Report AI 02-296, Artificial Intelligence Lab, University of Texas at Austin. Available from http://www.cs.utexas.edu/users/ml/papers/marlin-tr-02.pdf.

[Bouquet *et al*., 2003] P. Bouquet, F. Giunchiglia, F. van Harmelen, L. Serafini, and H. Stuckenschmidt. C-owl – contextualizing ontologies. In D. Fensel, K. Sycara, and J. Mylopoulos, editors, The Semantic Web - ISWC 2003, volume 2870 of Lecture Notes in

Computer Science (LNCS), pages 164–179, Sanibel Island (FL, USA), October 2003. Springer Verlag.

[Bray *et al*., 2009] Bray, Tim; Dave Hollander, Andrew Layman, Richard Tobin, Henry S. Thompson (December 2009). "Namespaces in XML 1.0". W3C. Retrieved 9 October 2010.

[Bray y Curtis, 1957] Bray J. R., Curtis J. T., 1957. An ordination of the upland forest of the southern Winsconsin. Ecological Monographies, 27, 325-349.

[Brindle, 1981] Brindle, A. (1981). Genetic algorithms for function optimization (Doctoral dissertation and Technical Report TR81-2). Edmonton: University of Alberta, Department of Computer Science.

[Brindle, 1991] A. Brindle (1991). Genetic algorithms for function optimization. Tesis doctoral, Universidad de Alberta, Canada.

[Browne *et al*., 2000] Browne, McCray and Srinivasan (2000). The Specialist Lexicon. Lister Hill National Center for Biomedical Communications, National Library of Medicine, Bethesda, MD, p. 1.

[Bunke y Sanfeliu, 1990] Bunke H. String matching for structural pattern recognition. In: Bunke H, Sanfeliu A (eds). Syntactic and Structural Pattern Recognition, Theory and Applications. World Scientific, 1990, pp 119–144.

[Castillo, 2002] Sergio Fernando Castillo Castelblanco. Composición de Servicios Mediante el Modelo de Agentes Móviles. Tesis doctoral.

[Cha, 2007] Cha, S.-H.: Comprehensive Survey on Distance/Similarity Measures between Probability Density Functions. International Journal of Mathematical Models and Methods in Applied Sciences. Issue 4, Volume 1, 300-307, 2007

[Chandrasekaran *et al*., 1999] Chandrasekaran, B., Johnson, T. R., Benjamins, V. R.(1999) Ontologies: what are they? why do we need them?. IEEE Intelligent Systems and Their Applications. 14(1). Special Issue on Ontologies: 20-26.

[Chaudhri *et al*., 1998] Chaudhri, V. K., Farquhar, A., Fikes, R., Karp, P. D. and Rice, J. P. 1998. OKBC: A Programmatic Foundation for Knowledge Base Interoperability. In Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI-98): 600-607. Madison, Wisconsin: AAAI Press/The MIT Press.

[Compton *et al*., 2012] Compton, M., Barnaghi, P., Bermudez, L., Garcia-Castro, R., Corcho, O., Cox, S., Graybeal, J., Hauswirth, M., Henson, C., Herzog, A., Huang, V., Janowicz, K., Kelsey, W. D., Le Phuoc, D., Lefort, L., Leggieri, M., Neuhaus, H., Nikolov, A., Page, K., Passant, A., Sheth, A., and Taylor, K. : The SSN Ontology of the W3C Semantic Sensor Network Incubator Group. Journal of Web Semantics.

[Cordón *et al*., 2001] Cordón, O., Herrera, F., Hoffman, F., Magdalena, L.: Genetic Fuzzy Systems. Evolutionary Tuning and Learning of Fuzzy Knowledge Bases. World Scientific, Singapore (2001).

[Craven *et al*., 2000] Craven, M.; DiPasquo, D.; Freitag, D.; McCallum, A.; Mitchell, T.; Nigam, K.; and Slattery, S. 2000. Learning to Construct Knowledge Bases from the World Wide Web. Artificial Intelligence, 118(1-2): 69-114.

[Cruz *et al*., 2009] Cruz, Isabel F., Palandri, Antonelli Flavio, and Stroe, Cosmin. (September 2009): AgreementMaker Efficient Matching for Large Real-World Schemas and Ontologies. In International Conference on Very Large Databases, Lyon, France, pages 1586-1589.

[Darwin, 1859] C. Darwin (1859). On the Origin of Species by Means of Natural Selection, Murray, London.

[Davis, 1985] L. Davis (1985). Applying adaptive algorithms to epistatic domains, en Proceedings of the International Joint Conference on Artificial Intelligence, 162-164.

[Davis, 1991] L. Davis (ed.) (1991). Handbook of Genetic Algorithms, Van Nostrand Reinhold, New York.

[De Bruijn *et al*., 2004] Jos de Bruijn, Douglas Foxvog, and Kerstin Zimmerman. Ontology mediation patterns library. Deliverable D4.3.1, SEKT, 2004.

[De Jong, 1975] K.A. De Jong (1975). An analysis of the behaviour of a class of genetic adaptive systems. Tesis doctoral, University of Michigan.

[Dean y Schreiber, 2004] Dean Mike, Schreiber Guus. OWLWeb Ontology Language Reference. http: //www.w3.org/TR/2004/REC-owl-ref-20040210/; 2004.

[DeJong *et al*., 1993] K. A. DeJong, W. M. Spears, and D. F. Gordon. Using genetic algorithms for concept learning. Machine Learning, 1(13):161–188, 1993.

[Deza y Deza, 2006] Deza E. and Deza M.M., Dictionary of Distances, Elsevier, 2006.

[Monev, 2004] Monev V., Introduction to Similarity Searching in Chemistry, MATCH Commun. Math. Comput. Chem. 51 pp. 7-38, 2004

[Dice, 1945] Dice, Lee R. (1945). "Measures of the Amount of Ecologic Association Between Species". Ecology 26 (3): 297–302. doi:10.2307/1932409. JSTOR 1932409.

[Ding *et al*., 2006] Li Ding, Pranam Kolari, Zhongli Ding, and Sasikanth Avancha, Using Ontologies in the Semantic Web: A Survey, book-chapter in Ontologies: A Handbook of Principles, Concepts and Applications in Information Systems, 2006.

[Do y Rahm, 2002] Do, H.-H.; Rahm, E. COMA: A system for flexible combination of schema matching approaches. In Proceedings of the 28th international conference on Very Large Data Bases, Hong Kong, China, 20–23 August 2002; pp. 610–621.

[Doan *et al*., 2004] Doan A., Madhavan, J., Domingos, P., Halevy, A.: Ontology Matching: A Machine Learning Approach. *Handbook on Ontologies in Information Systems. In: S. Staab and R. Studer (eds.)*, Invited paper. Pp. 397-- 416. Springer-Velag, (2004).

[Domingos *et al*., 2008] P. Domingos, D. Lowd, S. Kok, H. Poon, M. Richardson, and P. Singla. Just add weights: Markov logic for the semantic web. In Proceedings of the Workshop on Uncertain Reasoning for the Semantic Web, pages 1–25, 2008.

[Domínguez-Dorado, 2004] M. Domínguez-Dorado. Todo Programación. Nº 12. Págs. 16-20. Editorial Iberprensa (Madrid). DL M-13679-2004. Septiembre, 2005. Programación de algoritmos genéticos.

[Dou *et al*., 2004] Dejing Dou, Drew McDermott and Peishen Qi (2004). Ontology Translation on the Semantic Web S. Spaccapietra *et al*. (Eds.): Journal on Data Semantics II, LNCS 3360, pp. 35–57, 2004. c_Springer-Verlag Berlin Heidelberg 2004.

[Duda *et al*., 2001] Duda, R.O., Hart, P.E., and Stork, D.G., Pattern Classification, 2nd ed. Wiley, 2001.

[Dunn y Everitt, 1982] Dunn, G., Everitt, B.S., (1982), "An Introduction to Mathematical Taxonomy", Cambridge University Press.

[Euzenat *et al*., 2010] Euzenat, J, Shvaiko, P., Giunchiglia, F., Stuckenschmidt, H., Mao, M., Cruz, I. (2010): Results of the Ontology Alignment Evaluation Initiative 2010. In: Proceedings of the 5th International Workshop on Ontology Matching (OM-2010).

[Euzenat y Shvaiko, 2007] Euzenat, J. and P. Shvaiko (2007). Ontology Matching. Berlin / Heidelberg, Springer.

[Euzenat y Valtchev, 2004] Euzenat, J., Valtchev, P. Similarity-based ontology alignment in OWL-lite. In Proceedings of the 16th European Conference on Artificial Intelligence, Valencia, Spain, 22–27 August 2004; pp. 333–337.

[Euzenat, 2004] Jérôme Euzenat. An API for ontology alignment. In *Proc. 3rd international semantic web conference, Hiroshima (JP), pages 698–712, 2004.*

[Euzenat, 2006] Euzenat, J. (2006). An API for ontology alignment (version 2.1). Montbonnot, France, INRIA Rhône-Alpes.

[Faith *et al*., 1987] Faith, D.P., Minchin, P.R., Belbin, L., (1987), "Compositional dissimilarity as a robust measure of ecological distance", Journal of Plant Ecology, Volume 69, Numbers 1-3.

[Fellbaum, 1998] Fellbaum, C. (1998). WordNet: An Electronic Lexical Database. MIT Press. 3.0, Cambridge, MA.

[Fensel *et al*., 2000] D. Fensel, I. Horrocks, F. van Harmelen, S. Decker and M. Erdmann and M. Klein (2000). OIL in a Nutshell. In: *Proceedings of the 12th European Workshop on Knowledge Acquisition, Modeling, and Management ({EKAW}'00)*.

[Fensel y Musen, 2001] Fensel, D. and Musen, M. (2001). The semantic web: A brain for humankind. IEEE Intelligent Systems, pages 24-25.

[Fensel, 2002] Fensel, D. (2002). Language standardization for the Semantic Web: The long way from OIL to OWL. Consultado en: 10-11-2012. http://informatik.uibk.ac.at/users/c70385/ftp/paper/dwc.pdf.

[Fernández-Breis y Martínez-Béjar, 2002] J. Fernández-Breis and R. Martínez-Béjar. "A cooperative framework for integrating ontologies". International Journal of Human-Computer Studies, 56: 665–720, 2002.

[Fernandez *et al*., 2013] Fernandez, S.; Marsa-Maestre, I.; Velasco, J.R.; Alarcos, B. Ontology Alignment Architecture for Semantic Sensor Web Integration. *Sensors* 2013, 13, 12581-12604.

[Fernandez *et al*., 2012] Fernández S., Velasco J., Marsa-Maestre I. and López-Carmona M. (2012). FuzzyAlign-A Fuzzy Method for Ontology Alignment. In Proceedings of the International Conference on Knowledge Engineering and Ontology Development, pages 98-107. DOI: 10.5220/0004139500980107. ISBN: 978-989-8565-30-3. Barcelona. Spain.

[Fernandez *et al*., 2011] Susel Fernández, Juan R. Velasco, Miguel A. López-Carmona. Similitud difusa basada en nombres y relaciones taxonómicas de conceptos para el matching de ontologías. Jornadas de Ingeniería telemática. JITEL 2011. ISBN: 978-84-694-5948-5 88. Cantabria, España.

[Fernandez *et al*., 2010] Susel Fernández, Juan R. Velasco, Miguel A. López-Carmona. Sistema Basado en Reglas Difusas para el Mapeo de Ontologías. Congreso Español Sobre Tecnologías y Lógica Fuzzy (ESTYLF 2010). Huelva, España.

[Fernandez *et al*., 2009] Fernández, Susel; Velasco, Juan R.; López-Carmona, Miguel A.: A Fuzzy Rule-Based System for Ontology Mapping. 5925. In: PRIMA: Springer, 2009 (Lecture Notes in Computer Science). - ISBN 978-3-642-11160-0, S. 500-507.

[Fernandez *et al*., 2008] Susel Fernández, Andrés Navarro, Diego Casado, Juan R. Velasco. Técnicas de traducción de ontologías para la interconexión de servicios en escenarios de catástrofe. Congreso Internacional de Telemática y Telecomunicaciones. CITTEL'08. ISBN: 978-84-95227-61-4. La Habana. Cuba.

[Ferrer, 2005] Ferrer-i-Cancho, R.: The structure of syntactic dependency networks: insights from recent advances in network theory. In: L. V., A. G. (eds.) Problems of quantitative linguistics, pp. 60–75 (2005)

[FOAF, 2010] FOAF http://xmlns.com/foaf/0.1/

[Gangemi *et al*., 2003] A. Gangemi, N. Guarino, C. Masolo, and A. Oltramari. Sweetening WordNet with DOLCE. AI Magazine, (24(3)):13–24, 2003.

[Giunchiglia *et al*., 2004] F. Giunchiglia, P.Shvaiko, M. Yatskevich: S-Match: an algorithm and an implementation of semantic matching. In Proceedings of ESWS'04.

[Golbeck *et al*., 2003] Golbeck, J., Fragoso, G., Hartel, F., Hendler, J., y Oberthaler J., Parsia, B. (2003). The National Cancer Institute's Thésaurus and Ontology. Journal of Web Semantics. 1(1): 75-80.

[Goldberg *et al*., 1990] Goldberg, D. E., Korb, B., & Deb, K. (1990). Messy genetic algorithms: Motivation, analysis, and first results. Complex. Systems, 3, 493-530.

[Goldberg y Richardson, 1987] D.E. Goldberg, J.T. Richardson (1987). Genetic algorithms with sharing for multimodal function optimization. Genetic Algorithms and their Applications: Proceedings of the Second International Conference on Genetic Algorithms and Their Applications, 41-49.

[Goldberg, 1989] D.E. Goldberg (1989). Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, Reading, MA.

[Gomez-Perez *et al*., 2000] Gómez-Pérez, A. Moreno, A., Pazos, J., Sierra-Alonso, A. (2000) Knowledge Maps: An essential technique for conceptualisation, Data & Knowledge Engineering, 33(2), 169-190.

[Gomez-Perez *et al*., 2004] Ontological engineering. A Gomez-Perez, M Fernández-López, O Corcho - London *et al*, 2004

[Goodman y Kruskal, 1963] Goodman, L.A., Kruskal, W.H., (1963), "Measures of association for cross classifications III. Approximate sampling theory", Journal of the American Statistical Association 58, 310-364.

[Grefenstette y Baker, 1989] Grefenstette, J. J. & Baker, J. E. (1989). How genetic algorithms work: A critical look at implicit parallelism. Proceedings of the Third International Conference on Genetic Algorithms, 20-27.

[Grosof, 2001] Grosof, Benjamin. "Representing E-Business Rules for the Semantic Web: Situated Courteous Logic Programs in RuleML". Proc. Wksh on Information Technologies and Systems (WITS-01), held 2001 at the Intl. Conf. on Information Systems (ICIS). Describes SweetRules tool as well as RuleML.

[Grosso *et al*., 1999] W. Grosso, H. Eriksson, R. Fergerson, J. Gennari, S. Tu, and M. Musen. Knowl- edge Modelling at the Millennium|The design and evolution of Protege2000. In Proceedings of the 12th Knowledge Acquisition, Modelling, and Management(KAW'99), Ban®, Canada, October 1999.

[Gruber, 1993] T. Gruber. Ontolingua: A Translation Approach to Providing Portable Ontology Speci¯ cations. Knowledge Acquisition, 5(2):199-200, 1993.

[Guarino y Welty, 2001] Guarino, N., Welty, C. (2001). Identity and subsumption. In R. Green, C. A. Bean, and S. Hyon Myaeng (eds.), The Semantics of Relationships: An Interdisciplinary Perspective, Kluwer.

[Guarino, 1995] Nicola Guarino – "Ontologies and Knowledge Bases. Towards a terminological clarification" (1995) p. 1

[Guarino, 1998] Guarino, N. (1998). Formal Ontology and Information Systems. Proceedings of FOIS'98: 3-15

[Holland, 1975] J. Holland (1975). Adaptation in Natural and Articial Systems, University of Michigan Press, Ann Arbor.

[Horn, 1951] Alfred Horn (1951), "On sentences which are true of direct unions of algebras", Journal of Symbolic Logic, 16, 14–21.

[Horrocks *et al*., 2001] I. Horrocks *et al*., The Ontology Inference Layer OIL, tech. report, Vrije Universiteit Amsterdam; www.ontoknowledge.org/oil/ TR/oil.long.html (current 9 Mar. 2001).

[Horrocks *et al*., 2004] Ian Horrocks, Peter Patel-Schneider, Harold Boley, Said Tabet, Benjamin Grosof, and Mike Dean. SWRL: a semantic web rule language combining OWL and RuleML, 2004. http://www.w3.org/Submission/SWRL/.

[Horrocks y van Harmelen, 2001] Horrocks, I., van Harmelen, F. (2001) Reference Description of the DAML+OIL OntologyMarkup Language, draft report, www.daml.org/2000/12/reference.html (current Jan. 2002).

[Jaccard, 1901] Jaccard, P., (1901), "Étude comparative de la distribuition florale dans une portion des Alpes et des Jura", Bull Soc Vandoise Sci Nat 37:547-579.

[Janikow, 1993] C. Z. Janikow. A knowledge-intensive genetic algorithm for supervised learning. Machine Learning, 1(13):169–228, 1993.

[Janowicz y Compton, 2010] K. Janowicz and M. Compton. The Stimulus-Sensor-Observation Ontology Design Pattern and its Integration into the Semantic Sensor Network Ontology. In The 3rd International workshop on Semantic Sensor Networks 2010 (SSN10) in conjunction with the 9th International Semantic Web Conference (ISWC 2010), 2010.

[Jaro, 1995] Jaro, M. A. 1995. Probabilistic linkage of large public health data files (disc: P687-689). Statistics in Medicine 14:491–498.

[Jean-Mary *et al*., 2009] Jean-Mary Y., Shironoshita E.P., Kabuka, M. (2009). Ontology Matching with Semantic Verification. Journal of Web Semantics. Sci. Serv. Agents World Wide Web, doi: 10.1016/j.websem.2009.04.001.

[Jiang y Conrath, 1997] Jiang, J.J., Conrath, D.W.: Semantic similarity based on corpus statistics and lexical taxonomy (1997).

[Kalfoglou y Schorlemmer, 2003] Yannis Kalfoglou and Marco Schorlemmer. Ontology mapping: the state of the art. The Knowledge Engineering Review, 18(1):1–31, 2003.

[Kensche *et al*., 2007] D. Kensche, C. Quix, M. A. Chatti, and M. Jarke. GeRoMe: A generic role based metamodel for model management. *Journal on Data Semantics*. VIII: 82-117, 2007

[Kifer, 2008] Kifer, Michael (2008). "Rule Interchange Format: The Framework". In: Web Reasoning and Rule Systems. Lecture Notes in Computer Science.

[Krause, 1986] Krause E.F., Taxicab Geometry An Adventure in Non-Euclidean Geometry.

[Larrañaga y Poza, 1994] P. Larrañaga, M. Poza (1994). Structure learning of Bayesian network by genetic algorithms. E. Diday (ed.), New Approaches in Classification and Data Analysis, Springer-Verlag, 300-307.

[Lassila y Webick, 1999] Lassila, O., Webick, R. (1999) Resource Description Framework (RDF) Model and Syntax Specification. W3C Recommendation, Jan. 1999,www.w3.org/TR/PR-rdf-syntax (current Jan. 2002).

[Leacock y Chodorow, 1998] C. Leacock and M. Chodorow, "Combining local context and WordNet similarity for word sense identification," In WordNet: An Electronic Lexical Database, The MIT Press, 1998, pp. 265–283.

[Levenshtein, 1965] V. I. Levenshtein. Binary codes capable of correcting deletions and insertions and reversals. Doklady Akademii Nauk SSSR, pages 845–848, 1965.

[Lin, 1998] Lin, D.. An Information-Theoretic Definition of Similarity. In 15th International Conference on Machine Learning, Morgan Kaufmann, 1998, 296—304. San Francisco, CA, (1998)

[Lipscomb, 2000] Lipscomb, C. E. 2000. Medical Subject Headings (MeSH). Bull Med Libr Assoc 88(3): 265-266.

[Madhavan *et al*., 2001] Madhavan, J.; Bernstein, P.; Rahm, E. Generic schema matching with Cupid. In Proceedings of the 27th International Conference on Very Large Data Bases, Roma, Italy, 11–14 September 2001; pp 49–58.

[Maedche y Staab, 2001] Maedche, A., Staab, S. (2001) Ontology learning for the Semantic Web. IEEE Intelligent Systems, 16(2):72-79.

[Mamdani, 1974] Mamdani, E. H. (1974) Applications of fuzzy algorithm for control a simple dynamic plant. Proceedings of the IEE 121(12), 1585-1588.

[Marzal y Vidal, 1993] Marzal A, Vidal E. Computation of normalized edit distance and applications. IEEE Trans. Pattern Anal. and Machine Intelligence 1993; PAMI-2(15): 926–932.

[McCallum y Nigam, 1998] McCallum, A.; and Nigam, K. 1998. A Comparison of Event Models for Naïve Bayes Text Classification. AAAI-98 Workshop on "Learning for Text Categorization".

[Meilicke y Stuckenschmidt, 2007] C.Meilicke and H. Stuckenschmidt. Analyzing mapping extraction approaches. In Proceedings of the Workshop on Ontology Matching, Busan, Korea, 2007.

[Melnik *et al*., 2002]. S. Melnik, H.G. Molina and E. Rahm: Similarity Flooding: A Versatile Graph Matching Algorithm, In Proc 18th Int'l Conf. Data Eng. (ECDE'02) (2002) 117-128.

[Michalewicz y Janikow, 1991] Z. Michalewicz, C.Z. Janikow (1991). Handling constraints in genetic algorithms. Proceedings of the Fourth International Conference on Genetic Algorithms, 151-157.

[Michalewicz, 1992] Z. Michalewicz (1992). Genetic Algorithms + Data Structures = Evolution Programs, Springer-Verlag, Berlin Heidelberg.

[Miles y Brickley, 2005a] Alistair Miles and Ban Brickley. Skos core guide. Technical report, World Wide Web Consortium (W3C), http://www.w3.org/TR/2005/swbp-skos-coreguide, 2005.

[Miles y Brickley, 2005b] Alistair Miles and Ban Brickley. Skos core vocabulary. Technical report, World Wide Web Consortium (W3C), http://www.w3.org/TR/2005/swbp-skos-core-spec, 2005.

[Mitchell, 1997] Mitchell, T (1997). Machine Learning, McGraw Hill.

[Mizumoto y Zimmermann, 1982] Mizumoto, M., Zimmermann, H., Comparison of fuzzy reasoning methods, Fuzzy Sets and Systems, Vol. 8, pp. 253-283, 1982.

[Monge y Elkan, 1996] Monge, A., and Elkan, C. 1996. The field-matching problem: algorithm and applications. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining.

[Muhlenbein, 1989] Muhlenbein, H. (1989). Parallel genetic algorithms, population genetics and combinatorialoptimization. Proceedings of the Third International Conference on Genetic Algorithms, 416-421.

[Nelson, 1965] Nelson, Theodor Holm. "A File Structure for the complex, the changing and the indeterminate". En: ACM 20th National Conference, 1965.

[Neuhaus y Compton, 2009] H. Neuhaus and M. Compton The Semantic Sensor Network Ontology: A Generic Language to Describe Sensor Assets. In AGILE Workshop Challenges in Geospatial Data Harmonisation, 2009.

[Niles y Pease, 2001] I. Niles and A. Pease. Towards a standard upper ontology. In Proceedings of the International Conference on Formal Ontology in Information Systems (FOIS), pages 2–9, 2001.

[NLM, 2009] National Library of Medicine (2009). "Chapter 5 - Semantic Networks". UMLS Reference Manual. Bethesda, MD: U.S. National Library of Medicine, National Institutes of Health.

[Noessner *et al*., 2010] Noessner, J., Niepert, M., Meilicke, C. and Stuckenschmidt, H. (2010). Leveraging Terminological Structure for Object Reconciliation. The Semantic Web: Research and Applications, p 334–348.

[Noy y Musen, 1999] Noy, N. F, Musen, M. A. (October 1999): SMART: Automated Support for Ontology Merging and Align-ment. In: 12th Workshop on Knowledge Acquisition, Modelling and Management (KAW'99), Banff, Canada.

[Noy y Musen, 2001]. N.F. Noy, M.A. Musen: Anchor-PROMPT: using non-local context for semantic matching, In Proc. Of IJCAI2001 Workshop on Ontology and Information Sharing, (2001) 63-70.

[Noy y Musen, 2002] Noy, N. F, Musen, M. A. (August 2002): PROMPTDIFF: A Fixed-Point Algorithm for Compar-ing Ontology Versions. In: 18th National Conference on Artificial Intelligence (AAAI'02), Edmonton, Alberta, Canada.

[Noy y Musen, 2003] Noy, N. F, Musen, M. A. (2003): The PROMPT suite: Interactive tools for ontology merging and mapping. International Journal of Human-Computer Studies, 59(6), pp. 983–1024

[O'Neill, 2003] E. T. O'Neill, B. F. Lavoie, R. Bennett. Trends in the Evolution of the Public Web. D-Lib Magazine, Volume 9 Number 4, April 2003. Disponible en http://www.dlib.org/dlib/april03/lavoie/04lavoie.html.

[Obitko, 2007] Obitko, Marek, "Introduction to ontologies and semantic web", 2007. http://www.obitko.com/tutorials/ontologies-semantic-web

[Oommen, 1987] Oommen BJ. Recognition of noisy subsequences using constrained edit distances. IEEE Trans. Pattern Anal. and Machine Intelligence 1987; PAMI-9(5): 676–685.

[Pan *et al*., 2005] Pan, R., Ding, Z., Yu, Y., Peng, Y.: A Bayesian Network Approach to Ontology Mapping. The Semantic Web – ISWC 2005, Vol. 3729/2005, pp. 563—577. Springer Berlin / Heidelberg (October 2005)

[Pearson, 1900] Pearson, K. On the Criterion that a given system of deviations from the probable in the case of correlated system of variables is such that it can be reasonable supposed to have arisen from random sampling, Phil. Mag.,1900, 50, 157-172.

[Petrakis *et al*., 2006] G. Euripides, M. Petrakis, Giannis Varelas, A.H.P.R.: Design and evaluation of semantic similarity measures for concepts stemming from the same or different ontologies. In: In 4th Workshop on Multimedia Semantics (WMS'06), pp. 44–52 (2006)

[Porter, 1980] Porter, M.F. (1980) An Algorithm for Suffix Stripping, Program, 14(3): 130–137.

[Rahm *et al*., 2004] E. Rahm, H. H. Do, and S. Maßmann. Matching large XML schemas. SIGMOD Record, 33(4):26–31, 2004.

[Rahm y Bernstein, 2001] Erhard Rahm and Philip Bernstein. A survey of approaches to automatic schema matching. VLDB Journal, 10(4):334–350, 2001.

[RDF, 2004] Resource Description Framework (RDF): Concepts and Abstract Syntax. W3.org.

[RDFS, 1998] RDF Vocabulary Description Language 1.0: RDF Schema W3C Recommendation.

[Reeves, 1993] C. Reeves (1993). Modern Heuristic Techniques for Combinatorial Problems, Blackwell Scientic Publications.

[Refaeilzadeh *et al*., 2008] Payam Refaeilzadeh, Lei Tang, Huan Lui. K-fold Cross-Validation, Arizona State University, 6 de noviembre de 2008

[Resnik, 1995] Resnik, P.: Using information content to evaluate semantic similarity in a taxonomy. In: IJCAI, pp. 448–453 (1995).

[Resnik, 1999] Resnik, P.: Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. Journal of Artificial Intelligence Research 11, 95–130 (1999)

[Rijsbergen, 1979] Rijsbergen, V., C. J. (1979): Information Retrieval. Butterworths. Second Edition, London

[Roberts, 1986] Roberts, D.W. (1986) Ordination on the basis of fuzzy set theory. Vegetatio 66 (3): 123– 31.

[Rodriguez y Egenhofer, 2003] Rodriguez, M., Egenhofer, M.: Determining semantic similarity among entity classes from different ontologies. IEEE Transactions on Knowledge and Data Engineering 15(2), 442–456 (2003)

[Ross, 1976] Ross, S. (1976). A First Course in Probability. Macmillan.

[Salton *et al*., 1975] G. Salton , A. Wong , C. S. Yang, A vector space model for automatic indexing, Communications of the ACM, v.18 n.11, p.613-620, Nov. 1975

[Schaffer *et al*., 1989] J.D. Schaffer, R.A. Caruna, L.J. Eshelman, R. Das (1989). A study of control parameters affecting online performance of genetic algorithms for function optimization. J.D. Schaffer (ed.), Proceedings of the Third International Conference on Genetic Algorithms, Morgan Kaufmann, 51-60.

[Scharffe y de Bruijn, 2005] Scharffe, F. and J. d. Bruijn (2005). "A Language to Specify Mapping between Ontologies." Proceedings of the Internet Based Systems IEEE Conference (SITIS05): 267-271.

[Shvaiko y Euzenat, 2004] Shvaiko, Pavel and Euzenat, Jerome. A Survey of Schemabased Matching Approaches. Technical Report DIT-04-087, Informatica e Telecomunicazioni, University of Trento, 2004.

[Simons, 1987] Simons, P. (1987). Parts: A study in Ontology. Clarendon Press, Oxford.

[Sirag y Weisser, 1987] D.J. Sirag, P.T. Weisser (1987). Toward a unified thermodynamic genetic operator. Genetic Algorithms and their applications: Proceedings of the Second International Conference on Genetic Algorithms and Their Applications, 116-122.

[Smith, 1980] S. F. Smith. A learning system based on genetic adaptive algorithms. PhD thesis, Department of Computer Science, University of Pittsburgh, 1980.

[Sneath y Sokal, 1973] Sneath, P.H.A., Sokal, R.R., (1973), “Numerical Taxonomy: The Principles and Practice of Numerical Classification”, W.H. Freeman and Company, San Francisco.

[Sokal y Sneath, 1963] Sokal, R.R., Sneath P.H., (1963), “Principles of numeric taxonomy”, San Francisco, W.H. Freeman.

[Sorensen, 1948] Sørensen, T. (1948) A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on Danish commons. Biologiske Skrifter / Kongelige Danske Videnskabernes Selskab, 5 (4): 1-34.

[Sowa, 1991] Sowa John, editor. Principles of Semantic Networks: Explorations in the Representation of Knowledge. San Mateo: Kaufmann; 1991.

[Studer *et al*., 1998] Studer, R; Benjamins, R.; Fensel, D. "Knowledge Engineering: Principles and Methods". In: *Data and Knowledge Engineering, 1998, v.25, n.1-2, pp.161-197*.

[Suh y Van Gucht, 1987] Suh, J. Y. &. Van Gucht, D. (1987). Distributed genetic algorithms (Technical Report No. 225). Blooomington: Indiana University, Computer Science Department.

[Sussna, 1993] M. Sussna, "Word sense disambiguation for free-text indexing using a massive semantic network," In Proc. of the 2nd Intl Conf. on on Inform. and Knowl. Manage., pp. 67–74, 1993.

[Syswerda, 1991] G. Syswerda (1991). Schedule optimization using genetic algorithms. L. Davis (ed.). Handbook of Genetic Algorithms, Van Nostrand Reinhold, New York, 332-349.

[Takagi y Sugeno, 1985] Takagi, T. and M. Sugeno (1985). Fuzzy identification of systems and its application to modelling and control. IEEE Transactions on Systems, Man, and Cybernetics 15 (1). 116-132

[Tanimoto, 1957] Tanimoto, T.T. (1957) IBM Internal Report 17th Nov. 1957.

[Thrift, 1991] Thrift, P. (1991). Fuzzy Logic Synthesis with genetic algorithms. In: Proceedings 4th International Conference on Genetic Algorithms, Morgan Kaufmann, 509-513.

[Truran *et al*., 2010] Truran, D., Saad, P. Zhang, M., Innes, K. (2010) SNOMED CT and its place in health information management practice. Health Information Management Journal Vol 39 (2):37-39 ISSN 1833-3583 (Print) 1833-3575 (Online)

[Tversky, 1977] Tversky, Amos (1977). "Features of Similarity". Psychological Reviews 84 (4): 327–352.

[Underbrink *et al*., 2008] A. Underbrink, K. Witt, J. Stanley and D. Mandl Autonomous Mission Operations for Sensor Webs. In AGU Fall Meeting Abstracts, pp. C5+, 2008.

[URI, 2009] URI Planning Interest Group, W3C/IETF (21 September 2001). "URIs, URLs, and URNs: Clarifications and Recommendations 1.0". Retrieved 2009-07-27.

[Van Heijst *et al*., 1997] Van Heijst, G., Schreiber, A. T., Wielinga, B. J. (1997). Using explicit ontologies in KBS development, International Journal of Human-Computer Studies, 45: 183-292.

[Watson Wey y Jun Jae, 2010] Watson Wey, K., Jun Jae, K. (2010). Eff2Match results for OAEI 2010. In: *Proceedings of the 5th International Workshop on Ontology Matching (OM-2010)*.

[Whitley, 1989] Whitley, D. (1989). The Genitor algorithm and selection pressure: Why rank-based allocation of reproductive trials is best. Proceedings of the Third International Conference on Genetic Algorithms, 116-121.

[Winkler, 1999] Winkler, W. E. 1999. The state of record linkage and current research problems. Statistics of Income Division, Internal Revenue Service Publication R99/04. Available from http://www.census.gov/srd/www/byname.html.

[WOT, 2005] Web Of Trust RDF Ontology -WOT- http://xmlns.com/wot/0.1/

[Wu y Palmer, 1994] Wu, Z. and Palmer, M. Web semantics and lexical selection. In Proceedings of the 32nd Annual Meeting of the Associations for Computational Linguistics, 1994, 133-138

[XML Signature WG, 2008] XML Signature WG: http://www.w3.org/Signature/

[XML, 1998] Extensible Markup Language (XML) 1.0 (First Edition). W3.org.

[XMLS, 2004] W3C XML Schema Definition Language (XSD) 1.1 Part 1: Structures. W3C Recommendation.

[Xu *et al*., 2010] Xu, P., K., Wang, Y., Cheng, L., Zang, T. (2010). Alignment Results of SOBOM for OAEI 2010. In: *Proceedings of the 5th International Workshop on Ontology Matching (OM-2010)*.

[Yates y Neto, 1999] R. B. Yates and B. R. Neto. Modern Information Retrieval. ADDISON-WESLEY, New York, 1999.

[Zadeh, 1965] Zadeh, L.A.: Fuzzy Sets. Information and Control 8, pp 338-353, 1965.

[Zadeh, 1973] Zadeh, L.A.: Outline of a new approach to the analysis of complex systems and decision processes. IEEE Transactions on Systems, Man, and Cybernetics 3, 28-44