

Comparador de Estudios de Neuroimagen Basado en Ontologías

Trabajo Fin de Master

Master en Inteligencia Artificial Avanzada Sistemas
Inteligentes de Diagnóstico, Planificación y Control

ETSI Informática - UNED

Autor: Santiago Timón Reina

Director: D. Mariano Rincón Zamorano

Septiembre 2012

Resumen

Los estudios de neuroimagen son una de las herramientas más valiosas de las que disponemos actualmente para investigar e intentar comprender el cerebro y su funcionamiento. Dada la naturaleza multidisciplinar de la investigación neurológica existe una gran variedad de disciplinas, contextos y métodos, publicados en diferentes revistas.

Al realizar un estudio de neuroimagen, los expertos pasan por una fase de documentación donde realizan búsquedas extensivas que proporcione literatura relacionada con su investigación, empleando para ello herramientas de búsquedas basadas en indexación de términos como PubMed.

BrainMap es una base de datos centrada en estudios de imagen funcional, que proporciona una terminología, un esquema de codificación y un reporte de coordenadas de activación de forma estandarizada. Su principal objetivo es el de ofrecer un servicio para la búsqueda de artículos de investigación por su contexto y contenido, más allá de los habituales sistemas de búsqueda por indexación de término y por palabras clave. Parte del esquema de codificación de BrainMap se ha construido bajo la ontología Cognitive Paradigm Ontology, abriendo así la posibilidad de representar y modelar estudios de neuroimagen bajo dicho esquema y siguiendo además una terminología controlada y unas relaciones específicas.

El uso de ontologías en el dominio biomédico es muy amplio, ofreciendo valiosos modelos de conocimiento en diversas disciplinas. Gracias a los estándares de la Web Semántica y su implantación dentro de esta corriente, estos modelos pueden compartirse e interoperar entre diferentes comunidades y aplicaciones.

Con estas herramientas, este trabajo se centra en extender la ontología CogPO para incluir el esquema relevante de BrainMap y conocimiento de ontologías de dominio biomédico, posibilitando el modelado de estudios de neuroimagen de forma simbólica y, siguiendo este aproximación basada en ontologías, definir e implementar métricas para comparar los estudios modelados. El resultado final es una herramienta que permite al experto introducir estudios de neuroimagen y compararlos con el conjunto almacenado en BrainMap.

Agradecimientos

Quisiera agradecer al tutor de este trabajo, el Dr. Mariano Rincón, toda la confianza, dedicación, apoyo y conocimiento aportado.

A la Doctora Rosa María Molina, “la experta”, por toda la ayuda ofrecida.

A mis padres, por estar siempre a mi lado.

Y en especial a Alba, por su paciencia, comprensión y el apoyo que siempre ha dado. Pero sobre todo por su ayuda, por estar ahí siempre que la he necesitado.

A todos, muchas gracias.

Índice

1. Estado de la cuestión	1
1.1. Introducción	1
1.1.1. Hipótesis	2
1.2. BrainMap	4
1.2.1. Esquema de codificación	5
1.2.2. Software	6
1.3. Ontologías y Web Semántica	8
1.3.1. Estructura y relaciones: RDF.	10
1.3.2. Ontologías y OWL	12
1.3.3. Razonamiento en ontologías	14
1.4. Bio-ontologías	15
1.4.1. Basic Formal Ontology	17
1.4.2. Cognitive Paradigm Ontology	20
1.4.3. Foundational Model of Anatomy (FMA)	27
1.5. Similitud y ontologías	28
2. Métodos	30
2.1. Ontología de BrainMap	30
2.1.1. Identificación de clases y relaciones	31
2.1.2. Adaptación a las ontologías de dominio	33
2.1.3. Implementación en OWL	38
2.1.4. Ontología en el contexto de aplicación	40
2.2. Relación diferida de localizaciones anatómicas con FMA	42
2.2.1. FMA y Talairach	44
2.2.2. Extracción de conceptos del FMA a partir de etiquetas Talairach	45
2.3. Métricas para comparación de ontologías	47

2.3.1.	Consistencia Lógica	47
2.3.2.	Grafo Semántico	50
2.3.3.	Grafo VSM	51
2.3.4.	Entropía de la ontología	54
2.4.	Búsquedas en BrainMap y datos empleados	56
3.	Desarrollo de la herramienta	59
3.1.	Análisis	60
3.1.1.	Casos de Uso	60
3.2.	Diseño	60
3.2.1.	Arquitectura	62
3.2.2.	Diseño de clases	63
3.3.	Interfaz de usuario	69
3.3.1.	Ventana principal	69
3.3.2.	Diálogo Subjects	71
3.3.3.	Diálogo Conditions	71
3.3.4.	Diálogo Experiment	71
3.3.5.	Ventana de resultados	74
4.	Resultados	76
4.1.	Modelado del estudio a evaluar según BrainMap	76
4.2.	Validación	78
4.2.1.	Conjunto de datos	79
4.2.2.	Etiquetado del experto	80
4.2.3.	Métricas de evaluación	81
4.2.4.	Resultados Consistencia Lógica	83
4.2.5.	Resultados Grafo Semántico	85
4.2.6.	Resultados Grafo VSM	88
4.2.7.	Evaluación de la validación	88

5. Conclusiones y trabajo futuro	91
6. Anexos	98
6.1. Manual de usuario de la aplicación	98
6.1.1. Ventana principal	98
6.1.2. Introducción y comparación de un estudio	99
6.1.3. Introducción de un grupo de sujetos	100
6.1.4. Introducción de condiciones	100
6.1.5. Introducción de experimentos	103
6.1.6. Cálculo y resultados	104

Índice de figuras

1. Esquema de codificación de BrainMap	5
2. Introducción de condiciones en Scribe	9
3. Pantallas de Sleuth	10
4. Ejemplo simple de grafo RDF	11
5. Taxonomía de SNAP	20
6. Ejemplo de Paradigma	22
7. Diagrama de clases de la ontología. Los conceptos importados aparecen en gris	32
8. Base de conocimiento de la aplicación	41
9. Diagrama de actividad del proceso de generación del Abox	43
10. Diagrama de actividad del proceso	46
11. Detalle de un experimento en el workspace de Sleuth	58
12. Arquitectura de la aplicación	63
13. Diagrama de clases del paquete es.uned.comparator.core	65
14. Diagrama de clases del paquete es.uned.comparator.core.metrics	67
15. Diagrama de secuencia del caso de uso Añadir condición	68

16.	Diagrama de secuencia de la operación calcular.	68
17.	Ventana principal	70
18.	Diálogo de sujetos	72
19.	Diálogo de condiciones	73
20.	Diálogo de experimentos	74
21.	Ventana de resultados	75
22.	Aplicación de procesado de texto de Sleuth	80
23.	Conjunto de datos para Consistencia Lógica	83
24.	Panel de resultados tras ejecutar la métrica	84
25.	Resultados de Consistencia Lógica para los relevantes	85
26.	Conjunto de datos para Grafo Semántico	86
27.	Panel de resultados para la métrica Semantic Graph	87
28.	Resultados de Consistencia Lógica con los elementos relevantes	87
29.	Panel de resultados con Grafo VSM seleccionado	88
30.	Ventana principal desglosada	98
31.	Ventana de introducción de sujetos	101
32.	Ventana de introducción de condiciones	102
33.	Ventana de introducción de experimentos	103
34.	Ventana de resultados	105

Índice de cuadros

1.	Tabla de conceptos	26
2.	Tabla de relaciones	27
3.	Tabla de relaciones	39
4.	Consulta SPARQL para extraer el concepto FMA	45
5.	Casos de Uso	61
6.	Sujetos	77

7.	Instructions	77
8.	Experiments a	78
9.	Experiments b	78
10.	Algunas valoraciones del experto	82
11.	Resultados de las métricas para Consistencia Lógica	85
12.	Resultados de las métricas para Grafo Semántico	88
13.	Resultados de las medidas de evaluación	89

1. Estado de la cuestión

1.1. Introducción

En las últimas décadas, con los avances tecnológicos tanto en sistemas de captación de imagen médica como de computación y almacenamiento, se ha dado una mayor disponibilidad de sistemas MRI y también una mayor capacidad de almacenamiento. Por ello, se ha producido una fuerte explosión en la corriente de investigación centrada en neurociencia y neuroimagen, y como consecuencia gran cantidad de estudios publicados.

Fruto de este auge, BrainMap¹ surgió hace más de dos décadas con el objetivo de crear y mantener una base de datos estandarizada conteniendo estudios de neuroimagen funcional. Su principal objetivo es ofrecer un sistema de búsqueda basado en las características concretas de este tipo de estudios, su contexto y su contenido. Para ello se diseñó un esquema de codificación de metadatos estandarizado, de forma que los estudios almacenados y aquellos candidatos a serlo, sigan un sistema descriptivo común, aumentando así las capacidades de búsqueda e interoperabilidad con agentes Software.

Con esta cantidad de datos, ha ido creciendo la necesidad de compartir el conocimiento neurocientífico. Según esta iniciativa continúa expandiéndose dentro de la comunidad de neuroimagen, aumenta la necesidad de ontologías completas capaces de abarcar este reto [1, 2]. Ya desde 2002 se vio la necesidad de encontrar algún medio de estandarización de los experimentos de comportamiento[3].

Esta necesidad ha propiciado que numerosas universidades y centros de investigación dedicaran tiempo y esfuerzo en desarrollar terminologías y modelos conceptuales cada vez más completos y refinados. Estos modelos, y más concretamente las ontologías, han ido evolucionando a la vez que las tecnologías de representación del conocimiento.

En aplicaciones de integración de información, la ontología puede tener varios objetivos: puede proporcionar un vocabulario extensible y definido formalmente para su uso en anotaciones semánticas, emplearse para describir la estructura de fuentes de información y la información que almacenan, y puede proporcionar un modelo detallado del dominio, disponible para peticiones complejas.

Actualmente, la disponibilidad de herramientas y sistemas de inferencia relacionados con la Web Semántica han contribuido a la rápida expansión y

¹<http://brainmap.org/>

adopción de sus estándares, como es el caso de OWL, convirtiéndose en un estándar de facto para el desarrollo de ontologías en la mayoría de los campos de la investigación científica (biología, medicina, geografía, geología, ingeniería...).

Las ontologías empleadas en esas aplicaciones suelen estar desarrolladas específicamente para su propósito, o sin pensar en ninguna aplicación en particular. Muchas de las ontologías son el resultado de colaboraciones en una comunidad con la intención de mejorar el modelado y compartición de conocimiento.

Gran parte de las ontologías OWL se encuentran disponibles en la Web, pudiendo ser accesibles desde cualquier lugar a través de la URI que la identifica y así incorporar su conocimiento en el desarrollo de otras ontologías. Sin embargo, en la práctica, las aplicaciones se construyen sobre una ontología o un conjunto de ellas predeterminadas que son suficientemente conocidas y proporcionan el conocimiento relevante de los dominios de aplicación.

En resumen, el contexto actual en neurociencia muestra un escenario donde la compartición e interacción entre diferentes grupos de investigación es cada vez mayor, forzando la necesidad de contar con herramientas y formatos estandarizados que maximicen la interoperabilidad y, por tanto, abriendo el camino hacia la innovación en el campo de la informática aplicada a la neurociencia o neuroinformática.

1.1.1. Hipótesis

La preparación de un estudio de neuroimagen funcional requiere una exhaustiva búsqueda de documentación y literatura relacionada con los experimentos que los expertos se disponen hacer. Para ello, las herramientas más extendidas son los servicios de indexación de artículos científicos y más concretamente, al tratarse del dominio médico, servicios como PubMed².

Para conseguir buenos resultados, los neurocientíficos deben aprender el manejo avanzado de estos sistemas, así como tener claros los términos de búsqueda más adecuados para el área y objetivos que persiguen (zonas cerebrales, área de investigación, paradigmas empleados). Esto deriva en los siguientes problemas:

1. Las limitaciones propias de los sistemas basados en términos: dificultad para buscar áreas cerebrales muy específicas por ejemplo.

²<http://www.ncbi.nlm.nih.gov/pubmed>

2. El sesgo que los propios investigadores introducen al realizar sus búsquedas.

BrainMap constituye una fuente de datos muy valiosa que contiene multitud de estudios de neuroimagen, almacenados de forma estructurada y procesable por sistemas automáticos. Este sistema abre la posibilidad de realizar búsquedas automáticas basadas tanto en el contexto como el contenido esperado de los estudios a realizar, lo que proporciona las siguientes ventajas:

1. Elimina el sesgo y variabilidad que el usuario inserta al realizar una búsqueda.
2. Suprime la vaguedad de la definición por palabras clave.
3. Proporciona una representación del estudio del experto.

Por tanto, y a partir de los datos de BrainMap, es posible desarrollar un sistema capaz de obtener estudios similares al que el investigador se encuentra desarrollando.

Para realizar este tipo de comparaciones es necesario codificar los estudios de forma que sean procesables por la herramienta. La primera opción es crear un modelo directo, que partiendo de un estudio realice peticiones directas a la base de datos. Esta opción sin embargo es poco adecuada, ya que es muy improbable obtener correspondencias directas, completamente iguales entre diferentes estudios. Por tanto, la opción más adecuada, es la de un sistema que capture, de forma cuantitativa, las diferencias entre el conjunto de metadatos de los estudios.

Las ontologías proporcionan un potente medio para la codificación de la información, generando un modelo de conocimiento en el que enmarcar los estudios como un conjunto de conceptos relacionados. Partiremos de la ontología CogPO y centraremos parte del trabajo en extender su conocimiento, de forma que incluya el conjunto necesario de metadatos de BrainMap para modelar los estudios dentro de la ontología. Esto nos permitirá comparar estudios según sus definiciones lógicas dentro de la ontología, ayudándonos para ello de un razonador.

Disponiendo de los estudios modelados bajo la ontología desarrollada, es posible también definir e implementar un conjunto de métricas que permitan identificar las similitudes entre estudios. Para ello nos centraremos en las métricas de similitud adaptadas a ontologías, generalmente dentro de

los métodos de alineamiento de ontologías, adaptándolas a las necesidades concretas de nuestro modelo de conocimiento.

Al resultar un problema similar a los problemas de recuperación de la información, el sistema se evaluará por medio de métricas Precision & Recall, realizando la validación un experto en neurociencia.

La hipótesis general que se establece como base de este trabajo de investigación se puede formular como:

Gracias a la base de datos BrainMap y a la estructuración y modelado del contexto y contenido de estudios de neuroimagen, es posible crear un sistema basado en ontologías que, representando los estudios bajo los modelos desarrollados y ejecutando un conjunto de métricas, evalúe la similitud o consistencia entre un estudio y los más cercanos contenidos en BrainMap.

1.2. BrainMap

Los estudios de imagen funcional tienen un amplio impacto y aceptación dentro de prácticamente cualquier área de investigación del cerebro y el comportamiento. Los sistemas neuronales que soportan el lenguaje, música, razonamiento, atención, emoción, etc. así como los cambios sufridos por la edad y alteraciones externas, están siendo mapeados por miles de investigadores en todo el mundo. Dado que la imagen funcional es una técnica de base que puede ser utilizada para cubrir múltiples objetivos en diferentes campos, los estudios resultantes son de naturaleza heterogénea.

Con los sistemas más utilizados, como PubMed, los investigadores se encuentran con el problema de que la búsqueda y recopilación de literatura relacionada con su estudio resulta una labor compleja, siendo necesario conocer bien las opciones disponibles y elegir bien las palabras clave de la búsqueda. Estos servicios de indexación de literatura son necesariamente genéricos, ya que sirven a un amplio abanico de datos científicos y clínicos. Por ello, el contexto (el cuándo y el por qué de un estudio) es indexado vagamente, basándose en un conjunto de palabras clave desestructuradas dependientes de la elección del autor. El contenido (los descubrimientos) queda aun peor plasmado. Para los estudios de imagen funcional los resultados representan activaciones y desactivaciones de áreas concretas del cerebro. La descripción de todo ello por medio de palabras clave es realmente complejo, lo que implica que no sea un sistema adecuado para encontrar estudios relacionados conceptualmente y aun menos para analizar su contenido[4].

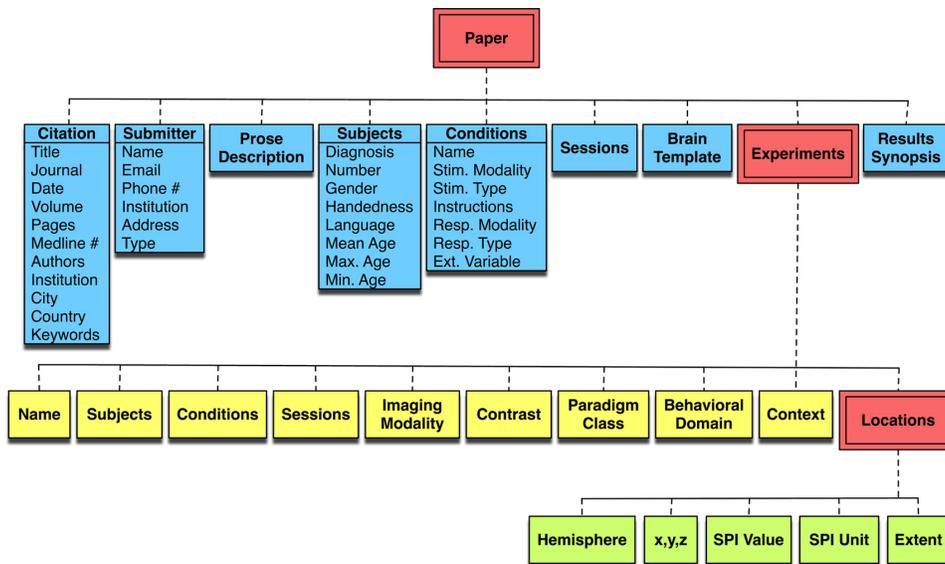


Figura 1: Esquema de codificación de BrainMap

La alternativa es un sistema de indexación que mapee tanto el contexto como el contenido en un espacio multidimensional de características, con un sistema propio de codificación.

Con esta idea surgió BrainMap, una base de datos de experimentos publicados de neuroimagen funcional y estructural con resultados mapeados espacialmente (x,y,z) en espacio Talairach o MNI. Tiene como objetivo el desarrollo de software y herramientas para la compartición y difusión de resultados de neuroimagen y permitir meta-análisis de estudios de la función del cerebro humano y su estructura en sujetos tanto sanos como enfermos.

Para conseguir este objetivo, BrainMap se vale de dos principios:

- Un esquema de codificación de experimento
- Y un espacio de coordenadas de activación estandarizado[5].

1.2.1. Esquema de codificación

BrainMap representa el contexto a partir de un sistema multiescala y multidimensional de palabras clave estructuradas que categorizan los experimentos en numerosos ejes. Estos ejes incluyen:

- La intención o contexto (los efectos de la enfermedad, mapeo de controles, efectos del fármaco...).
- Dominio de comportamiento (lenguaje, memoria...).
- Paradigma experimental.
- Modalidades sensoriales (visión, audición, tacto, gusto).
- Tipos de estímulos (palabras, objetos y patrones abstractos).
- Respuestas motoras por parte del cuerpo y su tipo.
- Tarea cognitiva (detección, discriminación y generación).
- Modalidad de adquisición de imagen.
- Categoría del diseño experimental (bloques o por eventos).
- Normalización de imagen.

El uso intensivo de palabras clave estructuradas es muy adecuado para la recuperación, rápida y exhaustiva, de todos los estudios en un dominio dado. También proporciona una forma eficiente y sencilla para comprender un estudio, ya que no todos los autores escriben con la suficiente claridad, lo que provoca que en ocasiones el texto no sea fácilmente comprensible, incluso por expertos en la materia.

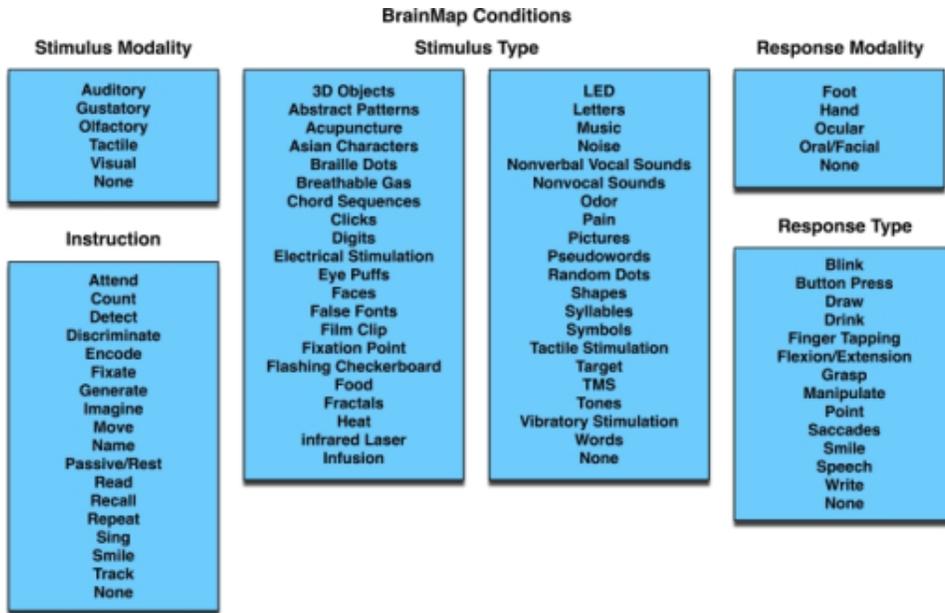
Con el esquema de codificación, es posible añadir una pequeña descripción del experimento, a la vez que se establecen estándares explicativos y convenciones de términos. Esta estrategia permite una rápida recuperación de estudios similares y su posterior filtrado por objetivos o criterios especificados.

Para facilitar el acceso a la base de datos de una forma amigable a un conjunto muy heterogéneo de usuarios, el proyecto BrainMap ha ido desarrollando un conjunto de herramientas Software, detalladas a continuación.

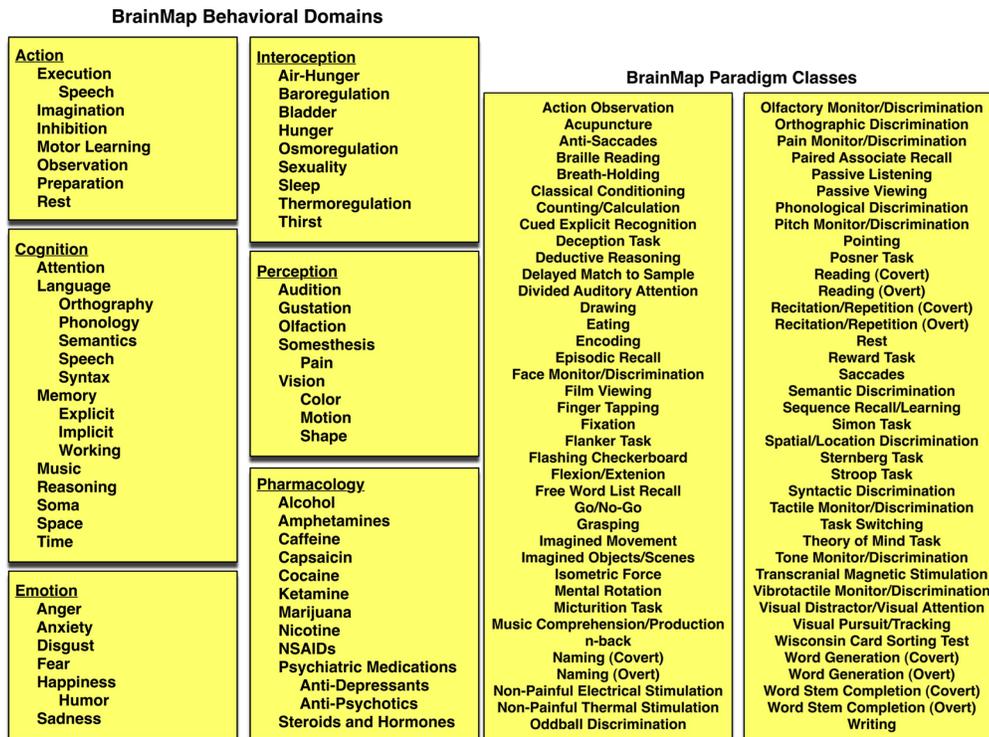
1.2.2. Software

Tras más de 20 años de desarrollo, BrainMap ha evolucionado hacia un proyecto más amplio cuyo software y datos han sido utilizados en numerosas publicaciones.

Para hacer accesible BrainMap se cuenta principalmente con tres aplicaciones Java de diferente propósito:



(a) Condiciones del experimento y sus propiedades



(b) Dominios de comportamiento

(c) Paradigmas

1. La adquisición y publicación en la base de datos de nuevos estudios
2. La búsqueda y recuperación de estudios
3. Herramienta para análisis de estimaciones de activación.

1.2.2.1. Scribe Scribe³ es la aplicación empleada para extraer las coordenadas de activación y los metadatos de un estudio de neuroimagen publicado, e introducirlo en la base de datos.

Consiste en una aplicación Java basada en escritorio, que guía al usuario para la introducción de los datos del estudio siguiendo el esquema de codificación de BrainMap.

Ha servido como modelo para la interfaz visual de la herramienta desarrollada, ya que para su procesado, es necesario contar con la mayoría de metadatos de forma similar a Scribe.

1.2.2.2. Sleuth Sleuth⁴ es la aplicación empleada para buscar estudios en BrainMap. Esta aplicación descarga los metadatos introducidos y muestra en una plano cerebral (x,y,z) las activaciones reportadas por el estudio en cuestión.

Debido a la restricción de realizar conexiones de forma programática a BrainMap, Sleuth ha resultado una herramienta clave para el desarrollo del trabajo. A partir de las búsquedas y el consiguiente detallado en el Workspace, se han obtenido los datos de los estudios que posteriormente se han modelado siguiendo la ontología desarrollada.

1.3. Ontologías y Web Semántica

El propósito final de la Web Semántica es permitir la compartición de datos de una forma efectiva entre grandes comunidades y que estos puedan ser procesados, además de por los propios usuarios, por herramientas de forma automática [6]. Lo que implica que un punto clave de la Web Semántica sea hacer que los agentes SW puedan entender la información presentada y las connotaciones semánticas que llevan implícitas. Esto se consigue a través de las ontologías, que dan significados definidos formalmente asociados a

³<http://brainmap.org/scribe/>

⁴<http://brainmap.org/sleuth/>

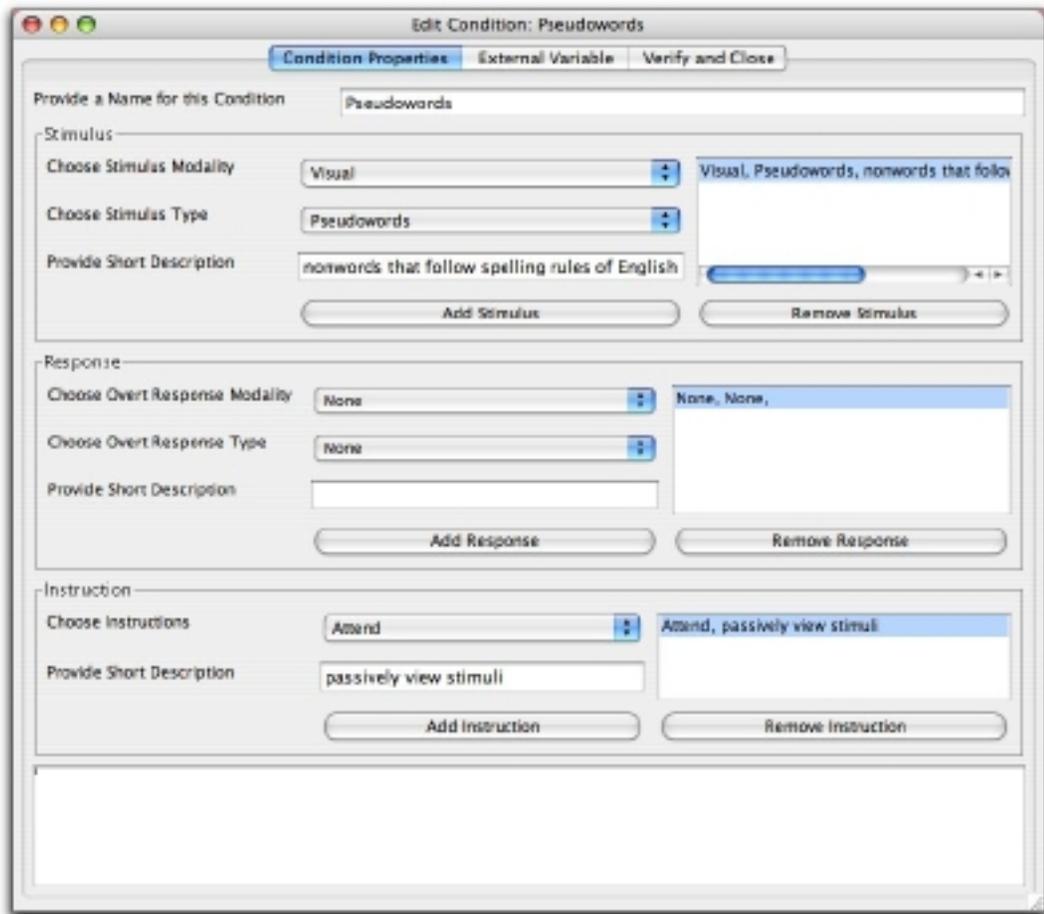
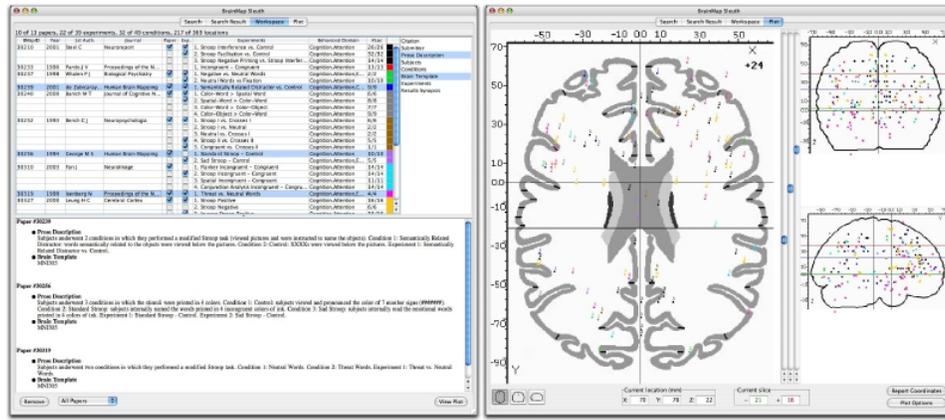


Figura 2: Introducción de condiciones en Scribe



(a) Apartado Workspace

(b) Panel Plot

Figura 3: Pantallas de Sleuth

los términos empleados en anotaciones, transformándolos en anotaciones semánticas. De esta forma, obtenemos la posibilidad de crear un conocimiento del dominio siguiendo interpretaciones semánticas, tal y como hacemos los humanos. Pero para crear un modelo coherente es necesario conocer y dominar las bases y herramientas necesarias para el modelado de conocimiento simbólico a partir del anotado semántico[7].

1.3.1. Estructura y relaciones: RDF.

Un detalle importante en el desarrollo de contenido compartible es su estructuración: la relación y jerarquía de sus fuentes. Una colección desestructurada es difícilmente compartible.

El lenguaje RDF⁵ (Resource Description Framework) es un lenguaje desarrollado para proporcionar un mecanismo flexible que facilita la descripción de los contenidos y las relaciones entre ellos. RDF es un modelo estándar para el intercambio de contenido en la Web.

Posee características que facilitan la mezcla de datos, incluso cuando los esquemas que los definen difieren, y soporta específicamente la evolución de los esquemas sin que sea necesario modificar las entidades que acceden a ellos.

⁵<http://www.w3.org/TR/rdf-concepts/>

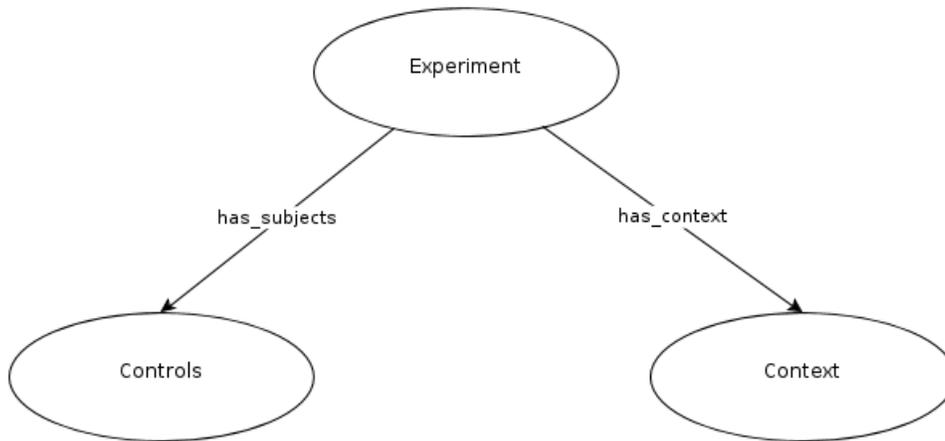


Figura 4: Ejemplo simple de grafo RDF

Una característica clave de RDF es el uso de los Internationalized Resource Locators (IRIs, una generalización de los URLs) para referirse a los recursos. De esta forma, se facilita enormemente la compartición y deslocalización de recursos a emplear en los modelos. Los IRIs, de forma similar a los URLs, son normalmente cadenas largas como <http://uned.es/Experiment>, por lo que existen varios mecanismos de abreviación.

La estructura interna de RDF es un grafo dirigido etiquetado, donde cada construcción sintáctica es una tripla de tres componentes: el sujeto, el predicado y el objeto. Esta tripla representa un arco (cuya etiqueta es el predicado) que conecta dos nodos (el sujeto y objeto), de forma que describe una relación binaria entre sujeto y objeto a través del predicado.

Un ejemplo: para describir la relación “El experimento tiene como sujetos Controls” tendríamos la tripla “Experiment hasSubjects Controls”, donde Experiment es el sujeto, hasSubjects es el predicado y Controls el objeto.

Un conjunto de triplas conforman un grafo RDF.

Para facilitar la distribución y compartición por la web se define el RDF/XML⁶. Dado el ejemplo de la figura anterior, añadiendo la propiedad hasContext:

```

<rdf:Description rdf:about="#Experiment">
  <has_subjects rdf:resource="#Controls"/>
  <has_context rdf:resource="#Context"/>

```

⁶<http://www.w3.org/TR/REC-rdf-syntax/>

</rdf:Description>

Donde #Experiment y #Controls y #Context son identificadores.

1.3.2. Ontologías y OWL

A pesar de ser reconocido como un lenguaje ontológico, RDF tiene varias limitaciones, como la incapacidad para definir restricciones de cardinalidad o uniones de clases. La necesidad de un lenguaje para la definición de ontologías más expresivo resultó en la propuesta de varios lenguajes, como SHOE, OIL y DAML+OIL.

Teniendo en cuenta que la arquitectura de la Web depende seriamente de estándares acordados, el consorcio W3C trabajó en la estandarización de un lenguaje ontológico, resultando el lenguaje OWL, el cual se encuentra actualmente en su segunda versión OWL2.

OWL aprovechó los trabajos anteriores en OIL y DAML+OIL y además está integrado con RDF. Esta integración implica una sintaxis basada en RDF, haciendo que las ontologías desarrolladas en OWL sean directamente accesibles desde aplicaciones web.

Una característica muy importante de OWL es que está basado en Lógica Descriptiva (DL en adelante), una familia de formalismos de representación del conocimiento basada en lógica descendiente de las Redes Semánticas y KL-ONE. Todos estos formalismos adoptan un modelo orientado a objetos.

Siguiendo la terminología de Platón o Aristóteles, el dominio se describía en términos de sus individuos, conceptos y roles. En RDF estos términos son referidos de otra forma: los conceptos son clases y los roles son las propiedades. Con esto, los individuos son los elementos básicos del dominio, las clases describen el conjunto de individuos que tienen características similares y las propiedades describen las relaciones entre pares de individuos (como `Experiment has_subjects Controls`).

Al igual que los nombres de clase atómicos, la lógica descriptiva permite como definición de clases la composición de clases atómicas y sus propiedades. Una DL dada se caracteriza por el conjunto de constructores que proporciona para construir las descripciones de clase.

OWL se basa en una DL muy expresiva llamada *SHOIN(D)* (el acrónimo derivado de las características del lenguaje). Los constructores de clase en OWL incluyen a los booleanos and, or y not, llamados en OWL `intersectionOf`, `unionOf` y `complementOf`, así como las formas restringidas de los

cuantificadores existencial (\exists) y universal (\forall), que son las restricciones someValuesFrom y allValuesFrom. En OWL la restricción del operador \exists (someValuesFrom) se usa para describir clases cuyas instancias están relacionadas por una propiedad a instancias de otra clase. En contraste, la restricción \forall (allValuesFrom) reduce los posibles objetos con una propiedad dada y se suele emplear como una restricción de alcance o tipo. Es importante tener en cuenta estas propiedades, pues son claves a la hora de modelar el conocimiento en una ontología.

Las características de la expresividad, resumidas en el acrónimo SHOIN, son las siguientes:

- \mathcal{S} es la abreviación de ALC con reglas transitivas. Esto implica que OWL permita que las propiedades sean transitivas. Por ejemplo si “*Henoc hasAncestor Cain*” y “*Cain hasAncestor Eva*” implica que “*Henoc hasAncestor Eva*”.
- \mathcal{H} implica la jerarquía de propiedades (roles en DL), por lo que se soportan propiedades con jerarquía.
- \mathcal{O} se refiere a los nominales. Capacitando las definiciones extensivas con el constructor oneOf.
- \mathcal{I} permite las propiedades inversas usando el constructor inverseOf. Una expresión muy útil de cara a inferir conocimiento.
- \mathcal{N} permite las restricciones de cardinalidad con los constructores minCardinality, maxCardinality y cardinality.
- (\mathcal{D}) indica el uso de los tipos de datos y valores de XML Schema.

Una ontología OWL consiste en un conjunto de axiomas. Como en RDF, los axiomas subClassOf y subPropertyOf se pueden emplear para definir una jerarquía de clases y propiedades. El término ontología suele usarse para referirse al esquema conceptual (TBox), pero en OWL una ontología puede consistir en una mezcla de axiomas del Tbox y Abox. En Lógica Descriptiva esto se conoce como una Base de Conocimiento (Knowledge Base). Las DLs tienen semánticas formales y pueden, de hecho, verse como un subconjunto de la lógica de primer orden, siendo sus individuos equivalentes a constantes, los conceptos como predicados unarios y los roles (propiedades) como predicados binarios. También pueden proporcionar un significado preciso sin

ambigüedades a las descripciones del dominio, lo que permite el desarrollo de algoritmos de razonamiento que proporcionen respuestas correctas a peticiones complejas sobre el dominio.

1.3.3. Razonamiento en ontologías

A primera vista, las ontologías se pueden concebir como algo similar a las bases de datos. A pesar de que existen algunas analogías claras, existen también muchas diferencias importantes. A diferencia de las bases de datos, en OWL la información perdida o no presente, se considera como desconocida en vez de falsa, y los axiomas se comportan como reglas de inferencia en vez de restricciones.

Otra diferencia importante es que OWL no tiene la restricción “Suposición de Nombres Únicos” (UNA Unique Name Assumption), que dos nombres sean diferentes no implica que se refieran a individuos diferentes. Por ello es posible inferir este hecho. Esta propiedad puede provocar inconsistencias en la ontología, porque a diferencia de los Sistemas Gestores de Bases de Datos (SGBD) las herramientas de modelado habitualmente no rechazan actualizaciones que resulten en una ontología parcial o totalmente inconsistente, normalmente sólo avisan de este hecho.

En las bases de datos, el esquema y restricciones pueden ser ignorados en tiempo de petición, pues en una instancia válida todas las restricciones tienen que estar ya satisfechas. Esto hace que la respuesta sea muy eficiente, puesto que para determinar si un valor es la respuesta a cierta pregunta, sólo es necesario mirar si está presente explícitamente en la base de datos.

En OWL el esquema juega un papel mucho más importante, por lo que se evalúa en tiempo de petición. Esta propiedad es muy potente ya que hace posible responder a peticiones tanto conceptuales como aquellas referidas a los datos. Se puede preguntar, por ejemplo, no sólo si Dumbledore es un Mago, sino comprobar también si cualquiera que tenga un Fénix como mascota debe ser un Mago. Este hecho hace que la respuesta a peticiones sea mucho más compleja, ya que en el caso del ejemplo sería necesario comprobar cada estado del mundo que sea consistente con los axiomas de la ontología.

En OWL, la respuesta a peticiones es por tanto análoga a la demostración de teoremas, y una respuesta suele ser denominada como una implicación. Por ello, OWL es más adecuado para aplicaciones donde el esquema juega un papel crucial, donde no sea posible afirmar que la información sobre el dominio sea completa pero que ésta tenga un alto valor.

Otro punto a tener en cuenta es el tamaño y complejidad que pueden llegar a tener las ontologías, como por ejemplo, la ontología de términos clínicos SNOMED, con más de 400.000 clases. Construir y mantener estas ontologías es muy costoso, por lo que disponer de medios y herramientas que faciliten las tareas de desarrollo y mantenimiento es muy importante.

Como hemos visto, las respuestas a peticiones no son procesos de chequeo, sino que requieren de razonamiento complejo. Por ello el razonamiento en ontologías es un punto clave tanto para el desarrollo de ontologías de alta calidad y su integración en las aplicaciones.

En vista de la complejidad del razonamiento en las ontologías OWL, sistemas de razonamiento DL muy optimizados como FaCT++ han probado ser muy eficientes (de hecho la disponibilidad de estos sistemas fue uno de los impulsos para basar OWL en DL). Herramientas de desarrollo como Protégé emplean estos razonadores para proporcionar feedback durante el desarrollo (como warnings sobre inconsistencias y sinónimos). En el caso de las inconsistencias es muy útil, ya que una clase inconsistente es aquella cuya descripción está “sobre-restringida” de forma que nunca podrá tener instancias. Suele ser una característica no intencionada en el diseño, derivado de interacciones entre axiomas. Para los sinónimos también lo es, ya que es posible que las descripciones en una ontología impliquen que dos clases tengan necesariamente el mismo conjunto de instancias, como que sean nombres alternativos para la misma clase. Esto es deseable en algunas situaciones, como para modelar el hecho de que “infarto de miocardio” es lo mismo que “ataque de corazón”. Pero también existe la posibilidad de que sea el resultado de interacciones entre descripciones inadvertidas, por lo que también es útil su advertencia.

El razonamiento también es importante cuando las ontologías se integran en las aplicaciones porque, como hemos visto, son necesarios para poder responder ante peticiones de datos como también conceptuales sobre la estructura del dominio.

1.4. Bio-ontologías

Hasta hace poco, el mayor problema de la bioinformática estaba relacionado con el almacenamiento, recuperación y análisis de datos moleculares como secuencias y estructuras proteicas. Sin embargo, según han ido avanzando los métodos aplicados al campo, estos datos se han ido volviendo cada vez más complejos, surgiendo la necesidad de disponer de la capacidad para describir

relaciones entre términos. Esto es debido a que el conocimiento biológico es de naturaleza compleja y por tanto no puede integrarse directamente en las bases de datos existentes, como por ejemplo datos moleculares.

Las ontologías ofrecen una solución para codificar este conocimiento de forma que este sea procesable por computadores, siendo su aplicación en el dominio biomédico relativamente nueva. Estas ontologías proporcionan por tanto una vía para asociar el conocimiento de un dominio biomédico con los datos almacenados en grandes bases de datos[8]. Sin embargo, las capacidades de las bio-ontologías no se limitan a relacionar datos, sino que también proporcionan grandes posibilidades a la hora de desarrollar aplicaciones en innumerables ámbitos de la investigación e implantación del dominio biomédico, siendo aplicables a cualquier situación que necesite de conocimiento complejo y/o razonamiento sobre él, como asistir en radiodiagnóstico[9], en planificación quirúrgica [10, 11] o en sistemas de gestión clínica[12] y asistencia al paciente[13].

El desarrollo e implantación de estas ontologías suponen una importante corriente de investigación que ha dado numerosos ejemplos. Algunos de ellos son el FMA (Anatomía), Gene Ontology (genética y procesos moleculares y biológicos) o Cognitive Atlas (ciencia cognitiva).

Con el objetivo de gestionar y fomentar el desarrollo, mantenimiento y difusión de las bio-ontologías, han surgido varias iniciativas entre las que cabe destacar el National Center for Biomedical Ontologies (NCBO)[14], un consorcio que une a investigadores en informática, biología, medicina y ontologías con el objetivo de permitir a los investigadores crear, difundir y gestionar conocimiento biomédico de forma computable empleando para ello ontologías.

El NCBO es especialmente importante por dos razones:

1. Proporciona herramientas como *bioportal*⁷, que permiten y facilitan el acceso a recursos basados en ontologías abstrayendo las tecnologías subyacentes.
2. Vigila porque las ontologías enviadas o pertenecientes al consorcio mantengan (dentro de lo posible) el mayor grado de ortogonalidad y alineamiento entre ellas, evitando solapamiento de términos.

⁷<http://bioportal.bioontology.org/>

Con todo esto en mente, parte de los objetivos que persigue el sistema a desarrollar se basan en la construcción de un modelo de conocimiento basado en un conjunto de bio-ontologías, concretamente en cuatro:

1. Basic Formal Ontology (BFO). Una upper-ontology muy utilizada en el conjunto de ontologías bajo el consorcio OBO[15], parte del NCBO.
2. Cognitive Paradigm Ontology. Una ontología que modela parte del esquema de codificación de metadatos de BrainMap.
3. Foundational Model of Anatomy. Modela el conocimiento anatómico del ser humano.
4. Cognitive Atlas. Modela el conocimiento relacionado con los procesos cognitivos y del comportamiento.

En las siguientes secciones se comentarán en detalle las características y necesidades que cubre cada una de ellas dentro del sistema a desarrollar.

1.4.1. Basic Formal Ontology

La ontología Basic Formal Ontology (BFO) surge de una orientación filosófica solapada con otras similares como DOLCE⁸ y SUMO⁹. Sin embargo, a diferencia de estas, se centra en la tarea de proporcionar una ontología formal (upper ontology) que puede emplearse en dar soporte a ontologías de dominio desarrolladas para la investigación científica, como biomedicina dentro del marco de la OBO Foundry. Por tanto BFO no contiene términos físicos, químicos, biológicos o similares que entren propiamente dentro de estos dominios de la investigación científica.

Las ontologías formales representan la realidad por medio de dos dimensiones esenciales. Primero, establecen cuáles son las categorías básicas de la realidad. Segundo, establecen las relaciones básicas que se sostienen entre los objetos pertenecientes a las categorías básicas de la realidad[16].

Las características de BFO se pueden resumir de la siguiente forma¹⁰:

1. BFO consiste en un conjunto de sub-ontologías (concebidas más propiamente con perspectivas de la realidad), de las cuales las más importantes son:

⁸<http://www.loa.istc.cnr.it/DOLCE.html>

⁹<http://www.ontologyportal.org/>

¹⁰<http://www.ifomis.org/bfo/overview>

- a) SNAP. Un conjunto de “ontologías-fotografías” (de ahí el nombre: snapshot) O_{ti} indexadas por tiempo.
- b) SPAN. Una única ontología videoscópica O_v .

Cada O_{ti} es un inventario de todas las entidades que existen en un momento dado.

O_v es un inventario (procedural) de todos los procesos que se desarrollan a lo largo del tiempo.

Cada O_{ti} es por tanto equivalente a la anatomía, mientras que O_v lo es a la fisiología.

Cada ontología de snapshot representa un análisis de las entidades que existen en un momento dado un instante presente. O_v es una vista general de la división de la totalidad de procesos. Procesos que son invisibles en la vista snapshot, lo que implica que las sustancias son invisibles en la vista SPAN.

2. Tanto SNAP como SPAN sirven como la base de un conjunto de sub-ontologías a diferentes niveles de granularidad. La misma porción de realidad que puede aparecer en situaciones con diferentes niveles de granularidad. En consecuencia, entidades de un nivel pueden agregar otras de otro nivel. Lo que importa como un proceso unitario en un nivel, puede ser parte de un proceso continuo a otro nivel.
3. Cada ontología representa una parte de la realidad dividida en categorías o universales.
4. Las ontologías descritas son parciales, es decir, son sólo una porción de la realidad que es visible a través de la ontología dada.
5. Las regiones espaciales y espaciotemporales admitidas por SNAP y SPAN son entidades abstractas que pueden o no alojar entidades concretas. Los emplazamientos/localizaciones y composiciones de SNAP y SPAN, referidos por expresiones como “en la habitación”, “en el pulmón”, etc. están sujetos a límites físicos específicos (paredes, suelos...). Los emplazamientos son porciones de espacio delimitados, que pueden estar limitados total (una habitación cerrada, una burbuja de aire dentro del cuerpo...) o parcialmente (un barril abierto, un nido...). Estos pueden mantener su identidad desde un instante al siguiente incluso cuando son proyectados en sucesión contra distintas regiones espaciales abstractas (de la misma forma que las sustancias mantienen su

identidad desde un instante al siguiente incluso a pesar de que son proyectadas en sucesión en distintos agregados de moléculas).

6. Las sustancias caen dentro del conjunto del mundo natural de cambios causales constantes. Por ejemplo, una cuasi-sustancia como un club de ajedrez pertenece al campo de las entidades socio-político-administrativas y cambia sólo como resultado de actos administrativos, como el de la admisión de un miembro nuevo.

Las cualidades, de forma similar, se encuadran dentro de los cambios constantes causales. Por ejemplo, una cuasi-cualidad como la deuda o la clasificación, pertenecen al conjunto de entidades socio-político-administrativas y cambia sólo como resultado de actos administrativos como el pago de una deuda. Todos los roles son cuasi-entidades: entidades que no están sujetas a cambios causales-energéticos constantes.

7. Las entidades dependientes dentro de la ontología SNAP están divididas entre relacionales (para entidades dependientes de un conjunto de entidades) y no relacionales (entidades dependientes de una entidad única).
8. SNAP distingue entre diferentes tipos de partes de sustancias:
 - a) fiat parts (partes delimitadas, como un brazo, una pierna, el tejido de la piel)
 - b) boundaries (límitrofes, la superficie de tu cuerpo)
 - c) structurals (estructurales, la organización de un cuerpo).

En virtud de las partes estructurales, es necesario que un cuerpo tenga algún color, temperatura y masa, incluso cuando el color específico, la temperatura y la masa del cuerpo no son partes del cuerpo directamente, sino dependent continuants que dependen del cuerpo para existir. De forma análoga, SPAN distingue partes estructurales de procesos, los cuales reflejan la diferencia entre, por ejemplo, los procesos cíclicos y no cíclicos, entre procesos de aceleración y deceleración, etc.

Como hemos visto, BFO reconoce una distinción básica entre dos tipos de entidad: entidades sustanciales o continuants, y entidades procedurales u occurrents.

La subontología SNAP representa a los continuantes, es decir, contiene las entidades que persisten a través del tiempo manteniendo su entidad.

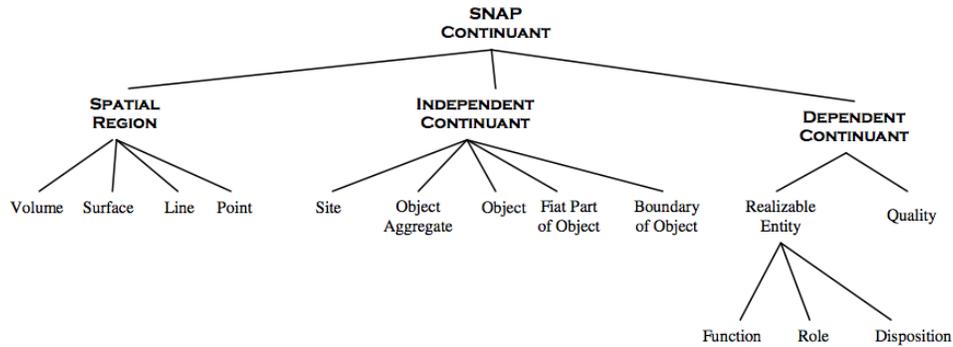


Figura 5: Taxonomía de SNAP

La importancia de la ontología BFO, más allá del sentido filosófico y metafísico que representa, reside en su amplia implantación dentro del conjunto de ontologías biológicas aceptadas o candidatas de la fundación OBO. Esta implantación implica que en muchos casos es deseable o directamente necesario seguir su modelo para extender el conocimiento contenido en ontologías a importar.

Dentro del conjunto de ontologías a utilizar en este proyecto, dos de ellas importan directamente la estructura de BFO, por lo que se adoptará este sistema de representación para incluir los nuevos términos que se precisen para cubrir los fines de representación del conocimiento.

1.4.2. Cognitive Paradigm Ontology

A partir de la taxonomía desarrollada y usada por el proyecto BrainMap, se desarrolló una ontología capaz de representar ciertas características de los paradigmas cognitivos empleados en la literatura de estudios fMRI y PET: “La Cognitive Paradigm Ontology” (CogPO)¹¹[17], que está siendo desarrollada para ser compatible con la Basic Formal Ontology y adaptarse dentro de lo posible con ontologías mayores como RadLex, NeuroLex, FMA, etc.

Los componentes clave de CogPO incluyen la representación de condiciones experimentales centradas en el estímulo presentado, las instrucciones dadas y las respuestas solicitadas. El uso de terminologías diferentes e incluso

¹¹<http://www.cogpo.org/>

enfrentadas puede impedir ciertos descubrimientos científicos. La categorización de los paradigmas de acuerdo a los estímulos, respuesta e instrucciones, ha mostrado que permiten técnicas de recuperación de datos avanzadas, por búsqueda de similitudes y contrastes a través de múltiples niveles de paradigmas.

El objetivo de CogPO es desarrollar, evaluar y distribuir una ontología de dominio de paradigmas cognitivos para su aplicación y uso dentro de la comunidad de neuroimagen. Para los propósitos de CogPO, el desarrollo se centra en representar cuál era la intención de los expertos al realizar un estudio y qué instrucciones se dieron a los sujetos, no lo que el sujeto percibió del estímulo o qué pensaban que estaban haciendo.

En vistas al futuro, CogPO pretende incluir más conceptos de BrainMap, de forma que sea posible representar con más detalle los experimentos. Sin embargo no está desarrollado aun, por lo que será necesario, con la Cognitive Paradigm Ontology como principal punto de partida, modelar el conocimiento del esquema de BrainMap que aun no se encuentra presente.

A continuación se desglosan los conceptos y relaciones más importantes, así como su significado y utilidad de cara a modelar experimentos de neuroimagen.

Estructura de CogPO

Como hemos dicho, CogPO pretende modelar los paradigmas cognitivos usados en estudios de neuroimagen funcional, basándose para ello en BFO y otras ontologías de referencia. Vamos a mencionar y resumir los conceptos principales de la ontología:

Behavioral Experimental Paradigm:

El primer término importante en CogPO es Behavioral Experimental Paradigm. Está definido como una subclase del término de la ontología IAO “planned process”, ya que se refiere a un proceso planificado que ha ocurrido, describiendo qué sujetos estaban dispuestos para escuchar, ver o realizar una tarea durante el experimento.

El paradigma cubre lo que se describe en artículos de neuroimagen como Tarea. No captura lo que puede haber pasado a un sujeto durante la sesión, como que el sujeto se haya dormido y respondido, o fallara el proyector de video y no se presentara el estímulo, etc. Estas desviaciones del protocolo quedan fuera del ámbito de CoPO.

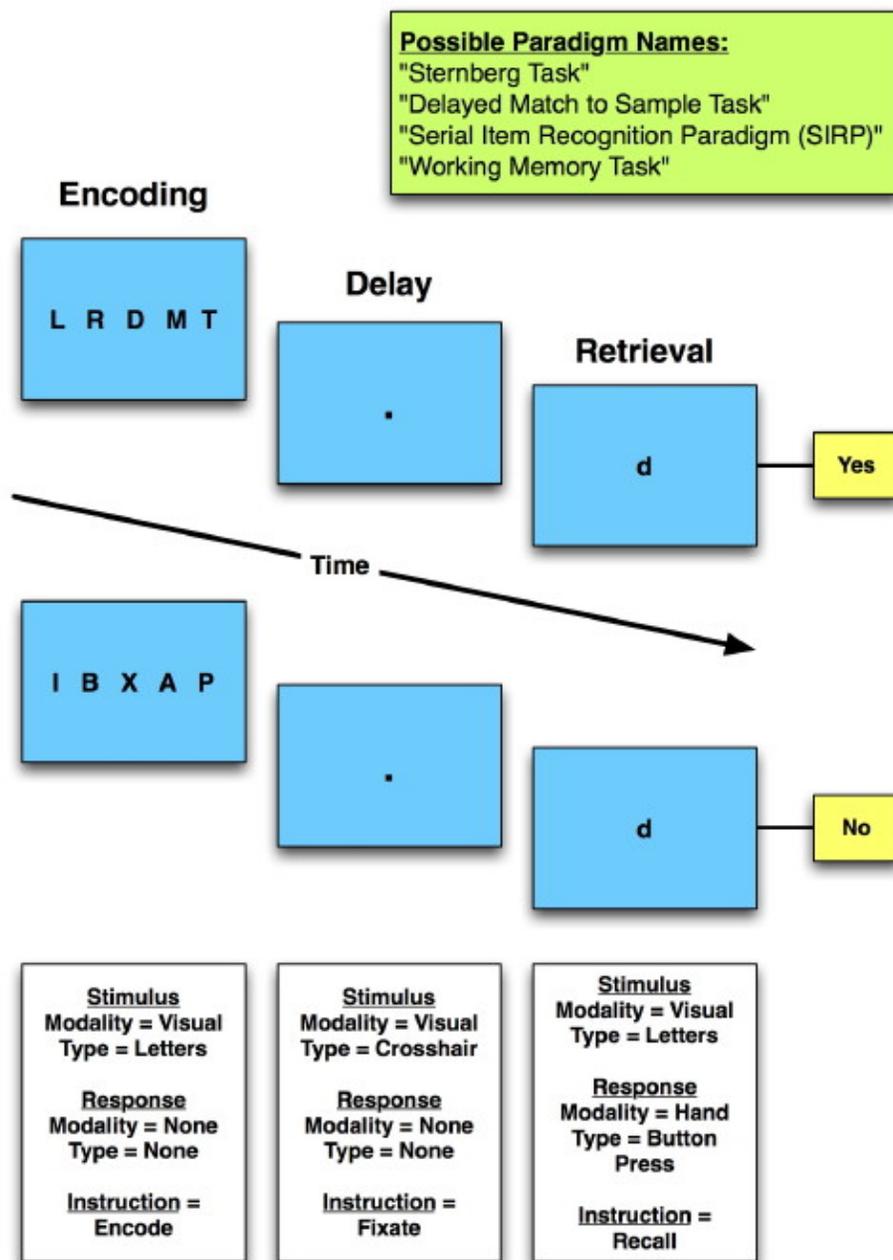


Figura 6: Ejemplo de Paradigma

Las subclases de Behavioral Experimental Paradigm son paradigmas reconocidos por los metadatos de BrainMap. Estos se pueden definir como restricciones lógicas cuando es posible aplicarlas. La lista de subclases se puede extender según la comunidad vaya reconociendo paradigmas estandarizados, por ejemplo: si se reconoce qué experimentos con tipos particulares de combinaciones de estímulo/respuesta/instrucción siempre activan el cíngulo anterior, esto se podría definir como una clase de paradigma.

Behavioral Experimental Paradigm Condition:

Dentro de una clase reconocida de paradigma, se dan variaciones en el estímulo empleado, en las respuestas solicitadas o en las instrucciones que se dan al sujeto. Por ello, el eje de CogPO es la clase Behavioral Experimental Paradigm Condition, otro “planned process” que debe incluir al menos:

- un estímulo
- una instrucción
- y un tipo de respuesta.

Siguiendo los estándares definidos en el diseño experimental en neuroimagen, cada paradigma experimental debe tener al menos dos condiciones. Por ejemplo, la clase Auditory Oddball Paradigm contiene paradigmas que tienen al menos dos condiciones, ambos teniendo un estímulo con modalidad auditiva, uno de los cuales requiere una respuesta y otro sin necesitarla.

En general, los estudios experimentales no usan repetidamente los mismos paradigmas, sino que desarrollan unos nuevos, algunos de los cuales pueden no encajar claramente dentro de una clase reconocida. Los paradigmas que se usan en un sólo estudio y nunca se vuelven a usar, se pueden describir como instancias de Behavioral Experimental Paradigm, con sus condiciones representadas como instancias de Behavioral Experimental Paradigm Condition. Una nueva clase de paradigma se puede definir cuando existen experimentos suficientes con condiciones similares descritas. Dentro de BrainMap, a partir de cinco instancias descritas y usando la misma combinación de condiciones se puede crear una nueva clase de paradigma.

Esta redundancia en condiciones (estímulo, instrucción, respuesta) será una de las situaciones a explotar por la herramienta.

Stimulus: El término estímulo es particularmente difícil de definir, ya que se emplea en muchos dominios diferentes. De hecho, cualquier cosa puede considerarse como estímulo, ya que cualquier cosa puede provocar

una respuesta por el sujeto, ya sean las luces de la habitación (algo irrelevante para el experimento) o la dosis de un fármaco (crítico en un experimento sobre los efectos farmacológicos).

Por ello, en CogPO se define la clase Stimulus Role, es decir, el papel que juega al ser empleado deliberadamente en el diseño experimental para provocar una respuesta del sujeto.

La clase Stimulus es una subclase de Object Aggregate de BFO, y es la clase que define las cosas que pueden hacer el papel de estímulo (tonos, música, caras, imágenes, pensamientos internos, etc.).

Esto se encuentra actualmente recogido en la lista finita de tipos de estímulos que el esquema de BrainMap reconoce, pero se puede extender cuando sea necesario. Existen dos subclases inmediatas de Stimulus:

1. Explicit Stimulus, el cual se genera bajo control del científico y existe de forma externa al sujeto al menos en algún momento.
2. Implicit Stimulus, el cual es generado por el sujeto.

Estímulos explícitos como las imágenes presentadas visualmente o la estimulación táctil son los más frecuentes en la literatura, sin embargo, los estímulos implícitos son importantes en estudios sobre fluctuaciones en el estado de reposo, memoria, meditación, etc.

Cada Explicit Stimulus tiene al menos un Stimulus Modality, el cual se refiere al sistema sensorial objetivo del estímulo. Por ejemplo, los tonos sonoros tiene la modalidad sensorial Auditory Modality, mientras que comida tendría la modalidad sensorial Gustatory Modality. Sin embargo, los videoclips tendrán tanto modalidad visual como auditiva.

Un Stimulus Modality es una Quality de BFO, ya que es la cualidad de estimular un sistema sensorial concreto, y no es el sistema sensorial en sí.

La relación entre un Explicit Stimulus y su Stimulus Modality se representa en CogPO por medio de la relación has_stimulus_modality, la cual es una subpropiedad de la relación de OBO bearer_of, que relaciona objetos y cualidades, roles o funciones de forma más general. La relación has_stimulus_modality queda restringida al dominio de Stimulus y el alcance de Stimulus Modality. Esta restricción se hace por medio de los axiomas rdfs:domain y rdfs:range, que especifican de qué tipo son el sujeto y objeto de una propiedad.

Response: El papel de la respuesta se lleva a cabo con la clase Response, el comportamiento explícito o implícito del sujeto en la condición experimental.

Una respuesta puede ser:

1. Overt Response: una respuesta hecha con una parte del cuerpo que se observable, como pulsar un botón.
2. Covert Response: realizada de forma interna, como nombrar objetos pensando.

Una respuesta puede ser considerada como la combinación de procesos desarrollados sobre el tiempo, por lo que se ha especificado como una subclase del concepto de BFO Process Aggregate.

Cada Overt Response tiene una Response Modality que debe ser la parte del cuerpo empleado en ejecutar la respuesta.

Por tanto, la subclase Response Button Press debe tener como Response Modality la parte del cuerpo que se ha usado para pulsar el botón.

Estos términos de partes del cuerpo se han importado desde el FMA empleando el proceso MIREOT[18] y tienen como superclase fiat object part de BFO. Un fiat object part es una parte de un objeto que no se encuentra limitada por discontinuidades físicas: el límite preciso donde la muñeca se convierte en mano o el pie en tobillo no está determinado.

Instructions: El término Instructions es de tipo Action Specification de la ontología IAO. Actualmente son definiciones de una palabra importados del listado de BrainMap (Attend, Discriminate, Encode...) y cada uno incluye una definición y ejemplos. No tienen ninguna otra limitación o relaciones en este momento, sin embargo pueden ser restringidos a tipos de respuesta particulares o condiciones si es necesario.

Las relaciones empleadas en CogPO incluyen las siguientes desde la ontología Relationship Ontology (RO): has_part, has_participant y la extensión a RO para incluir bearer_of e inherent_in para las cualidades y los roles. Las relaciones entre Behavioral Experimental Paradigm y algunos Behavioral Experimental Paradigm Condition es has_part. Se propusieron nuevas relaciones que son específicas a los dominios y rangos de los términos de CogPO: una Behavioral Experimental Paradigm Condition has_stimulus algún tipo de estímulo que viene de la clase Stimulus, y has_response de algo

Concepto	Superclase	Definición	Restricciones
Behavioral Experimental Paradigm	OBI:Planned_process BFO:procesual_entity	Describe los aspectos de comportamiento del experimento: qué estímulo es presentado al sujeto, bajo qué condiciones y qué respuestas se van a dar	Debe tener como parte has_part al menos un Behavioral Experimental Paradigm Condition
Behavioral Experimental Paradigm Condition	OBI:Planned_process BFO:procesual_entity	Una combinación planificada de estímulos e instrucciones relacionadas con las respuestas de los sujetos del experimento	Al menos una instrucción, un estímulo y una respuesta
Stimulus Role	BFO:role	El role de un estímulo en un experimento es atribuido al objeto que se presentan al sujeto en un experimento de forma controlada	
Response Role	BFO:role	El role de la respuesta se relaciona con el comportamiento (explícito o implícito) que se percibe del sujeto en una condición experimental	
Stimulus	BFO:ObjectAggregate	El objeto o conjunto de objetos, internos o externos al sujeto, con los que se pretende generar una respuesta en el sujeto como parte de la condición experimental	
Response	BFO:ObjectAggregate	El comportamiento intemo o externo que se percibe del sujeto en una condición experimental	
Instructions	IAO:'action specification', BFO: generically_independent_continuant	Son la entidad portadora de la información que configura las reglas para el comportamiento deseado de los sujetos. Una directriz explícita que guía el comportamiento de los sujetos durante el experimento.	
Stimulus Modality	BFO:quality	La cualidad de la percepción sensorial de un estímulo explícito	Inheres_in only Explicit Stimulus
Response Modality	BFO:FiatObjectPart	partes del cuerpo empleados para realizar acciones que juegan el papel de una respuesta	

Cuadro 1: Tabla de conceptos

que venga de la clase Response. Has_stimulus y has_response son subpropiedades de has_part. Has_stimulus_modality, que describe las relaciones entre estímulos y su modalidad, es una subpropiedad de bearer_of, mientras que has_response_modality es una subpropiedad de has_part (ya que las Response Modalities son partes del cuerpo para realizar la respuesta).

Como resumen de la estructura de CogPO, las tablas 1 y 2 contienen los conceptos y relaciones más importantes.

Relación	Superpropiedad	Dominio	Rango
Has_stimulus	Has_participant	Behavioral Experimental Paradigm Condition	Stimulus
Has_response_type	Has_part	Behavioral Experimental Paradigm Condition	Response
Has_instructions	Has_participant	Behavioral Experimental Paradigm Condition	Instructions
Has_stimulus_modality	Bearer_of	Explicit Stimulus	Stimulus Modality
Has_response_modality	Has_participant	Overt Response	Stimulus Modality

Cuadro 2: Tabla de relaciones

1.4.3. Foundational Model of Anatomy (FMA)

El Foundational Model of Anatomy¹²[19] es una ontología de código libre disponible como base de conocimiento para sistemas biomédicos. Para ello consiste en la representación de clases y las relaciones necesarias para la representación simbólica de la estructura fenotípica del cuerpo humano. Forma parte de los recursos de información integrados en el marco distribuido de la Anatomy Information System de la Universidad de Washington.

A pesar de no ser una base de conocimiento puramente centrada en el conocimiento neuro-anatómico, su estudio es relevante al ser una de las primeras y más importantes, cuyo contenido e iniciativa han sentado base para innumerables pasados y futuros trabajos.

Su propósito es hacer disponible información anatómica modelada de forma simbólica con fines educativos, clínicos, biomédicos y todas las áreas médicas. Intenta asegurar consistencia y el empleo de estándares actuales para la representación de la anatomía, siendo una fuente de información no diseñada para un tipo de aplicación concreta.

Su diseño consta de los siguientes componentes:

1. Taxonomía anatómica. Clasifica entidades anatómicas según las características que comparten y por las que pueden distinguirse unas de otras.
2. Abstracción estructural anatómica. Especifica las relaciones espaciales y de oposición que existen en las entidades representadas en At.

¹²<http://sig.biostr.washington.edu/projects/fm/>

3. Abstracción de transformación anatómica. Especifica las transformaciones morfológicas de las entidades durante el desarrollo prenatal y el ciclo de vida posterior.
4. Meta-conocimiento. Especifica los principios, reglas y definiciones en función de qué clases y relaciones se representan en los otros tres componentes.

En cuanto a su contenido, el FMA se compone de aproximadamente 75000 clases y de unos 12000 términos; más de 2.1 millones de relaciones de más de 168 tipos de relaciones en un modelo simbólico coherente, siendo una de las fuentes más grandes de conocimiento anatómico digital.

El componente más completo es la taxonomía anatómica (At).

En la jerarquía, la clase de mayor nivel es la entidad Anatomical Structure. Estas entidades incluyen macromoléculas biológicas, células y sus partes, porciones de tejidos, órganos y sus partes, así como los sistemas orgánicos y las regiones corporales. Las moléculas biológicas se introdujeron principalmente para ilustrar la continuidad estructural para las estructuras principales, como el tórax, a las macromoléculas biológicas como myosina.

La ontología del FMA está implementada originalmente en Protégé Frame, sin embargo gracias al proyecto FMAInOWL existe representación en OWL con completo soporte actualmente[20].

En este trabajo, la parcela de conocimiento cubierta por el FMA es clara: proporciona los términos anatómicos derivados de las activaciones registradas en un estudio. De esta forma podemos relacionar directamente los experimentos con instancias del FMA.

Como se puede ver, el FMA es una ontología muy extensa, lo que lleva a problemas en abarcabilidad a la hora de llevar a cabo tareas de razonamiento. Por ello se utilizará un módulo conteniendo sólo el conocimiento de la anatomía cerebral, el NeuroFMA obtenido directamente de forma automática del propio FMA[21].

1.5. Similitud y ontologías

En el contexto de la Web Semántica, las ontologías se desarrollan para definir de forma inequívoca términos. De forma que estos términos puedan ser compartidos de forma interoperable entre diferentes sistemas. Sin embargo,

en la práctica, es muy común que cada sistema implemente una ontología propia, llevando a colisiones entre las definiciones que diferentes ontologías hacen sobre un mismo término.

En el contexto de las ontologías biomédicas este problema es aun más común. Esto es debido a que dominios diferentes suelen solaparse, provocando que ocurra también en las ontologías que pretenden modelar dicho conocimiento. Por ejemplo, la definición de célula en la ontología Cell Ontology con la definición existente en la ontología FMA.

Por ello, este problema viene acompañando al desarrollo e implantación de ontologías desde sus inicios, lo que ha llevado a crear una corriente de investigación centrada en detectar y buscar formas de solucionarlo: la alineación de ontologías u ontology matching.

En el dominio del procesamiento del lenguaje natural existen dos principales aproximaciones de medidas de similitud:

- Uno de ellos basado en Vector Space Models (VSM)[22].
- Y otro empleando la frecuencia de aparición de las palabras[23].

La mayoría de aproximaciones en medidas de similitud en ontologías se basan en la estructura jerárquica de los conceptos, la taxonomía [Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy]. Sin embargo es evidente que, dada la expresividad de OWL, esta aproximación no es suficiente, por lo que han surgido sistemas que tienen en cuenta las relaciones entre conceptos más allá de la herencia.

De esta forma, la obtención de las mejores correspondencias de conceptos entre ontologías se limita a comparar distancias entre los mismos. Algunas de las formas para obtener estas medidas son:

- Terminológica, comparando las etiquetas de las entidades.
 - String-based, haciendo la comparación terminológica a través de las diferencias estructurales.
 - Terminológica con léxicos, haciendo el matching con las relaciones encontradas en léxico.
- Estructura interna, comparando la estructura interna de las entidades (el valor de rango o cardinalidad de sus relaciones).

- Estructura externa, comparando las relaciones de las entidades con otras entidades.
 - Estructura taxonómica. Comparando la posición de las entidades dentro de una taxonomía.
 - Estructura externa con ciclos. Una comparación de estructura externa robusta frente a ciclos.
- Comparación extensa, comparando la extensión conocida de las entidades, como el conjunto de otras entidades que están ligadas a las mismas (como individuos de una clase).
- Comparación semántica, que compara las interpretaciones o modelos de las entidades.

2. Métodos

En esta sección se planteará la base teórica sobre la que se construirá la aplicación.

Para realizar el desarrollo necesitamos contar con una ontología que permita modelar los estudios de neuroimagen según el esquema de codificación de BrainMap. En la sección 2.1 se detalla el proceso para modelar una ontología de BrainMap basándonos en CogPO.

Una vez obtenida la ontología, es necesario pensar en cuáles serán las técnicas que nos permitan comparar los diferentes estudios modelados bajo esta ontología. La sección 2.2 describe las métricas planteadas para abordar este problema.

2.1. Ontología de BrainMap

Como hemos visto. CogPO cubre parte del esquema de metadatos definidos por BrainMap para modelar los paradigmas más aceptados dentro de la comunidad de neuroimagen. Sin embargo, la parte cubierta por CogPO es insuficiente, pues deja fuera gran parte de los términos necesarios para definir un estudio.

2.1.1. Identificación de clases y relaciones

A partir del conjunto completo del esquema de codificación de metadatos, se han identificado los términos o conceptos mínimos no presentes en CogPO con los que modelar un estudio de neuroimagen según BrainMap. Estos conceptos y su descripción son los siguientes:

Experiment representa un experimento parte de un estudio, por lo que la instancia de un estudio tendrá un conjunto de experimentos. Un experimento tiene como parte: un paradigma y un contexto, unos sujetos y un conjunto de localizaciones anatómicas resultado de las activaciones.

fMRI Study modela un estudio de neuroimagen, es decir, el concepto raíz del que surgirán las relaciones que dan el modelo lógico a un estudio dado.

Experiment Context representa el contexto del experimento. Tendrá como subclases el conjunto de contextos definidos en BrainMap.

Anatomical Location representa una entidad anatómica relacionada con la activación observada en un experimento.

Subjects define el concepto de sujetos (como conjunto) que participan en un experimento.

Los sujetos tienen las siguientes propiedades: una enfermedad, un género, un total, edad máxima, mínima y media.

Disease representa una enfermedad dentro del contexto de BrainMap. Tendrá como subclases las definidas en los metadatos.

El desarrollo de ontologías en OWL es muy similar al Diseño Orientado a Objetos, por lo que resulta de utilidad emplear las mismas herramientas que ofrece este paradigma, como UML. A la hora de pensar en los conceptos que poblarán la ontología podemos valernos del diagrama de clases para identificar las clases, subclases y sus relaciones. La figura 7 muestra el diagrama de clases generado, donde se encuentran los conceptos a modelar junto aquellos importados de CogPO y BFO que serán superclase de los nuevos, así como las relaciones que tendrán los conceptos entre ellos; es decir, las propiedades a modelar.

Por tanto, como se aprecia en el diagrama, el conjunto de propiedades será el siguiente:

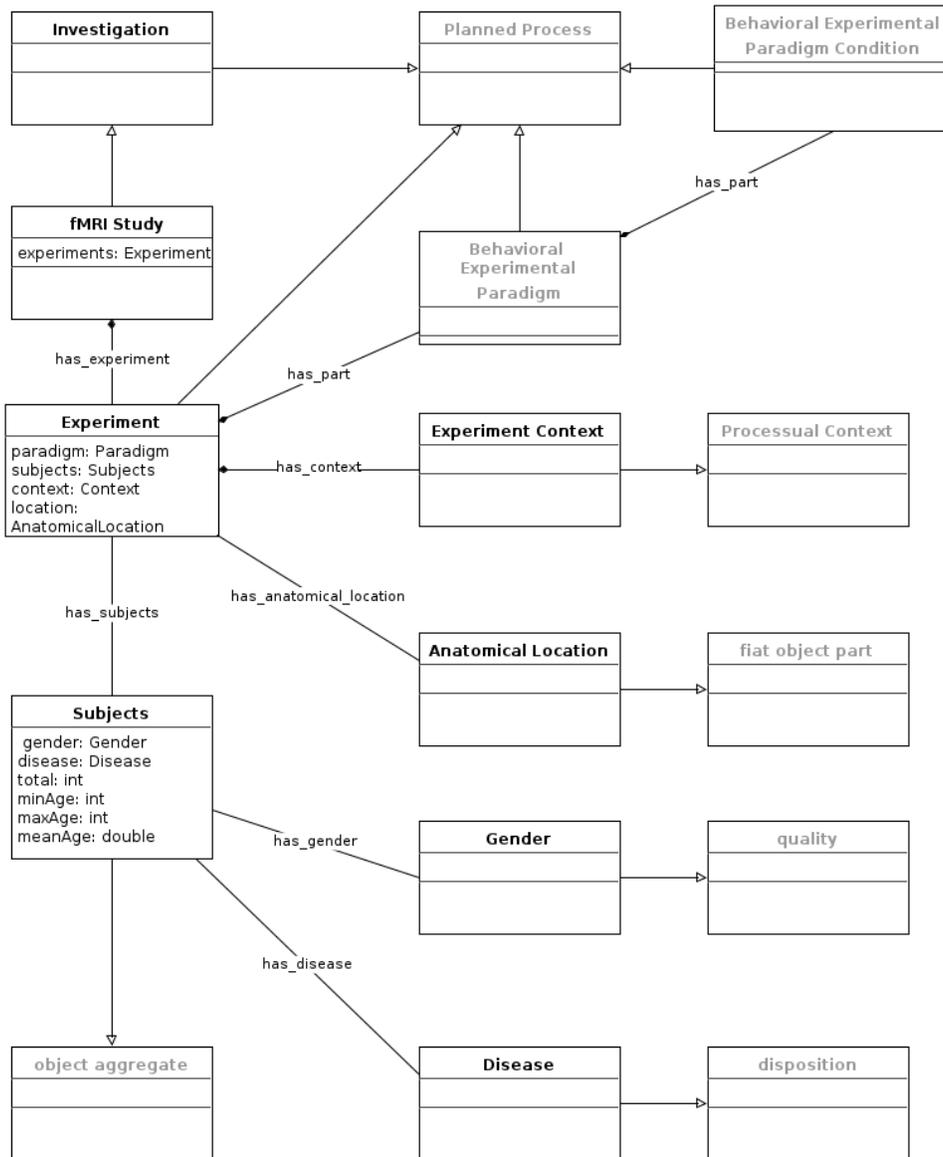


Figura 7: Diagrama de clases de la ontología. Los conceptos importados aparecen en gris

has_experiment relaciona un estudio con un experimento, indicando que este último es parte del primero. Un estudio tiene uno o más experimentos.

has_context relaciona un experimento con un contexto.

has_anatomical_location esta relación indica que un área cerebral (localización anatómica) se ha visto activada en el experimento. Un experimento tendrá por tanto un conjunto de activaciones.

has_subjects relaciona un experimento con un grupo de sujetos.

has_disease relaciona unos sujetos con la enfermedad diagnosticada.

has_gender relaciona unos sujetos con el género.

Una vez tenemos identificados el conjunto de conceptos y relaciones necesarios para representar la parcela de realidad de interés, el siguiente paso es acomodarlos dentro de la taxonomía de las ontologías de referencia que modelan parte del dominio. En nuestro caso, es necesario seguir la taxonomía de CogPO y, por tanto, de BFO e IAO.

2.1.2. Adaptación a las ontologías de dominio

Con el conjunto de clases y relaciones identificados ya se podría desarrollar una ontología para representar los experimentos directamente, sin necesidad de encuadrarlo en un modelo formal. Con esto obtendríamos una ontología de aplicación perfectamente usable, pero con el importante inconveniente de que fuera del contexto de la aplicación apenas tendría la flexibilidad necesaria para adaptarse a nuevos modelos o escalar debidamente al identificar nuevas características a añadir. Esta situación es de mayor importancia si el dominio de aplicación es el biomédico y más concretamente, la neurociencia.

Como se ha ido detallando, existe una potente comunidad de desarrollo y formalización de este tipo de conocimiento en forma de ontologías, ofreciendo gran cantidad de posibilidades de cara a modelar nuevo conocimiento, pero también incluso obligando a adaptarse a su línea. Por todo esto, es de vital importancia que la ontología a desarrollar se encuadre y extienda el conocimiento ya formalizado por la comunidad científica, de forma que el beneficio sea doble:

1. Gran parte de las clases y relaciones estarán definidas y testadas.

2. La ontología será más correcta y tendrá más posibilidades de ser aceptada.

Como hemos visto, CogPO pretende ser una ontología de dominio que modela los paradigmas experimentales empleados en los estudios de neuroimagen funcional, adaptando el conjunto equivalente del esquema de codificación de metadatos de BrainMap de forma lógica en OWL.

El objetivo de la ontología a desarrollar es proporcionar un modelo lógico que represente el subconjunto suficiente de BrainMap capaz de modelar estudios de neuroimagen. Por ello, CogPO es la ontología de dominio que marca el modelo básico de la ontología a desarrollar.

El siguiente paso consiste en acomodar correctamente los conceptos y relaciones dentro de la taxonomía de CogPO y, por consiguiente, de la taxonomía de BFO e IAO. A continuación vamos realizar este proceso, indicando bajo qué clases se agruparán y justificando esta decisión.

fMRI Study

La clase fMRI Study representa un estudio de neuroimagen tal y como lo modela BrainMap, por lo que estará relacionada con un número finito de experimentos. Conceptualmente, podemos considerar un estudio fMRI como una investigación dentro de un contexto determinado.

La ontología IAO define el término investigation, definido como un proceso planeado formado por partes como planificación, estudio, etc. Por tanto, este concepto está a su vez subsumido dentro de la clase planned process de BFO.

Este concepto es el más adecuado para adoptar la clase fMRI Study como subclase.

Experiment

Esta es sin duda la clase más compleja de las que modelará la ontología, dado que será portadora de mucha información.

Podemos considerar un experimento como un proceso planeado y ejecutado en el que participan sujetos que serán sometidos a uno o varios paradigmas, obteniendo como resultado la captación de un conjunto de activaciones en el cerebro de los sujetos. El experimento se desarrolla dentro de un contexto experimental.

Esta definición conlleva a las siguientes implicaciones lógicas:

1. Al considerarse un proceso planificado, será subclase de “IAO:planned process”.

2. Participan sujetos, por lo que estará relacionada con la clase “Subjects” por medio de la relación “hasSubjects”.
3. Los sujetos están sometidos a paradigmas que *son parte* del experimento. Esto implica que estará relacionada con subclases de “Behavioral Experimental Paradigm” por medio de la relación “hasPart”.
4. Fruto de someter a los sujetos a los paradigmas surgen activaciones cerebrales, es decir, localizaciones anatómicas. Un experimento entonces estará relacionado con varias “Anatomical Locations” por medio de la relación “hasAnatomicalLocation”.
5. El experimento tiene un contexto experimental, por lo que este es parte del experimento, utilizando la relación “hasContext”.

Experiment Context

Según BrainMap, el contexto experimental se puede definir como el propósito con el que el experimento se realizó, clasificado por el tipo de efecto apreciado. Para un experimento dado se pueden aplicar múltiples contextos.

Los experimentos son entidades procedurales (processual entities) que existen en un determinado momento. Bajo esta rama de la subontología SPAN de BFO encontramos también la clase processual context, que se define como las entidades que caracterizan otros ocurrentes. Esta definición encaja perfectamente con el cometido que tendrá Experiment Context de cara a los experimentos, por lo que processual context será su superclase.

Teniendo en cuenta el conjunto de contextos definidos por BrainMap, estos son añadidos a su vez como subclases de Experiment Context: Age Effects, Disease Effects, Drug Effects, Experimental Design Effects, Gender Effects, Handedness Effects, Learning, Linguistic Effects, Normal Mapping y Treatment Effects.

Subjects

El concepto “Subjects” puede verse como una agregación de objetos (cada sujeto), es decir, la suma de cada uno de ellos, los cuales a su vez son entidades físicas. Este tipo de conceptos se encuentran definidos en la rama SNAP de BFO, siendo la clase más adecuada object aggregate, que encaja directamente con esta definición.

El concepto Subject es también una entidad portadora de numerosa información. De acuerdo con BrainMap, como mínimo se debe añadir el diagnóstico,

género, edad mínima, máxima y media, así como el total de sujetos que forman el grupo. Esto implica que es necesario pensar qué relaciones llevara el modelo de dicho concepto, de forma similar a Experiment:

1. Será subclase de BFO:object aggregate.
2. Debe tener un género asociado. Para ello se relaciona con la clase Gender por medio de hasGender.
3. Los sujetos tienen un diagnóstico o enfermedad. Por medio de la relación hasDisease se relaciona con subclases de Disease.
4. Para los datos numéricos se añadirán relaciones hasTotal, minAge, maxAge y meanAge.

Gender

El género sin duda es un adjetivo o cualidad de un sujeto, en nuestro caso del concepto agregado Subjects. El concepto cualidad se encuentra también representado (y de notoria importancia) en la rama SPAN de BFO por medio de la clase quality.

Los sujetos participantes en un experimento pueden ser femeninos, masculinos o ser un grupo mixto. Para representar este hecho se añaden las subclases de Gender:

- Female
- Male
- Y Mixed.

Disease

De nuevo, un concepto definido y adaptado en muchas ontologías relacionadas, obviamente, con la enfermedad. La mayoría de estas ontologías se centran en cubrir alguno de sus tipos exhaustivamente y por tanto perdiendo generalidad o introduciendo ruido y otras los abarcan en exceso. Dado el objetivo de este concepto dentro de la representación de BrainMap, no parece tener sentido importar ninguna de estas, pero sí inspirarse en su definición bajo Basic Formal Ontology.

Siguiendo la aproximación de la Infectious Disease Ontology[24] y la de Ontology for General Medical Science¹³, el concepto Disease se encuadraría

¹³<http://code.google.com/p/ogms/>

bajo la clase *disposition* de BFO, cuya definición es una entidad realizable, la cual si cesa de existir, su portador cambia físicamente, así como su realización depende en virtud de la configuración física que el portador presente bajo ciertas circunstancias.

De nuevo, bajo este concepto es necesario introducir como subclases el conjunto de enfermedades que BrainMap acepta en su esquema de codificación.

Es necesario remarcar que este diseño permite que en un futuro, si fuera necesario, los conceptos definidos en esta ontología se importen o relacionen con ontologías del dominio de la enfermedad como las aquí mencionadas. De momento, para la representación de BrainMap no ha sido necesario.

Anatomical Location

Este concepto es también muy importante, ya que gracias a él será posible conseguir dos objetivos básicos:

1. Introducir conceptos anatómicos desde el FMA
2. Y que estos sean relacionados desde los experimentos como sus activaciones.

El primer paso es identificar cuál será la superclase de *Anatomical Location* dentro de CogPO/BFO. Al ser un concepto de naturaleza material y continuante (su definición perdura en el tiempo), es necesario irse a la rama SPAN de BFO. Dentro de ella, la clase *fiat object part* es sin duda la más adecuada, ya que una localización anatómica es una entidad material sin límites definidos físicamente.

El segundo es poblar el conjunto de subclases de *Anatomical Location*. Sin duda, la fuente de conocimiento sobre anatomía cerebral más extensa es la ontología FMA. Debido a su extensión, incluso de su módulo NeuroFMA, resulta inviable importarla directamente, ya que introduciría gran cantidad de términos no relacionados con el problema e impediría realizar tareas de inferencia de forma rápida y liviana.

Para evitar este problema se ha seguido, al igual que en CogPO, la técnica MIREOT o Minimum Information to Reference an External Ontology Term[18]. Este método define que el conjunto mínimo para referenciar de forma consistente un término externo es el siguiente:

- URI de la ontología fuente, la ontología que contiene el término externo a ser importado.

- URI del término fuente, el término específico a importar.
- URI de la superclase directa del término referido en la ontología que importa.

A pesar de tener un sistema que evita importar términos innecesarios, prácticamente todo el cerebro es susceptible de ser observado en una activación. Esto requeriría introducir a mano esos términos y además produciría un problema similar. Para evitarlo se ha seguido una estrategia incremental y de “importación diferida”, introduciendo los términos necesarios sólo cuando se requiera, es decir, cuando un experimento lo reporte como activación. Los detalles de implementación de esta estrategia se encuentran en apartado 2.2, tras conocer los detalles de la implementación en OWL.

2.1.3. Implementación en OWL

Como ya hemos visto en el apartado dedicado a la Web Semántica, OWL es un lenguaje orientado a la creación de ontologías por medio de la Lógica Descriptiva. En el contexto actual, la elección natural de cara a desarrollar una ontología es sin duda este lenguaje. No sólo por las ventajas que el propio lenguaje ofrece, sino también por su implantación dentro de las nuevas ontologías de dominio y por sus capacidades de interoperabilidad y modularidad gracias a su directiva `<owl:imports>`. Además, usando este lenguaje, podremos definir las clases usando potentes construcciones lógicas, consiguiendo que sean algo más que declaraciones: definiciones inequívocas de conocimiento.

Empezaremos definiendo las relaciones identificadas. En OWL estas relaciones serán de dos tipos:

1. *Object Properties* (relaciones entre individuos)
2. Y *Datatype Properties* (relaciones entre un individuo y un tipo de datos o literal).

2.1.3.1. Relaciones Al igual que RDF, las relaciones (propiedades) en OWL pueden definir qué tipos de clases aceptan como sujeto y qué tipos de clase como objeto. Esto es el dominio y rango de la propiedad. En algunas de nuestras propiedades, al estar pensadas para un tipo concreto de relación, será necesario especificar esta restricción.

La siguiente tabla resume la implementación de estas propiedades.

Nombre	Tipo	Superpropiedad	Dominio	Rango
has_experiment	Object Property	has_part	fMRI Study	Experiment
has_context	Object Property	has_part	Experiment	Experiment Context
has_subjects	Object Property	has_participant	Experiment	Subjects
has_disease	Object Property	bearer_of	Subjects	Disease
has_gender	Object Property	bearer_of	Subjects	Gender
has_anatomical_location	Object Property	has_participant	Experiment	Anatomical Location
max_age	Datatype Property		Subjects	int
min_age	Datatype Property		Subjects	int
mean_age	Datatype Property		Subjects	double
total_subjects	Datatype Property		Subjects	int

Cuadro 3: Tabla de relaciones

2.1.3.2. Clases Las clases en OWL son descripciones lógicas de la realidad. En algunas de las clases a definir se aprovechará la expresividad de OWL para establecer un modelo lógico que permita más tarde realizar inferencias sobre el conocimiento contenido en la ontología.

La tabla 5 contiene el conjunto de clases definidas, así como su superclase y posibles restricciones.

A continuación desglosamos las clases más importantes, de las cuales algunas heredarán conceptos directos de la ontología o se añadirán dinámicamente en la aplicación. El conjunto completo se encuentra desglosado en una tabla en el anexo.

fMRI Study un estudio de neuroimagen debe tener experimentos, por lo que se añade la restricción $\exists hasExperiment.Experiment$

Experiment un experimento debe estar muy bien definido, pero además, de cara a realizar inferencia, es útil especificar de forma lógica qué es equivalente a un experimento de neuroimagen:

$$\begin{aligned}
 Experiment \equiv & \exists hasPart.Behavioral\ Experimental\ Paradigm \sqcap \\
 & \exists hasPart.Experiment\ Content \sqcap \exists hasAnatomicalLocation.Anatomical\ Location \sqcap \\
 & \exists hasSubjects.Subjects
 \end{aligned}$$

Subjects Los sujetos, en el sentido de un experimento de neuroimagen definido por BrainMap, deben tener un género, un diagnóstico, un total

de sujetos, un mínimo y máximo de edad y una media de edad. La expresión lógica añadida como restricción de superclase es la siguiente.

$$\exists hasGender.Gender \sqcap \exists hasDisease.Disease \sqcap \exists totalSubjects.Integer \\ \exists minAge.Integer \exists maxAge.Integer \exists meanAge.Double$$

Añadiendo este conjunto de clases y sus equivalencias y restricciones obtenemos un modelo como el detallado en el diagrama de clases de la figura 7.

2.1.4. Ontología en el contexto de aplicación

Hasta aquí hemos desglosado la identificación de los conceptos y relaciones de la ontología, así como su implementación en OWL. El resultado es una ontología que modela gran parte del esquema de BrainMap, abriendo su uso en diferentes aplicaciones.

En el problema planteado, la ontología tendrá dos funciones básicas: proporcionar la terminología del dominio y permitir la ejecución de procesos de inferencia para la resolución de problemas.

Por tanto, la base de conocimiento se organiza como indica la figura 8. La base terminológica o Tbox está determinada por las ontologías de dominio (CogPO y sus imports) y la ontología BrainMap. En ella se encuentran las declaraciones de los conceptos, sus relaciones y los axiomas lógicos que declaran el conocimiento del dominio.

Una vez tenemos la base de conocimiento determinada, el siguiente paso es plantear su uso dentro de la aplicación:

- Qué datos se van a manejar
- Cómo se van a añadir a la base de conocimiento
- Cómo se van a gestionar dentro de ella
- Y qué inferencias se van a realizar.

Los datos finales, en el marco de la Web Semántica, se corresponden con individuos o instancias de las clases definidas en la ontología (Tbox). Por tanto, cabe pensar que la Abox (el conjunto de hechos o aserciones en base a la terminología) estará formada por instancias que representan el desglose de los metadatos obtenidos de un estudio según BrainMap.

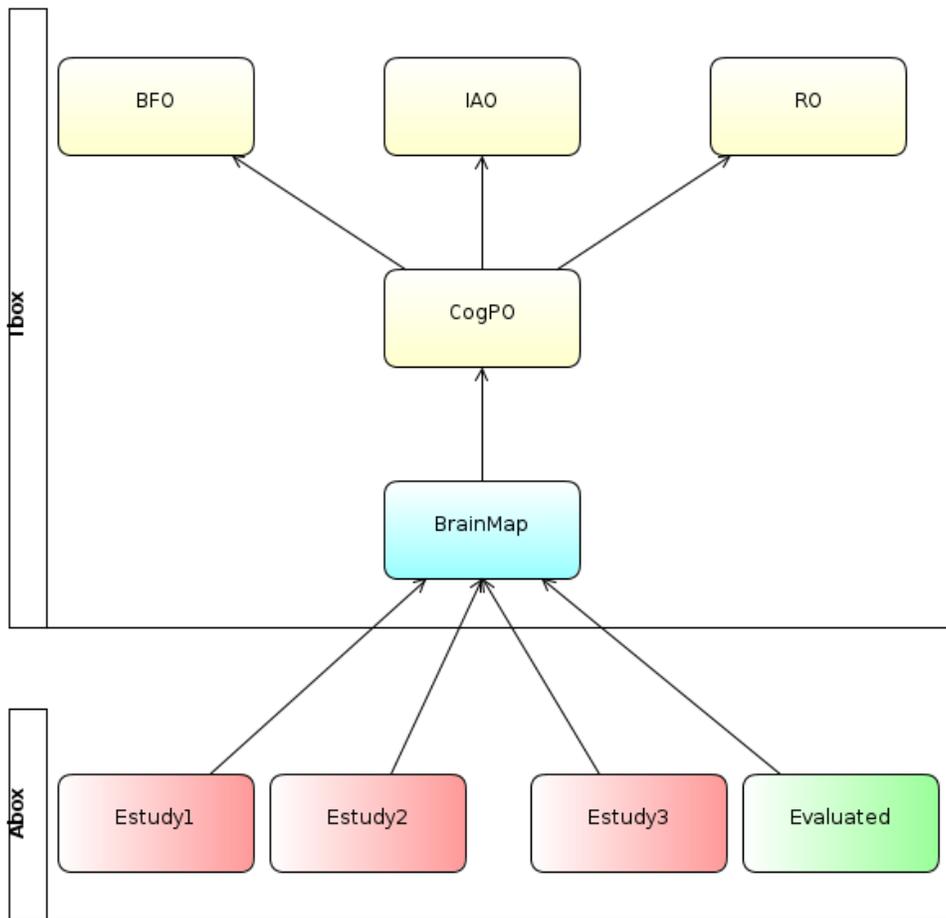


Figura 8: Base de conocimiento de la aplicación

Esto implica que la capa factual se modificará dinámicamente según las necesidades de la aplicación. Es decir, la aplicación contará con un conocimiento “estático” definido en BrainMap, y según los datos necesarios, se crearán ontologías que importen BrainMap y añadan hechos en base a los datos. Más tarde, los hechos de estas ontologías serán evaluados por medio de medidas de similitud. Este proceso se puede observar en el diagrama de la figura 9 y resumir en los siguientes pasos:

1. En función de los datos que el usuario ha introducido del estudio.
 - a) Crear una ontología que modele dicho estudio según BrainMap, importando esta última y añadiendo clases e instancias para representar los hechos que surgen de los datos.
 - b) En función de los datos clave del estudio, realizar una query a BrainMap y con los estudios resultantes hacer lo mismo que en a).
2. Por cada ontología representando un estudio obtenido de BrainMap.
 - a) Realizar medida de comparación entre la ontología evaluada (estudio introducido) y la ontología objetivo (estudio obtenido de BrainMap).

Con este proceso estamos aprovechando el modelo formal surgido de la extensión de CogPO para modelar estudios de neuroimagen funcional y, de esta forma, evaluarlos cuantitativamente, evitando el sesgo y pérdida de información que introducen las búsquedas por palabras clave, así como aprovechando toda la información, tanto de contexto como contenido, que un estudio lleva consigo.

2.2. Relación diferida de localizaciones anatómicas con FMA

Una de los datos importantes que contiene un experimento es el conjunto de activaciones cerebrales detectadas. Las localizaciones de las activaciones reportadas en los experimentos de BrainMap se muestran tanto en formato (x,y,z) como con etiqueta del atlas Talairach. Será esta etiqueta la que nos permita obtener los conceptos de FMA y representarlos en la ontología por medio de la propiedad `hasAnatomicalLocation`, que relaciona un experimento con una localización anatómica.

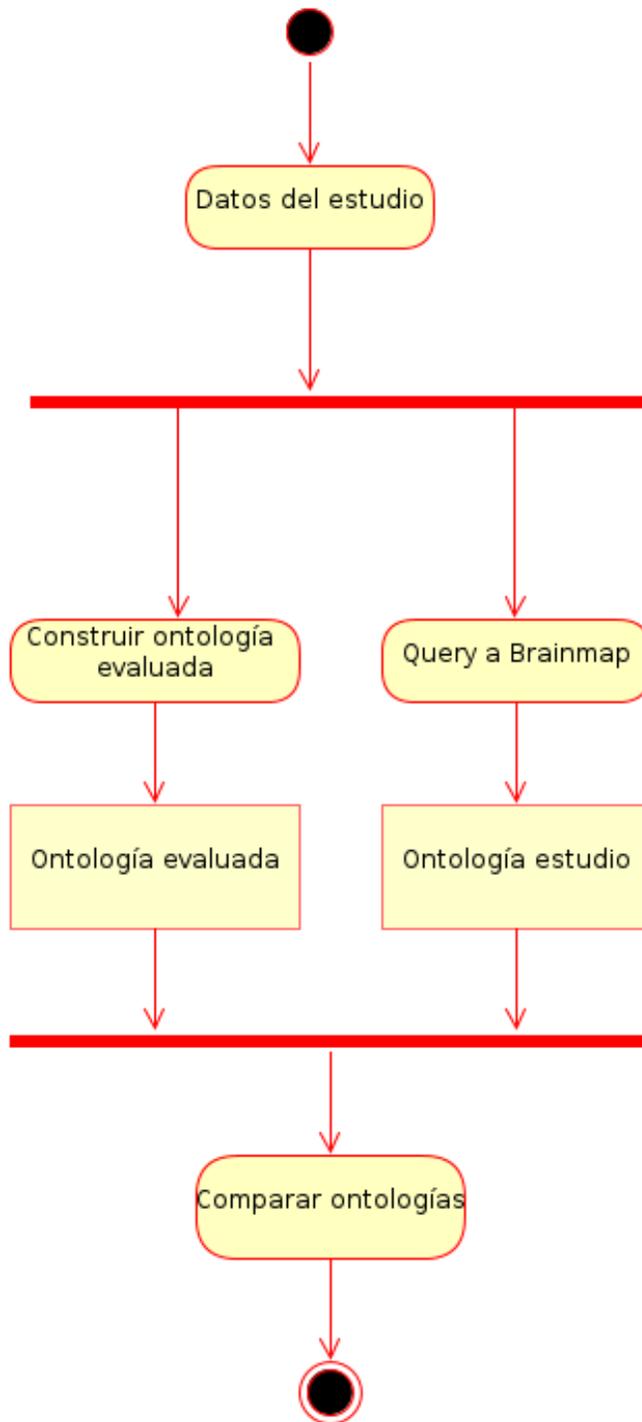


Figura 9: Diagrama de actividad del proceso de generación del Abox

Como se ha visto a lo largo del trabajo, esta información anatómica se debe extraer del FMA por medio del procedimiento MIREOT e introducir en la ontología BrainMap. Teniendo en cuenta que el módulo de la anatomía cerebral, NeuroFMA, cuenta con 15783, la realización de forma manual de este proceso resulta inviable. Para solventarlo se ha diseñado y programado un script que, aprovechando las relaciones de FMA con términos del atlas Talairach, añade automáticamente las localizaciones reportadas en un experimento si estas no se encuentran presentes en la ontología. Este script se ejecuta por cada estudio modelado.

2.2.1. FMA y Talairach

El FMA es una ontología formal que modela el conocimiento anatómico por medio de conceptos y relaciones. Dentro del dominio neurocientífico, existen diferentes terminologías definidas por varios consorcios y aplicaciones que se solapan en la definición de muchos de los conceptos neuroanatómicos.

Para maximizar la alineación e interoperabilidad (varias de las terminologías son de aplicaciones de procesamiento de imagen médica o altas) el FMA define el concepto Neuro Term, que a su vez es una Attribute entity, es decir un atributo relacionado con los conceptos que aportan algún tipo de información adicional. Las equivalencias que ofrece dentro de Neuro Term son:

- Attribute Entity.
 - Neuro Term.
 - AAL term.
 - FreeSurfer term (equivalencia con etiquetas del software de procesamiento de imágenes FreeSurfer).
 - Neurolex term (equivalencia con la taxonomía Neurolex).
 - Talairach term (equivalencia con las etiquetas del atlas Talairach).

En nuestro caso, el concepto a explotar son las instancias de Talairach term. Estas se relacionan con los subconceptos de Anatomical Entity por medio de la relación **Talairach**. Las instancias de Talairach term portan la información siguiente:

FMAID el identificador dentro del FMA, toda entidad definida esta ontología define uno.

```

PREFIX fma:<http://sig.uw.edu/fma#>
SELECT ?clase ?TalairachTerm
FROM <http://localhost/fma>
WHERE {
  ?TalairachTerm fma:name name
  ?clase fma:Talairach ?TalairachTerm.
}

```

Cuadro 4: Consulta SPARQL para extraer el concepto FMA

name la etiqueta tal y como se define en el atlas Talairach.

Talairach_DI el identificador definido por el atlas Talairach.

Por tanto, es posible extraer el concepto neuroanatómico que FMA define dada una etiqueta o ID de Talairach.

2.2.2. Extracción de conceptos del FMA a partir de etiquetas Talairach

Para encontrar el concepto de FMA a partir de una etiqueta de Talairach es necesario responder a la pregunta *qué concepto tiene como Talairach una instancia con name=etiqueta*. Al hablar de ontologías, tenemos dos métodos para encontrar la respuesta: por medio de un razonador o por medio de una consulta SPARQL¹⁴. Dado el tamaño del NeuroFMA, la primera opción es inviable, ya que el coste computacional resulta inasumible. Por tanto, la solución es realizar una consulta SPARQL:

Para evitar problemas con el tamaño de la ontología, se ha utilizado la Triple Store virtuoso¹⁵, indexando de esta forma el grafo RDF del NeuroFMA.

Para referenciar las localizaciones reportadas en un experimento en Sleuth, se almacenan las etiquetas mostradas en un fichero de texto plano, de forma que el script acceda a ellas secuencialmente.

El proceso completo se puede observar en el diagrama de actividad.

¹⁴<http://www.w3.org/TR/rdf-sparql-query/>

¹⁵<http://virtuoso.openlinksw.com/>

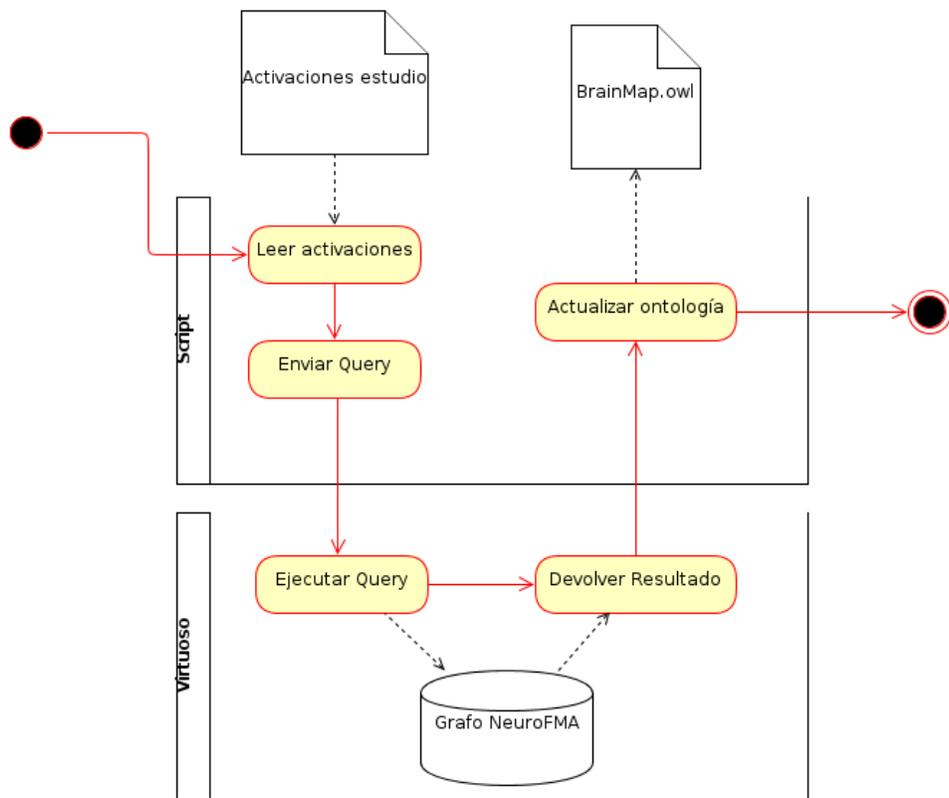


Figura 10: Diagrama de actividad del proceso

2.3. Métricas para comparación de ontologías

Para poder realizar comparaciones entre la ontología obtenida del estudio y las resultantes de la búsqueda en BrainMap, es necesario definir una o más medidas de similitud.

El problema de la alineación y extracción de módulos en ontologías es un tema abierto, donde la comunidad científica ha aplicado diferentes aproximaciones para su automatización y evaluación. Principalmente, utilizando diferentes medidas de similitud a partir de los componentes de la ontología.

Con el objetivo de comprobar y evaluar la adaptación de diferentes aproximaciones, se han implementado varias métricas inspiradas en algunas de las últimas desarrolladas en la literatura relacionada.

A diferencia de los problemas de alineación o modularización, nuestro proceso de comparación es un problema cerrado y centrado en dominio concreto, utilizando una única ontología. Sin embargo, estas medidas también son de utilidad para comparar conceptos dentro de una ontología.

Nuestro caso consiste en una única ontología que modela, de forma aumentada y enriquecida, el esquema de codificación de la base de datos BrainMap. Al ser un problema bien definido, es posible reducir la ambigüedad de ciertos métodos particularizándolos en el dominio de la ontología, esto es, evaluar los conceptos y relaciones más relevantes dentro de la ontología de aplicación.

A continuación se detallarán los métodos adoptados, así como su base teórica y especificaciones de cara a la implementación.

2.3.1. Consistencia Lógica

La primera de las medidas de similitud planteadas se basa en el empleo inteligente de los axiomas lógicos definidos en la ontología BrainMap y la evaluada. A diferencia del resto de métricas implementadas, inspiradas en métodos de alineamiento de ontología basados en grafo, este planteamiento está completamente inspirado y orientado al modelo de conocimiento representado por la ontología, por lo que se aleja del concepto de métrica en ese contexto.

Dado un conjunto de datos de estudio, si éste se modela especificando como axiomas equivalentes cada uno de los datos, es posible emplear un razonador que clasifique instancias de otros estudios y evaluar posteriormente el número de clasificaciones exitosas. Para realizar el razonamiento, es necesaria una

ontología temporal que importe tanto la ontología evaluada como la que representa el estudio extraído de BrainMap.

Por ejemplo:

Teniendo un concepto de sujetos llamados Controles, cuyos datos son (por simplicidad mantenemos un número pequeño de atributos):

- Género: Mujer
- Diagnóstico: Normal
- Total: 10

Se representará en la ontología como una subclase de Subjects, con el axioma de clase equivalente $Controls \equiv \exists hasGender.Female \sqcap \exists hasDisease.Normal \sqcap hasTotal = 10$.

Si nos paramos a pensar en el contexto del problema, donde los diferentes estudios pueden ir desde la diferenciación absoluta hasta sutiles diferencias, es muy improbable que se den conceptos exactamente iguales, por lo que este sistema de inferencia no ofrecería resultados aceptables.

Para concebir la consistencia lógica como un valor no absoluto normalizado en $[0, 1]$, es posible plantear la partición del axioma que define un concepto como la intersección de todas sus propiedades.

Siguiendo el ejemplo anterior, la subclase de Subjects, Controls, puede definirse como una clase que a su vez subsume un conjunto de subclases, cada una conteniendo uno de los operandos del axioma que lo define. Es decir, de forma jerárquica:

- Subjects
 - Controls
 - $Controls_1 \equiv \exists hasGender.Female$.
 - $Controls_2 \equiv \exists hasDisease.Normal$.
 - $Controls_3 \equiv hasTotal = 10$.

Por tanto, cualquier individuo que esté relacionado con Female por medio de hasGender será inferido como miembro de Controls₁. Si este mismo individuo cumple con $\exists hasDisease.Normal$ será inferido también como Controls₂ y, por tanto, 2/3 consistente con la definición de Controls.

El siguiente paso es contextualizar este sistema al comparar dos ontologías de estudio.

Tal y como está modelado BrainMap, hay tres conceptos fuertemente relacionados que consideraremos como los principales nodos de información de un estudio. Estos nodos son:

- **Subjects.** Debido a que cada estudio suele contar como mínimo con dos grupos diferentes de sujetos, y a que este concepto requiere de mucha información.
- **Experiment.** Es el principal concepto dentro del modelo de BrainMap, cada estudio realiza un conjunto de experimentos, que a su vez están caracterizados por muchos atributos (sujetos, contexto, paradigma...).
- **Condition.** Los paradigmas experimentales empleados en los experimentos dependen de las condiciones que los expertos definen en su diseño, considerando un mínimo de instrucciones, estímulos y respuestas.

Por tanto, serán estos conceptos los que la aplicación revisará al realizar las inferencias de clasificación. Una vez identificado el proceso y los conceptos más relevantes del dominio podemos formular el procedimiento formalmente.

Teniendo las definiciones:

\mathcal{E} Ontología evaluada, resultante del modelado del estudio a comparar.

\mathcal{B} Ontología resultante al modelar un estudio obtenido de BrainMap.

C Conjunto de conceptos de interés definidos en \mathcal{E} . Para evaluar el problema, los conceptos más relevantes son Experiment, Subjects y Condition, ya que son los que añadirán subclases variadas, adaptando su definición lógica para representar los datos del estudio evaluado.

A Conjunto de subclases de un concepto $c \in C$ que representan un subaxioma.

I_c Conjunto de individuos definidos en \mathcal{B} del tipo $c; c \in C$.

inferred Función que devuelve 1 si el individuo es inferido como tipo del concepto o 0 si no es así.

$$inferred(i, a) = \begin{cases} 1 & \text{si } i \text{ es inferido como tipo } a \\ 0 & \text{en caso contrario} \end{cases}$$

La consistencia de un individuo respecto a un concepto C_i es:

$$CLi(i) = \frac{\sum_j \text{inferred}(i, a_j)}{|A|}; i \in I_c, a \in A$$

La consistencia de un concepto es:

$$CLc(c) = \frac{\sum_j CLi(i_j, c)}{|I_c|}; i \in I_c, c \in C$$

Y la consistencia de la ontología es:

$$CL(e) = \frac{\sum_i CLc(c_i)}{|C|}; c \in C$$

2.3.2. Grafo Semántico

El segundo de los métodos consiste en una adaptación de la metodología de alineamiento planteada en[25].

De forma similar, el punto de partida es el grafo RDF que representa a los objetos de la ontología. A partir del mismo, centrándonos en los individuos y descartando las relaciones que no sean las definidas por el dominio, obtenemos un subgrafo de la ontología que representa el conjunto de individuos que pueblan la ABox y, por tanto, los datos referentes al estudio. Al tratarse de individuos, nos interesan las relaciones (arcos) definidas para el dominio, desechando las propias del lenguaje (OWL).

El grafo resultante es un multigrafo dirigido y etiquetado:

$$\mathcal{G} = (V, E)$$

Donde:

- V es un conjunto finito de vértices, representando los individuos definidos en la ontología.
- E es el conjunto de de arcos que representan las relaciones entre los individuos (v_m, e_i, v_n)

- C es el conjunto de conceptos tipo de los individuos.

Dados dos grafos \mathcal{G} y \mathcal{G}' , la métrica de similitud es la suma de la similitud máxima entre individuos del mismo tipo normalizada en $[0, 1]$ por el número total de clases:

$$Sim(G, G') = \frac{\sum \max Sim(v, v')}{|C|}$$

Dados dos vértices $v \in G, v' \in G'$ la medida de similitud:

$$Sim(v, v') = \frac{\sum_i \sum_j (r_i = r_j)}{|R_1| + ||R_1| - |R_2||}$$

donde $r_i \in R_1, r_j \in R_2$ son una relaciones.

El denominador R_1 normaliza en $[0, 1]$ mientras que la diferencia entre el número de relaciones (arcos salientes) de dos vértices penaliza el valor. De esta forma se evita considerar como iguales dos vértices con diferentes números de relaciones.

2.3.3. Grafo VSM

En este método de nuevo se toma como base la representación de grafo resultante de las ontologías, siguiendo una aproximación de similitud semántica basada en la topología o estructura[26].

El proceso original se puede resumir en dos puntos:

1. Para calcular la similitud semántica de dos entidades se toman como referencia semántica los predicados RDF/OWL comunes. Los objetos se describen y comparan en base a la forma de relacionarse con otros objetos en términos de estos predicados. Una vez identificados se modelan como un Vector Space Model (VSM).
2. Una vez obtenida la representación, se aplica un algoritmo de comparación de grafos.

Al igual que en la anterior definición de métrica, tomamos como predicados el conjunto de relaciones definidas en el dominio.

De grafo a VSM

En una ontología OWL, se compararán las entidades teniendo en consideración sus relaciones con el resto de entidades presentes en la ontología. Dado que las relaciones pueden ser de naturaleza diferente, éstas se modelan como un espacio vectorial. Para este espacio vectorial se toman tantas dimensiones como relaciones definidas en BrainMap.

De esta forma, podemos definir formalmente la relación entre dos nodos del modelo como:

Dado cualquier par de nodos n_1 y n_2 de un grafo dirigido y etiquetado RDF representando la ontología \mathcal{O} , la relación entre ellos $rel(n_1, n_2)$, se define como el vector:

$$\{arc(n_1, n_2, p_1), \dots, arc(n_1, n_2, p_N)\}$$

Donde:

- arc es una función que devuelve:
 - 1 si existe un arco etiquetado con el predicado p_1 desde n_1 a n_2 o
 - 0 en caso contrario.
- p_i es un predicado del conjunto de predicados definidos en BrainMap \mathcal{P} (por ejemplo $\{\text{brainmap:hasExperiment, brainmap:hasSubjects}\}$).

$$rel(n_1, n_2) = \{arc(n_1, n_2, p_1), \dots, arc(n_1, n_2, p_N)\} | n_1, n_2 \in \mathcal{G} \wedge \forall i \in [0; N], p_i \in \mathcal{P}$$

El modelo resultante es básicamente una matriz de adyacencia, donde cada componente contiene un vector con las relaciones del nodo correspondiente a la componente.

Por ejemplo, tomando el grafo RDF de la figura 1 $\mathcal{G}_{\mathcal{A}}$ representando la ontología $\mathcal{O}_{\mathcal{A}}$, con el conjunto de predicados siendo $\mathcal{P} \{\text{brainmap:hasContext, brainmap:hasSubjects}\}$, obtendríamos un sencillo espacio vectorial bidimensional. El conjunto de espacios vectoriales describen una matriz tridimensional \mathcal{A} que representa el grafo $\mathcal{G}_{\mathcal{A}}$, siendo el orden de las filas Experiment, Subjects y por último Context:

$$A = \begin{pmatrix} (0, 0) & (1, 0) & (0, 1) \\ (0, 0) & (0, 0) & (0, 0) \\ (0, 0) & (0, 0) & (0, 0) \end{pmatrix}$$

Similitud entre grafos VSM

Una vez hallada la forma de representar las ontologías a través de un Vector Space Model, el siguiente paso es definir el proceso que calcule la medida de similitud entre dos ontologías dadas. Para ello se emplea el algoritmo adaptado del caso más general de similitud entre vértices de grafos dirigidos[27] al Vector Space Model definido, de la misma forma que[26]. Recordemos que lo único que variará es el conjunto de predicados evaluados.

El proceso consiste en aplicar de forma iterativa la siguiente ecuación:

$$S_{k+1} = BS_kA^T + B^T S_kA, k = 0, 1, \dots$$

Donde:

- S_k es la $N_B * N_A$ matriz de similitud de las entradas s_{ij} en la iteración k
- A y B son las matrices tridimensionales $N_B * N_B * N_P$ y $N_A * N_A * N_P$ que representan a los grafos \mathcal{G}_A y \mathcal{G}_B respectivamente.
 - N_A y N_B son el número de filas de A y B
 - P es el número de predicados seleccionados como dimensiones del VSM.

Inicialmente S_0 está inicializada a 1, considerando que todos los nodos son iguales. Al aplicar la ecuación de forma iterativa esos valores se van ajustando hasta converger, punto en el cual el proceso finalizaría obteniendo la matriz.

Para mantener la matriz normalizada en $[0, 1]$, se dividen todos los componentes por la norma Frobenius de la matriz, definida como la raíz cuadrada de la suma de las raíces absolutas de todos sus elementos.

Con este proceso obtenemos la matriz de similitud entre los nodos de las ontologías, sin embargo, para nuestra herramienta precisamos que la medida de similitud sea un escalar. Para ello, se considera como medida de similitud final a la norma matricial:

$$Sim(A, B) = \frac{\sum_i \sum_j s_{ij}}{S_i \times S_j}$$

2.3.4. Entropía de la ontología

A diferencia de las anteriores medidas, esta métrica no tiene como objetivo comparar dos ontologías, sino evaluar cuantitativamente la cantidad de contenido informativo que ofrece una ontología.

El concepto de entropía tiene su origen en la física (segunda ley de la termodinámica) y mecánica. A partir del trabajo de Shannon[28], la entropía fue incorporándose al dominio de la teoría de la información.

La entropía, según fue definida por Shannon, representa la cantidad de información contenida en un mensaje o, desde el punto de vista del receptor, la incertidumbre sobre el mensaje que envió la fuente antes de su recepción. Se define como

$$H(X) = - \sum_i^n p(i) \log p(i)$$

Donde:

- $p(i)$ es la posibilidad de recibir el mensaje i
- X una variable aleatoria discreta.

Shannon demostró que $\log p(i)$ es la única función que satisface todos los requerimientos para medir la información.

Para poder emplear el concepto de entropía, es necesario contar con una función de probabilidad $p(x_i)$ que sea aplicable a la ontología, tomando como representación el grafo y unidades de información los vértices. [29, 30] define esta función como el grado del vértice evaluado (el número de arcos salientes) dividido por el sumatorio de todos los grados del conjunto de vértices V .

$$p(v_i) = \frac{\text{deg}(v_i)}{\sum_v \text{deg}(v)} ; v_i, v \in V$$

Sin embargo, esta aproximación está limitada al considerar todos los arcos iguales.

Para paliar estas limitaciones, [31] plantea una reformulación que separa la noción de *entropía a nivel de lenguaje* $H_L(X)$ y *entropía a nivel de dominio* $H_D(X)$. La primera estima la cantidad de información transportada por los arcos que representan los predicados a nivel de lenguaje, como

<owl:ObjectProperty>. La segunda se centra en las relaciones específicas del dominio, es decir, las definidas dentro de una ontología.

La entropía total sería:

$$H(X) = H_L(X) + H_D(X)$$

Esta reformulación resulta interesante en el contexto de este trabajo, ya que define el concepto de entropía de dominio tal y como se busca implementar. Desechando así la semántica e información aportada por el lenguaje y centrándose en la información que aportan las relaciones propias del dominio.

La entropía a nivel de dominio calcula por tanto la entropía asociada a los arcos del dominio.

De nuevo, hay que tener en cuenta que no todas las relaciones son iguales, por lo que es necesario llevar a cabo un sistema que evalúe la información que aporta cada una. Para esto, la solución adoptada es asociar pesos a las relaciones.

Siendo D el conjunto de relaciones definidas por el dominio, asumimos que los elementos \sum_D que aparecen de forma más frecuente, reparten la información de forma igualitaria, por lo que el peso asociado debería ser menor.

En el momento de construir el grafo, para cada $d \in D$ se define una función de peso $w()$ que asigna un peso a cada elemento de D . Estos pesos se definen como:

$$w(d) = \frac{1}{|D_d|}$$

Donde:

$$D_d = \{x \times \sigma_D \times y \mid etiqueta_d(d) = \sigma_D\}$$

Lo cual implica que los pesos se determinan en función del número de elementos de D para los cuales la etiqueta σ_D es la misma. Así, los pesos de los arcos están normalizados entre 0 y 1, de forma que los arcos que aparecen más frecuentemente reciben un peso menor y los arcos que aparecen menos se les asigna un peso mayor.

Finalmente, la función de probabilidad $p(i)$ empleada para calcular la entropía a nivel de dominio $H_D(X)$ es:

$$p(i) = \frac{\text{pesosDelNodo}(i)}{\sum_{v \in V} \text{pesosDelNodo}(v)}$$

Donde la función $\text{pesosDelNodo}()$ para cada $v \in V$ es tal que:

$$\text{pesosDelNodo}(v) = \sum_{r \in R} w(f)$$

Donde:

- R es el conjunto de arcos de D desde v .

Por tanto, se suman los pesos de los arcos salientes de v y dividen por el sumatorio de los pesos de los arcos salientes de todos los elementos de V .

Este concepto de entropía será empleado por la aplicación para evaluar la cantidad de información que un estudio (ontología) contiene, a título individual.

2.4. Búsquedas en BrainMap y datos empleados

Para obtener los estudios candidatos a ser similares al evaluado, es necesario realizar una búsqueda en BrainMap que proporcione los datos para construirlos dentro del modelo de la ontología.

Desafortunadamente, BrainMap carece de una API pública que permita a aplicaciones de terceros interactuar de forma programática. Tampoco ha sido posible obtener libre acceso al Sistema de Gestión de Base de Datos, a pesar de haber realizado contactos solicitándolo.

Debido a esta importante restricción, surgen dos importantes interrogantes:

1. Qué tipo de queries son las más adecuadas para obtener estudios similares.
2. Cómo se van a ejecutar las queries para visualizar los datos de los estudios.

Para el primer problema se ha debatido con el experto qué datos considera más significativos. Estos son los siguientes:

- Diagnóstico de los sujetos.
- Área cerebral reportada como activación.
- Paradigma empleado en los experimentos.
- Estímulo empleado en las condiciones de los paradigmas de los experimentos.

Como no es posible definir los datos concretos de cada criterio en tiempo de ejecución, es necesario identificarlos previamente. Por tanto, el caso de uso a evaluar será el estudio de neuroimagen sobre Trastornos Compulsivos de la Alimentación (TCA) en el que el experto se encuentra trabajando actualmente. Estos son:

- Diagnóstico.
 - Anorexia nervosa
 - Bulimia nervosa
- Área cerebral.
 - Amígdala .
 - Cortex Prefrontal Ventral.
 - Cortex Prefrontal Dorso Lateral.
 - Cortex Cingulado Anterior.
 - Caudado.
 - Putamen.
- Estímulo.
 - Fotos/imágenes.

Con los criterios de búsqueda determinados, la opción más habitual es realizar una búsqueda como una intersección de todos ellos. Sin embargo, existen dos inconvenientes para esta aproximación:

1. BrainMap no está lo suficientemente poblado
2. Y los estudios de neuroimagen pueden ser muy heterogéneos.

● Experiments

3. Food > Non-Food, Binge Purging Type Anorexia Nervosa Patients

Context: Disease Effects

Imaging Modality: fMRI (1.5T)

Subject Groups: Binge Purging Type Anorexia Nervosa Patients

Conditions: Food, Non-Food

Activations: High-Level Control

Sessions: Contrast: Stimulus Type, Instruction, Contrast Description: Stimulus type: food vs. non-food pictures.

Instruction: in the food condition subjects imagined eating the food displayed, and in the non-food condition subjects imagined using the non-food items displayed.

Paradigm Class: Passive Viewing, Imagined Objects/Scenes

Behavioral Domain: Perception.Vision, Cognition.Attention

Locations

ID	X (mm)	Y (mm)	Z (mm)	Converted Talairach X(mm)	Converted Talairach Y(mm)	Converted Talairach Z(mm)	SPI Value	SPI Unit	Extent (mm ³)	Talairach Labels
1	-4	-56	-36	-4	-56	-36				Left Cerebellum, Posterior Lobe, Cerebellar Tonsil, Gray Matter.*
2	25	-63	-20	25	-63	-20				Right Cerebellum, Posterior Lobe, Declive, Gray Matter.*
3	22	-4	53	22	-4	53				Right Cerebrum, Frontal Lobe, Sub-Gyral, Gray Matter, Brodmann area 6

Figura 11: Detalle de un experimento en el workspace de Sleuth

Para obtener el máximo de estudios posibles se realizarán búsquedas especificando un criterio a la vez.

Una vez identificados los criterios, el siguiente paso es realizar las queries para obtener los datos de los estudios. BrainMap ofrece dos opciones:

1. La interfaz web BrainMapWeb¹⁶. Una aplicación web para realizar búsquedas en BrainMap, pero que carece de visualización 3D y capacidades de manipulación de datos.
2. La aplicación Sleuth, que como ya hemos visto es la herramienta ideada para realizar búsquedas complejas.

Lógicamente la opción más adecuada es Sleuth, ya que ofrece una interfaz categorizada donde muestra los datos en modo texto, como muestra la figura 10.

Tras ejecutar las queries con los criterios mencionados, se han seguido los datos mostrados por cada estudio en Sleuth para codificar cada uno como una ontología. Esta ontología importa BrainMap y añade como individuos concretos los datos seleccionados. Como ya se ha mencionado, en el caso de las localizaciones cerebrales se sigue un proceso semiautomático descrito en el anexo 7.1.

Es importante remarcar que, debido a la imposibilidad de realizar este proceso de forma automática y bajo demanda de la aplicación, el conjunto de

¹⁶<http://apps.rii.uthscsa.edu/bmapWeb/>

datos es muy reducido, ya que el tiempo necesario para buscar y codificar los estudios es relativamente alto y una dedicación excesiva iría en detrimento del desarrollo del proyecto.

3. Desarrollo de la herramienta

Para ejemplificar la hipótesis planteada en el trabajo, se ha desarrollado una aplicación de escritorio basada en Java¹⁷. La elección de este lenguaje de programación atiende principalmente a las siguientes razones:

- Dispone de una API muy amplia, cubriendo la mayoría de operaciones básicas necesarias.
- Tiene una fuerte integración y orientación al entorno Web.
- Existe gran apoyo de la comunidad Open Source con iniciativas como la Apache Software Foundation¹⁸.
- Es independiente de la plataforma.
- Es un lenguaje que ofrece muchas facilidades para emplear patrones de diseño.
- Y el más importante de todos, las librerías de Web Semántica más potentes están desarrolladas en Java.

El proceso ideal sería conseguir una aplicación modular, con código fácilmente reutilizable y escalable, que permita realizar cambios de forma sencilla y sin esfuerzo en un futuro, así como trasladar parte o la totalidad de su funcionalidad a un entorno Web. Con este fin se ha seguido un proceso cuidado de diseño, empleando patrones actuales.

El resultado es una aplicación que permite, desde una interfaz de usuario amigable, la introducción de los datos de un estudio de neuroimagen a evaluar y ejecutar el proceso de comparación.

¹⁷<http://www.java.com/>

¹⁸<http://www.apache.org/>

3.1. Análisis

Sin entrar en un análisis profundo sobre requisitos funcionales, vamos a identificar las necesidades y operaciones que la aplicación debe cubrir.

Básicamente, la herramienta deberá de ser capaz de realizar las siguientes tareas:

1. Acceder a ficheros OWL para generar una representación interna del modelo.
2. Permitir al usuario introducir los datos de un estudio. En el contexto de BrainMap estos son:
 - a) Sujetos
 - b) Condiciones
 - c) Experimentos
3. Actualizar el modelo según los datos introducidos por el usuario.
4. Realizar el cálculo de las métricas definidas.
5. Presentar al usuario los resultados de forma gráfica y amigable.

3.1.1. Casos de Uso

En vista de las funcionalidades identificadas, surge una serie de casos de uso de la herramienta. La siguiente tabla muestra el conjunto de casos de usos. Puesto que el actor que los inicia siempre es el usuario de la aplicación, estos se muestran simplificados en nombre, descripción, datos necesarios y salida generada:

3.2. Diseño

Para modularizar el código de la aplicación, se ha seguido el patrón de diseño Model View Controller (MVC), un patrón ampliamente aceptado e implantado dentro de la Ingeniería del Software consistente en separar el acceso a los datos, la lógica de negocio y la forma en que se muestra al usuario. Los componentes del mismo son los siguientes:

Nombre	Descripción	Datos	Salida
Añadir sujetos	Añade un nuevo grupo de sujetos al modelo de la ontología	Nombre, género, diagnóstico, edad mínima, máxima y media	Una subclase de Subjects y el conjunto de axiomas como subclase de la misma
Añadir condición	Añade una nueva condición al modelo de la ontología	Estímulo, respuesta e instrucción	Una subclase de Condition y el conjunto de axiomas como subclase de la misma
Añadir experimento	Añade un nuevo experimento al modelo de la ontología	Contexto, sujetos, paradigma, condición y localizaciones	Una subclase de Experiment y el conjunto de axiomas como subclase de la misma
Editar sujetos	Edita un grupo de sujetos existente y lo modifica en la ontología	Una subclase de Subjects ya definida en la ontología evaluada	Una subclase de Subjects y el conjunto de axiomas como subclase de la misma
Editar condición	Edita una condición existente y la modifica en la ontología	Una subclase de Condition ya definida en la ontología evaluada	Una subclase de Condition y el conjunto de axiomas como subclase de la misma
Editar experimento	Edita un experimento existente y lo modifica en la ontología	Una subclase de Experiment ya definida en la ontología evaluada	Una subclase de Experiment y el conjunto de axiomas como subclase de la misma
Eliminar sujetos	Elimina de la ontología un grupo de sujetos	Una subclase de Subjects ya definida en la ontología evaluada	
Eliminar condición	Elimina de la ontología una condición	Una subclase de Condition ya definida en la ontología evaluada	
Eliminar experimento	Elimina de la ontología un experimento	Una subclase de Experiment ya definida en la ontología evaluada	
Comparar estudios	Comienza la comparación del estudio con el conjunto de estudios de BrainMap	El conjunto de conceptos definido en la ontología evaluada	Un conjunto de resultados de métricas

Cuadro 5: Casos de Uso

Modelo	Es la representación de los datos y las reglas que determinan el acceso y las actualizaciones de estos datos.
Vista	Renderiza los contenidos de un modelo, especificando exactamente como el modelo y los datos deben ser presentados. Si el modelo de datos cambia, la vista cambia su presentación según la necesidad. Existen dos estrategias para conseguir esto: <ol style="list-style-type: none"> 1. Empleando un modelo de notificación, en el que la vista se registra al modelo para recibir notificaciones 2. Otra que sigue un modelo de petición, en el cual la vista es responsable de llamar al modelo cuando necesita recuperar los datos actuales.
Controlador	Traduce las interacciones del usuario con la vista en acciones que normalmente se traducen en cambios del modelo.

El conjunto de clases definidas en el modelo, suele tener correspondencia con la mayoría de los datos que maneja la aplicación. En nuestro caso, será la propia ontología utilizando para ello el modelo definido en la librería owl-api.

3.2.1. Arquitectura

Al emplear el patrón MVC, la arquitectura resultante es algo parecido a una arquitectura de tres capas:

1. Presentación.
2. Lógica de negocio.
3. Acceso a los datos.

Sin embargo, la correspondencia no es directa, ya que la Vista (equivalente a la capa de presentación) suele estar acoplada al Controlador, y este a su vez con la capa de acceso a datos. Por tanto la arquitectura resultante es la mostrada en la figura 12.

Como se puede apreciar en la figura, la librería OWL-API tiene un papel doble.

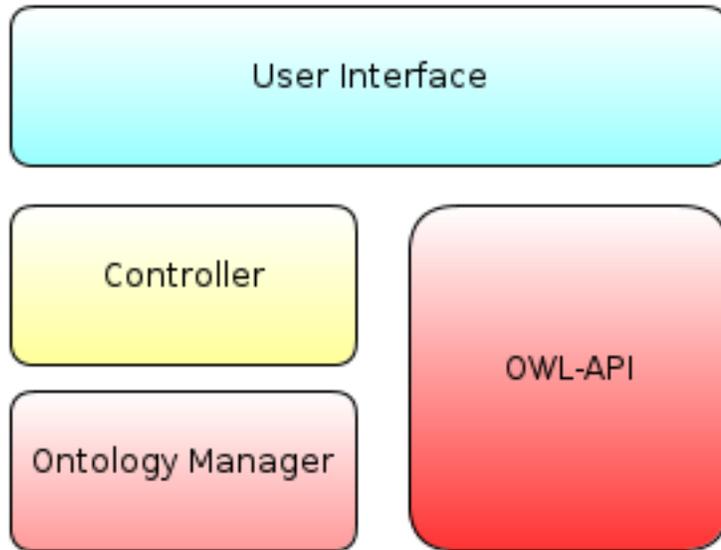


Figura 12: Arquitectura de la aplicación

1. Por un lado, ofrece la capa de acceso a datos (ficheros de ontologías), ya que contiene el conjunto de operaciones necesarios para leer y escribir en este formato, abstrayendo al programador de esta tarea.
2. Ofrece también una abstracción muy completa del modelo de OWL2, evitando la necesidad de definir clases como OWLClass, Property... Por ello, se emplea como capa de modelo en la aplicación.

Por otro lado, el componente OntologyManager implementará un conjunto de operaciones puente entre las operaciones de acceso básicas de OWL-API y el resto de componentes de la interfaz.

3.2.2. Diseño de clases

Como ya se ha mencionado, es muy importante mantener una separación clara entre los diferentes componentes de la aplicación (interfaz visual, operaciones lógicas, acceso a datos...). Con este fin se ha cuidado el diseño de paquetes y clases de forma que sea posible manipular estos componentes de la forma más independiente posible.

Los paquetes definidos en la implementación son los siguientes:

es.uned.comparator.core Contiene las operaciones básicas de la aplicación: acceso, representación y modificación de ontologías.

es.uned.comparator.core.metrics Contiene el conjunto de clases necesarias para la implementación de las métricas definidas en el apartado 2.2.

es.uned.comparator.controller Contiene las clases controladoras de la aplicación: las acciones relacionadas con los botones de la interfaz de usuario.

es.uned.comparator.ui Contiene las clases principales de la interfaz de usuario.

es.uned.comparator.ui.components Contiene componentes comunes y reutilizables de la interfaz de usuario.

es.uned.comparator.ui.renderers Clases de renderizado para ciertos componentes de la aplicación.

es.uned.comparator.ui.resources Contiene el conjunto de recursos de imágenes e iconos para la interfaz de usuario.

Por simplicidad y contención, describiremos únicamente el paquete core, ya que es el que contiene la implementación de los métodos descritos en este trabajo. De igual forma, los diagramas de clases no muestran todos los atributos y operaciones codificados en las clases, sólo aquellos considerados más importantes.

Paquete core

El paquete core implementa el núcleo de operaciones lógicas y acceso a datos de la aplicación. Se ha tomado esta decisión porque el tener este conjunto aglutinado en un paquete permitirá que en un futuro pueda integrarse en otros entornos y aplicaciones.

Este paquete a su vez contiene otro paquete, denominado metrics, donde se implementan el conjunto de métricas definidas en el apartado 2, así como otras clases de utilidad. La figura 13 muestra el diagrama de clases.

OntologyManager Es el punto principal de interacción con el modelo de las ontologías, implementando el conjunto de operaciones que cargan, alteran y serializan estas. De esta forma encapsula gran parte de la funcionalidad de OWL-API, desacoplando la dependencia de la librería.

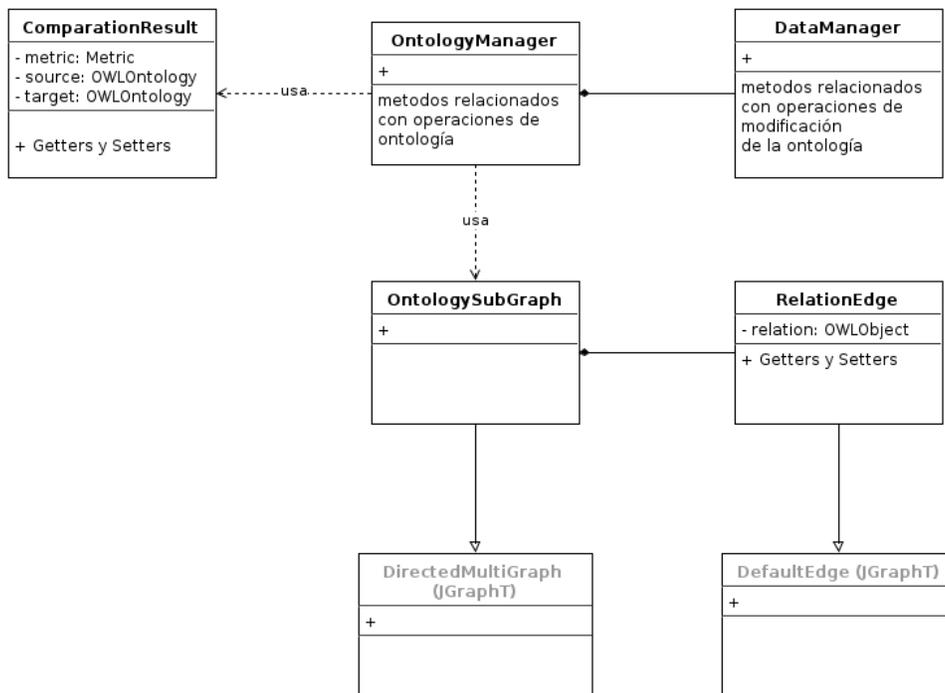


Figura 13: Diagrama de clases del paquete es.uned.comparator.core

Es decir, las clases de la aplicación no interactúan directamente con operaciones de OWL-API, sino que lo harán con `OntologyManager`, reduciendo así la dependencia de la librería y evitando problemas en caso de cambios en la misma (que se extenderían por toda la aplicación). En caso de necesidad también facilita el cambio por otra.

DataManager Se encarga de las operaciones de modificación de la ontología evaluada. Es una partición de la funcionalidad de `OntologyManager` para evitar el crecimiento de esta última.

OntologySubGraph Modela el subgrafo obtenido de una ontología. Para ello hereda de la clase `DirectedMultiGraph` de la librería `JGraphT`.

RelationEdge Hereda de la clase `DefaultEdge` de `JGraphT` para representar el arco dentro de un `OntologySubGraph`, de forma que pueda llevar asociada la propiedad OWL de la que surge este arco.

ComparisonResult Una clase de modelo empleada para representar el resultado de computar una métrica, relacionándola con la ontología evaluada y la ontología objetivo. Después de la ejecución de una métrica la vista renderiza los resultados a partir de los objetos de esta clase.

El subpaquete de `metrics` es uno de los más importantes de la aplicación, ya que en él se implementan el conjunto de clases que soportan la funcionalidad de las medidas definidas. Para conseguir un mayor desacoplamiento en otros componentes, este paquete ha seguido un diseño por interfaces, de forma que es posible implementar patrones de inicialización como `Factory`, evitando la dependencia sobre implementaciones concretas de las métricas. La figura 14 muestra el diagrama de clase de este paquete.

Metric Es la interfaz que define la funcionalidad básica que una métrica, en el contexto de la aplicación, debe tener. Cualquier clase que la implemente debe proporcionar un nombre (utilizado para la factoría).

LogicConsistency Implementa la funcionalidad de la métrica de consistencia lógica descrita en el apartado 2.2.1.

VectorSpaceModel Implementa la métrica de grafo VSM definida en el apartado 2.2.3.

SemanticGraph Implementa la métrica de grafo semántico descrita en el punto 2.2.2.

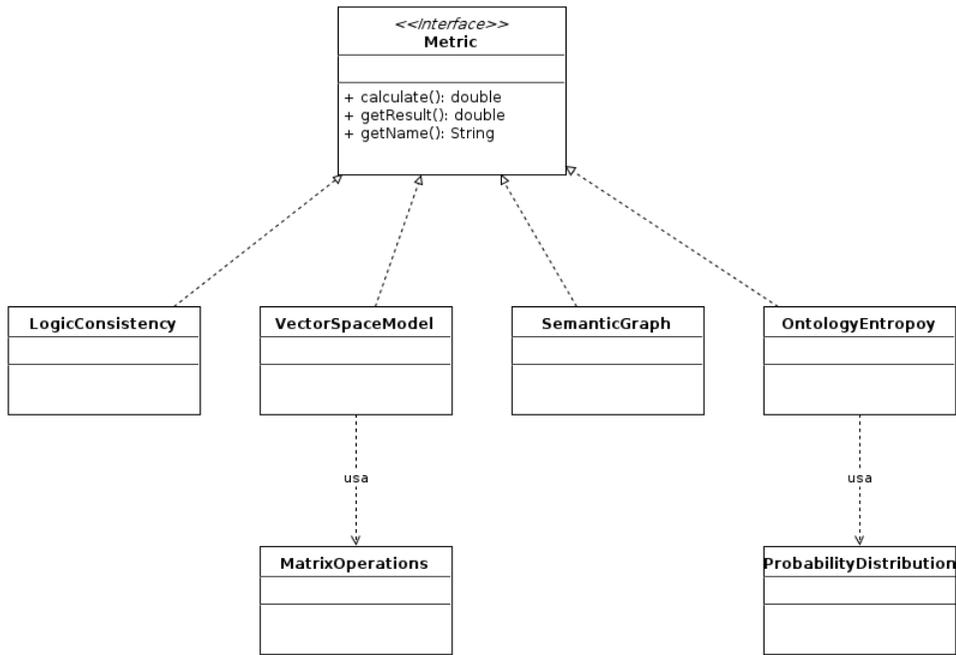


Figura 14: Diagrama de clases del paquete es.uned.comparator.core.metrics

OntologyEntropy Implementa la funcionalidad de la métrica entropía de una ontología descrita en el apartado 2.2.4.

MatrixOperations Una clase de utilidad centrada en implementar las operaciones con matrices necesarias para la métrica VectorSpaceModel (producto escalar, multiplicación, normas matriciales, etc.)

ProbabilityDistribution Implementa la función de probabilidad para un grafo de ontología descrita 2.2.4.

Diseño de casos de uso

Como hemos visto en el apartado de análisis de casos de uso, las operaciones a realizar por la herramienta son de naturaleza similar. Para observar la estructura e interacciones de diferentes partes del diseño, ejemplificaremos el diagrama de secuencia de dos operaciones:

1. La adición de una condición.
2. El proceso de comparación.

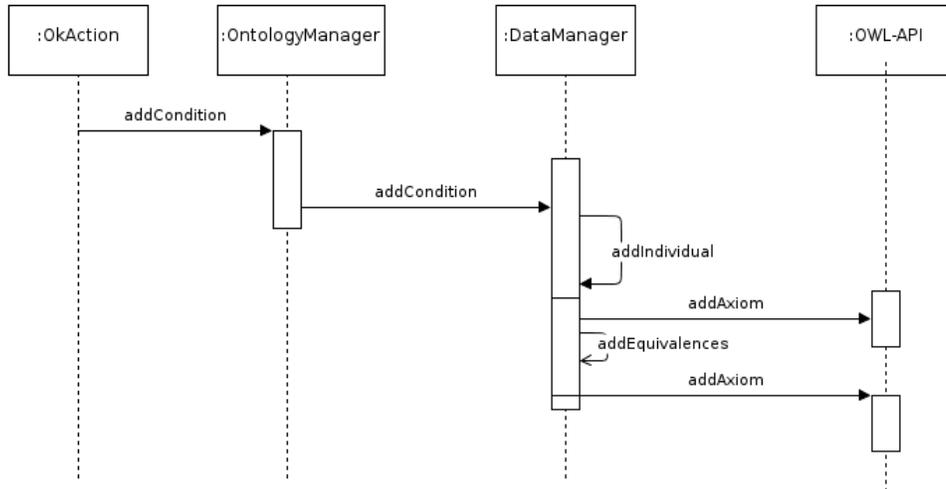


Figura 15: Diagrama de secuencia del caso de uso Añadir condición

En el primer caso, se observa la comunicación entre clases de control (OkAction) con las de gestión de la ontología, que a su vez utilizan las operaciones ofrecidas por la librería OWL-API.

El caso del proceso de cálculo es más interesante, ya que precisa de la interacción entre diferentes capas de la aplicación. En el diagrama de la figura 16 se ha utilizado un código de colores para distinguir su pertenencia a cada capa:

- Marrón, capa de control.

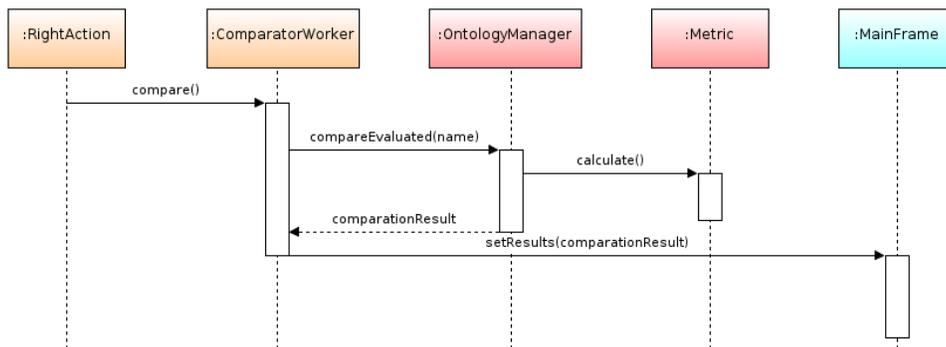


Figura 16: Diagrama de secuencia de la operación calcular.

- Rojo, capa núcleo y acceso a datos.
- Azul, capa de vista.

Hasta aquí queda detallado el proceso de diseño software de la herramienta. En el siguiente apartado se muestra el diseño de la interfaz visual de la aplicación.

3.3. Interfaz de usuario

Para el diseño de la interfaz de usuario se han seguido dos máximas:

Simplicidad Dentro de la medida de lo posible, la interfaz debe proporcionar un diseño que permita añadir la información de forma simple e intuitiva.

Similitud con Scribe Dado que los datos a recoger son gran parte de los definidos en el esquema de codificación de metadatos de BrainMap, lo más lógico es basarse en su herramienta para la introducción de estudios: Scribe.

Par la implementación de la interfaz de usuario se ha empleado la API Swing, incluida en la distribución de Java J2SE.

3.3.1. Ventana principal

La ventana principal muestra los tres principales conceptos (correspondientes con la ontología) que definen un estudio:

- Los sujetos
- Condiciones
- Experimentos.

Para ello se sirve de tres paneles que contienen los botones de operación y la lista de items añadidos.

Estos paneles son componentes reutilizables implementados bajo la clase “OperationPanel”, dentro del paquete `es.uned.comparator.ui.components`. Todos los componentes visuales reutilizables se encuentran alojados en este paquete.

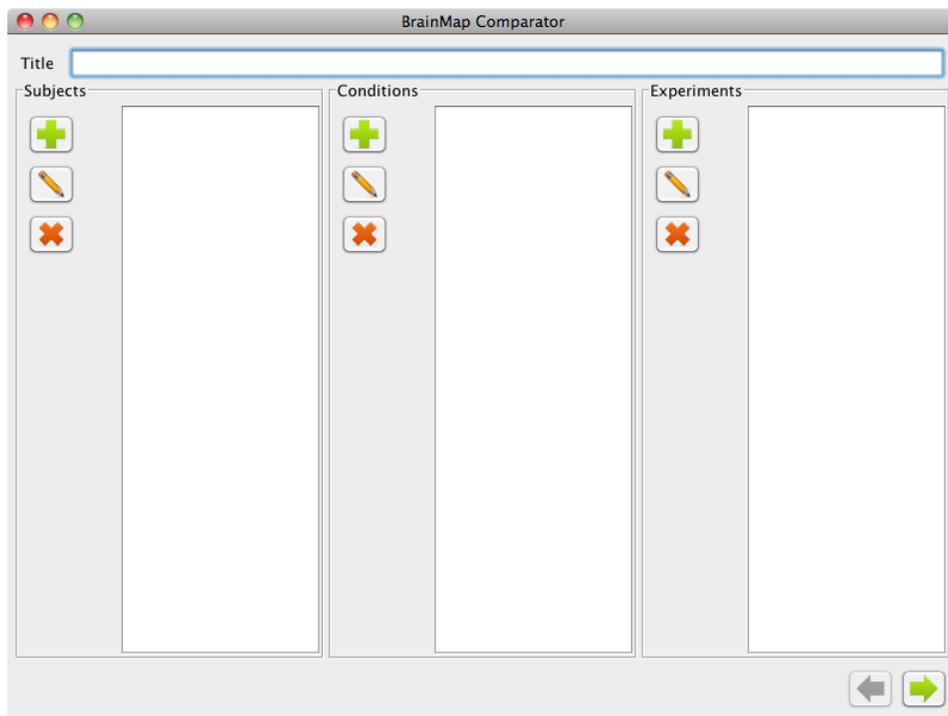


Figura 17: Ventana principal

La funcionalidad que ofrece cada botón es la misma, cambiando sólo el diálogo a mostrar, la lista a modificar (perteneciente al panel) y la superclase de los items (Subjects, Behavioral Experimental Paradigm Condition o Experiment). El renderizado de este tipo de objetos en la interfaz visual se hace por medio de los renderers (paquete `es.uned.comparator.ui.renderers`), que obtienen la anotación de tipo `rdfs:label` de la clase a mostrar.

Siguiendo esta aproximación, se evita la necesidad de codificar repetidamente las acciones relacionadas con los botones o la colocación de los componentes visuales.

3.3.2. Diálogo Subjects

Al hacer click en el botón  o  del panel *Subjects* se abre el diálogo New/Edit Subject Group. Este diálogo permite la introducción por parte del usuario del conjunto de datos ya definidos anteriormente.

A partir de estos datos, una vez pulsado OK, el controlador añade dentro de la ontología evaluada una nueva subclase a Subjects y, a su vez, una Subclase con axioma de clase equivalente por cada uno de los datos.

Este comportamiento de nuevo es común, por lo que está definido en la clase “AbstractEditorDialog”, de la que todos los diálogos heredan.

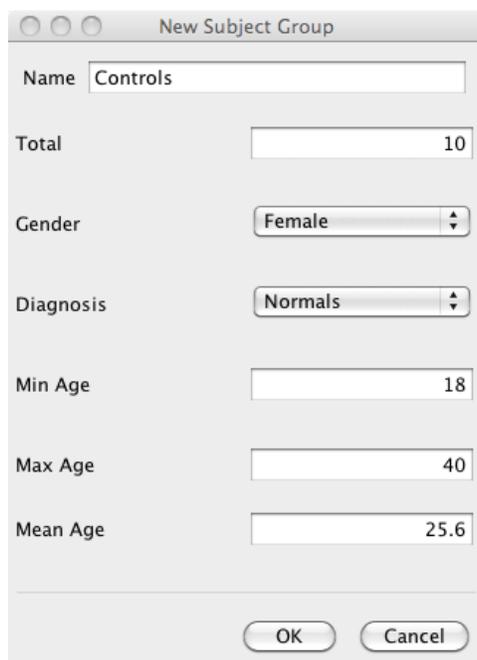
3.3.3. Diálogo Conditions

De forma similar, al hacer click en los botones correspondientes del panel *Conditions*, se abre el diálogo de introducción de condiciones. En este caso, los datos dependen completamente de los conceptos definidos en la ontología BrainMap, por lo que deben mostrarse en opciones desplegadas para ser seleccionadas por el usuario.

De nuevo, esta situación se da de forma repetida, por lo que se ha implementado como el panel reutilizable SelectorPanel, compuesto por el desplegable, botones para añadir/quitar y la lista de selección.

3.3.4. Diálogo Experiment

El diálogo de experimentos es el más complejo, ya que tiene que mostrar conceptos tanto definidos en la ontología de dominio como los que el usuario ha introducido al añadir sujetos y condiciones.



The image shows a dialog box titled "New Subject Group". It contains several input fields and dropdown menus. The "Name" field is set to "Controls". The "Total" field is set to "10". The "Gender" dropdown menu is set to "Female". The "Diagnosis" dropdown menu is set to "Normals". The "Min Age" field is set to "18". The "Max Age" field is set to "40". The "Mean Age" field is set to "25.6". At the bottom of the dialog box, there are two buttons: "OK" and "Cancel".

Field	Value
Name	Controls
Total	10
Gender	Female
Diagnosis	Normals
Min Age	18
Max Age	40
Mean Age	25.6

Figura 18: Diálogo de sujetos

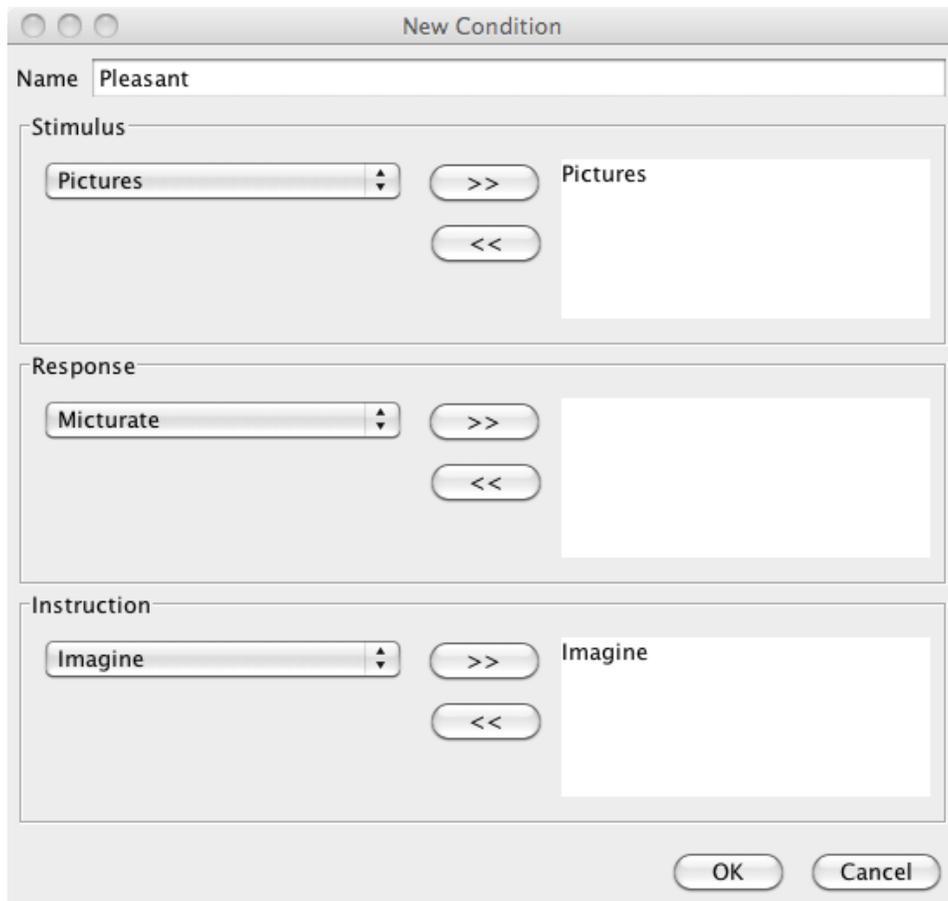


Figura 19: Diálogo de condiciones

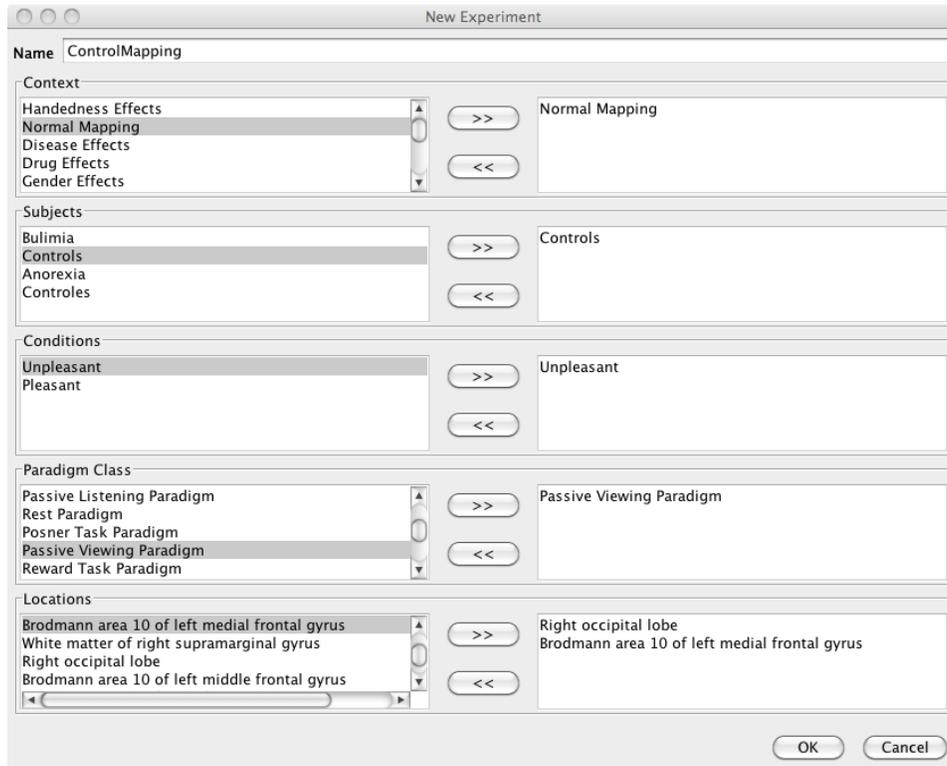


Figura 20: Diálogo de experimentos

La aproximación es la misma que en el resto de diálogos, por medio de un panel reutilizable se evita tener que programar toda la serie de componentes. Sólo cambia el título y la lista de conceptos que se obtienen de la ontología.

3.3.5. Ventana de resultados

Tras pulsar el botón  la aplicación realiza el proceso de comparación con las ontologías y muestra la ventana de resultados. Esta consiste en un conjunto de gráficas, una por cada métrica, implementadas por medio de la librería JFreeChart.

Cada gráfica muestra el valor de métrica que ha obtenido la ontología comparada con el estudio introducido.

El usuario puede volver a la ventana principal y editar su estudio pulsando el botón .

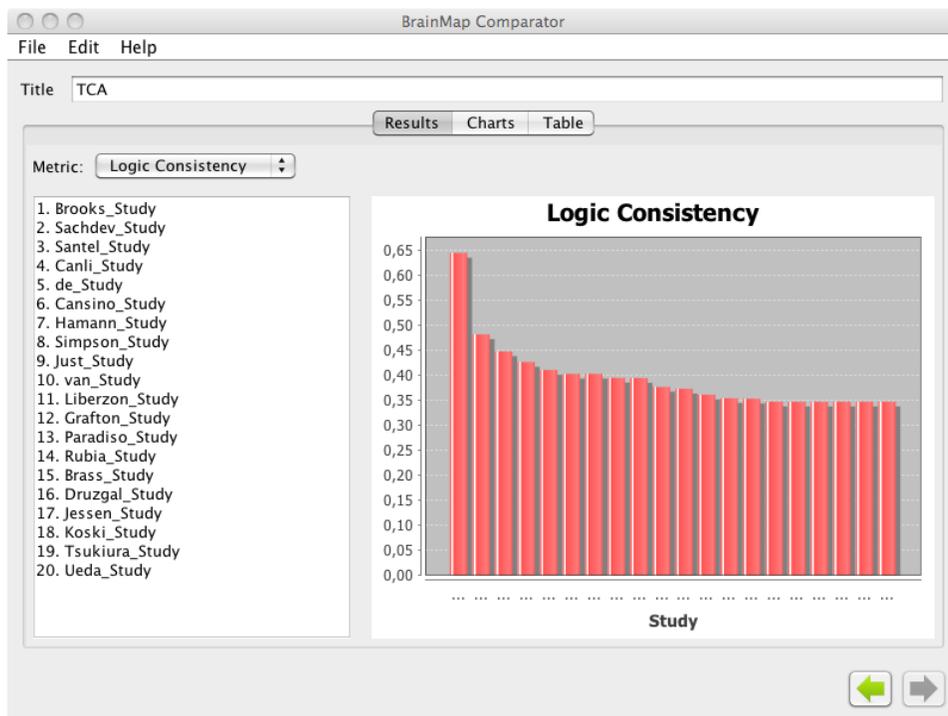


Figura 21: Ventana de resultados

El manual de usuario completo se encuentra en el anexo 7.1.

4. Resultados

El proceso de validación de la herramienta consiste en ejecutar el proceso de cálculo y comparar el resultado ofrecido con la opinión y supervisión del experto. Para ello, se ha contado con la colaboración de un experto Psiquiatra que actualmente está desarrollando un estudio de neuroimagen para comprobar la valencia emocional en Trastornos de la Conducta Alimentaria.

La evaluación ha consistido por tanto en introducir como estudio evaluado el estudio sobre TCA, realizar el proceso de comparación y por último la validación de los resultados de la herramienta contra los criterios del experto.

4.1. Modelado del estudio a evaluar según BrainMap

El estudio modelo consiste en una evaluación de la respuesta emocional de los sujetos ante imágenes agradables, desagradables y neutras obtenidas del test International Affective Picture System (IAPS). Los grupos de sujetos están compuestos por controles, pacientes con diagnóstico Anorexia Nervosa y pacientes con diagnóstico Bulimia Nervosa. Los experimentos consisten en mostrar, en diferentes bloques (placentera, neutra, displacentera), imágenes del IAPS para observar la activación ante la respuesta emocional entre los diferentes grupos de sujetos.

Siguiendo el modelo definido en la ontología desarrollada, las características introducidas en la herramienta son las siguientes:

- Subjects
 - Healthy Controls
 - Anorexia Nervosa Patients
 - Bulimia Nervosa Patients
- Instructions
 - Pleasant
 - Unpleasant

- Experiments
 - Controls pleasant
 - Controls unpleasant
 - Anorexia+Bulimia pleasant
 - Anorexia+Bulimia unpleasant

Cada uno de los conceptos se desglosan en su tabla correspondiente.

Dato	Valor
Name	Healthy Controls
Total	25
Gender	Female
Diagnosis	Normals
Min Age	18
Max Age	25
Mean Age	23

Dato	Valor
Name	Anorexia Nervosa Patients
Total	11
Gender	Female
Diagnosis	Anorexia Nervosa
Min Age	19
Max Age	35
Mean Age	27

Dato	Valor
Name	Bulimia Nervosa Patients
Total	8
Gender	Female
Diagnosis	Bulimia Nervosa
Min Age	20
Max Age	30
Mean Age	25

(a) Controles

(b) Anorexia

(c) Bulimia

Cuadro 6: Sujetos

Dato	Valor
Stimulus	Pictures
Response	
Instruction	Imagine

Dato	Valor
Stimulus	Pictures
Response	
Instruction	Imagine

(a) Pleasant

(b) Unpleasant

Cuadro 7: Instructions

Dato	Valor	Dato	Valor
Context	Control Mapping	Context	Control Mapping
Subjects	Healthy Controls	Subjects	Healthy Controls
Conditions	Pleasant	Conditions	Unpleasant
Paradigm	Passive Viewing Paradigm	Paradigm	Passive Viewing Paradigm
Locations	Left Amygdala, Right Amygdala, Right Thalamus, Left Hippocampus	Locations	Left Amygdala, Right Amygdala, Right Thalamus, Left Hippocampus, Left Pallidum

(a) Controls pleasant

(b) Controls unpleasant

Cuadro 8: Experiments a

Dato	Valor	Dato	Valor
Context	Disease Effects	Context	Disease Effects
Subjects	Anorexia Nervosa Patients Bulimia Nervosa Patients	Subjects	Anorexia Nervosa Patients Bulimia Nervosa Patients
Conditions	Pleasant	Conditions	Unpleasant
Paradigm	Passive Viewing Paradigm	Paradigm	Passive Viewing Paradigm
Locations	Left Amygdala, Right Amygdala, Right Thalamus, Left Hippocampus	Locations	Left Amygdala, Right Amygdala, Left Hippocampus

(a) Anorexia+Bulimia pleasant

(b) Anorexia+Bulimia unpleasant

Cuadro 9: Experiments b

4.2. Validación

El proceso de validación ha consistido en los siguientes pasos:

1. Ejecución de la herramienta.
2. Etiquetado del conjunto de datos por parte del experto.
3. Cálculo de las métricas de evaluación

Exceptuando el proceso de etiquetado del conjunto de datos, el proceso de validación está automatizado por medio de un script. La herramienta serializa en formato csv los resultados calculados, de forma que el script sólo tiene que

comprobar cada uno contra los datos etiquetados, descartando los fallidos, para calcular las métricas de validación.

A continuación, se comentará los detalles sobre el conjunto de datos empleados, el proceso de etiquetado, las métricas de validación empleadas y sus resultados.

4.2.1. Conjunto de datos

Una de los principales problemas en el desarrollo de este trabajo ha sido la imposibilidad de acceder directamente al sistema de base de datos de BrainMap. Esto ha supuesto que el conjunto de datos disponibles para realizar las pruebas sea limitado, principalmente por dos motivos.

1. BrainMap no es accesible de forma programática, por lo que los estudios resultado de una búsqueda candidatos a formar parte de los datos se han tenido que codificar de forma semiautomática.
2. BrainMap no cuenta con un conjunto lo suficientemente grande (aproximadamente 2000 estudios), sobre todo de cara a obtener resultados al buscar paradigmas, experimentos o enfermedades raras o poco comunes en la corriente de investigación. De hecho, sólo cuenta con tres estudios relacionados con los TCA.

Como se ha comentado, el programa Sleuth permite realizar búsquedas bajo ciertos criterios en BrainMap, mostrando los resultados en forma de texto estructurado por un conjunto de palabras clave, la mayoría correspondientes al esquema de codificación.

Trasladar el conjunto de estudios obtenidos en Sleuth a su representación ontológica de forma manual es un proceso que consume gran cantidad de tiempo. Por ello, para poder obtener un conjunto lo suficientemente grande, se ha desarrollado un a aplicación diseñada para parsear el contenido textual de Sleuth y convertirlo en una ontología que representa el estudio seleccionado. De esta forma, pasamos de un proceso completamente manual a uno semi automático, donde sólo es necesario copiar el texto relevante de Sleuth y pegarlo en los componentes de la aplicación.

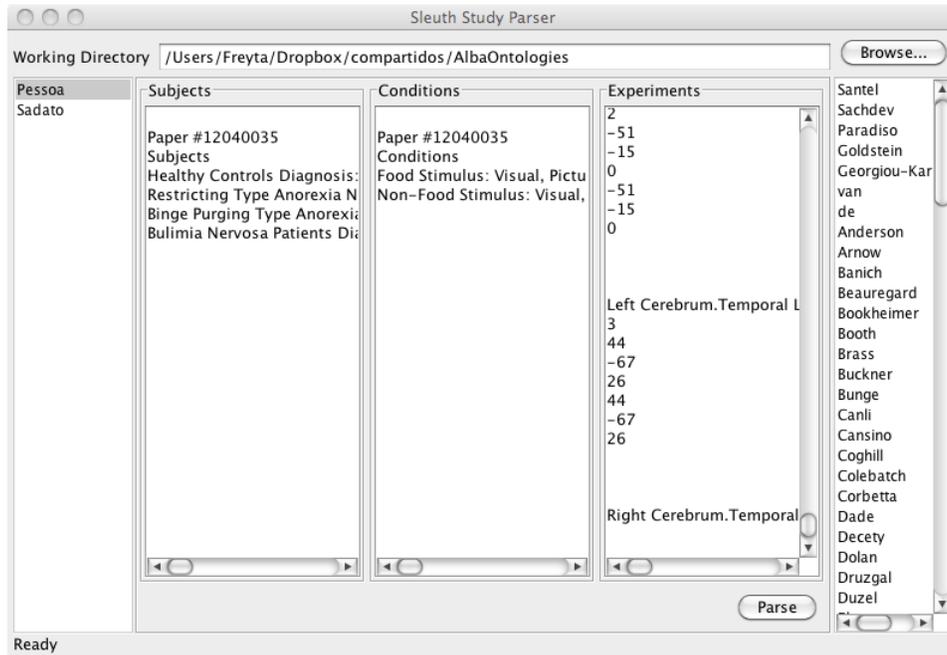


Figura 22: Aplicación de procesado de texto de Sleuth

Con todo esto, se ha conseguido un conjunto de datos de prueba consistente en 116 estudios, obtenidos en Sleuth según los criterios definidos en el apartado 2.3. Hay que tener en cuenta, que por inconsistencias en las descripciones que algunos estudios ofrecen en Sleuth, el procesado de algunos estudios no resulta completo, produciendo fallos en el proceso de cálculo de las métricas. La herramienta detecta estos problemas, descartando su evaluación para la métrica en la que falle, por lo que el dataset varía de unas métricas a otras.

En el contexto del caso de estudio, basado en TCA, existe muy poca bibliografía contenida en BrainMap, por lo que dentro del conjunto extraído sólo tres estudios se encuadran en este área, pero se dan algunos coincidentes en contexto emocional.

4.2.2. Etiquetado del experto

Para poder valorar la eficiencia de la herramienta, es necesario contar con la valoración que daría un experto para cada estudio. Para ello, el experto ha clasificado, a partir de la información de Sleuth, cada uno de los estudios del conjunto de datos. La clasificación consiste asignar una de las tres etiquetas:

1. Poco similar.
2. Similar.
3. Muy Similar.

Por ejemplo, el estudio sobre TCA de Brooks et al. cuya descripción (simplificada) es:

- Sujetos.
 - Anorexia.
 - Bulimia.
 - Controles.
 - Paradigma.
 - Passive Viewing.
 - Estímulo.
 - Fotos sobre comida o no comida.

Se ha clasificado como *Muy Similar*.

El objetivo de la herramienta es mostrar al experto los estudios de neuroimagen relevantes según el modelo de su estudio, por lo que resulta en una clasificación binomial relevante/no relevante. El concepto de relevancia se ha definido como aquellos que son similares o muy similares. Por tanto, el acierto o fallo de la herramienta dependerá de si considera relevante un estudio similar/muy similar, o no, así como su descarte en caso de no serlo.

Lógicamente, los estudios centrados en TCA obtienen una valoración de similitud alta. Cabe destacar que aunque los tres cubren este área, hay uno que destaca, siendo clasificado como muy similar (Brooks), principalmente por el conjunto de sujetos empleados y las condiciones de los experimentos.

4.2.3. Métricas de evaluación

Para evaluar el resultado de cada métrica, se ha contextualizado la herramienta dentro de la búsqueda de estudios relevantes, es decir, como si de un buscador de estudios similares se tratara.

Partiendo de este contexto, se usarán algunas de las métricas más empleadas en tareas de Recuperación de Información (IR): Precision, Recall y F-Measure[32]. Siguiendo la definición de[33]:

Estudio	Valoración
Santel	Similar
Georgiou	Poco similar
Paradiso	Poco similar
Sachdev	Similar
Goldstein	Poco similar
Brooks	Muy similar

Cuadro 10: Algunas valoraciones del experto

Precision Mide el número de items identificados correctamente como un porcentaje del número de items identificados. Es decir, mide cuántos de los items que el sistema ha identificado son realmente correctos, sin importar si también ha fallado al obtener items correctos. Cuanto mayor sea la precisión, mejor es el sistema asegurando que lo identificado es correcto. Se define como

$$Precision = \frac{Correcto + 1/2Parcial}{Correcto + Fallo + Parcial}$$

donde *Correcto* es el número de estudios clasificados correctamente, *Parcial* el número estudios cercanos a su clasificación correcta (por exceso o por defecto) y *Fallo* el número de estudios clasificados de forma incorrecta.

Recall Mide el número de items identificados correctamente como un porcentaje del numero total de items: mide cuantos de los items que deberían haber sido identificados lo fueron realmente, sin importar el número de fallos cometidos. Cuanto más Recall tenga un sistema mejor será de cara a no perder items correcto. Se define como:

$$Recall = \frac{Correcto + 1/2Parcial}{Correcto + Perdidos + Parcial}$$

F-measure Esta medida[34] se suele usar junto a Precision-Recall, como una media ponderada de las dos. Se define como:

$$F - measure = \frac{(\beta^2 + 1)P \cdot R}{(\beta^2 R) + P}$$

Donde β refleja el peso de Precision contra Recall. El caso más habitual es darle la misma importancia a Precision y Recall con un peso de 0.5, por lo que la ecuación queda como

$$F_1 = \frac{P \cdot R}{0,5(P + R)}$$

Para el término *Parcial*, la definición adoptada es la de un item erroneo dentro de un margen de $\pm 0,05$. Este concepto resulta útil especialmente para el refinado de las métricas, permitiendo identificar estudios límite.

4.2.4. Resultados Consistencia Lógica

Tras descartar los estudios fallidos, el conjunto de datos empleados en la métrica de Consistencia Lógica se muestran en la figura 23.

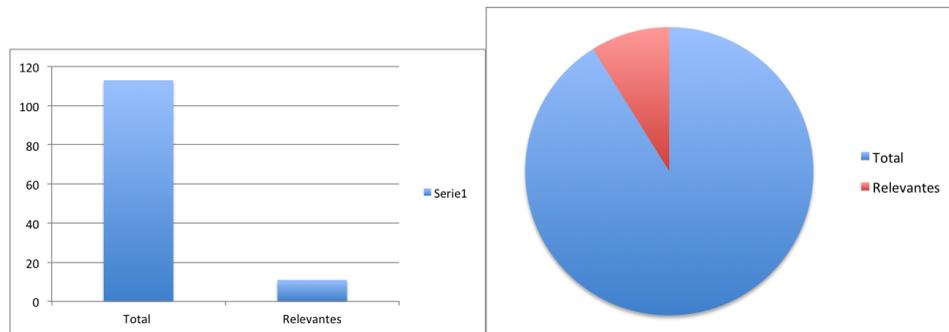


Figura 23: Conjunto de datos para Consistencia Lógica

El resultado de ejecutar la métrica en la aplicación se muestra en la figura 24. En ella se puede observar un número importante de estudios considerados relevantes, de los cuales la mayoría lo son. Sin embargo, algo menos de la mitad han resultado sobreestimados, de forma que no resultan relevantes en el contexto del estudio sobre TCA evaluado por la herramienta.

Una vez obtenidos los datos de la ejecución, se ejecuta el script de validación para calcular los valores de las medidas de evaluación.

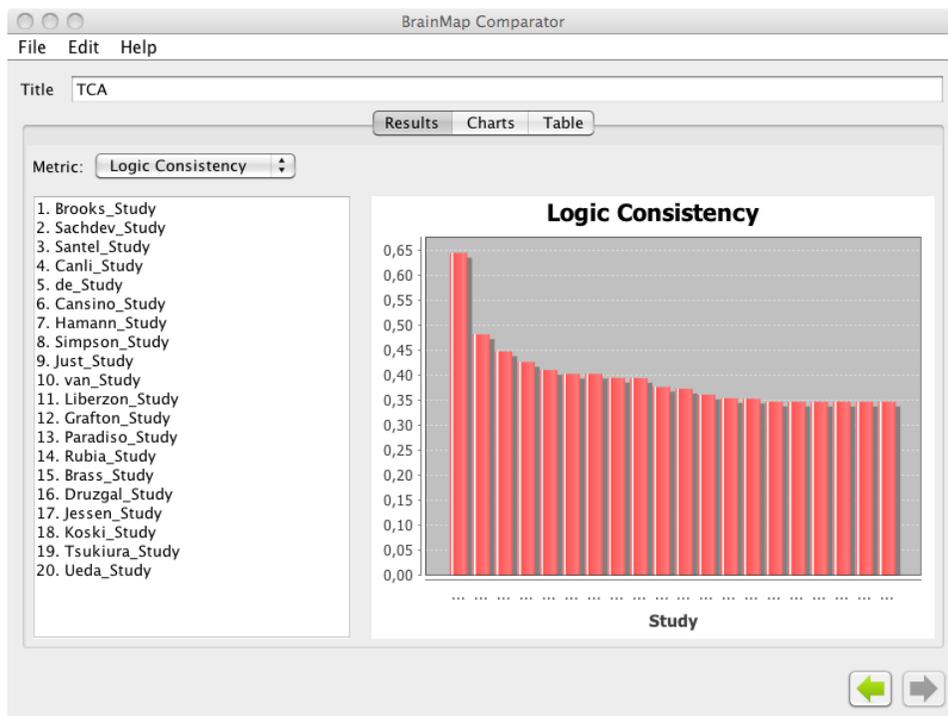


Figura 24: Panel de resultados tras ejecutar la métrica

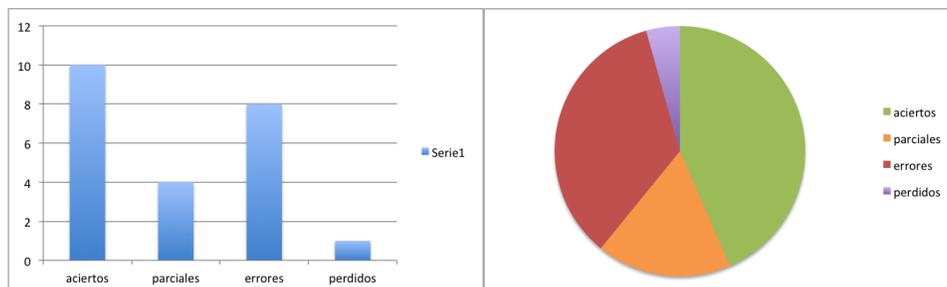


Figura 25: Resultados de Consistencia Lógica para los relevantes

El cuadro 11 muestra los resultados tras aplicar las métricas de evaluación.

Precision	Recall	F-Measure
0.6	0.75	0.667

Cuadro 11: Resultados de las métricas para Consistencia Lógica

Observando los resultados, se puede apreciar que la métrica de Consistencia Lógica funciona razonablemente bien identificando estudios que realmente son relevantes, pero sobreestima gran cantidad que no lo son, perdiendo precisión. Más tarde se comentarán las posibles fuentes de los problemas detectados.

4.2.5. Resultados Grafo Semántico

Al igual que con Consistencia Lógica, cierto número de estudios no se han modelado correctamente por inconsistencias en Sleuth. Tras su filtrado el conjunto de datos es el mostrado en la figura 26.

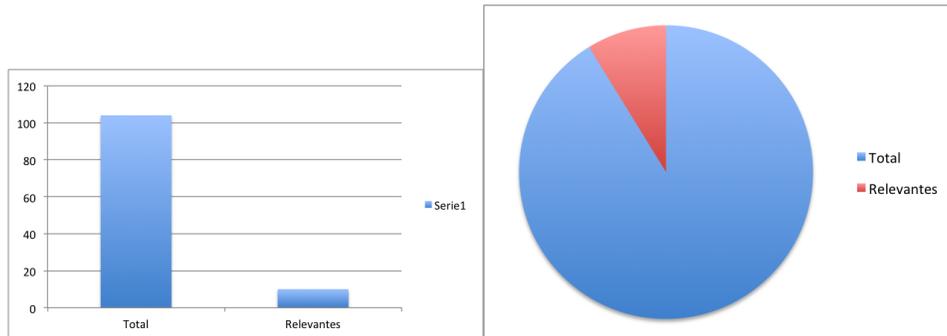


Figura 26: Conjunto de datos para Grafo Semántico

Tras la ejecución, en el panel de resultados, seleccionando la métrica grafo semántico, se puede observar que sólo recupera dos estudios, pero ambos muy relevantes ya que son estudios muy parecidos centrados en el estudio de los TCA, por lo que la precisión es muy alta. Sin embargo, pierde gran cantidad de estudios que, aunque en menor medida, resultan relevantes, lo que se deduce en un recall muy bajo. Estos resultados indican que la métrica es muy restrictiva.

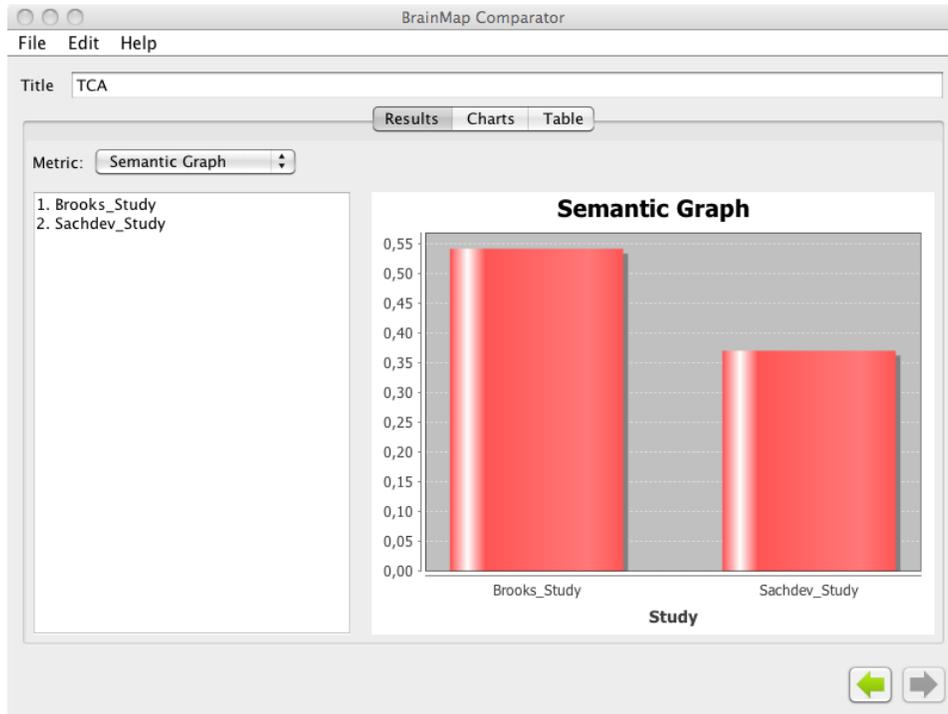


Figura 27: Panel de resultados para la métrica Semantic Graph

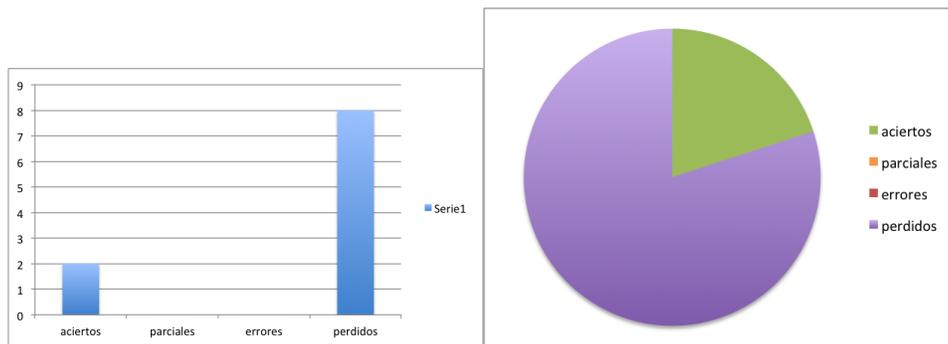


Figura 28: Resultados de Consistencia Lógica con los elementos relevantes

Precision	Recall	F-Measure
1	0.2	0.333

Cuadro 12: Resultados de las métricas para Grafo Semántico

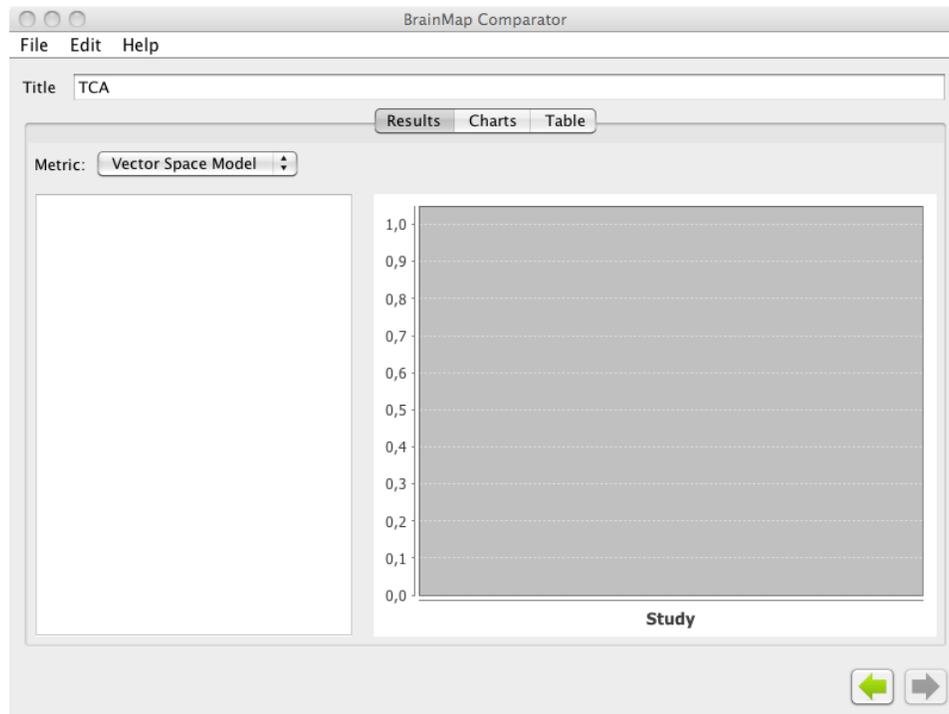


Figura 29: Panel de resultados con Grafo VSM seleccionado

4.2.6. Resultados Grafo VSM

Los resultados de esta métrica han sido realmente pobres, siendo incapaz de clasificar ninguna de las instancias, por lo que se omite su validación con las medidas de evaluación.

4.2.7. Evaluación de la validación

Hasta aquí se ha realizado el proceso validación de las métricas empleando medidas típicas del área IR.

Métrica	Precision	Recall	F-Measure
Consistencia Lógica	0.6	0.75	0.667
Grafo Semántico	1	0.2	0.333

Cuadro 13: Resultados de las medidas de evaluación

Los resultados obtenidos varían de unas a otras, destacando especialmente la métrica Consistencia Lógica, ya que ha sido capaz de recuperar gran cantidad de estudios relevantes. Por tanto podemos afirmar que, a pesar de ser memorables, la eficiencia de dicha métrica es suficiente como para resultar de utilidad.

Por otro lado, la métrica Grafo Semántico ha mostrado muy eficiente en términos de precisión, clasificando correctamente como relevantes los estudios recuperados. Sin embargo, no muestra un buen comportamiento al obviar la mayoría de los estudios relevantes, por lo que a pesar de no ofrecer resultados erróneos, deja muchos sin recuperar.

Observando los resultados para Consistencia Lógica, la mayoría de estudios están evaluados obteniendo un resultado que ronda el valor de 1/3, mientras que los más parecidos no consiguen obtener una diferencia muy significativa (Brooks, por ejemplo obtiene un 64% de consistencia). Esto se debe principalmente a dos problemas de representación e interpretación, que se dan entorno a los conceptos de las condiciones y la comparación de las localizaciones de los experimentos, y a que los sujetos participantes pueden parecerse mucho, aportando casi ese tercio de información.

En el caso de las condiciones, debido a la representación actual de BrainMap, estas apenas aportan información. Un ejemplo muy claro son las condiciones definidas en el estudio del experto. A pesar de ser diferentes en la realidad, utilizando la representación de los conceptos disponibles, resultan iguales. Para este caso sería necesario añadir los conceptos emocionales y relacionarlos con los objetos para obtener conceptos como *Emotional Picture* o *Pleasant Pictures*, con el consecuente trabajo ingeniería ontológica.

Esto implica también que habría que redefinir las métricas, para valorar el nivel de subsumisión dentro de una taxonomía[35], por ejemplo, ya que *Emotional Picture* sería subclase de *Picture*, concepto que podría seguir utilizándose en algunos contextos.

El conjunto de activaciones también distorsiona el resultado de las métricas, ya que los resultados de activación pueden no ser absolutos o utilizar atlas

distintos, como es el caso del estudio de TCA que nos ha servido de modelo. En este estudio el atlas empleado es el Harvard-Oxford Atlas, que tiene un nivel de granularidad muy bajo comparado con el Talairach (atlas base de BrainMap).

Una posible solución es emplear el conocimiento neuroanatómico del FMA para relativizar el matching de localizaciones entre experimentos. Es decir, sería posible aprovechar las relaciones mereológicas y espaciales del FMA para valorar la proximidad o solapamiento de dos activaciones.

En el caso de la métrica Grafo Semántico, los resultados muestran que la métrica resulta muy restrictiva, penalizando en exceso las diferencias que se puedan dar. Esto se debe principalmente a que, además de verse afectada por los problemas mencionados, disminuye de forma significativa ante diferencias en el número de arcos (relaciones) entre los nodos comparados. Este planteamiento es correcto para evitar que estudios con gran cantidad de experimentos, por ejemplo, reincidan en la evaluación de alguna característica (como un estímulo), resultando en una sobrevaloración. Sin embargo, en vistas de los resultados será necesario ajustar el sistema de penalización.

Los resultados de la métrica basada en la interpretación del grafo como un Vector Space Model se muestran desalentadores. Sin embargo, estos tienen sentido dado el contexto de su aplicación. Esta medida fue planteada para la alineación de conceptos entre ontologías explotando la información estructural y relacional del modelo. Para adaptar parte de la aproximación, se evalúan sólo las relaciones especificadas por el dominio y teniendo en cuenta como nodos a los individuos. El interpretar los individuos de la ontología como nodos del grafo provoca que se den grandes variaciones entre estudios.

En cuanto a la última métrica implementada, Entropía, los valores de cada estudio se corresponden con la cantidad de información que contiene cada uno, obteniendo que Van (4.186) es el que más información ofrece, lo cual es correcto teniendo en cuenta sus 14 experimentos. Sería interesante usar esta medida para cuantificar el grado de diferencia de cantidad de información entre dos estudios, y también puede ser empleada para implementar métricas relacionadas con conceptos como información mutua o distancias basadas en ella[36].

En vista de los resultados obtenidos resulta obvio que no deben valorarse todos los conceptos por igual, ya que no todos tienen la misma relevancia. Dependiendo del contexto, no se deben valorar con el mismo peso los datos sobre la edad del paciente y los estímulos de una condición dada. En un principio se pensó en implementar un sistema de pesos configurables por el

usuario, sin embargo se descartó ya que desvirtuaría el objetivo final de resultar en un proceso transparente para el experto, evitando molestos ajustes técnicos.

Teniendo en cuenta la disponibilidad de un conjunto de datos etiquetados por el experto, una línea futura interesante sería la implementación de un sistema de aprendizaje automático que ajuste la importancia de cada concepto de cara al cálculo de las métricas. Dada la alta dimensionalidad de las características que representarían tal escenario, la aproximación más adecuada sería un sistema basado en Support Vector Machine[37]. Con esto se obtendría un cálculo más ajustado de las métricas, valorando de forma más similar a lo que el experto considera relevante, y permitiría implementar un sistema de feedback de forma que los usuarios actualizaran el conjunto de datos etiquetado con valoraciones sobre los resultados ofrecidos por la herramienta.

En resumen, el planteamiento de estas métricas son un primer paso dentro de la aplicación de modelos de conocimiento como sistemas de búsqueda directa dentro del contexto neurocientífico, donde el propio estudio constituye la query a resolver, y ha demostrado la utilidad de utilizar ontologías como sistema de codificación enriquecido.

5. Conclusiones y trabajo futuro

A lo largo de este trabajo hemos visto como la aplicación e implantación de sistemas basados en ontologías son una tendencia creciente dentro del ámbito de las ciencias biológicas, especialmente la biomedicina. La comunidad científica cada vez es más colaborativa y estas soluciones ayudan a la interoperabilidad estandarizando la terminología y conocimiento disponible.

También se ha mostrado como BrainMap es un servicio interesante no solo para la asistencia al experto en la fase de documentación, sino que además el amplio contenido estructurado abre la puerta a la creación de aplicaciones de terceros que exploten la información de diversas formas. La aplicación aquí planteada abre una línea de explotación del conocimiento, codificado como ontologías, para sacar partido de esta información.

Hemos visto cómo se debe realizar el desarrollo de ontologías dentro de la comunidad, adaptando y maximizando la compatibilidad y la alineación a partir de las ontologías de dominio. Obteniendo como resultado una ex-

tensión de la ontología CogPO de forma que cubra la mayor cantidad de metadatos de BrainMap posible.

Con esta ontología, ha sido posible modelar estudios de neuroimagen funcional de forma simbólica y, a su vez, utilizar esta representación para realizar tareas de búsqueda cuantitativa. Estas búsquedas tienen en cuenta el contexto y el contenido de los experimentos, lo que permite que los resultados sean de mayor valor para el experto.

Para hacer posible este tipo de búsqueda ha sido necesario contar con un conjunto de medidas de comparación. Aprovechando la naturaleza de la representación lógica de las ontologías, la primera de ellas ha consistido en una medida de consistencia lógica entre el modelo de un estudio evaluado y las instancias del estudio a comparar. De la misma forma, aprovechando la representación implícita de la ontología como un grafo dirigido etiquetado, se han implementado medidas de similitud basadas en este tipo de representación, pero adaptadas al contexto de la representación de estudios. También, basándonos en el concepto de entropía, se ha diseñado una medida de cantidad de información que un estudio contiene, lo cuál también es útil para discriminar estudios en función de la cantidad de contenido.

Para ejemplificar y probar la viabilidad de estos métodos, se ha desarrollado una herramienta que permite a un usuario experto introducir de forma amigable los datos de sus estudio siguiendo el modelo definido en la ontología, pero abstrayendo al usuario de este hecho. A partir de los datos introducidos modela el estudio dentro de la ontología y realiza las comparaciones con el resto de estudios obtenidos de BrainMap. Los resultados se muestran en forma de lista junto a gráficas de barras con el resultado de la métrica empleada, de forma que al usuario le sea fácil identificar las diferencias.

Dada la imposibilidad de acceder de forma programática a BrainMap, el conjunto de datos experimentales ha sido muy limitado. Aun así, los resultados se muestran prometedores para dos de las métricas implementadas, siendo evidentes los cambios según los estudios son más parecidos unos a otros.

En vista de los resultados, queda demostrado la viabilidad de la aproximación planteada al problema de la comparación cuantitativa de estudios de BrainMap, siendo mejorables con ajustes razonables. Por tanto, los objetivos planteados de este trabajo:

1. Desarrollar una ontología que modele el esquema de codificación de metadatos de BrainMap.

2. Plantear un conjunto de medidas que habilite la comparación de estudios diferentes modelados bajo dicha ontología.
3. Desarrollar una herramienta que ejemplifique y permita:
 - a) Introducir un estudio de neuroimagen, descrito a partir de los metadatos de BrainMap.
 - b) Representar los estudios bajo la ontología y ejecutar las medidas planteadas.
 - c) Mostrar los resultados al usuario de forma gráfica y clara.

Se han visto cumplidos. Queda abierto el trabajo de seleccionar y adaptar nuevas medidas, así como el de mejorar la eficiencia e implementación de las ya contenidas en la herramienta.

En vistas al futuro, el uso de conocimiento de diferentes dominios, como la anatomía, abre la puerta a la implementación de numerosas opciones dentro de la herramienta, que vayan más allá del sistema de comparación.

La información anatómica relacionada por medio de conceptos del FMA puede ser explotada para realizar inferencias complejas adaptadas. Por ejemplo, dadas las activaciones de dos experimentos, inferir las proyecciones que dichos tejidos cerebrales extienden, ayudando dentro del contexto de una enfermedad. También en el proceso de comparación, al evaluar las activaciones de dos estudios, es posible realizar un “suavizado” a partir de las relaciones espaciales y mereológicas del FMA, en vez de un matching absoluto. Por ejemplo, un estudio reporta una zona concreta de la amígdala y otro estudio reporta un tejido próximo a la amígdala, por lo que podría plantear un proceso que evalúe cuantitativamente la proximidad.

Además de las relaciones espaciales y mereológicas, el FMA cuenta con información de conectividad entre entidades anatómicas cerebrales. Esta información puede emplearse para crear un grafo que represente la red de conectividad cerebral implícita, donde cada activación reportada se marque como un nodo de interés. Este modelo abre la puerta a la comparación de activaciones entre diferentes estudios partiendo del grafo de conectividad, y también a la posibilidad de realizar operaciones de minería de datos bajo dicha representación, como la identificación de biomarcadores tipo hubs, donde los nodos más incidentes serían identificados.

De la misma forma, es interesante la inclusión de relaciones con conceptos de cognición desde alguna de las ontologías disponibles. Cognitive Atlas ofrece

un modelo muy completo de este dominio, existiendo algún solapamiento con CogPO.

Una de las limitaciones encontradas durante el desarrollo de la ontología y la herramienta, es la necesidad de introducir el concepto de emociones, sobre todo de cara a estudios de neuroimagen centrados en enfermedades psiquiátricas. En este contexto, otra ontología interesante es la Emotion Ontology (EMO)[38], actualmente en desarrollo bajo colaboración de los responsables de CogPO.

Además de las mejoras en términos de representación, al contar con un conjunto de datos etiquetados, sería posible implementar un sistema de aprendizaje automático basado en Support Vector Machine de forma que los conceptos evaluados en las métricas se ajusten automáticamente de acuerdo a la experiencia.

En vista de todas estas mejoras, es necesario tener en cuenta las limitaciones que este tipo de representación del conocimiento conlleva, principalmente de coste computacional. Para abarcar todas ellas es necesario plantear la herramienta como un sistema distribuido, como podría ser un sistema basado en servicios web y aplicaciones de usuario. Con esta aproximación se desacoplan los diferentes sistemas de razonamiento (uno por cada ontología de dominio) de la aplicación final, de forma que el coste computacional estaría contenido para cada caso.

Referencias

- [1] Bug, W. J., et al. The Nifstd and Birnlex vocabularies: building comprehensive ontologies for neuroscience. *Neuroinformatics*, 2008
- [2] Larson, S. D., et al. Ontologies for neuroscience: what are they and what are they good for?. *Frontiers in Neuroscience*, 2009
- [3] Toga, A. W. Neuroimage databases: the Good, the Bad and the Ugly. *Nature Reviews Neuroscience*, 2002
- [4] AR Laird, JJ Lancaster, PT Fox. BrainMap: The Social Evolution of a Human Brain Mapping Database. *Neuroinformatics*, 2005
- [5] Peter T. Fox and Jack L. Lancaster. Mapping context and content: the BrainMap model. *Nature Reviews Neuroscience*, 2002
- [6] T. Berners-Lee et al. The semantic Web.
- [7] Ian Horrocks. Ontologies and the Semantic Web. *Communications of the ACM*, 2008
- [8] Jonathan B. L. Bard and Seung Y. Rhee. Ontologies in Biology: design, applications and future challenges.
- [9] JLV Mejino Jr, DL Rubin, JF Brinkley. FMA-RadLex: An application ontology of radiological anatomy derived from the foundational model of anatomy reference ontology. *AMIA Annual Symposium*, 2008
- [10] A Mechouche, C Golbreich, B Gibaud. Towards an hybrid system using an ontology enriched by rules for the semantic annotation of brain MRI images. *AMAIA 2007*
- [11] A Mechouche, X Morandi, C Golbreich. A hybrid system for the semantic annotation of sulco-gyral anatomy in mri images. *Medical Image Computing*, 2008
- [12] D Sonntag. Towards dialogue-based interactive semantic mediation in the medical domain. *The 7th International Semantic Web Conference*, 2008
- [13] Chuan-Jun Su, Chun Wei Peng. Multi-agent ontology-based Web 2.0 platform for medical rehabilitation. *Expert Systems with Applications* 2012

- [14] Daniel L. Rubin, Suzanna E. Lewis et al. National Center for Biomedical Ontology: Advancing Biomedicine through Structured Organization of Scientific Knowledge. OMICS 2006
- [15] Barry Smith, Michael Ashburner, et al. The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration. *Nature Biotechnology*, 2007.
- [16] Andrew D. Spear. *Ontology for the Twenty First Century: An Introduction with Recommendations*.
- [17] Jessica A. Turner & Angela R. Laird. *The Cognitive Paradigm Ontology: Design and Application*. *Neuroinform*, 2011
- [18] Mélanie Courtot, Frank Gibson et al. MIREOT: The minimum information to reference an external ontology term. *Applied Ontology* 2011
- [19] C. Rosse and J. L. V. Mejino. A reference ontology for biomedical informatics: the foundational model of anatomy. *J. of Biomedical Informatics*, 2003.
- [20] C Golbreich, S Zhang, O Bodenreider. Migrating the FMA from Protégé to OWL. *AMAIA Symposium*, 2008
- [21] Marianne Shaw et al. Generating Application Ontologies from Reference Ontologies. *AMIA Symposium*, 2008
- [22] Jones K.S. and Willett P. *Readings in Information Retrieval*. Morgan Kaufmann, 1997
- [23] Hotho A., Maedche A., and Staab S. *Ontology based Text Document Clustering*. *KI*, 2002
- [24] Albert GOLDFAIN, Barry SMITH and Lindsay G. COWELL. Dispositions and the Infectious Disease Ontology. *Formal Ontology in Information Systems*, 2010
- [25] Jérôme Euzenat, and Petko Valtchev. Similarity-based ontology alignment in OWL-Lite. *ECAI*, 2004
- [26] Rubén Tous and Jaime Delgado. A Vector Space Model for Semantic Similarity Calculation and OWL Ontology Alignment. *Database and Expert Systems Applications*, 2006

- [27] Vincent D. Blonde et al. A Measure of Similarity between Graph Vertices: Applications to Synonym Extraction and Web Searching. Society for Industrial and Applied Mathematics, 2004
- [28] C. E. Shannon. A mathematical theory of communication. Bell System Technical Journal, 1948.
- [29] Jacques Calmet, Anusch Daemi. From entropy to ontology. Cybernetics and systems, 2004.
- [30] Anusch Daemi and Jacques Calmet. From Ontologies to Trust through Entropy. Proc. of the Int. Conf. on Advances in Intelligent Systems-Theory and Applications, 2004
- [31] Paul Doran et al. Evaluating Ontology Modules Using an Entropy Inspired Metric. Web Intelligence and Intelligent Agent Technology, 2008
- [32] W. Frakes and R. Baeza-Yates, editors. Information retrieval, data structures and algorithms. Prentice Hall, 1992.
- [33] Diana Maynard, Wim Peters and Yaoyong Li. Metrics for Evaluation of Ontology-based Information Extraction. Evaluation of Ontologies for the Web, 2006
- [34] C. van Rijsbergen. Information Retrieval. Butterworths, 1979.
- [35] Jay J. Jiang, David W. Conrath. Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy. Proceedings of International Conference Research on Computational Linguistics, 1997
- [36] Dekang Lin. An Information-Theoretic Definition of Similarity. Proceedings of the 15th international conference on Machine Learning, 1998
- [37] Cortes, Corinna; and Vapnik, Vladimir N. Support-Vector Networks. Machine Learning, 1995.
- [38] Janna Hastings, Werner Ceusters³, Barry Smith⁴, Kevin Mulligan. Dispositions and Processes in the Emotion Ontology. ICBO: International Conference on Biomedical Ontology, 2011

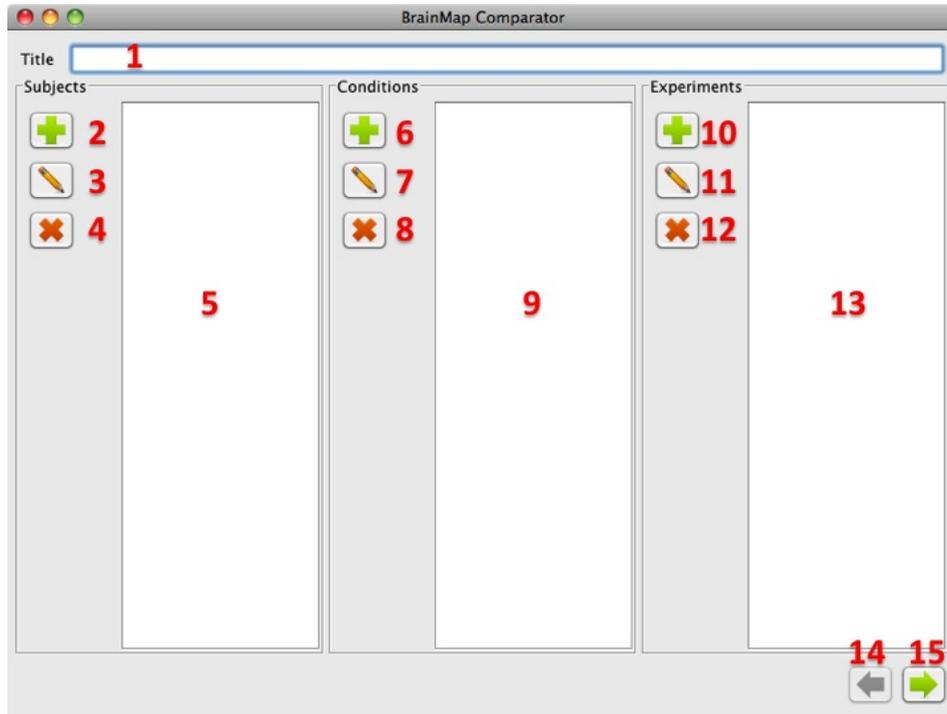


Figura 30: Ventana principal desglosada

6. Anexos

6.1. Manual de usuario de la aplicación

BrainMap comparator es la aplicación diseñada para realizar el proceso de comparación de un estudio con el conjunto de estudios definidos en la base de datos BrainMap.

Con este fin, la aplicación está pensada para que el proceso de introducir un estudio a evaluar sea sencillo e intuitivo.

6.1.1. Ventana principal

El punto de partida de la aplicación es la ventana principal, donde se encuentran disponibles las opciones necesarias para el modelado de un estudio. Estas opciones se desglosan en la figura 30.

1. Campo de nombre de estudio.
2. Añadir grupo de sujetos.
3. Editar grupo de sujetos.
4. Eliminar grupo de sujetos.
5. Lista de sujetos introducidos.
6. Añadir condición.
7. Editar condición.
8. Eliminar Condición.
9. Lista de condiciones introducidas.
10. Añadir experimento.
11. Editar experimento.
12. Eliminar experimento.
13. Lista de experimentos introducidos.
14. Botón volver (activado solo en ventana de resultados).
15. Botón siguiente (calcular).

6.1.2. Introducción y comparación de un estudio

El proceso de evaluar un estudio es el siguiente:

1. Introducir nombre del estudio.
2. Introducir los diferentes grupos de sujetos que participan en los experimentos (apartado 7.1.3).
3. Introducir las condiciones empleadas en los paradigmas de los experimentos (apartado 7.1.4).
4. Introducir los experimentos del estudio (apartado 7.1.5).
5. Pulsar siguiente.

6.1.3. Introducción de un grupo de sujetos

Tras pulsar el botón de añadir un grupo de sujetos  (ventana principal.2) aparece el diálogo de introducción de sujetos. Los campos a introducir (todos obligatorios) son los siguientes:

1. Nombre del grupo.
2. El total de personas pertenecientes al grupo.
3. Género de los sujetos. Las opciones son:
 - a) Femenino.
 - b) Masculino.
 - c) Mixto.
4. Edad mínima del grupo.
5. Edad máxima del grupo.
6. Edad media obtenida del grupo.

Una vez introducidos todos los datos, al pulsar OK se añade el nuevo grupo a la lista de grupos introducidos (ventana principal.5).

6.1.4. Introducción de condiciones

Tras pulsar el botón de añadir condición  (ventana principal.6) aparece el diálogo de introducción de condiciones. Los campos que definen la condición se introducen seleccionando de la lista correspondiente (obtenida de los conceptos de la ontología) el ítem deseado y pulsando >>. En el caso de que se haya introducido uno por error es posible eliminarlo de la selección por medio del botón <<.

Los campos a introducir son los siguientes:

1. Nombre de la condición.
2. Estímulo de la condición.
3. Respuesta esperada del sujeto.

The image shows a standard Mac OS-style dialog box titled "New Subject Group". It features a title bar with three window control buttons (red, yellow, green) on the left. The main area contains several input fields and dropdown menus:

- Name:** A text input field.
- Total:** A text input field.
- Gender:** A dropdown menu with "Female" selected.
- Diagnosis:** A dropdown menu with "Anorexia Nervosa" selected.
- Min Age:** A text input field.
- Max Age:** A text input field.
- Mean Age:** A text input field.

At the bottom of the dialog, there are two buttons: "OK" and "Cancel".

Figura 31: Ventana de introducción de sujetos

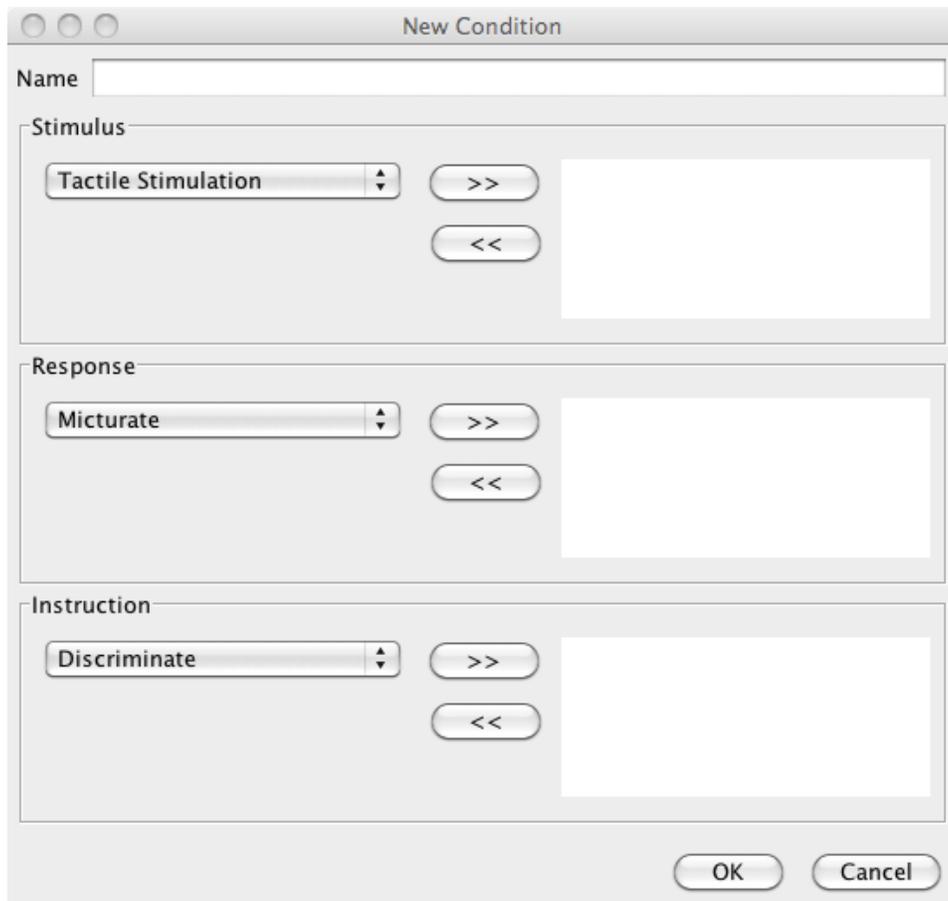


Figura 32: Ventana de introducción de condiciones

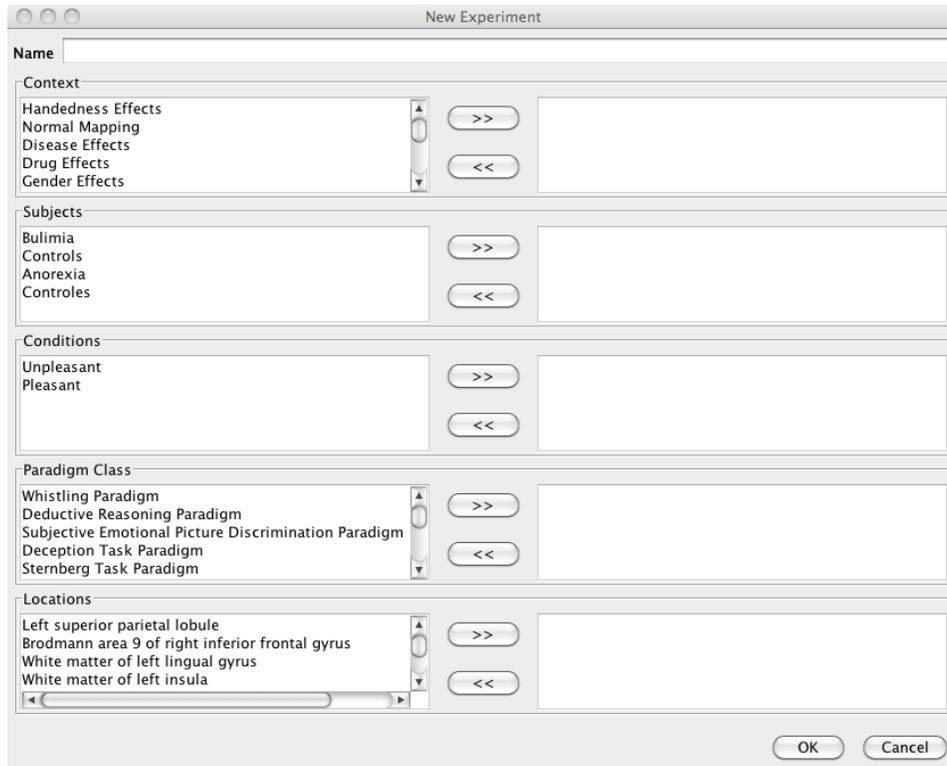


Figura 33: Ventana de introducción de experimentos

4. Instrucción dada al sujeto.

Una vez introducidos todos los datos, al pulsar OK se añade la nueva condición a la lista de condiciones introducidas (ventana principal.9).

6.1.5. Introducción de experimentos

Tras pulsar el botón de añadir experimento  (ventana principal.10) aparece el diálogo de introducción de experimentos. En este diálogo los campos provienen tanto de la ontología como de los grupos y condiciones ya introducidas. El conjunto de campos a introducir es:

1. Nombre del experimento.
2. Contexto del experimento.

3. Sujetos que participan en el experimento.
4. Condiciones del experimento.
5. Paradigma del experimento.
6. Localizaciones anatómicas de las activaciones observadas en el experimento.

Tras introducir los campos y pulsar OK se añade a la lista de experimentos (ventana principal.13).

6.1.6. Cálculo y resultados

Tras tener todos los datos del estudio introducidos, el siguiente paso es ejecutar el proceso de comparación. Para ello basta con pulsar el botón siguiente  (ventana principal.15).

Una vez finalizado este proceso, la aplicación muestra la ventana de resultados. Los resultados a su vez se muestran como gráficas, en el que aparece cada estudio comparado con el valor obtenido en cada métrica. Estas métricas son:

1. Consistencia Lógica. La consistencia lógica que muestra el estudio introducido con cada estudio de la base de datos.
2. Grafo semántico. La similitud estructural semántica que tiene el estudio introducido con cada estudio de la base de datos.
3. Vector Space Model. La similitud estructural modelada como un VSM que tiene el estudio introducido con cada estudio de la base de datos.
4. Entropía. La entropía que tiene cada estudio. Esta medida indica la cantidad de información que ofrece un estudio dado.

Es posible volver a la pantalla de edición de estudio y editar sus datos pulsando el botón volver  (ventana principal.14).

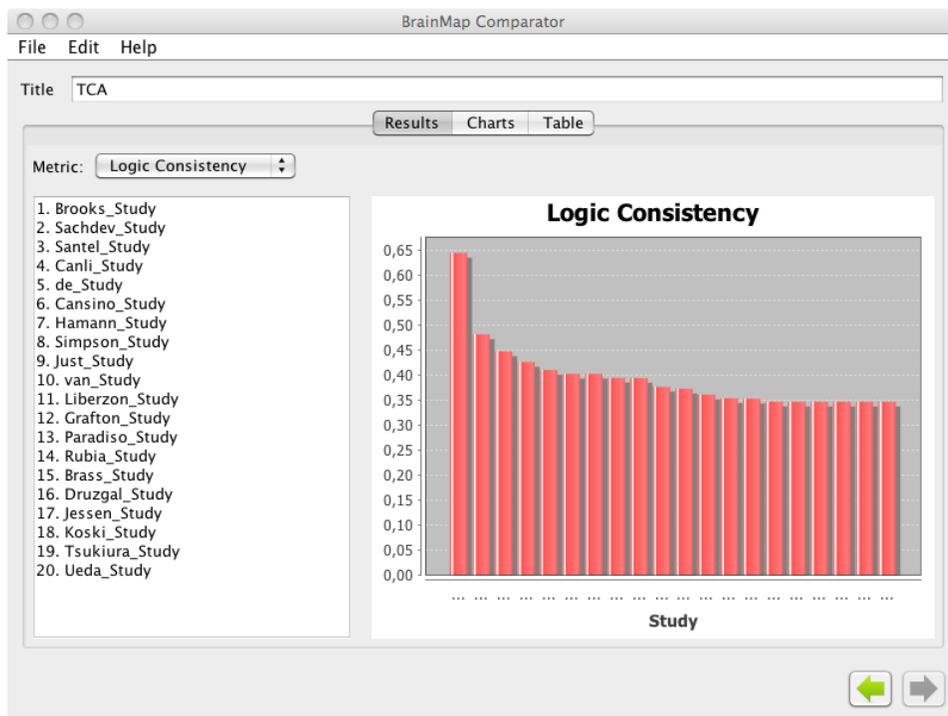


Figura 34: Ventana de resultados