Universidad Nacional de Educación a Distancia

Máster en Lenguajes y Sistemas Informáticos

---

# Estudio y propuesta para enriquecimiento de información utilizando fuentes Open Data: Una experiencia con videos educativos multidominio

---

**Febrero 2014**

**Autor: Jesús Río Álvarez**

**Directora: Ana García Serrano**

# ÍNDICE

# Lista de Figuras



# Lista de Tablas

# RESUMEN

Existe hoy en día un aumento constante de la demanda de información por la mayoría de las personas o usuarios. La ubicuidad tecnológica actual del usuario unido a los enormes avances en dispositivos móviles, computadoras y tecnologías web, genera nuevas necesidades de información. Por otra parte, el imparable proceso de apertura de datos (*Open Data*) bajo estándares de Web Semántica pone a disposición de los desarrolladores y usuarios una cantidad ingente de información y recursos, inimaginables hace algunos años. La llegada de nuevos estándares como HTML5 permite además la evolución en el mundo del desarrollo Web y abre un gran abanico de posibilidades para la reutilización de datos y la creación de interfaces de usuario.

La combinación de todos estos factores junto con las actuales técnicas de Recuperación de Información (IR) hace posible el desarrollo de nuevas aplicaciones que combinen fuentes de datos muy diversas para generar experiencias satisfactorias de usuario.

En esta tesis fin de máster (TFM) se analizan algunos trabajos relevantes relacionados con la anotación semántica de vídeos y las herramientas y tecnologías relacionadas con Linked Open Data(LOD), se estudia su grado de implantación y se muestran las posibilidades de combinación de dichas tecnologías para crear entornos enriquecidos de información en una propuesta tanto metodológica como práctica.

La propuesta se aplica en un escenario de educación a distancia proporcionado por la UNED. En esta vertiente práctica, se desarrollan un conjunto de funcionalidades necesarias para crear un entorno enriquecido de información: a partir de un catálogo de videos, se realizarán diferentes tareas de extracción de información en cada video y enriquecerle con contenidos relacionados extraídos de fuentes de datos Open Data. Finalmente se analizan los resultados obtenidos y se plantean futuras líneas de investigación. El corpus utilizado está formado por videos AVIP con clases de diferentes dominios y que han sido transcritos.

Nuevas combinaciones de datos pueden crear nuevos conocimientos e ideas, que pueden llevar a nuevos campos de aplicación. Es muy probable que en un futuro muy cercano haya desarrollos que tengan su origen en ideas inesperadas que surgen de la correcta combinación de diferentes conjuntos de datos abiertos.

# ABSTRACT

There is, nowadays, a growing information demand from most part of people and users. The current technological ubiquity and the advances in mobile devices, computers and web technologies, create new information needs. In addition, the unstoppable Open Data process following the Web Semantic principles provides a huge amount of information for developers and users. The arrival of standards like HTML5 allows the evolution of the web technologies and creates a broad range of possibilities for data reuse and the creation of new user interfaces.

The combination of these factors with current Information Retrieval (IR) techniques, makes possible the development of applications that create new user experiences connecting different data sources.

In this master's thesis we analyze state of the art works related to semantic annotation of videos and Linked Open Data (LOD) technologies, and we show the different ways in which these resources can be combined to create information enriched environments.

The approach described in this work has been implemented in a distant education scenario provided by the UNED. A video corpus from the UNED has been enriched with related Linked Open Data sources using extraction information techniques. Finally we show the process results and we propose future research.

# INTRODUCCIÓN

Este trabajo fin de master es una investigación aplicada sobre:

- La identificación de unidades de información a enriquecer
- El entrelazado de información proveniente de fuentes de datos Open Data (OD)
- La viabilidad de enriquecer información en un entorno real: videos educativos AVIP.

Se inicia a partir de las ideas mostradas por *Jaap Blom* en su artículo de 2012 [1] donde se describe una aplicación que reproduce videos de noticias y a la vez muestra información adicional relacionada con el contenido que va apareciendo en el video. El conjunto es una experiencia novedosa y sorprendente, una especie de *mashup* dinámico donde información interesante e inesperada proveniente de diversas fuentes se muestra sincronizadamente a la vez que se reproduce un video. Dicho artículo muestra las enormes posibilidades de la aplicación de tecnologías de datos enlazados (LOD) en combinación con la detección de entidades (NER) aunque la recuperación de información relacionada se hace exclusivamente a partir del servicio web de Freebase (en inglés) en lugar de recurrir a diferentes recursos, como en el trabajo de esta tesis de máster.

También es posible mejorar la forma en la que se genera la información enriquecida. En el trabajo original de *Jaap Blom* dicha información se codifica mediante un objeto JSON que se pasa como entrada a la librería JavaScript *Porpcorn.js*[1], en nuestro trabajo se ha utilizado un enfoque similar al utilizado en el proyecto *Chrooma+* [2] utilizando las nuevas características de video de HTML5: el elemento de pista [2](track) y el nuevo formato de subtítulos WebVTT[3]

En esta tesis fin de master se ha ido aún más allá y se ha pretendido desarrollar una metodología que describa como construir una aplicación de datos entrelazados. La construcción de una aplicación de datos enlazados es una tarea compleja ya que requiere del conocimiento y uso acertado de diferentes tecnologías. Para esta tesis se han recogido ideas del trabajo realizado en el grupo de investigación de M. Wald [3]. Este trabajo constituye un ejemplo representativo del estado actual de las técnicas LOD y destaca por la acertada combinación de tecnologías. Utiliza los videos disponibles de YouTube como origen del enriquecimiento, se utiliza JENA, la API JAVA para la construcción de aplicaciones de Web Semántica, usa una combinación de 10 extractores de entidades nombradas diferentes (NER), recupera información enlazada de la DBPedia y detalla de una forma muy específica tanto la arquitectura utilizada como el proceso de enriquecimiento llevado a cabo.

Su principal carencia es su escasa utilización de fuentes LOD, ya que se limita a recuperar propiedades estándar de las URIs de la DBPedia sin explorar más a fondo las posibilidades de varias fuentes de datos disponibles, como se realiza en este trabajo.

La utilización de la DBPedia en las aplicaciones de datos entrelazados es recurrente, y dicho recurso se revela como una herramienta fundamental ya que actúa como *hub* o concentrador

[1] http://popcornjs.org
[2] http://www.html5rocks.com/es/tutorials/track/basics
[3] http://dev.w3.org/html5/webvtt

de la nube LOD. Es por ello que cualquier equipo de trabajo que afronte una tarea donde intervengan datos entrelazados debe conocer a fondo los detalles de construcción, diseño y utilización de la DBpedia [4]. Su descripción se complementa en este trabajo con la descripción de Yago [5], una ontología ligera y extensible de gran calidad y amplia cobertura que se integra con la DBPedia. La inclusión de categorías YAGO en la DBPedia ofrece nuevas posibilidades en el descubrimiento de información y la inferencia de nueva información a partir del conocimiento proporcionado por la ontología puede mejorar enormemente los procesos de enriquecimiento de datos. En este trabajo fin de master se ha utilizado la DBpedia tanto para identificar las distintas entidades detectadas como para recuperar información adicional sobre las mismas. Además es posible utilizar las categorías de Yago para aumentar la precisión de los procesos de enriquecimiento.

En [6] se utiliza la DBpedia como un vocabulario controlado para unificar diversos servicios de la BBC, y pone de manifiesto la importancia de identificar de manera inequívoca cada recurso. Como se verá, en esta tesis fin de master la identificación de las entidades de forma inequívoca se revelará como una cuestión fundamental en el proceso de enriquecimiento de datos, y se utilizará, con licencia para investigación, uno de los recursos para el castellano más eficientes y eficaces, el recurso Stilus (http://www.daedalus.es/en/products/stilus/).

El auge actual de las redes sociales ofrece grandes posibilidades con relación a la utilización de datos entrelazados. Servicios como *Facebook*, *Twitter*, *Flickr* o similares constituyen fuentes valiosas de información que pueden ser utilizadas por aplicaciones basadas en LOD. Aunque en esta tesis fin de master no se hace un tratamiento específico de datos provenientes de redes sociales, se ha incluido una descripción de al menos un trabajo interesante para reflejar adecuadamente el estado del arte de estas tecnologías. En [7] se combina la utilización de recursos LOD con contenido web generado por las redes sociales (en este caso Facebook) y muestra el gran potencial existente en estas fuentes de información.

Utilizando técnicas de Linked Data es posible recuperar información sobre diversas entidades mediante la realización de consultas SPARQL sobre diversos conjuntos de datos. Según se muestra en [8], es posible utilizar las convenciones que utiliza la Wikipedia para organizar su información de una forma muy provechosa para el enriquecimiento de unidades de información como son las entidades. Por ejemplo, las entidades de la Wikipedia pueden pertenecer a diversas categorías (especificadas mediante la propiedad *dcterms:subject*) y dichas categorías se presentan con el formato *"<clase semántica> (of|in) <información de localización>*". De esta forma, si la entidad Alabama pertenece a la categoría "States of the United States" es posible inferir que Alabama es un estado y que pertenece a los Estados Unidos. En [9] se utiliza el sistema de categorías de la Wikipedia y su estructura jerárquica para realizar alineamientos de ontologías.

Basándonos en dichas asunciones sobre los sistemas de categorías de la Wikipedia ha sido posible en este TFM recuperar información relacionada sobre ciertas entidades. Por ejemplo, dado un nombre de un país y sabiendo que en la Wikipedia existe la categoría "*Científicos_de_Pais*" se puede parametrizar consultas SPARQL que recuperen nombres de científicos de cualquier país detectado.

Por otra parte en [10] se destaca la importancia de consultar determinadas propiedades de una entidad dependiendo de su tipo. Aunque CAF-SIAL se centra en un problema muy delimitado (solo tiene en cuenta el tipo de entidad "Persona") su enfoque basado en diferentes procesos dependiendo del tipo de una entidad es muy acertado y así los procesos de enriquecimiento en este trabajo, se han estructurado según el tipo de las entidades detectadas. De esta forma los distintos procesos de enriquecimiento, consultarán diferentes propiedades de las entidades a partir del tipo asociado a una entidad con lo que se ha desacoplado la definición de los procesos de enriquecimiento del proceso de recuperación de información, dotando a la propuesta final de mayor flexibilidad.

Uno de los puntos clave de este trabajo es la correcta identificación de las entidades o desambiguación, es decir, se debe asignar de forma adecuada la URI correspondiente a cada entidad detectada. Sindice[1] [11] es un servicio online que se encarga de rastrear la Web Semántica e indexar los recursos encontrados en cada fuente de datos, para asignar una URI a una entidad de un texto en inglés. Esta asignación no es trivial, y se realiza en este estudio mediante una combinación de funcionalidades sobre diferentes recursos: Stilus NER, DBPedia Lookup Service y el mencionado Sindice (para el caso de información en inglés). Resulta que en este trabajo es posible obtener un identificador para la mayoría de las entidades reconocidas en un video, aunque esta asignación puede verse afectada por problemas de desambiguación de entidades.

Esta memoria está estructurada en torno a tres partes. La primera está dedicada a la revisión del estado del arte, la segunda a la descripción de la metodología y la investigación concreta realizada sobre el enriquecimiento de información multimedia utilizando fuentes Open Data y otros recursos, así como a la presentación del experimento realizado sobre el catálogo de videos educativos de la UNED (AVIP), actualmente transcritos, y la tercera parte está dedicada a los anexos con aspectos concretos del desarrollo realizado. Finalmente se incluyen unas conclusiones y una discusión sobre los posibles trabajos futuros.

[1] http://www.sindice.com

# PARTE I: ESTADO DEL ARTE

# Capítulo 1: Trabajos relacionados

Esta tesis fin de master se plantea en el ámbito del tratamiento de información multimedia, para desarrollar una metodología que permita la explotación de recursos y datos abiertos en la web. La viabilidad e interés de esta propuesta se confirma tras el estudio del artículo del 2012 [1] escrito por *Jaap Blom* un ingeniero de software del Departamento de I+D del Instituto Holandés para el Sonido y la Visión[1]: el organismo público holandés encargado de preservar y hacer disponible la herencia audiovisual del país. En dicho artículo se describe la creación de un sistema de enriquecimiento de videos a partir de Open Data y se aplica a 1500 videos sobre noticias de los años 20 a los 80.

El principal objetivo del trabajo de tesis fue identificar los problemas existentes para enriquecer información con LOD y desde el punto de vista aplicado, contextualizar los videos mientras son visionados proporcionando al usuario información interesante o inesperada sobre los temas de los que trata un video. Es un planteamiento tecnológico que presenta tanto una vertiente teórica como práctica, ya que es necesario obtener la información adecuada (en relación con un video), combinarla convenientemente y presentarla de forma interesante para los usuarios.

Para avanzar a partir del trabajo de [1], en el que (1) a partir de las transcripciones de los videos se eliminan las palabras que no aportan ningún significado (*stopwords*) y se ordenan las restantes por importancia, (2)se envían a un tesauro y a Freebase[2] que devuelve conceptos relacionados con los proporcionados, y (3) que a su vez se utilizan para hacer consultas a diversas fuentes de datos LOD, como Google Maps, la Wikipedia, la web del Museo de Ámsterdam y la del Rijksmuseum, se decidió hacer un experimento con el repositorio AVIP de videos educativos de la UNED y otras fuentes LOD más acordes. En [1] los resultados se muestran alineados temporalmente en el momento en que aparece en el video el concepto (utilizando HTML5 y la librería JavaScript Popcorn.js[3]), y aunque en este trabajo solo se ha desarrollado un sencillo prototipo para mostrar las funcionalidades implementadas, ya que el interfaz estaba fuera de los objetivos planteados, la alineación temporal también se consigue, para la presentación de resultados.

Se organiza la presentación del estado del arte en torno a las siguientes preguntas:

1. ¿En qué contexto enriquecer información?
2. ¿Cómo se combina y enriquece la información?
3. ¿Cómo se presenta el contenido enriquecido al usuario?
4. ¿Cómo gestionar la ambigüedad en procesos de enriquecimiento de información?
5. ¿Cómo evaluar la calidad del contenido enriquecido?
6. ¿Qué aporta el proyecto Linked Open Data?

[1] http://www.beeldengeluid.nl/en
[2] http://www.freebase.com
[3] http://popcornjs.org

## 1.1 ¿En qué contexto enriquecer información?

Es posible enriquecer prácticamente cualquier tipo de información. Durante los años 2012 y 2013 fundamentalmente, se desarrollaron trabajos de enriquecimiento de contenidos multimedia como el caso de videos provenientes de *YouTube* [3], o documentos web, como el esfuerzo realizado por la BBC [6] para integrar y enlazar el contenido web existente en sus portales públicos con la nube LOD.

Las redes sociales también son un terreno propicio para aplicar las técnicas relacionadas con LOD. La ingente cantidad de información que los usuarios de estas redes aportan es susceptible de ser enriquecida y enlazada para aportar nuevas funcionalidades. El proyecto *Chrooma+* [2] utiliza *tweets* de *Twitter* que contienen un *hashtag* concreto para realizar anotaciones de metadatos. Otros trabajos se centran en obtener información a partir de los mensajes existentes en *Facebook* apoyándose en tecnologías LOD, de esta forma, el equipo de Dong-Po Deng [7] ha sido capaz de recuperar observaciones ecológicas presentes en hilos de Facebook mediante técnicas de recuperación de información que utilizan principios de LOD y en [12] se enriquece información contenida en *tweets*.

Además es posible adaptar los resultados obtenidos de acuerdo a las preferencias de los usuarios: *RailGB* [13] muestra a un usuario con algún tipo de discapacidad las estaciones de metro más cercanas a su posición que mejor se adaptan a sus capacidades de movilidad.

A continuación se detallan aquellos aspectos más relevantes para este TFM, de los trabajos anteriores.

### 1.1.1 Aplicaciones con Synote

Para el enriquecimiento de videos a partir de fuentes LOD, resulta de especial interés el trabajo del grupo del investigador Mike Wald que ha desarrollado Synote[1] [14] [15] [16] [17] [18], un sofisticado sistema de anotación multimedia. Un ejemplo de un proceso de enriquecimiento basado en Synote a partir de videos provenientes de *YouTube* combinado con la detección de entidades es el trabajo de Li et al [3].

La principal aportación de [3] es ofrecer una estrategia combinada para extraer entidades nombradas de los vídeos, el alineamiento temporal de dichas entidades y la creación de un interfaz de usuario para navegar entre los vídeos enriquecidos mostrando la información proveniente de las fuentes LOD. Para mostrar las aplicaciones prácticas los autores realizaron una aplicación mediante la integración y extensión de dos sistemas, Synote y NERD:

- Synote[2] es un sistema de anotación multimedia basado en web que permite asociar notas a ciertas partes de un recurso. Es una aplicación gratuita y soporta la creación de notas, enlaces, etiquetas, imágenes y subtítulos asociados a fragmentos multimedia.
- NERD es un entorno que unifica la aplicación de 10 extractores de entidades nombradas (NER) disponibles en la red: AlchemyAPI, DBpediaSpotlight, Evri, Extractiv, Lupedia,

[1] http://www.synote.org/synote
[2] http://www.synote.org

OpenCalais, Saplo, Wikimeta, Yahoo! Content Extraction and Zemanta. Proporciona además una ontología que incluye un conjunto de axiomas para alinear las taxonomías de estas herramientas.

El proceso de enriquecimiento de datos se realiza de la siguiente forma (ver Ilustración 1):

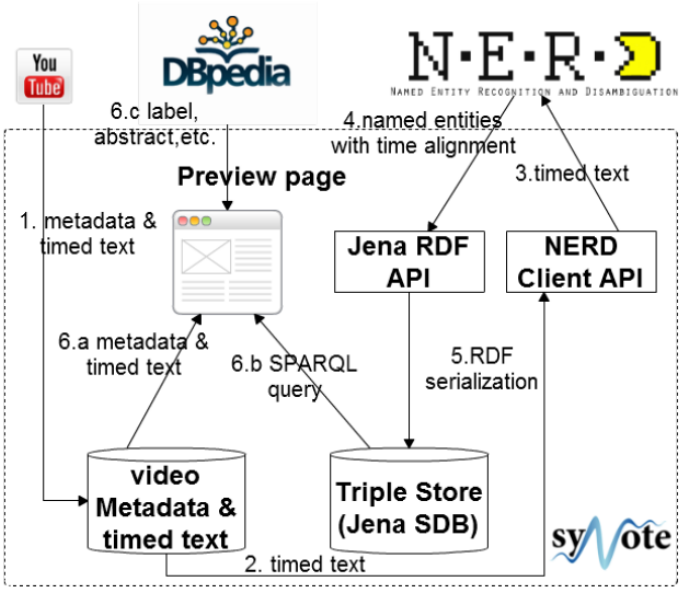



Ilustración 1: Integración de las arquitecturas de Synote y NERD

1) Se comienza con la creación de una grabación en Synote a partir de un video de YouTube. A continuación el sistema recupera los metadatos y transcripciones (si están disponibles) mediante la API de YouTube.
2) El texto con la alineación temporal es enviado a la API cliente de NERD.
3) Posteriormente se almacena el texto en el servidor NERD. Se realiza entonces la detección de entidades a partir del contenido completo de las transcripciones usando de forma combinada los 10 sistemas NER disponibles en NERD.
4) NERD devolverá una lista de entidades nombradas, junto con su tipo, una URI para su desambiguación y una referencia temporal que indica el punto donde dichas entidades aparecen en el fichero de transcripciones.
5) Al recibir la respuesta de NERD, Synote construye las URIs de los fragmentos multimedia y utiliza la API RDF de Jena (Jena es la API de Java para construir aplicaciones de Web Semántica y Linked Data) para serializar las anotaciones de los fragmentos en tripletes RDF utilizando los vocabularios NERD, Ontology for Media Resource, Open Annotation y StringOntology. Finalmente el interfaz de usuario muestra de forma integrada las entidades nombradas, los fragmentos multimedia, el video de YouTube y sus subtítulos.
6) Las entidades nombradas y los metadatos relacionados extraídos de los subtítulos se obtienen mediante consultas SPARQL (6a, 6b). Si una entidad se ha desambiguado con una URI de la DBPedia (6c) se realiza una consulta SPARQL para recuperar información sobre dicha entidad (propiedades label, abstract y depiction de la DBPedia).

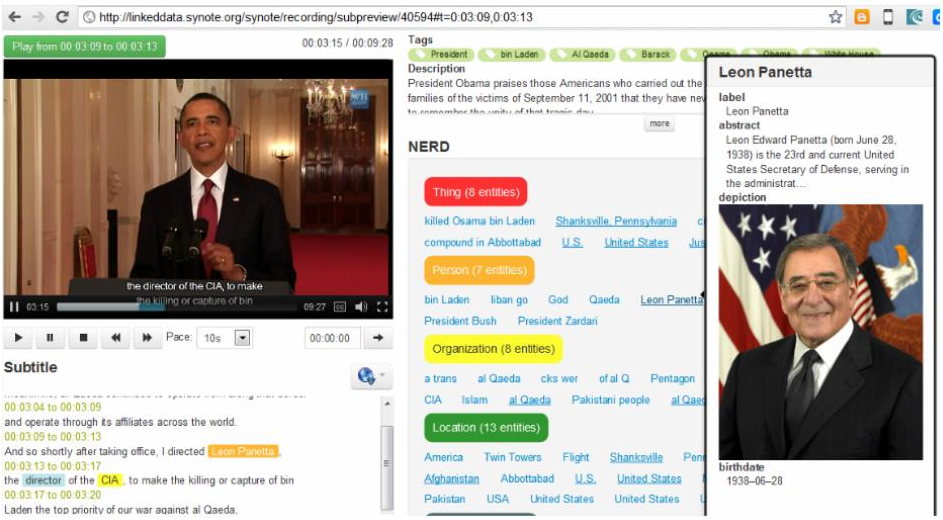



**Ilustración 2: Interfaz de usuario de Synote enriquecido con NERD Y DBPedia**

En la imagen anterior se puede ver la aplicación en funcionamiento. La columna de la derecha muestra la entidades encontradas agrupadas según las categorías de NERD (Cosa, Persona, Organización, Localización, Fecha,…). En la parte izquierda se puede ver el video de YouTube con sus subtítulos interactivos. Las entidades encontradas se muestran en un color determinado por su categoría. Si una entidad ha sido desambiguada con una URI de la DBPedia aparece subrayada y cuando se selecciona dicha entidad se muestra una ventana emergente con información adicional proveniente de la DBPedia. Para las entidades de tipo Persona se muestra su fecha de nacimiento, y para las Localizaciones su latitud y longitud.

Una cuestión no resuelta en esta área, es la evaluación de los procesos de anotación semántica basada en LOD (enriquecimiento). El hecho de que este tipo de sistemas se basen en el descubrimiento de nueva información hace que su evaluación sea muy compleja hasta el momento, debido a la carencia de un "Gold Estándar" o conjunto de resultados de prueba que permita comprobar que la nueva información recuperada es correcta o no con respecto al gold standard.

Los autores realizan un estudio que consiste en la obtención de 60 videos provenientes de 3 canales diferentes de YouTube (Gente y Blogs, Deporte, Ciencia, Tecnología) y del cálculo de ciertas variables: número de documentos de cada categoría, número de entidades encontradas en cada video, número de palabras en cada video, etc. Dicha evaluación refleja que los videos muestran un comportamiento diferente en términos de entidades reconocidas dependiendo de su género, de esta forma, los videos de "Ciencia y Tecnología" tienden a incluir entidades de tipo "Persona" y "Organización" mientras que los de "Deporte" mencionan más "Localizaciones", "Fechas" y "Cantidades". El grupo de "Gente y Blogs" tiende a presentar menos información útil.

Finalmente indicar que este trabajo constituye un ejemplo representativo del estado actual de las tecnologías de LOD y destaca por la acertada combinación de tecnologías y sistemas ya existentes (YouTube, Jena, Synote, NERD). Quizás su enfoque se queda corto en la utilización de fuentes LOD ya que se limita a recuperar propiedades estándar de las URIs de la DBPedia

sin explorar más a fondo las posibilidades de otras fuentes de datos disponibles. Sin embargo, en este TFM sí se estudiará la agregación de diferentes fuentes de datos abiertos.

### 1.1.2 Agregación de información basada en Wikipedia

Otro ejemplo interesante de enriquecimiento de datos a partir de información web, es el realizado por la BBC en sus portales públicos [6] donde se utilizan tecnologías de Linked Data en combinación con DBPedia y MusicBrainz.

La BBC publica grandes cantidades de contenido web: tanto texto como audio y video. Este contenido se agrupaba tradicionalmente en forma de microsites independientes con un dominio muy específico (alimentación, música, jardinería,…) y no se realizaba una integración con el resto de la web de la BBC[1] ni por supuesto con el resto de la Web. Como resultado, se proporcionaba información en el contexto de un dominio muy especializado pero no era posible mostrar información que englobara diferentes dominios, por ejemplo, es posible encontrar quién presenta el programa "Top Gear" pero no es posible saber que más programas ha presentado esa persona.

Otro problema que aparece durante la agregación de fuentes de datos es que al manejar diferentes dominios temáticos (alimentación, música, noticias,…) mantenidos por diferentes equipos, es difícil el entrelazado entre diferentes servicios y sitios web de programas. La existencia de diferentes dominios plantea problemas adicionales de desambiguación, por ejemplo el hecho de que Madonna sea una artista (en MusicBrainz) y una actriz o persona (en Wikipedia).

Para afrontar todos estos problemas la BBC, la Universidad de Berlin y Rattle Research han trabajado en proporcionar un vocabulario controlado basado en la DBPedia que permita añadir palabras clave a páginas web existentes. Grandes organizaciones como la BBC han invertido grandes recursos a lo largo de los años en crear su contenido web, por lo que el rediseño completo de los mismos es una tarea inasumible. Para conseguir el entrelazado dichos contenidos mediante una transición suave y reduciendo el impacto en los sistemas existentes se optó por aplicar las tecnologías de Linked Data. Los objetivos planteados eran los siguientes:

- Desarrollar un nuevo servicio que unifique el estilo de todas las emisoras de radio, canales de televisión y programas (bbc.co.uk/programmes) y asegure que los usuarios y desarrolladores de terceros puedan acceder desde los datos de la BBC a otros conjuntos de datos de la nube Linked Data siguiendo enlaces relacionados.
- Desarrollar una oferta musical (bbc.co.uk/music/beta) construida a partir de estándares web abiertos e integrada con el servicio de programas.
- Modernizar los elementos de navegación simples (Conjuntos de palabras clave) de páginas web antiguas para soportar una navegación semántica y contextual.
- Proporcionar un conjunto común de identificadores web para mejorar la clasificación de todo el contenido online de la BBC y crear equivalencias entre múltiples vocabularios.

[1] http://www.bbc.co.uk

Desarrollos del servicio de Programas de la BBC (BBC Programmes[1]) proporcionaron una URI por cada programa emitido en la BBC, permitiendo que otros equipos de la BBC incorporasen dichas URIs en sus sitios web de canales de televisión y radio. Por otra parte, el servicio de música (BBC Music) generó una URI basada en identificadores de MusicBrainz por cada artista y cada canción que la BBC reproduce. Sin embargo un artista existe en varios dominios (un cantante puede ser además actor) por lo que era necesario disponer de un mecanismo que creara equivalencias entre varios identificadores de dominios diferentes.

Tanto "BBC Programmes" como "BBC Music" proporcionaban por tanto identificadores web persistentes diseñados bajo los principios de Web Semántica.

El valor de los sitios web no está en los metadatos implícitos que proporciona el modelo de dominio en el que se incluya sino en la forma en la que éste modelo del dominio se entrelaza con otros dominios, es decir: los enlaces son más importantes que los propios nodos. De esta forma se pueden diseñar nuevas experiencias de usuario a través de las diferentes web, aunque para ello aún será necesario utilizar el contenido y los sistemas existentes previamente (contenido legacy).

El contenido existente en la BBC fue creado con un sistema de auto-categorización llamado CIS. Dicho sistema dispone de una jerarquía de términos con cuatro clases de alto nivel: gente, asuntos, marcas y lugares. Dicho sistema fue utilizado originalmente para anotar noticas regionales en un sistema de gestión de contenido por lo que su vocabulario estaba centrado en contenido regional británico (bandas locales, gente, eventos, localizaciones…). Este sistema fue utilizado para anotar los programas de la BBC de forma automática basándose en su descripción textual. Por ejemplo en el programa cuya descripción era "una mirada hacia los Juegos Olímpicos de Pekin: un avance de las esperanzas del boxeo Británico" el término "Pekín" se convertiría en el enlace entre este programa y otros programas sobre Pekín, además de enlazar noticias de la BBC sobre Pekín. Este enfoque permitía el entrelazado de diferentes servicios de la BBC manteniendo su desarrollo completamente independiente.

Sin embargo, este modelo presentaba varias desventajas: un sistema de categorías es difícil de mantener y es complicado cubrir cada una de las entidades que puedan ser de interés. Además en CIS no existen relaciones entre términos, por lo que no sería posible relacionar por ejemplo "Pekín" con "Los Juegos Olímpicos de Pekín". Para ofrecer una experiencia de usuario satisfactoria es necesario disponer de relaciones ricas entre diferentes conceptos del vocabulario. Además los términos de CIS solo disponen de un identificador interno por lo que no es posible enlazarlos a recursos fuera de la BBC.

Teniendo en cuenta estas cuestiones en [6] presentan cómo utilizaron un conjunto común de identificadores para la BBC. El proyecto DBPedia que ofrece información extraída de la Wikipedia a la Web Semántica y que se ha convertido en el estándar de-facto para entrelazar recursos de LOD, era la elección obvia para obtener identificadores universales y a la vez entrelazarse con la Web de Datos. Además la DBPedia no sólo ofrece URIs universales para un gran rango de conceptos sino además datos estructurados sobre esos conceptos y sus relaciones, que pueden ser utilizados por algoritmos de enlazado. De esta forma la DBPedia se

[1] http:// bbc.co.uk/programme

convierte en el vocabulario controlado para conectar los distintos dominios de la BBC (Música, Noticias, Programas)

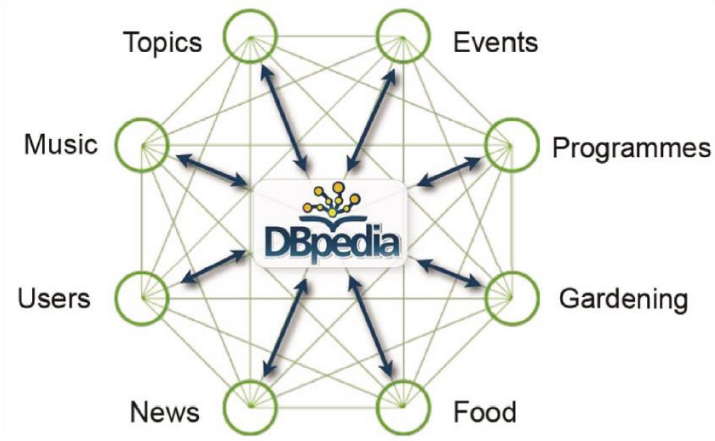



Ilustración 3: Enlazado de dominios de la BBC

Para lograr el enlazado de conceptos entre diferentes dominios desarrollaron un sistema que enlazara de forma automática los conceptos CIS con los de la DBPedia. CIS contiene 150.000 términos aproximadamente que cubren cuatro dominios distintos: marcas, localizaciones, gente y asuntos. Cada dominio tiene su propia jerarquía de términos y las categorías se expresan en SKOS. Para el entrelazado de términos sólo era posible utilizar la propia taxonomía de CIS ya que por varias razones no era posible acceder a documentos etiquetados con términos CIS. El objetivo entonces consistía en encontrar el emparejamiento más probable basándose en la búsqueda de términos de CIS en la DBPedia y realizar tareas de desambiguación sobre esos emparejamientos usando información de clasificación.

Este problema es conocido en el área de alineamiento de ontologías, sin embargo, era necesaria una solución adaptada a las especiales características del problema, teniendo en cuenta la limitada información de los términos CIS disponibles. El enfoque adoptado consistió en un agrupamiento por similitud basado en el contexto de los conceptos. El término “apple” es muy ambiguo, pero dado el contexto de términos “microsoft” y “google”, el significado de apple refiriéndose a la empresa Apple se hace más claro. El algoritmo desarrollado en este trabajo consta de dos partes: la búsqueda de etiquetas en la DBPedia y desambiguación basada en el contexto.

La DBPedia contiene 2,5 millones de conceptos de forma que la mayoría de las búsquedas de términos de CIS en la DBPedia devuelven varios conceptos candidatos. Por ejemplo, la búsqueda de “Shakespeare” devuelve 50 resultados. Para encontrar el término más probable el sistema utiliza una búsqueda ponderada de etiquetas usando el número de enlaces entre artículos de la Wikipedia como indicador del peso, ya que este valor da una medida de la relevancia del artículo: el artículo sobre “William Shakespeare” recibe 6000 enlaces mientras que el artículo sobre “Nicholas Shakespeare” solo 18.

Para desambiguar los posibles términos candidatos se identifican contextos de similitud de términos CIS creando clusters o grupos y encontrando contextos adecuados en la DBPedia. Se utiliza el sistema de categorías de CIS y los textos entre paréntesis para generar los clusters,

por ejemplo para el concepto de CIS “María (1985 ComediaDeSituación)” se crean los clusters “televisión” (por su categoría en CIS), “1985” y “ComediaDeSituación” (texto entre paréntesis). El algoritmo crea estos clusters para todos los términos CIS e identifica los emparejamientos entre categorías DBPedia, clases y plantillas para cada cluster basándose en múltiples emparejamientos posibles. Los contextos identificados se utilizan para desambiguar parejas por cada concepto CIS. En el ejemplo anterior sobre series de televisión el algoritmo es capaz de rechazar el resultado con un mayor ranking: ”María (Santa Madre)” para el término de búsqueda “María” porque su clase DBPedia rechaza “Algo sobre María” y “El Show de María Tylor Moore” y acepta el resultado “María (1985_TV_Series)” basándose en la categoría de la DBPedia “Series de televisión americanas de 1980”.

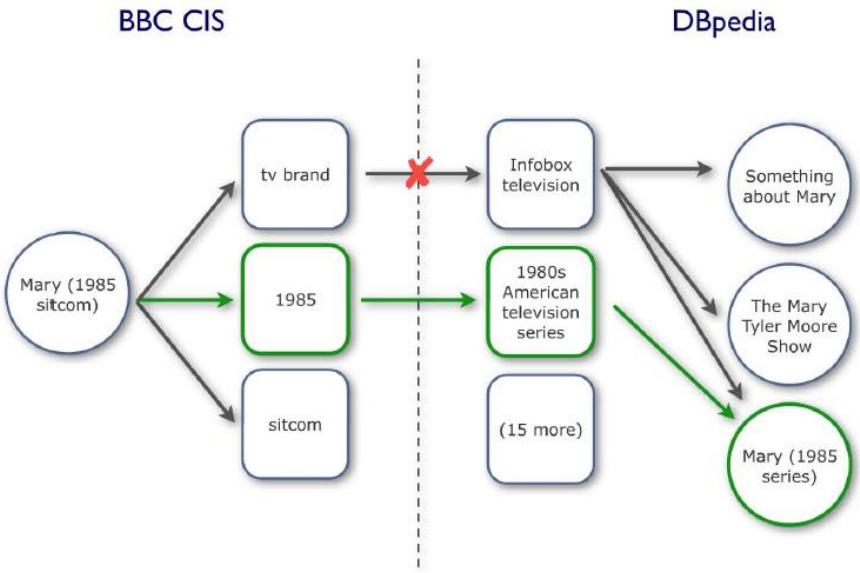



Ilustración 4: Desambiguación basada en el contexto

Una vez que se ha realizado el enlazado de conceptos entre CIS y la DBPedia es posible procesar los documentos existentes en la BBC tales como artículos de noticias o websites editoriales para enriquecer la información que proporcionan.

El sistema desarrollado, Muddy Boots, realiza un análisis sintáctico de los documentos de la BBC y usa un sistema de reconocimiento de entidades nombradas (NER) para obtener las entidades más importantes de cada documento. Estas entidades que constan simplemente de texto y no incorporan ninguna información semántica o de clasificación se emparejan a recursos de la DBPedia mediante un algoritmo de lógica difusa que compara títulos de recursos de la DBPedia con los de las entidades extraídas. Para cada término se genera una lista de recursos DBPedia y se evalúa cada uno mediante una desambiguación contextual seleccionando el que obtenga la mejor evaluación. Esto permite a Muddy Boots hacer un mapeo entre las entidades de los documentos y sus entidades asociadas de la DBPedia. El paso final es identificar si los recursos de la DBPedia corresponden a la categoría PERSONA o COMPAÑÍA, para ello se examinan los predicados de cada recurso, por ejemplo, si un recurso incorpora el predicado “fecha de nacimiento” es muy probable que se trate de una PERSONA.

El interfaz de usuario de cualquier herramienta de anotación es un aspecto crítico para lograr la creación de metadatos de calidad. Además, dicha herramienta debe incorporar sugerencias

automáticas relevantes para hacer el proceso de anotación lo más ágil posible. Estas fueron los principios que guiaron el desarrollo de la herramienta de enlazado de contenido de la BBC.

Se utilizó el sistema *Muddy Boots* como fuente de sugerencias automáticas basadas en el conjunto de datos de la DBPedia. La herramienta incorpora un buscador con autocompletado de palabras clave basadas en recursos de la DBPedia. La visualización de resúmenes de las páginas de DBPedia facilita a los editores de la BBC la elección de conceptos sugeridos. Los términos seleccionados se añaden a la lista de conceptos de la URL editada.

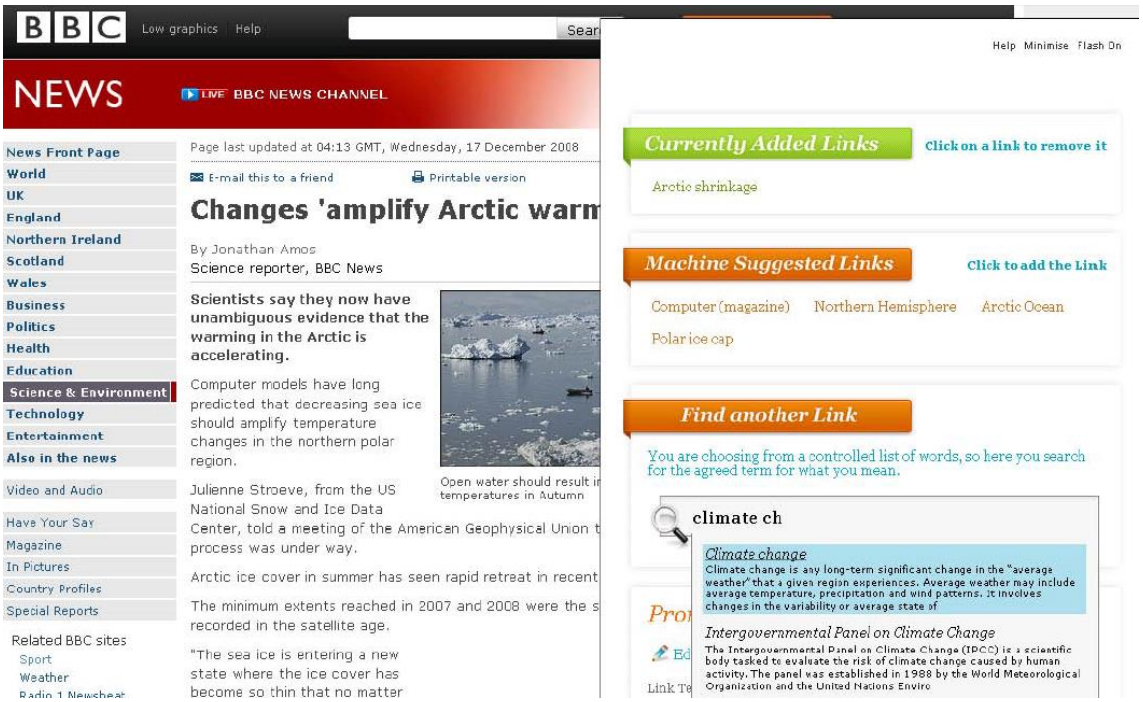



**Ilustración 5: Herramienta de Enlazado de Contenido en BBC News**

La carencia de datos estructurados enlazados hacía muy difícil la presentación de un website de la BBC coherente. La creación de una navegación a través de los diferentes dominios requiere del entrelazado de un modelo de datos basado en identificadores comunes. Obviamente esto es más difícil de conseguir cuando se trabaja sobre contenido desestructurado.

Una solución a este problema fue la creación de páginas de agregación de contenido estructurado y desestructurado. Estas páginas juntan el contenido modelado de "*BBC Programmes*" (que incluye identificadores CIS mapeados a la DBPedia) y el contenido desestructurado de los artículos de BBC News. Estas páginas de agregación o páginas de términos son muy populares en webs como la del New York Times o CNN por su habilidad para concentrar los motores de búsqueda sobre una determinada palabra clave. Para la BBC este tipo de páginas sirven además como nodos de navegación para facilitar el paso entre diferentes dominios.

Para crear las páginas de agregación utilizaron el vocabulario de la DBPedia que presenta numerosas ventajas:

- Interoperabilidad con otros datos de dominio específico utilizados por la BBC como *MusicBrainz*.

- Es más fácil de mantener que los vocabularios internos existentes.

- DBPedia proporciona datos adicionales: descripciones cortas de términos, geo-localización y datos temporales útiles para enriquecer las páginas de agregación.

- DBPedia ofrece relaciones jerárquicas y asociativas de gran calidad.

- Los textos descriptivos de la DBPedia se pueden utilizar como material de entrenamiento para sistemas de categorización automática.

Al utilizar la DBPedia como vocabulario es posible unir los datos estructurados que usan identificadores de Linked Open Data (como *MusicBrainz*) con el contenido no estructurado usando sistemas de auto-categorización basados en la DBPedia.

Las páginas de agregación permiten a un usuario navegar desde una determinada persona, lugar o asunto a cualquier otra área de la BBC, pero una vez allí existen pocos enlaces a otros contenidos relacionados. Para mejorar esta situación se crea una "bolsa de navegación" de forma dinámica basándose en la extracción de entidades (realizada por *Muddy Boots*) del contenido de la página en la que el usuario se encuentra. Los conceptos detectados se comparan con páginas de agregación existentes y se enlazan con ellas. De esta forma se pueden enlazar páginas con contenido relacionado.

### 1.1.3 Agregación de información desde Redes Sociales

Con el auge de las redes sociales han surgido grandes posibilidades de enriquecimiento de datos a partir del contenido generado por los usuarios de estas redes. Un ejemplo de ello es el trabajo de 2013 [7], donde se detecta información ecológica en hilos de Facebook y se publican como Linked Data.

En [7], se considera a los usuarios como sensores que informan de lo que sucede a su alrededor. El auge de las redes sociales ofrece una gran oportunidad de involucrar a un gran número de voluntarios a que contribuyan con sus datos diversos propósitos científicos. Por ejemplo, muchas personas interesadas en pájaros suelen registrar en foros online que tipo de aves observan y dónde. Sin embargo, el contenido generado por usuarios (CGU) suele tener una forma desestructurada, como por ejemplo texto y fotos y su semántica suele ser muy pobre y ambigua. Aunque el CGU es considerado un gran recurso potencial para proyectos científicos, el procesar ese contenido no estructurado es una tarea compleja. Además, para recopilar CGU de una forma eficiente y sistemática se hacen necesarias herramientas que faciliten la semántica de CGU, conectándola con fuentes LOD. Si entidades nombradas de CGU se pudieran identificar y conectar a entradas de LOD, la semántica de las entidades nombradas se desambiguaría de forma que el CGU sería más fácil de procesar.

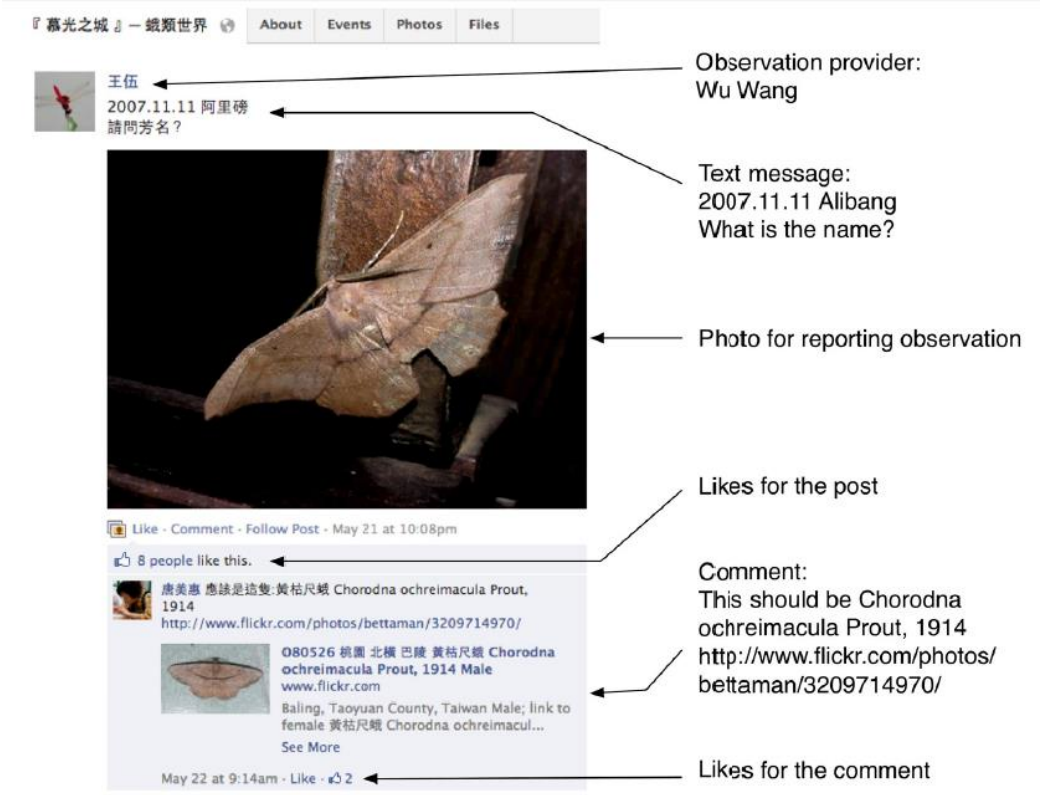



**Ilustración 6: Observación ecológica publicada en el grupo de Facebook "EnjoyMoth"**

En [7] se describe el desarrollo de un recurso Linked Open Data (LOD) para Nombres Geográficos Taiwaneses (LOD TGN). El objetivo es enriquecer la semántica del contenido generado por usuarios mediante la utilización de recursos LOD, de forma que por ejemplo el contenido de los textos introducidos en Facebook puedan ser más reutilizables. Se puede considerar este estudio como el desarrollo de un método de anotación semántica geoespacial para textos de Facebook, mediante el uso de recursos LOD.

Las coordenadas geométricas de un lugar no son de mucha utilidad para la mayoría de las personas, sino que tendemos a utilizar el nombre del lugar. La Ilustración 6 muestra un post de Facebook informando de la observación de una mariposa (*ChorodnaOchreimaculaProut*) en una fecha y lugar concreto (*Alibang*).

Para catalogar el hilo de Facebook como una observación ecológica es necesaria una herramienta que reconozca nombres de especies y nombres de lugares. Para ello se ha construido una herramienta de procesamiento de lenguaje natural para extraer información de observaciones ecológicas de noticias de Facebook. Las observaciones reconocidas se publican según los principios de LOD utilizando una ontología que organiza de forma semántica los datos extraídos (ver Ilustración 7) y se ha publicado dicha información en la Web utilizando un servidor D2R.

Un servidor D2R es una herramienta para publicar bases de datos relacionales en la Web Semántica que forma parte de la plataforma D2RQ (http://d2rq.org). Permite que navegadores RDF y HTML puedan navegar por el contenido de la base de datos y admite consultas a la base de datos utilizando SPARQL.

El recurso de Datos Abiertos Enlazados de Ecología (LODE) es un conjunto de datos validado como proyecto LOD. LODE integra 4 bases de datos distribuidas previamente:

- *Firedb*: Ecología de fuegos del Instituto de Investigación de Bosques de Taiwan (TFRI)
- *Taif*: Herbario del TFRI
- *Flyhorse*: Colección de insectos del TFRI
- *TaiBNET*: Catalogo de vida de Taiwan

LODE utiliza el documento de metadatos del Catálogo de Datos de Investigación del TFRI que a su vez usa el Lenguaje de Metadatos Ecológicos (EML). La Ilustración 8 muestra como los nombres de especies extraídas de los hilos de Facebook están enlazadas a LODE usando la propiedad *owl:sameAs.*

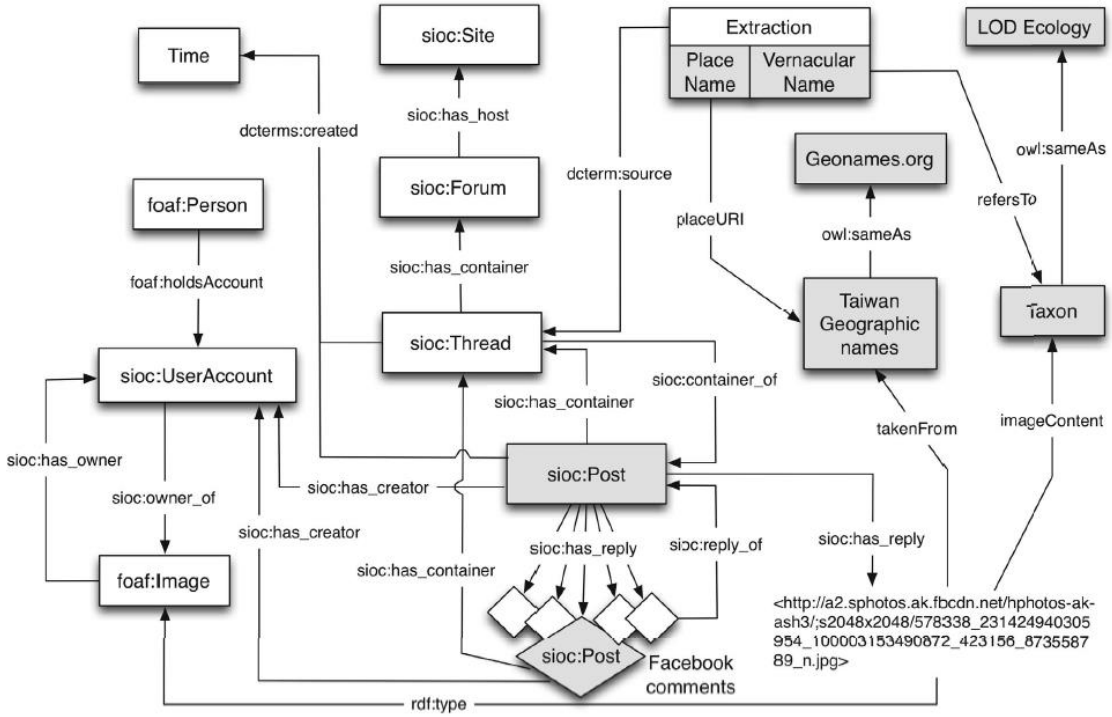



**Ilustración 7: Ontología para formalizar la información extraída de un hilo de Facebook**

GeoNames.org es un importante recurso LOD que, aunque proporciona nombres de lugares multilingües, no es lo suficientemente detallado para detectar el nombre de lugares en redes sociales. *TaiwanGazeteer* es otra fuente de nombres de lugares de Taiwán. Para poder enlazar recursos de CGU a nombres geográficos de Taiwán necesitaron transformar el *TaiwanGazeteer* en un recurso LOD. Usando los principios de LOD modificaron este recurso para generar el *LOD TaiwanGeographicNames* (LOD TGN), además se ha enlazado las entradas del LOD TGN a las de GeoNames utilizando enlaces de tipo *owl:sameAs*.

Para mejorar la entrada de datos del usuario en los hilos de Facebook, en este trabajo se ha desarrollado una extensión del navegador utilizando JavaScript. La extensión consiste en un motor que reconoce nombres de lugares y nombres de especies en el texto introducido por el usuario utilizando técnicas de LOD. Si se detecta que un nombre encaja con el nombre de una

entidad de los recursos LOD, la extensión mostrará al usuario un interfaz con sugerencias para permitirle que identifique el lugar exacto o la especie concreta a introducir.

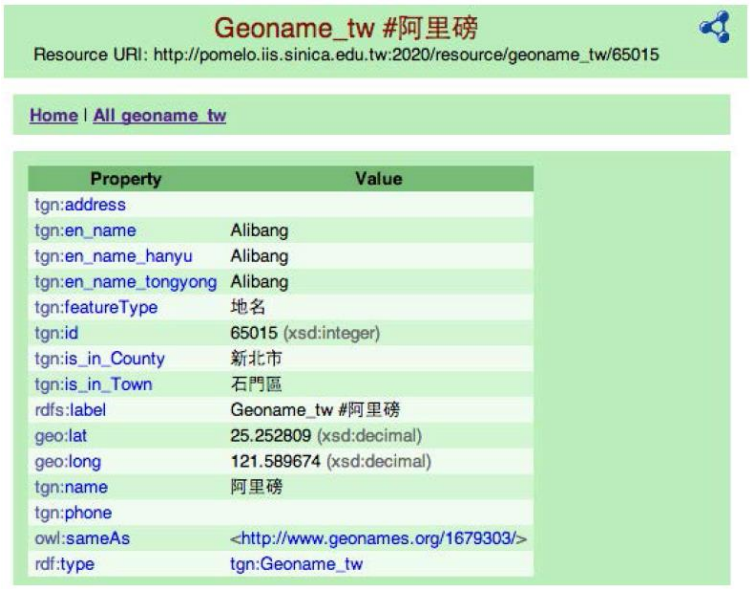
Chorodna ochreimacula

Resource URI: http://140.109.28.64:2020/resource/taxon/344731

Home | All Image All Taxon

| Property | Value |
|---|---|
| dwc:canonicalScientificName | Chorodna ochreimacula |
| cs-prop:distributionMap | <http://taibif.tw/vgd/cs/show_convex.php?spcode=344731> |
| dwc:family | Geometridae |
| dbpowl:family | <http://ecowlim.tfri.gov.tw/lode/resource/taibnet/Family/Geometridae> |
| is cs-prop:imageContent of | <http://a1.sphotos.ak.fbcdn.net/hphotos-ak-ash3/622006_397243850335048_1868357350_o.jpg> |
| is cs-prop:imageContent of | <http://a1.sphotos.ak.fbcdn.net/hphotos-ak-snc7/477962_128076690662709_1931419059_o.jpg> |
| is cs-prop:imageContent of | <http://a4.sphotos.ak.fbcdn.net/hphotos-ak-ash4/415298_441217595892480_1668370545_o.jpg> |
| is cs-prop:imageContent of | <http://a4.sphotos.ak.fbcdn.net/hphotos-ak-ash4/471981_147272025405404_1454404199_o.jpg> |
| is cs-prop:imageContent of | <http://sphotos-d.ak.fbcdn.net/hphotos-ak-ash3/s2048x2048/599284_333611233389165_1955047680_n.jpg> |
| gs:isExpectedIn | <http://pomelo.iis.sinica.edu.tw:2020/resource/geoname_tw/65015> |
| rdfs:label | Chorodna ochreimacula |
| is cs-prop:refersTo of | <http://140.109.28.64:2020/resource/extraction/177883715557195_422581304420767-1> |
| owl:sameAs | <http://ecowlim.tfri.gov.tw/lode/resource/taibnet/Species/Chorodna_ochreimacula> |
| dwc:taxonID | 344731 |
| rdf:type | dwc:Taxon |
| dwc:vernacularName | 圓枯尺蛾 |
| dwc:vernacularName | 黃枯尺蛾 |

Ilustración 8: Enlace al nombre de la especie conectada a LODE mediante un owl:sameAs



Geoname_tw #阿里磅

Resource URI: http://pomelo.iis.sinica.edu.tw:2020/resource/geoname_tw/65015

Home | All geoname_tw

| Property | Value |
|---|---|
| tgn:address | |
| tgn:en_name | Alibang |
| tgn:en_name_hanyu | Alibang |
| tgn:en_name_tongyong | Alibang |
| tgn:featureType | 地名 |
| tgn:id | 65015 (xsd:integer) |
| tgn:is_in_County | 新北市 |
| tgn:is_in_Town | 石門區 |
| rdfs:label | Geoname_tw #阿里磅 |
| geo:lat | 25.252809 (xsd:decimal) |
| geo:long | 121.589674 (xsd:decimal) |
| tgn:name | 阿里磅 |
| tgn:phone | |
| owl:sameAs | <http://www.geonames.org/1679303/> |
| rdf:type | tgn:Geoname_tw |

Ilustración 9: Entrada del LOD TGN enlazada a geonames.org por atributos *owl:sameAs*

Este trabajo es interesante por combinar de forma acertada la utilización de recursos LOD con contenido web generado por las redes sociales y muestra el gran potencial existente en la combinación de información.

También es posible realizar enriquecimientos de información a partir de los datos de microblogs. En [12] se realiza la anotación semántica de señales sociales provenientes de Twitter. Para ello se combina la información en tiempo real que proporcionan los *tweets* de Twitter con la capacidad de consulta y análisis que aporta la Web Semántica. Se presenta la arquitectura "Linked Open Social Signals" que recopila anota, filtra y envía a los usuarios

interesados las señales sociales relevantes sobre un determinado tema. La arquitectura propuesta consiste en recopilar los *tweets* mediante la Twitter Streaming API[1], a continuación se realizan tareas de extracción de información para anotar los *tweets* con las entidades, *hashtags* y URLs que se mencionan en ellos. El resultado se almacena en RDF sobre el cual se realizarán consultas SPARQL que combinadas con datos de la nube LOD permitirán filtrar los mensajes de más interés para los usuarios. Estas técnicas se han puesto en práctica en un escenario relacionado con la reforma sanitaria de EEUU impulsada por Barack Obama.

[1] http://api.twitter.com

## 1.2 ¿Cómo se combina y enriquece la información?

Los procesos de enriquecimiento que se pueden llevar a cabo pueden ser muy diversos: el enfoque más sencillo consiste en identificar las entidades sobre las que trabajar y realizar un enriquecimiento básico de datos recuperando algunas propiedades de cada entidad de la DBPedia o cualquier otro conjunto de datos disponible mediante la ejecución de consultas SPARQL sobre los datasets existentes.

Puede ocurrir que las fuentes de datos que se deseen consultar existan pero no estén disponibles en un formato adecuado para la Web Semántica (Excel, PDF, CSV,…), en ese caso es posible recopilar dicha información y publicarla de acuerdo a los estándares LOD aunque esto requiere un cierto esfuerzo. Ya se ha descrito el trabajo de la BBC [6] para incorporar datos etiquetados. Otro ejemplo es *RailGB* [13] donde eran necesarios datos sobre los elementos de accesibilidad de las distintas estaciones de Metro de Londres. Dicha información existía pero los formatos no eran adecuados para su interacción con la nube LOD (unos datos estaban disponibles en formato Excel, otros datos se obtenían de diversas APIs, etc). Esto obligó al equipo de *RailGB* a recopilar dicha información obteniendo volcados de datos de accesibilidad de la API *Transport for London*[1] en formato RDF/N3. Además, al no existir una ontología adecuada para describir infraestructuras de estaciones de metro fue necesario crear la ontología TF (*Tube Facility*) y finalmente se publicaron los datos basándose dicha ontología como un endpoint de SPARQL.

Existen varias alternativas para publicar Linked Data: publicar ficheros RDF en un servidor web, embeber RDF en documentos HTML mediante RDFa, generar ficheros RDF dinámicamente mediante scripts que corren del lado servidor o montar un endpoint SPARQL que sirva como punto de entrada a un almacén de tripletes RDF. Existe una opción adicional consistente en publicar como Linked Data los datos existentes en una base de datos relacional utilizando un servidor D2R[2]. En D2R se realiza un mapeo declarativo entre el esquema de la base de datos y los términos RDF a utilizar. A partir de este mapeo, el servidor D2R publica un endpoint SPARQL con la información existente en la base de datos. Este método fue el utilizado por el equipo de Dong-Po Deng [7] para publicar observaciones ecológicas provenientes de hilos de Facebook en la nube LOD, como ya se ha descrito.

Aparte de realizar tareas básicas de recuperación de propiedades de un conjunto de datos como la DBPedia, también es posible realizar tareas más sofisticadas de enriquecimiento utilizando el sistema de categorías de la Wikipedia como en [8], que se describe con más detalle a continuación.

El número de websites de datos abiertos editados de forma colaborativa como Wikipedia o GeoNames ha crecido mucho en los últimos años. Estas bases de datos están interconectadas bajo los principios de Linked Data. GeoNames y la Wikipedia ocupan un lugar muy importante en la Web de Datos ya que actúan como concentradores de enlaces de datos abiertos. Ha

[1] http://publishmydata.com/datasets/transport-for-london
[2] http://d2rq.org/d2r-server

habido varios intentos de enlazar estas dos bases de datos entre sí, descubriendo enlaces entre ellas. El número de enlaces entre ambos sistemas es de 385.754 (a fecha de Mayo 2012).

El sistema de categorías de la Wikipedia es un recurso muy utilizado en diversos trabajos para alineamiento de ontologías. El auge de Linked Data ha generado multitud de conjuntos de datos, aunque a menudo estos conjuntos de datos están débilmente conectados, BLOOMS [9] es un trabajo que se basa en la idea de enlazar información presente en la nube LOD con la ayuda de datos generados por la comunidad y disponibles en la Web para lo que utiliza la Wikipedia y su sistema jerárquico de categorías.

### 1.2.1 Entrelazado de datos con la DBpedia

En muchos trabajos relacionados con LOD, la DBPedia [4] se erige como un recurso fundamental dentro de la nube de Linked Data ya que funciona como un concentrador o *hub* de prácticamente cualquier aplicación de datos enlazados.

La DBPedia constituye un esfuerzo colaborativo para obtener información estructurada proveniente de la Wikipedia y ofrecer esta información disponible en la Web. Se ha convertido en un estándar de facto para tareas de clasificación. En [4] se describe la forma en que se construyen los conjuntos de datos de la DBPedia y cómo se publican estos datos para su consumo masivo.

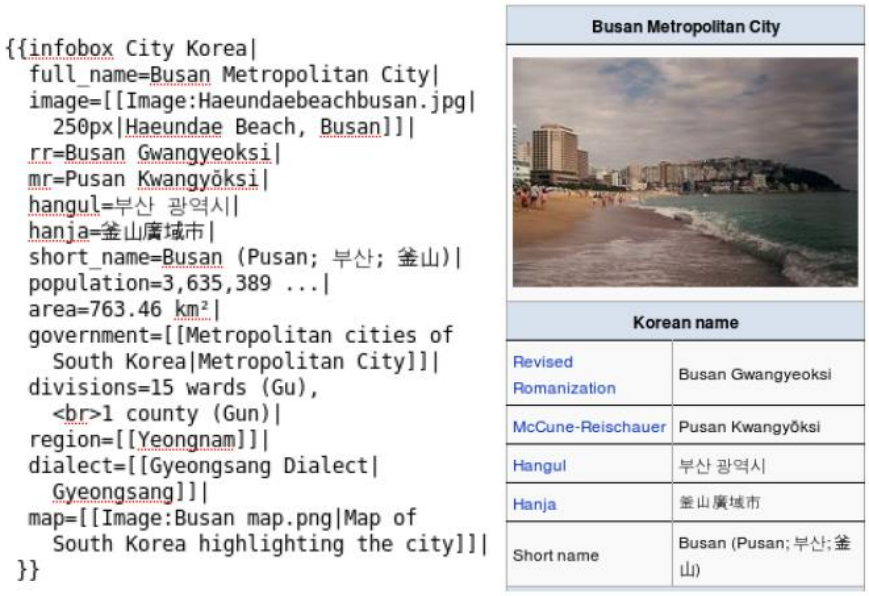
Uno de los problemas que la Web Semántica debía resolver era disponer de un modelo de representación y gestión de información estructurada para afrontar de una manera uniforme cuestiones de inconsistencia, ambigüedad, incertidumbre, procedencia de datos y conocimiento implícito. El proyecto DBPedia ha sido capaz de generar dicho corpus a partir de los datos existentes en la Wikipedia centrándose en los siguientes aspectos:

- Se ha desarrollado un entorno de extracción de información que convierte el contenido de la Wikipedia en datos RDF.
- Se ofrece el contenido de la Wikipedia como un gran conjunto de datos RDF multidominio: contiene 103 millones de tripletes RDF.
- Se ha realizado un enlazado entre la DBPedia y otros conjuntos de datos generando como resultado un conjunto de datos global de unos 2 billones de tripletes RDF.
- Se han desarrollado una serie de interfaces de forma que el dataset de la DBPedia puede ser accedido y enlazado a través de diferentes servicios web.

Los artículos de la Wikipedia consisten principalmente en texto libre, pero también contienen información estructurada: plantillas infobox (datos tabulares que aparecen en muchas páginas de la DBPedia), información sobre categorías, imágenes, geo-coordenadas, enlaces a páginas externas y enlaces a diferentes versiones de la Wikipedia.

Mediawiki es el software utilizado para generar la Wikipedia. Debido a la naturaleza de este sistema Wiki, todas las ediciones, enlaces y anotaciones con metadatos se incluyen en los textos de los artículos añadiendo determinadas construcciones sintácticas. Mediante el análisis

de estas construcciones es posible obtener la información estructurada que constituye la DBPedia.



```
{{infobox City Korea|
  full_name=Busan Metropolitan City|
  image=[[Image:Haeundaebeachbusan.jpg|
    250px|Haeundae Beach, Busan]]|
  rr=Busan Gwangyeoksi|
  mr=Pusan Kwangyŏksi|
  hangul=부산 광역시|
  hanja=釜山廣域市|
  short_name=Busan (Pusan; 부산; 釜山)|
  population=3,635,389 ...|
  area=763.46 km²|
  government=[[Metropolitan cities of
    South Korea|Metropolitan City]]|
  divisions=15 wards (Gu),
    <br>1 county (Gun)|
  region=[[Yeongnam]]|
  dialect=[[Gyeongsang Dialect|
    Gyeongsang]]|
  map=[[Image:Busan map.png|Map of
    South Korea highlighting the city]]|
 }}
```

**Busan Metropolitan City**

**Korean name**

| | |
|---|---|
| Revised Romanization | Busan Gwangyeoksi |
| McCune-Reischauer | Pusan Kwangyŏksi |
| Hangul | 부산 광역시 |
| Hanja | 釜山廣域市 |
| Short name | Busan (Pusan; 부산; 釜山) |

Ilustración 10: Plantilla infobox de Wikipedia de una ciudad SurCoreana y su codificación

El proceso de extracción de plantillas infobox detecta este tipo de construcciones y reconoce su estructura utilizando técnicas de reconocimiento de patrones. Selecciona plantillas significativas que son traducidas a tripletes RDF. Los enlaces de MediaWiki son convertidos en URIs y determinadas estructuras comunes son transformadas a tipos de datos.

El dataset de la DBPedia proporciona información sobre más de 4 millones de elementos, incluyendo al menos 832.000 personas, 639.000 lugares, 372.000 trabajos creativos etc. Contiene además 27,6 millones de enlaces a páginas web externas, 45 millones de enlaces a otros datasets RDF externos, 67 millones de enlaces a categorías de la Wikipedia y 41,2 millones de categorías YAGO (datos correspondientes a Febrero de 2014).

Aunque el dataset de la DBPedia está formado por 103 millones de tripletes RDF se ofrece dicho dataset para su descarga en diferentes ficheros RDF más pequeños. Algunos de ellos son:

- *Artículos*: Descripciones de 1.95 millones de conceptos que incluyen un título y un corto resumen (abstract).
- *Infoboxes*: Datos en forma de atributos sobre los conceptos que han sido obtenido a partir de los infoboxes de la Wikipedia.
- *Enlaces externos*: Links a páginas externas sobre un concepto.
- *Categorías de artículos*: Enlaces de conceptos a categorías utilizando SKOS
- *Categorías*: Información cuyo concepto es una categoría y cómo se relacionan las categorías entre sí.
- *Tipos Yago*: Dataset que contiene sentencias de tipo *rdf:type* para todas las instancias de la DBPedia usando el sistema de clasificación de YAGO.
- *Personas*: Información sobre personas representadas con el vocabulario FOAF

- *Enlaces RDF*: Enlaces entre DBPedia y GeoNames, Musicbrainz, Proyecto Gutenberg, bibliografía de la DBLP, etc.

Cada uno de los recursos pertenecientes a la DBPedia se identifica por una URI con la forma http://dbpedia.org/resource/Nombre, donde "Nombre" proviene de la url del artículo de la Wikipedia que tiene la forma http://en.wikipedia.org/wiki/Nombre de forma que es posible identificar de manera inequívoca cada concepto.

Es posible utilizar el dataset de la DBPedia mediante tres mecanismos de acceso:

1) **Datos Enlazados**: Método de publicación de datos RDF en la Web basado en los identificados de recurso (URI) y en el protocolo HTTP. Las URIs están diseñadas para devolver información relevante acerca de un recurso, normalmente una descripción RDF que contiene todos los datos relativos al recurso. Dicha información suele incluir URIs a recursos relacionados que pueden ser recuperables. Las URIs devuelven descripciones RDF cuando son accedidas por agentes de Web Semántica y HTML cuando son accedidas por un navegador web.
2) **Endpoint de SPARQL**: DBPedia ofrece un endpoint SPARQL[1] sobre el que se pueden realizar consultas SPARQL.
3) **Volcados RDF**: Consiste en serializaciones de tripletes disponibles para ser descargados. Es la opción más adecuada si se desea obtener grandes volúmenes de datos.

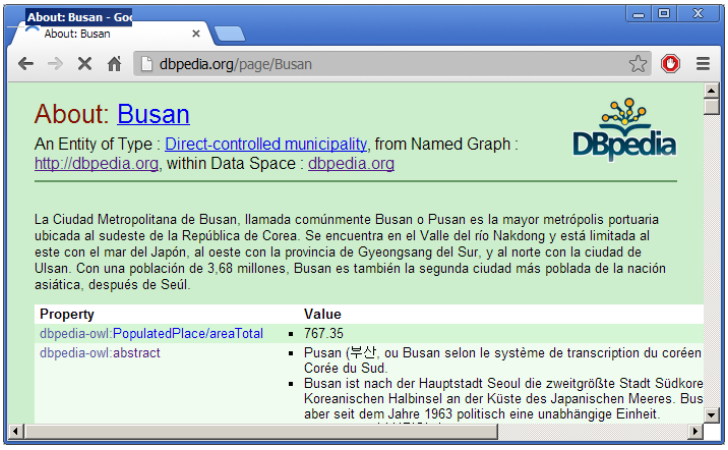



**Ilustración 11: Vista HTML de una URI de la DBPedia**

Para permitir el descubrimiento de nueva información más allá de la DBPedia, ésta dispone de conexiones con otros conjuntos de datos usando enlaces RDF. Este esfuerzo de entrelazado de la DBPedia es parte del proyecto Linked Open Data del W3C con el objetivo de que grandes datasets y ontologías como US Census, Geonames, MusicBrainz, Wordnet y otras sean interoperables con la Web Semántica. La DBPedia supone el punto de unión de todos estos datasets. En la siguiente figura se muestra ver una visión general de los diferentes datasets entrelazados por la DBPedia.

[1] http://dbpedia.org/sparql

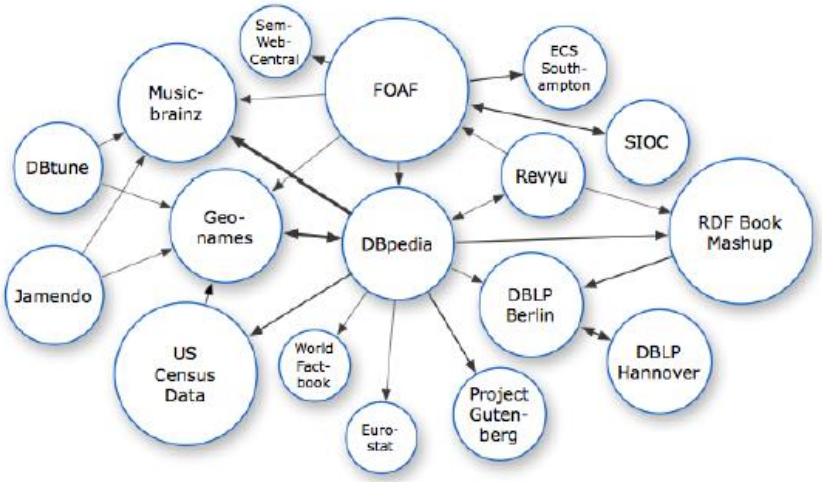



Ilustración 12: Algunos Datasets entrelazados por la DBPedia

**YAGO** extrae información estructurada de la Wikipedia aunque solamente 14 tipos de relaciones tales como: *subClassOf*, *type*, *familyNameOf*, *locatedIn* desde diferentes fuentes de información, como el sistema de categorías de la Wikipedia y sus redirecciones, pero no realiza una extracción a partir de los datos de las infobox. Además para determinar las relaciones de tipo clase-subclase no utiliza el sistema de categorías jerárquico de la Wikipedia sino los enlaces a las categorías en la jerarquía de WordNet.

Otro enfoque interesante es el utilizado por Freebase[1]. Se basa en la construcción de una inmensa base de datos online que los usuarios pueden editar de forma similar a como se editan los artículos de la Wikipedia. Se están realizando trabajos para enlazar datos entre la DBPedia y Freebase de forma que el conjunto enriquezca el sistema global de datos enlazados.

Se ha utilizado con éxito la DBPedia en diversas áreas de lingüística computacional [19]: Enlazado de Entidades, Desambiguación del sentido de las palabras, Respuesta a Preguntas y Extracción de Relaciones.

### 1.2.2 Enriquecimiento de entidades con Yago

Es posible utilizar la información disponible en la Wikipedia en combinación con otros recursos para obtener información adicional. Por ejemplo la inclusión de una ontología como YAGO [5] en la DBPedia ha abierto nuevas posibilidades en la inferencia de nueva información. CAF-SIAL [10] utiliza YAGO en combinación con la Wikipedia para determinar las propiedades más relevantes de una entidad.

YAGO[2] es una ontología ligera y extensible de gran calidad y amplia cobertura que se construye a partir de entidades y relaciones entre ellas que constituyen hechos. Esta información se extrae automáticamente de la Wikipedia y WordNet utilizando una combinación de métodos heurísticos y sistemas basados en reglas. El resultado es una base de conocimiento que supera a WordNet en calidad y cantidad, añadiendo conocimiento sobre

[1] http://www.freebase.com
[2] http://www.mpi-inf.mpg.de/yago-naga/yago

personas, organizaciones, productos etc. Tiene una precisión del 95% y está basado en un modelo lógico extensible y compatible con RDFS.

Para la construcción de YAGO se utilizaron las "páginas de categorías" de la Wikipedia. Las páginas de categorías son listas de artículos que pertenecen a una categoría específica (Ej: Zidane está en la categoría "*Jugadores de futbol franceses*"). Estas listas proporcionan candidatos de entidades (Ej: *Zidane*), candidatos de conceptos (Ej: *EsUn(Zidane, Jugador de Futbol))* y candidatos de relaciones (Ej: *esCiudadanoDe(Zidane, Francia)*). En una ontología los conceptos se organizan en forma de taxonomía para su posterior uso. Las categorías de la Wikipedia también se organizan en una jerarquía pero ésta no es muy útil para usos ontológicos, ya que por ejemplo *Zidane* se incluye en la categoría "*Futbol en Francia*", pero *Zidane* es un jugador de futbol y no un tipo de *Futbol*. Por el contrario WordNet proporciona una jerarquía diseñada cuidadosamente de miles de conceptos, pero los conceptos de la Wikipedia no tienen una correspondencia obvia en WordNet.

En [10] se presentaron nuevas técnicas que permiten a YAGO enlazar estas dos fuentes con una precisión prácticamente perfecta (95%). Permite que YAGO incorpore la ingente cantidad de entidades de la Wikipedia y al mismo tiempo la taxonomía de conceptos de WordNet. Actualmente YAGO incorpora 900.000 entidades y 5 millones de relaciones entre ellas.

El modelo de datos en que se basa YAGO debe ser capaz de representar entidades, hechos, relaciones entre hechos y propiedades de relaciones. RDFS permite expresar propiedades de relaciones pero con una semántica muy básica (no soporta transitividad). Este es el motivo por el cual se ha utilizado una ligera extensión de RDFS para el modelo YAGO.

Por ejemplo para representar que Einstein ganó el Premio Nobel se dice que "*AlbertEinstein*" tiene la relación "*HaGanadoPremio*" con la entidad "*PremioNobel*". Números, fechas y otros literales se representan igualmente como entidades, de esta forma podríamos representar su fecha de nacimiento como:

AlbertEinstein BORNINYEAR 1879

1) Las entidades son entes ontológicos abstractos independientes del lenguaje. Un lenguaje utiliza palabras para referirse a esas entidades. En YAGO las palabras son también entidades. Esto hace posible expresar que una cierta palabra se refiere a una cierta entidad y permite tratar la sinonimia y la ambigüedad:

   "Einstein" MEANS AlbertEinstein

2) Entidades similares se agrupan en clases. Por ejemplo la clase "*physicist*" engloba a todos los físicos. Cada entidad es una instancia de al menos una clase y se representa con la relación tipo (TYPE):

   AlbertEinstein TYPE physicist

3) Las clases son además entidades. Cada clase es una instancia de otra clase. Las clases se organizan en una jerarquía taxonómica expresada por la relación SUBCLASSOF:

   Physicist SUBCLASSOF scientist

4) Las relaciones son también entidades. Esto hace posible representar propiedades de relaciones (como la transitividad) en el modelo. La siguiente línea establece que la relación *SubClassOf* es transitiva:

subclassOf TYPE transitiveRelation

5) Un triplete de entidad, relación y otra entidad es un hecho que además posee un identificador. Los identificadores de hechos son también entidades, de forma que es posible representar que un cierto hecho se encontró en una cierta URL. Suponiendo que el hecho (Einstein BornInYear 1879) tiene el identificador #1 podríamos representar que ese hecho se encontró en la Wikipedia de la siguiente forma:

#1 FoundIn http://www.wikipedia.org/Einstein

A nivel semántico, cualquier ontología YAGO debe contener al menos las relaciones *type, subClassOf, domain, range y subRelationOf.*Las entidades comunes deben contener al menos las clases *entity, class, relation, acyclicTransitiveRelation* y deben existir clases para todos los literales.

El modelo de YAGO es muy similar a RDFS. En RDFS las relaciones reciben el nombre de propiedades. Al igual que YAGO, RDFS reconoce las *propiedades domain, range, subClassOf y subPropertyOf*. RDFS además utiliza identificadores de hechos que pueden ser utilizados como argumentos para otros hechos.

Las relaciones en YAGO están prefijadas. Sus propiedades (como dominio y rango) se pueden ver en la Tabla 1. YAGO está diseñado para ser extensible de manera que se puedan añadir a la ontología nuevos hechos provenientes de nuevas fuentes. Para ello cada hecho está etiquetado con un valor de confianza: los hechos de YAGO tienen una confianza de 1.0, los hechos extraídos con otras técnicas pueden tener valores de confianza menores.

| Relation | Domain | Range | # Facts |
|---|---|---|---|
| subClassOf | class | class | 126,792 |
| type | entity | class | 2,011,072 |
| context | entity | entity | ∼40,000,000 |
| describes | word | entity | 997,061 |
| bornInYear | person | year | 189,950 |
| diedInYear | person | year | 93,827 |
| establishedInYear | entity | year | 14,602 |
| locatedIn | object | region | 60,354 |
| writtenInYear | book | year | 4,399 |
| politiciansOf | organization | person | 3,618 |
| hasWonPrize | person | prize | 1,024 |
| means | word | entity | 2,166,891 |
| familyNameOf | word | person | 181,926 |
| givenNameOf | word | person | 177,291 |

Tabla 1: Cobertura de YAGO

Con más detalle:

- **La relación "type".** Los individuos de YAGO se extraen de la Wikipedia, ya que ésta contiene muchos más que WordNet. Cada título de página de la Wikipedia es candidato a convertirse en un individuo en YAGO. Para establecer la clase de cada individuo se utiliza el sistema de categorías de la Wikipedia. Existen de diferentes tipos: categorías

conceptuales que identifican una clase por la entidad de la página (Ej: Einstein está en la categoría *"Ciudadanos nacionalizados en los EEUU"*) otras sirven a propósitos administrativos, otras incorporan información relacional (Ej: *Nacidos en 1879*) y otros indican simplemente una afinidad temática (Ej: *Físicos*). Para establecer la clase de un individuo sólo las categorías conceptuales son candidatas. Las categorías administrativas y relacionales son muy escasas y pueden ser excluidas a mano.

- **La relación "subClassOf".** Como se comentó anteriormente, el sistema jerárquico de categorías de la Wikipedia está diseñado desde un punto de vista temático y no es de mucha utilidad desde un punto de vista ontológico por lo que YAGO construye su jerarquía de clases a partir de WordNet.

  Cada synset de WordNet se convierte en una clase en YAGO excluyéndose los nombres propios que se convertirán en individuos. La jerarquía de clases en la que se basa la relación *SubClassOf* se construye a partir de la relación de hiponimia de WordNet. Una clase es una subclase de otra si el primer synset es un hipónimo del segundo. Las clases más bajas en la jerarquía extraídas de la Wikipedia se conectan a clases de más alto nivel extraídas de WordNet mediante un algoritmo de mapeo y desambiguación, de esta forma la clase de la Wikipedia "*Gente Americana en Japón*" se convierte en una subclase de la clase WordNet "*Gente*".

- **La relación "means".** Tanto la Wikipedia como WordNet incluyen información sobre el significado de las palabras. Por ejemplo en WordNet las palabras "centro urbano" y "metrópolis" pertenecen al synset "ciudad". Esta información se incorpora a YAGO creando una entidad por cada nombre de WordNet (Ej: físico) y estableciendo una relación MEANS entre cada palabra o synset y la clase correspondiente. Ej: "físico", MEANS físico.

  YAGO utiliza las páginas de redirección de la Wikipedia para obtener nombres alternativos para las entidades. Las páginas de redirección de la Wikipedia redireccionan al usuario a la página de Wikipedia correcta. Por ejemplo si el usuario teclea "*Einstein, Albert*" existe una redirección a "*Albert Einstein*". Por cada página de redirección de la Wikipedia, YAGO incluye un hecho MEANS: Ej: ("*Einstein, Albert*", MEANS , *Albert Einstein*).

- **Otras relaciones.** Basándose en las categorías de la Wikipedia se extraen también las relaciones: *BornInYear, DiedInYear, EstablishedInYear, LocatedIn, WrittenInYear, PoliticianOf, y HasWonPrize*. Para la extracción de las relaciones *BornInYear* se usan las categorías que terminan con la palabra "_births". Por ejemplo si una página está en la categoría "1879_births" podemos asumir que el individuo es una persona nacida en 1879. De forma análoga se extraen el resto de relaciones.

  YAGO está diseñado para ser extendido con nuevos hechos. La relación *FoundIn* identifica la URL de donde se extrajo el hecho y *ExtractedBy* identifica la técnica mediante la cual un hecho fue extraído. Estas relaciones permitirán a las aplicaciones utilizar solo hechos extraídos mediante determinadas técnicas o desde determinadas fuentes.

  YAGO almacena para cada entidad, aquellas a las que está enlazado en la página de la Wikipedia correspondiente. Por ejemplo "*Albert Einstein*" esta enlazado a la "*Teoría de la*

*Relatividad*". Esta información se refleja en la relación *CONTEXT* y facilita a las aplicaciones las tareas de desambiguación.

El modelo de YAGO es independiente de cualquier formato de almacenamiento físico, utiliza un formato de texto simple donde se mantiene una carpeta por cada relación y cada carpeta contiene ficheros que listan las parejas de entidades. Además se proporcionan con YAGO programas de conversión para convertir la ontología a diferentes formatos de salida como XML, RDFS o a una tabla de base de datos Oracle o MySQL.

Para evaluar la precisión de una ontología sus hechos deben ser cotejados con una base de comparación que es muy difícil de obtener, por lo que en [10] se realizó una evaluación manual. Se presentaron hechos de la ontología elegidos aleatoriamente a usuarios humanos y se les pidió que evaluaran si los hechos eran correctos. La evaluación mostró muy buenos resultados (95% de precisión media). Especialmente la relación TYPE y el enlace entre WordNet y Wikipedia (*SubClassOf*) resultaron ser muy acertados.

La Tabla 3 muestra el número de hechos existentes en cada relación de YAGO. No es fácil comparar la cobertura de YAGO con la de otras ontologías ya que las diferentes ontologías difieren en su estructura, sus relaciones y su dominio pero basta con considerar que las ontologías independientes del dominio más importantes suelen contener del orden cientos de miles de hechos mientras que YAGO incorpora millones.

| Relation | # evaluated facts | Accuracy | |
|---|---|---|---|
| subClassOf | 298 | 97.70% | ± 1.59% |
| type | 343 | 94.54% | ± 2.36% |
| familyNameOf | 221 | 97.81% | ± 1.75% |
| givenNameOf | 161 | 97.62% | ± 2.08% |
| establishedInYear | 170 | 90.84% | ± 4.28% |
| bornInYear | 170 | 93.14% | ± 3.71% |
| diedInYear | 147 | 98.72% | ± 1.30% |
| locatedIn | 180 | 98.41% | ± 1.52% |
| politicianOf | 176 | 92.43% | ± 3.93% |
| writtenInYear | 172 | 94.35% | ± 3.33% |
| hasWonPrize | 122 | 98.47% | ± 1.57% |

**Tabla 2: Precisión de YAGO**

| Relation | Domain | Range | # Facts |
|---|---|---|---|
| subClassOf | class | class | 126,792 |
| type | entity | class | 2,011,072 |
| context | entity | entity | ∼40,000,000 |
| describes | word | entity | 997,061 |
| bornInYear | person | year | 189,950 |
| diedInYear | person | year | 93,827 |
| establishedInYear | entity | year | 14,602 |
| locatedIn | object | region | 60,354 |
| writtenInYear | book | year | 4,399 |
| politiciansOf | organization | person | 3,618 |
| hasWonPrize | person | prize | 1,024 |
| means | word | entity | 2,166,891 |
| familyNameOf | word | person | 181,926 |
| givenNameOf | word | person | 177,291 |

**Tabla 3: Cobertura de YAGO en número de hechos**

### 1.2.3 Entrelazando información con Wikipedia

En [8] se describe un método automático de descubrimiento de enlaces para identificar correspondencias entre GeoNames y la Wikipedia y se propone utilizar información de distancias para evaluar la calidad de estos enlaces. Sin embargo, debido a la especial naturaleza de estos conjuntos de datos que pueden contener cierto número de errores, la información obtenida por distancias podría no ser demasiado fiable. Existen varios casos además, donde es difícil justificar lo apropiado de los enlaces. Las categorías de la Wikipedia usan un estilo común para representar la información de categorías que facilita la extracción de su información semántica. Por ejemplo existen muchas categorías con el formato *"<clase semántica> (of|in) <información de localización>*". De esta forma la página de la Wikipedia sobre "Alabama" (http://dbpedia.org/page/Alabama) que incluye la categoría (dcterms:subject) "States of the United States" nos permite inferir que se trata de uno de los "Estados" de los "Estados Unidos".

El procedimiento utilizado para encontrar parejas entre Wikipedia y GeoNames consiste en:

1) **Comparación de nombres de página de la Wikipedia y nombres de GeoNames.** La Wikipedia utiliza ciertas convenciones para identificar la información y asegurar la desambiguación de entidades. Por ejemplo, se añade información de desambiguación justo después del nombre de una entidad geográfica para discriminar diferentes entradas geográficas para "Roma". Ej: "Roma, Italia", "Roma Iowa". Como esa información no existe en GeoNames eliminaron la información de desambiguación (todas la información después de la primera ',') durante las comparaciones.
2) **Extracción de información de las categorías.** Se extrae la clase semántica y la información de localización comparando la categoría de la Wikipedia con la plantilla *"<clase semántica> (of|in) <información de localización>*". Para poder asociar una clase semántica de la Wikipedia con una clase de GeoNames prepararon manualmente una lista de correspondencias entre ambas.

| Descripción General | Clase semántica Wikipedia | Clase de GeoNames |
|---|---|---|
| División administrativa y ciudades | Populated places, villages, Neighborhoods | ADM1, ADM2, ADM3, PPL, PPLA, PPLA2 |
| Aeropuertos | Air force, air base, airfields, air bases, airfield, airports | AIRQ,AIRB, AIRF, AIRP,AIRH |
| Bahías y golfos | Lochs, gulfs, coves, bays | BAY,BAYS,COVE,GULF |

Tabla 4: Correspondencias entre clases semánticas de la Wikipedia y clases de GeoNames

La información de localización extraída de las categorías de la Wikipedia se aplicó a la selección de parejas de candidatos, utilizando listas de países y códigos administrativos como información de localización. La generación de dichas listas requiere la comparación de la cadena de texto de localización de la entrada de GeoNames con una clase que describa la división administrativa.

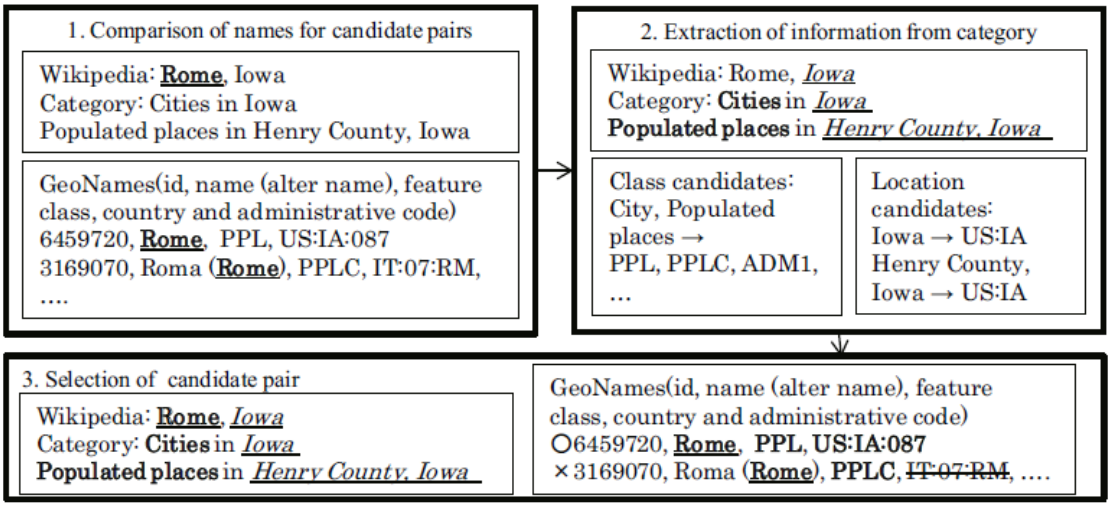



Ilustración 13: Búsqueda de parejas candidatas en Wikipedia y GeoNames

3) **Selección de parejas candidatas.** Se seleccionan parejas apropiadas de entre las candidatas, utilizando la información de la clase de GeoNames estimada a partir de la clase semántica y los códigos administrativos y de país obtenidos de las cadenas de localización.

   Además se utilizan métodos para excluir parejas no apropiadas y mantener la consistencia entre las parejas candidatas:

   - Múltiples páginas Wikipedia para una única entrada de GeoNames: Se asume que diferentes páginas de la Wikipedia representan diferentes entidades geográficas por lo que se excluyen estas parejas de la lista de candidatas para conseguir una buena precisión.
   - Múltiples entradas GeoNames para una única página de Wikipedia: En este caso se calculan las distancias entre las entradas de GeoNames. Si una pareja de entradas de GeoNames están separadas por más de 5 Km, esta pareja se marca como inapropiada y se excluye de la lista de candidatas.

Tras las pruebas realizadas para descubrir enlaces entre Wikipedia y GeoNames se encontraron parejas para 51.016 páginas de la Wikipedia. Posteriormente se eligieron aleatoriamente 200 parejas y se analizaron manualmente, encontrándose los siguientes tipos de errores:

- **Variaciones en nombres**: Algunos nombres de entidades no estaban representados en inglés en GeoNames, lo cual hace muy difícil encontrar su pareja en una página de Wikipedia. Se excluyeron este tipo de parejas aplicando análisis de inconsistencia para este tipo de entradas.
- **Fallos al estimar el código administrativo apropiado**: En varios casos el método falló al intentar encontrar el código administrativo de una localización. Estos errores condujeron a desajustes entre páginas de Wikipedia y entradas de GeoNames con diferentes códigos administrativos. Además el texto de información de localización fue insuficiente para seleccionar el código administrativo de una página Wikipedia para algunos países. Ej: pocas

categorías Sudafricanas contenían textos de localización con códigos de categoría administrativa de la Wikipedia.

El método propuesto en este trabajo no se basaba en información de coordenadas geográficas, por lo que se puede utilizar este método para evaluar la consistencia de la información de coordenadas geográficas en las diferentes bases de datos. Detectaron de esta forma tres tipos de inconsistencias:

- Información geográfica inconsistente para parejas apropiadas: este tipo de error puede ocurrir cuando una entidad geográfica cubre un área muy grande.
- Errores en la Wikipedia y/o en GeoNames.
- Errores debidos al método de detección que puede generar parejas inconsistentes.

Estos resultados demuestran que este método de descubrimiento de enlaces puede ser útil para encontrar errores en las bases de datos mencionadas.

Además durante el proceso de descubrimiento de enlaces se encontraron numerosas parejas que no pueden ser representadas usando la links de tipo *owl:SameAs*, utilizados en la DBPedia. Se encontraron varios casos donde una única página de la Wikipedia se corresponde a varias entradas en GeoNames:

- Entidades geográficas con múltiples puntos: Por ejemplo un río. En Wikipedia la tabla infobox de un río contiene su fuente y las coordenadas de su nacimiento. GeoNames contiene diferentes puntos de un río con el mismo nombre.
- Entidades geográficas con múltiples clases en GeoNames: Cada entrada en GeoNames se corresponde con una única clase. Si el nombre de una división administrativa es además el nombre de una ciudad es necesario utilizar dos entradas de GeoNames con el mismo nombre para dos clases distintas. Ej: “Milolii, Hawaii” tiene dos entidades GeoNames (5851041: División administrativa y 5851402: Lugar poblado).
- Páginas Wikipedia con múltiples entidades geográficas. Varias páginas de la Wikipedia contienen información sobre entidades geográficas múltiples. Ej: algunas páginas sobre cadenas montañosas contienen información sobre varias montañas.

En todos estos casos es aceptable construir links de tipo “Seealso” entre ambos sistemas aunque esto puede causar algunos problemas cuando se construyen redes de tipo “SameAs”.

En este trabajo de 2013, se ha mostrado cómo el método utilizado puede servir para detectar inconsistencias entre dos fuentes de datos y describe el tipo de problemas que pueden surgir al crear una aplicación basada en datos abiertos.

## 1.3 ¿Cómo se presenta el contenido enriquecido al usuario?

La presentación de información en una aplicación de datos entrelazados es muy importante. El diseño de una buena interfaz de usuario (IU) es clave para que una aplicación basada en Linked Data sea usable ya que se debe integrar de forma adecuada fuentes de datos diversas y mostrar al usuario información relevante en cada momento.

### 1.3.1 Interfaces de usuario en aplicaciones web

Existen diferentes formas de mostrar información enriquecida al usuario en una aplicación web y el objetivo de cada interfaz de usuario puede ser muy diverso. En [3] se muestran de forma integrada videos de *Youtube*, transcripciones e información LOD creando un entorno muy atractivo para el usuario. El objetivo es construir un sistema de anotación multimedia basado en web que permite asociar notas a ciertas partes de un recurso audiovisual.

También es posible utilizar una IU para facilitar la entrada de datos por parte del usuario. En el trabajo de Dong-Po Deng [7] se permite la entrada de datos en los hilos de Facebook mediante una extensión del navegador que contiene un motor de procesamiento de lenguaje natural (NLP) que reconoce nombres de lugares y nombres de especies de animales a partir del texto introducido por el usuario utilizando técnicas LOD. Cuando se detecta uno de estos nombres se muestran sugerencias para identificar el lugar exacto o la especie concreta mencionada.

Otra posibilidad consiste en ofrecer sugerencias a los editores de contenido en un portal Web para enriquecer y entrelazar la información existente tal como hace el sistema *Muddy Boots* en los portales de la BBC [6].

En [12] se describe una arquitectura que permite recuperar *tweets* de Twitter, detectar entidades en ellos y realizar tareas de enriquecimiento para lograr filtros muy específicos. Mediante un mecanismo similar a los *feeds* de RSS los usuarios pueden suscribirse a conceptos de su interés que internamente se modelan como consultas SPARQL. Para abstraer al usuario de conocimientos de Web Semántica se le ofrece una interfaz de usuario donde puede modelar los conceptos mediante una herramienta grafica que internamente traducirá dicho concepto a una consulta SPARQL

También es posible utilizar un interfaz para ayudar al usuario en la desambiguación de entidades como en [10] que se concentra en:

- Un **Marco de Agregación de Conceptos** (MAC) para presentar la información más relevante a partir de recursos LOD.
- Un **mecanismo de búsqueda de palabras clave** que oculte al usuario final la lógica de la búsqueda semántica subyacente. Recuérdese, que DBPedia y Yago son considerados estándares de facto para tareas de clasificación. DBPedia es usado habitualmente para el descubrimiento de hechos. Dicha información se extrae de la Wikipedia y se estructura en forma de propiedades definidas por la ontología de la DBPedia. Este vocabulario se asocia además con etiquetas de clasificación de Yago para identificar el tipo (persona, lugar, organización, etc.) del recurso.

El ámbito de actuación en [10] se limita a la DBPedia y a Yago, y solo utiliza el tipo de recurso Persona para la realización de una prueba de concepto. El MAC se encarga de agregar la información más relevante relacionada con la persona en cuestión. En la siguiente figura se puede ver su esquema de funcionamiento: La capa o nivel denominado de **Aspectos Inferidos** se encarga añadir la información más relevante relativa a la persona en cuestión. Esta información es obtenida a partir de la lista de propiedades relacionadas, recuperadas en la capa de **Agregación de Propiedades**. Las propiedades se extraen de las bases de conocimiento en la capa de **Agregación de Bases de Conocimiento**.

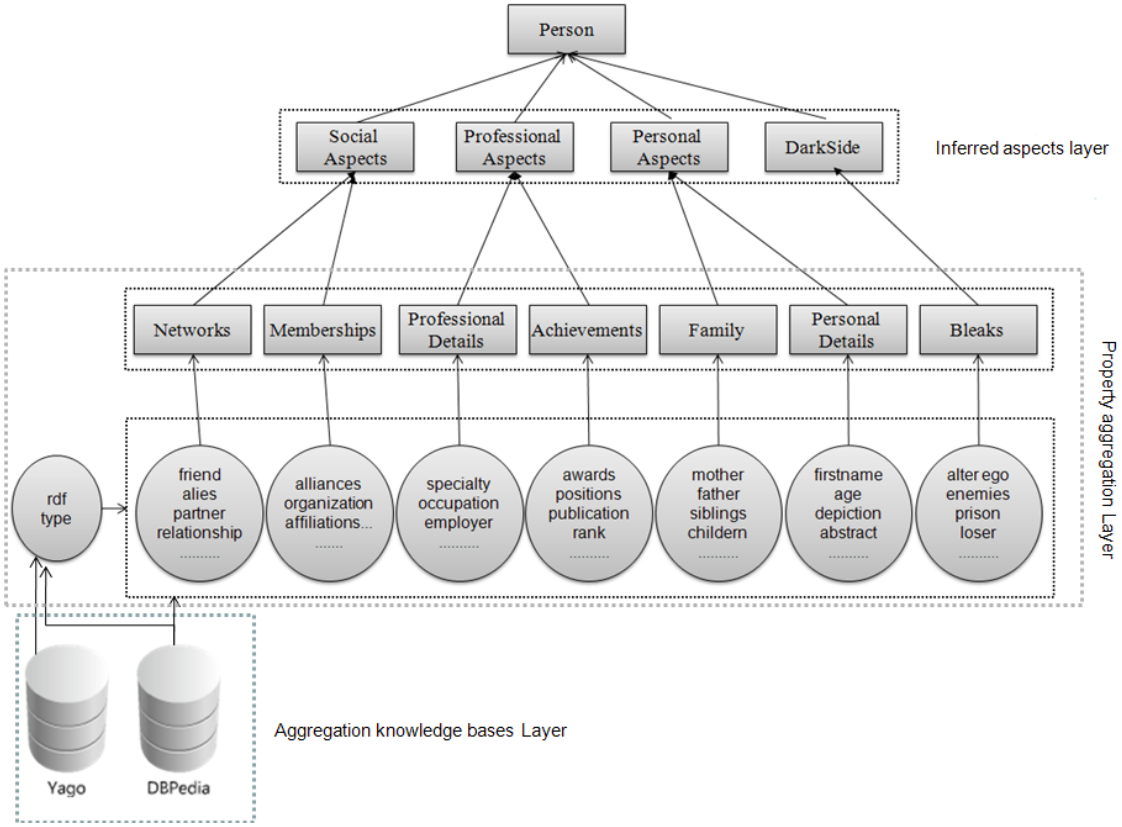



Ilustración 14: Marco de Agregación de Conceptos (MAC)

1) En la capa de Agregación de Bases de Conocimiento, se generan dos bases de datos de conocimiento, un volcado de propiedades de la DBPedia y un volcado de clasificaciones de Yago. El volcado de propiedades de la DBPedia se ha construido a partir de consultas SPARQL que devuelven las diferentes propiedades de cada tipo de Persona. Por ejemplo la consulta SPARQL que devuelve todas las propiedades del tipo de persona "*Artist*" es:

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdfschema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdfsyntax-ns#>
SELECT DISTINCT ?p
WHERE {
?s ?p ?o .
?s rdf:type<http://DBpedia.org/ontology/Artist> .
}
```

De esta forma es posible almacenar el conjunto de propiedades que contiene cada tipo de persona y hacer una selección de las más representativas. A continuación se puede ver el número de propiedades de cada tipo de persona:

| Tipo de persona | Número de propiedades | Propiedades seleccionadas |
|---|---|---|
| Artist | 2111 | 409 |
| Journalist | 186 | 55 |
| Cleric | 419 | 76 |
| BritishRoyalty | 252 | 47 |
| Athlete | 2064 | 496 |
| Monarch | 337 | 50 |
| Scientist | 421 | 126 |
| … | … | … |

El volcado de clasificaciones de YAGO se ha realizado consultando subclases de la clase Persona mediante una consulta SPARQL siguiente:

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdfschema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdfsyntax-ns#>
SELECT DISTINCT ?s
WHERE {
       ?s rdfs:subClassOf
       <http://DBpedia.org/class/yago/Person100007846> .
}
```

2) En la capa de **Agregación de Propiedades**, se identifica el tipo de profesión de la persona y se agregan sus propiedades representativas. Para ello se obtiene el tipo de recurso (tipo RDF) de la DBPedia. Como se ha realizado una agrupación de las propiedades asociadas a cada profesión, dada una persona en concreto se agregarán las propiedades asociadas típicas de su profesión (Ej: Un actor de cine presentará propiedades como "*haActuadoEn*" mientras que un director de cine presentará propiedades como "*haDirigidoLaPelicula*"). Si la profesión obtenida (Ej: Científico Austríaco) no tiene correspondencia en la DBPedia se intenta mapear a una clase Yago. Si este mapeo es posible, al ser Yago un sistema de clasificación jerárquico, es posible inferir información adicional de su clase padre. Ej: Científico Austríaco es una subclase de Científico, por lo que la persona en cuestión es un "*Científico*" que quizás sí que tenga un mapeo directo a la DBPedia.
3) La información referente a una persona puede estar organizada desde diferentes enfoques: vida personal, vida profesional, aspectos sociales, etc. Inspirándose en la forma que tiene Google de mostrar información sobre una persona, se han identificado varios conjuntos de hechos (capa de **Aspectos Inferidos**) que podrían ser mostrados para cada persona: social, profesional, personal y lado oscuro (que engloba hechos negativos acerca de la persona en cuestión).

El sistema (Ilustración 15) se divide en cuatro módulos: **Manager de consultas**, que traduce las consultas de términos del usuario en consultas SPARQL y desencadena la acción del **módulo de sugerencias** que a partir de la consulta SPARQL generada en el paso anterior, recupera resultados de la DBPedia y ofrece al usuario los términos encontrados para que seleccione uno de ellos. El **Módulo de Recuperación de Información** se encarga de obtener la URI de la DBPedia correspondiente al término buscado. Si no es posible, se realiza una consulta al servicio web de Sindice (Índice de Web Semántica) para obtener la URI. Posteriormente se

desreferencia la URI para obtener la descripción RDF del recurso. Finalmente el proceso de agregación de conceptos seleccionará lo que se mostrará como resultado al usuario. A partir del **Módulo de Búsqueda de Propiedades**, el usuario buscará entre las diferentes propiedades de un recurso recuperado por el Módulo de Recuperación de Información. Cuando un usuario introduce un término de búsqueda, se consultará a WordNet para recuperar el synset correspondiente al término introducido. Esto permitirá presentar al usuario las propiedades que encajen con dicho synset.

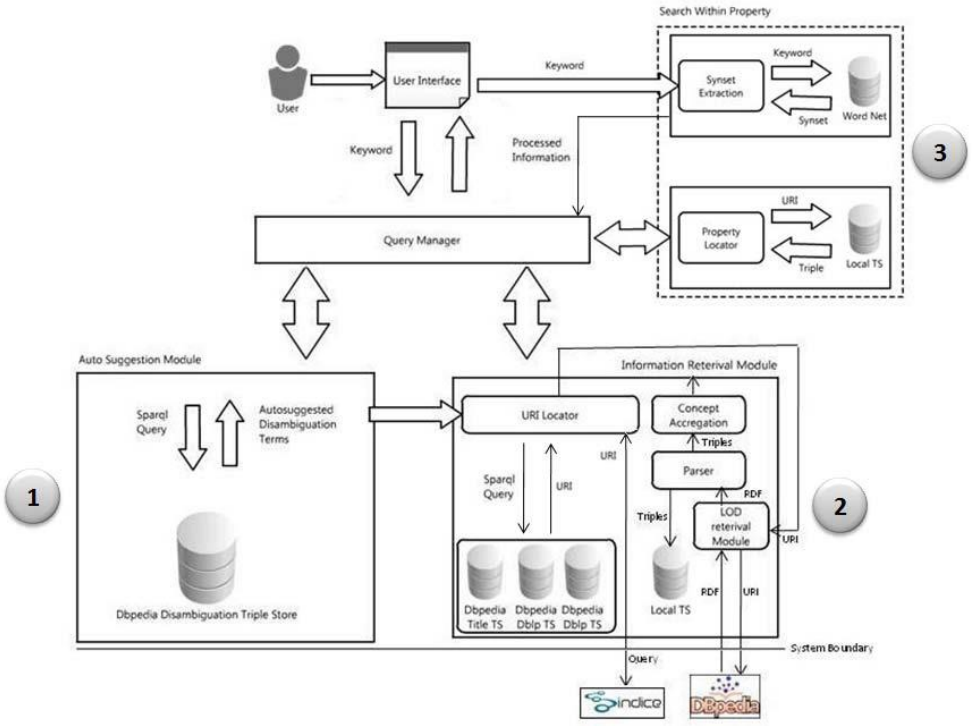



Ilustración 15: Arquitectura del sistema CAF-SIAL

### 1.3.2 Dispositivos móviles

Existen también trabajos que aplican los principios de Linked Data en aplicaciones móviles. En RailGB [13] se utilizan las técnicas LOD en una aplicación orientada a teléfonos móviles para proporcionar información sobre elementos de accesibilidad en el metro de Londres a personas con discapacidad. La información se personaliza dependiendo del grado de discapacidad de cada usuario y su localización. Este aspecto como no es un objetivo de este TFM, simplemente se identifica como de interés.

### 1.3.3 Interfaces de usuario basadas en HTML5

La llegada del HTML5 proporciona a los desarrolladores de aplicaciones web un nuevo abanico de posibilidades. En el campo del contenido multimedia, HTML5 ofrece un soporte nativo de diversos formatos de video y la posibilidad de acceder mediante lenguaje de script tanto a los distintos elementos multimedia (audio, videos…) como a sus subtítulos, controlar su reproducción, responder a sus eventos de reproducción, etc.

Con HTML5 los elementos multimedia dejan de ser contenido externo que debe ser reproducido con programas adicionales (*plugins*) para ser elementos reconocidos por el propio navegador. Gracias al nuevo formato de subtítulos WebVTT de HTML5 incluso es posible incluir anotaciones o metadatos en dichos subtítulos. Esta característica comienza a ser utilizada en algunos proyectos, por ejemplo en *Chrooma+* [2] se enriquece contenido audiovisual a partir de datos provenientes de los servicios web de Google Maps, Twitter y Wikipedia, tal y como se ha investigado para realizar el trabajo descrito en la Parte II de esta memoria de TFM.

El contenido multimedia es una parte fundamental de la Web actual. Plataformas audiovisuales como *Youtube* o *Vimeo* se cuentan entre los sitios web más utilizados hoy en día. Sin embargo en este tipo de plataformas el contenido audiovisual se muestra de forma aislada. El enfoque de *Chrooma+* aboga por mejorar la experiencia de usuario en la reproducción de contenido multimedia integrando dicho contenido con información adicional proveniente de la Web: texto, imágenes, videos, mapas… En este enfoque es clave la utilización de las nuevas características que proporciona HTML5 y el nuevo formato de anotación WebVTT que permite mayor flexibilidad y extensibilidad en las anotaciones de metadatos.

Las nuevas características de HTML5 facilitan la tarea de mostrar contenido multimedia enriquecido. En concreto, la etiqueta *<track>* permite la inclusión explícita de subtítulos y anotaciones sincronizadas con la reproducción de un video. El atributo "*kind*" de dicho elemento establecido al valor "*metadata*" permite especificar que dichas anotaciones no sean mostradas por los navegadores para no interferir con los subtítulos de un video, pero sí serán accesibles por componentes JavaScript. Dado que se pretende mostrar las anotaciones en momentos específicos de tiempo el elemento *<track>* proporciona un evento JavaScript (*oncuechange*) que se desencadena cuando se inicia o finaliza un nuevo subtítulo. La captura de dicho evento permite mostrar las anotaciones en los momentos temporales deseados.

```
<video id="demo" controls >
        <source src="demovideo.webm" type="video/webm">
        <track src="metadata.vtt" label="Metadata" kind="metadata" default>
</video>
```

Los navegadores actuales no soportan ningún formato de video que contenga metadatos incorporados. Es por ello que el enfoque *Chrooma+* utiliza un formato de anotación externo (WebVTT) combinado con formatos de video estándar soportados por los navegadores. WebVTT permite incorporar los subtítulos de un video y de forma sincronizada incluir anotaciones en formato JSON de una forma muy flexible. Actualmente las anotaciones se incluyen manual mente aunque el equipo de *Chrooma+* está trabajando en una herramienta de anotación automática.

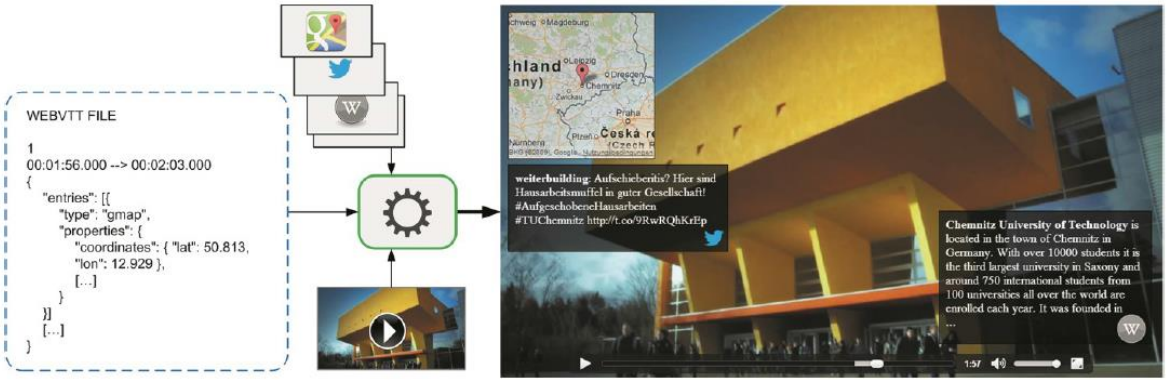



**Ilustración 16: Visión general de la arquitectura y mashup de video resultante.**

El fichero WebVTT contiene varios segmentos (cues) que incluyen un identificador, una marca temporal y el cuerpo del segmento que puede incluir cualquier tipo de información en formato JSON. Utilizando jQuery [1] es posible implementar un script que capture el evento "*oncuechange*" para mostrar de forma sincronizada las anotaciones correspondientes a cada momento del video. Además es posible utilizar diversas API de servicios web para enriquecer la información. El equipo de *Chrooma+* ha desarrollado una demo[2] que mediante los servicios web de Google Maps, Twitter y la Wikipedia muestra mapas de localizaciones, Tweets que contienen un *hashtag* concreto especificado en las anotaciones o descripciones de diversos conceptos recuperados de la Wikipedia.

El resultado final es una experiencia de usuario atractiva y sorprendente que muestra las posibilidades de esta tecnología. En este trabajo fin de master se ha utilizado este mismo mecanismo para embeber anotaciones y metadatos en ficheros WebVTT aunque para hacer que el resultado final esté más orientado a la Web Semántica y las técnicas LOD la información enriquecida se ha recuperado mediante procesos de enriquecimiento basados en consultas SPARQL.

[1] www.jquery.com
[2] http://chroomaplus.eu/www2013-demonstrator

## 1.4 ¿Cómo gestionar la ambigüedad en procesos de enriquecimiento de información?

Otra de las cuestiones críticas que influyen en la calidad de un proceso de enriquecimiento basado en LOD, es la asignación de URIs a las entidades detectadas en un texto. Lo más habitual es utilizar las URIs de la DBPedia como identificadores universales de los conceptos detectados.

Esta función también puede ser realizada por un NER obteniéndose el identificador de la entidad en el mismo momento de su detección, aunque otros trabajos [10] utilizan un buscador semántico como Sindice [11]. Otra opción es la utilización del servicio de búsqueda de la DBPedia: *DBPedia Lookup Service(DLS)* que a partir de un texto de búsqueda devuelve los términos de la DBpedia a los que puede referirse dicho texto. *PoolParty* [20] es una herramienta de gestión de tesauros que utiliza esta técnica para asociar conceptos del tesauro con conceptos de la DBPedia. Esto le permite enriquecer el tesauro con información relevante proveniente de la nube LOD para mantener la información actualizada.

*TagMe!* [21] es una aplicación de etiquetado social de imágenes de Flickr que también utiliza el DLS: permite a los usuarios asociar etiquetas a zonas específicas de una imagen y categorizar dichas etiquetas. *TagMe!* utiliza el DLS para mapear etiquetas y categorías a URIs de la DBPedia para definir claramente el significado de los nombres de etiquetas seleccionados mejorando el rendimiento de las búsquedas realizadas con dichas etiquetas.

A la hora de asignar URIs a las entidades detectadas en un texto surge el problema de la desambiguación. A menudo existen varias URIs que tengan relación con una entidad dada por lo que existe una ambigüedad acerca de que URI asignar a un término a partir de una lista de candidatos. La BBC [6] resuelve esta cuestión mediante un clustering de similitud basado en el contexto de los conceptos o entidades. Por ejemplo, el término "*apple*" es muy ambiguo, pero dado el contexto de términos "*microsoft*" y "google", el significado de "*apple*" refiriéndose a la empresa Apple parece evidente. Esta suposición es la base del algoritmo de entrelazado de términos empleado por la BBC, que utiliza la información de clasificación para construir clusters de similaridad (conceptos de la misma categoría) que permiten la desambiguación del significado de un término.

DBPedia Spotlight[1] [22] es una herramienta capaz de anotar menciones a fuentes de la DBPedia en un texto. Permite, por tanto, el entrelazado de información no estructurada con la nube LOD a través de la DBPedia. Para ello realiza tareas de extracción de entidades nombradas y desambiguación. Un proceso de etiquetado automático de conceptos está formado por dos fases: el reconocimiento de las sentencias a etiquetar (reconocimiento de frases) y la búsqueda de identificadores no ambiguos que describan el significado de las sentencias (desambiguación). El etiquetado de conceptos se puede realizar utilizando diferentes focos de anotación, es decir, qué segmentos del texto deben ser anotados. Diferentes problemas de lingüística computacional como Desambiguación del Sentido de las Palabras (WSD), Reconocimiento de Entidades Nombradas (NER) o Extracción de palabras

[1] http://dbpedia-spotlight.github.io/demo/

clave, establecen diferentes focos de anotación. DBPedia Spotlight es capaz de adaptar el estilo de anotación en tiempo de ejecución lo que permite al usuario influenciar el proceso de etiquetado seleccionando distintas estrategias en la identificación de frases. Los resultados del etiquetado de conceptos están muy influenciados por la estrategia de reconocimiento de sentencias utilizada [23]. DBPedia Spotlight constituye, por tanto, un sistema de etiquetado de conceptos de propósito general adaptable a lo que cada usuario considera que es un término. Además mientras la mayoría de sistemas de anotación restringen su funcionamiento a un reducido número de tipos de entidades (persona, organización, lugar,…) DBPedia Spotlight es capaz de anotar entidades de cualquiera de las 320 clases definidas en la ontología de la DBPedia [24].

Si el número de elementos a resolver no es muy grande y es conocido de antemano es posible combinar la utilización de Sindice y el Servicio de Resolución de Correferencia *sameAs.org*[1] con un trabajo manual consistente en asignar las URIs resolviendo la desambiguación a partir de los resultados devueltos por ambos sistemas, este fue el enfoque elegido el *RailGB* [13] que se describe brevemente a continuación.

### 1.4.1 Gestión de URIs de información adicional

En *RailGB* los usuarios pueden buscar las estaciones de metro cercanas a un punto concreto de Londres filtrando aquellas que disponen de determinadas infraestructuras: ascensores, escaleras mecánicas, puntos de ayuda etc. *RailGB* mostrará las estaciones en un mapa así como la disponibilidad de las diferentes infraestructuras. La descripción e imagen de cada estación se recupera del endpoint SPARQL de la DBPedia.


Ilustración 17: Capturas de pantalla de RailGB

A partir de una consulta, el **Agente de Consultas** realiza consultas SPARQL basadas en los requisitos de búsqueda de cada usuario y el módulo de **Mezcla de Datos** presenta la información en un interfaz web móvil. Los datos que utiliza *RailGB* son los siguientes:

- Disponibilidad de las distintas infraestructuras de accesibilidad en cada estación de metro.
- Localización física de cada estación en términos de longitud y latitud.
- Nombres de lugares de interés de Londres y su localización

[1] www.sameAs.org

Para el desarrollo de este proyecto necesitaron convertir la información disponible a datos estructurados y republicarlos mediante un endpoint SPARQL. Se ha utilizado además el Servicio de Resolución de Correferencias (CRS) sameAs.org[1] para conectar adecuadamente el nuevo dataset con la DBPedia. Al no existir una ontología adecuada para describir las infraestructuras de las estaciones de metro fue necesario crear la ontología Tube Facility[2] (TF) de forma que es posible identificar por ejemplo una estación mediante una URI del tipo *http://m.railgb.org.uk/id/tube/stationid*. El dataset resultante se incluye como un endpoint SPARQL[3].

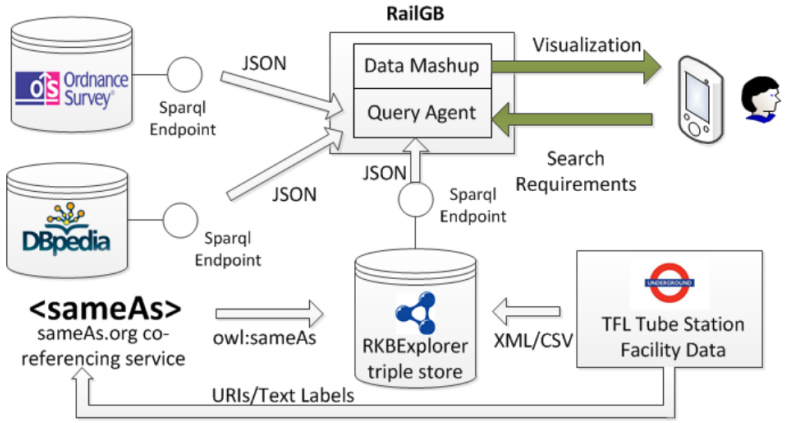



Ilustración 18: Arquitectura de RailGB

Para obtener información adicional de cada estación se enlazó el dataset de RailGB con la nube LOD asociando cada estación con su URI correspondiente en la DBPedia mediante enlaces de equivalencia (*owl:sameAs*). Para esta tarea se utilizó el servicio sameAs.org y Sindice. SameAs.org es capaz de descubrir URIs correferentes de la DBPedia, Freebase…, aunque fue necesario un trabajo manual para realizar tareas de desambiguación y encontrar la mejor URI de cada estación.

Para evaluar el funcionamiento del sistema se hicieron pruebas con usuarios reales en Londres. Las pruebas fueron satisfactorias ya que el sistema detectó las estaciones deseadas con unos tiempos de respuesta aceptables. Se detectó que la principal debilidad del sistema es la exactitud de los datos utilizados sobre las infraestructuras debido a la utilización de datos no actualizados.

RailGB es una aplicación basada en LOD que ayuda a personas con discapacidad a localizar las estaciones de metro que disponen de determinadas infraestructuras en Londres, siendo de interés su constatación de la gran dificultad que existe para encontrar fuentes de datos públicas de calidad que contengan datos de accesibilidad de una forma estructurada.

[1] http://www.sameas.org
[2] http://www.railgb.org.uk/ns/2012/9/tubefacility.owl
[3] http://oad.rkbexplorer.com/sparql

### 1.4.2 Localización de URIs basada en Sindice

La Web Semántica puede ser vista como una inmensa base de conocimiento formada por fuentes de datos que proporcionan información en forma de ficheros RDF o a través de endpoints SPARQL. En la Web Semántica no existe una única base de datos de información sino que cualquiera puede contribuir haciendo disponibles sus fuentes de datos en un espacio web público y al utilizar identificadores (URIs) y términos compartidos, su información puede ser agregada en nuevos servicios útiles. El problema que aparece es cómo encontrar de forma automática información relevante sobre un cierto recurso o entidad, dado el elevado número de colecciones de datos existentes.

Sindice[1] [11] es un servicio online que sirve como respuesta a dicha cuestión, se encarga de rastrear la Web Semántica e indexar los recursos encontrados en cada fuente de datos. Para ello recupera documentos RDF de la Web de Datos y los indexa como URIs, IFPs (Propiedades funcionales inversas) y palabras clave. OWL define una propiedad funcional inversa como un tipo de predicado que relaciona un sujeto y un objeto de forma biunívoca de forma que no existe ningún otro sujeto que se relacione con ese mismo objeto por esa misma propiedad. Ej: "Barack Obama (sujeto) es el actual presidente (predicado) de los EEUU (objeto).

Aunque existe un interfaz web para probar la funcionalidad de Sindice, su principal aportación es una API dirigida a los desarrolladores de aplicaciones de Web Semántica para localizar datos relevantes. La Ilustración 20 muestra los resultados al realizar una consulta con los términos "*Tim Berners-Lee*". La API de Sindice devuelve la misma información en formatos más adecuados para su procesamiento automático: RDF, XML, JSON.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<rdf:RDF xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">

<rdf:Description rdf:about="http://www.w3.org/People/Berners-Lee/card#i">
  <rdfs:seeAlso rdf:resource='http://www.w3.org/People/Berners-Lee/card'/>
  <rdfs:seeAlso rdf:resource='http://danbri.org/foaf.rdf'/>
  <rdfs:seeAlso rdf:resource='http://heddley.com/edd/foaf.rdf'/>
  <rdfs:seeAlso rdf:resource='http://www.eyaloren.org/foaf.rdf'/>
  <rdfs:seeAlso rdf:resource='http://people.w3.org/simon/foaf'/>
  <rdfs:seeAlso rdf:resource='http://www.ivan-herman.net/foaf.rdf'/>
</rdf:Description>

</rdf:RDF>
```

Ilustración 19: Ejemplo de resultados devueltos por Sindice

Sindice actúa únicamente como localizador de recursos RDF devolviendo enlaces a fuentes de datos remotas pero no actúa como motor de consultas. Sindice está más cerca conceptualmente a un motor de búsqueda Web tradicional adaptado a conceptos de Web Semántica que a otros motores de búsqueda de Web Semántica como SWSE[2] o Swoogle[3]. Al proporcionar únicamente enlaces a las fuentes de los recursos, Sindice evita gran parte de aspectos que los motores de Web Semántica deben afrontar: confianza, políticas de

[1] www.sindice.com
[2] http://www.swse.org
[3] http://swoogle.umbc.edu

consolidación de entidades globales, denegaciones de servicio por consultas con excesiva complejidad o excesivos datos, etc.

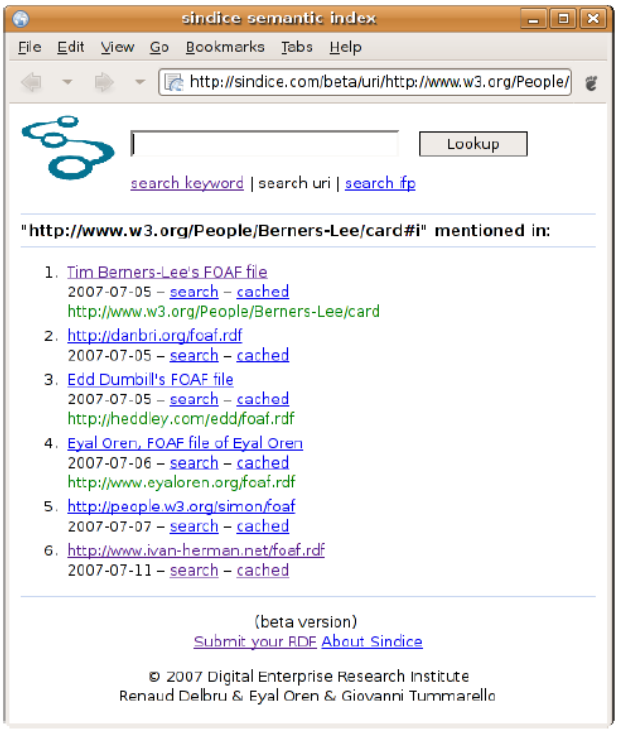



**Ilustración 20: Resultados devueltos por Sindice para la consulta "Tim Berners-Lee"**

El diseño de Sindice permite a las aplicaciones dar al usuario un control total sobre las fuentes de datos través de una API para interactuar con la Web Semántica. Sindice ofrece tres servicios básicos a aplicaciones clientes:

1) Análisis sintáctico de ficheros y endpoints SPARQL cuando se realizan los rastreos o cuando se le llama de forma explícita.
2) Búsqueda de recursos (identificados por su URI o por una combinación de una propiedad funcional-inversa y un valor identificativo) que devuelve URLs o documentos RDF donde esos recursos o entidades aparecen.
3) Búsqueda de descripciones de texto completo que devuelve las URLs de las fuentes donde esos recursos aparecen.

Para cumplir con estos requisitos, la API de Sindice consta de cuatro métodos:

- *Index(url) =>nil*: Parsea e indexa un documento o un endpoint SPARQL dada una url
- *lookup(uri) => url[]*: Dada una URI, busca un recurso y devuelve una lista de fuentes donde ese recurso aparece.
- *lookup(ifp, valor) => url[]*: Busca un recurso identificado unívocamente por una pareja propiedad-valor y devuelve una lista de fuentes donde ese recurso aparece.
- *lookup(texto) => url[]*: Busca un texto y devuelve una lista de fuentes donde ese recurso aparece.

Adicionalmente existen tres requisitos no funcionales: Minimización del tamaño del índice, minimización de los tiempos de búsqueda para que pueda ser utilizado por las aplicaciones cliente y permitir continuas actualizaciones del índice para mantenerlo actualizado.

Sindice está formado por varios componentes independientes que funcionan de forma paralela para lograr el rastreo, la indexación y la ejecución de consultas.

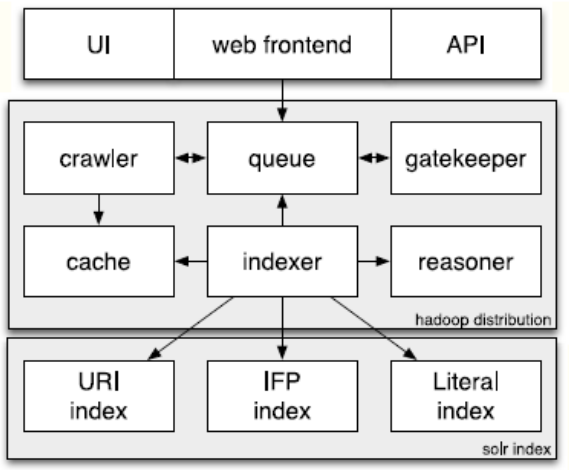



Ilustración 21: Arquitectura de Sindice

1) El frontal Web es el punto de entrada de la aplicación y se compone de un interfaz de usuario (UI) para el acceso humano y una API HTTP para el acceso de las aplicaciones.

2) El *crawler* recopila datos RDF de la Web y los añade a la cola de indexación (*queue*). Si recibe un "*ping*" (llamada explicita) de un usuario o aplicación para analizar nuevos documentos, estos se añaden también a la cola. El Guardián (*Gatekeeper*) evalúa cada entrada de la cola y decide si indexa cada documento y con qué prioridad. El indexador extrae URIs, IFPs y palabras clave de cada documento (utilizando el Razonador para la extracción IFP) y los añade a su índice respectivo. Durante una búsqueda los componentes de interfaz solo necesitan enviar la consulta al índice adecuado, obtener los resultados y generar la salida con el formato deseado.

3) Los tres índices almacenan apariciones de URIs, IFPs y literales en documentos RDF. El índice de URIs contiene una entrada por cada por cada recurso URI y lista las URLs de los documentos donde dicha URI aparece. El índice IFP es similar excepto que en vez de utilizar URIs explícitas se utilizan parejas propiedad-valor como índice clave apuntando también a una lista de documentos. Este índice permite la búsqueda de recursos con diferentes URIs que realmente identifican la misma entidad del mundo real. El índice de literales contiene una entrada por cada *token* apuntando de nuevo a una lista de documentos.

4) Dado que el único patrón de acceso consiste en "*dado un recurso devolver las fuentes donde aparece*", se ha modelado el sistema como un índice inverso de apariciones de URIs en documentos. El proceso de indexación de Sindice realiza las siguientes tareas:

**a.** **Planificar**: El servicio de indexación recibe un feed RSS con la localización del grafo RDF a indexar. La url del grafo se envía al Planificador que actúa como un recolector de URLs esperando a ser procesadas para evitar la sobrecarga de los servidores web. Solo se revisitan fuentes de datos cuando se supera un umbral de tiempo de espera y sólo se analiza la respuesta de la fuente si el código hash del contenido ha cambiado.

b. **Extracción de grafos**: Las URLs del planificador pueden ser recursos RDF o endpoints de SPARQL. En el primer caso se obtiene el fichero y se envía al analizador sintáctico (*parser*). En el segundo caso se envía a la base de datos las consultas necesarias para obtener su contenido completo.

c. **Análisis sintáctico de grafos** (*Graphparsing*): El *parser* recibe el grafo RDF y verifica su validez. Después extrae las URIs y las envía al planificador siguiendo los principios de Linked Data para tratar cada URI como un hiperenlace a más información.

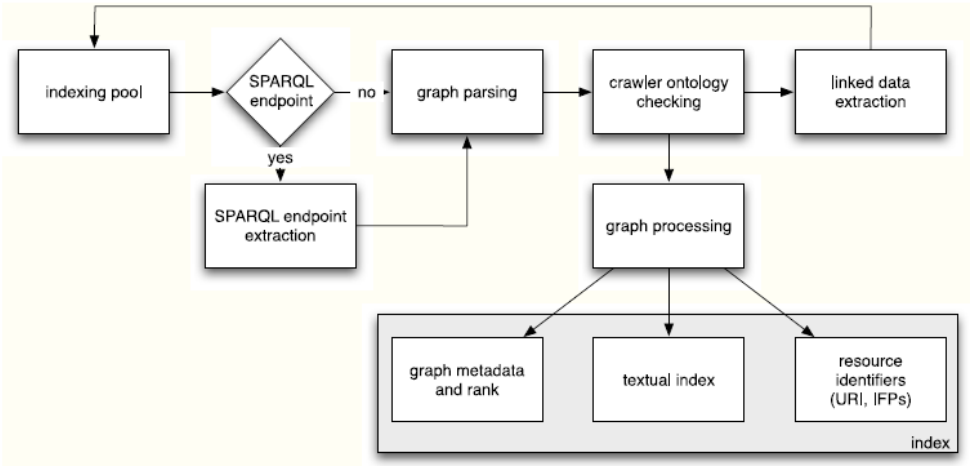



Ilustración 22: Proceso de indexación de Sindice

d. **Procesamiento de grafos**: El procesador de grafos extrae e indexa el texto y los identificadores de recursos del grafo. También extrae los identificadores indirectos en forma de parejas propiedad-valor para todas las propiedades funcionales inversas mencionadas en el grafo. La extracción IFP requiere dos pasos adicionales: la expansión recursiva del grafo con la información de su esquema y la realización de un proceso de inferencia OWL para identificar los IFPs existentes en estos esquemas.

Un grafo no menciona de forma explícita cuales de sus propiedades son funcionales inversas sino que el grafo se referirá a una o varias definiciones de esquemas que de forma recursiva dependerán de otros esquemas. Sindice obtiene la definición de cada propiedad, desrreferencia su URI e importa la definición obtenida en el grafo RDF. Esta recuperación se repite de forma recursiva. Posteriormente se realiza un proceso de inferencia basándose en OWL para encontrar propiedades funcionales inversas.

5) Durante la ejecución de las consultas se recuperan los datos del índice y se realiza un ranking o evaluación de los resultados obtenidos, para ordenarlos por su relevancia de acuerdo a varias métricas. Para ello, para cada fuente se realiza un cálculo medio de los valores de los metadatos siguientes:

- *Nombre de host*: Se prefieren fuentes cuyo nombre de host coincida con el del recurso.

  Por ejemplo se considera que para el recurso http://eyaloren.org/foaf.rdf#me la fuente http://eyaloren.org/foaf.rdf es más importante que una fuente externa como http://g1o.net/g1ofoaf.rdf .

- *Ranking externo*: Se prefieren fuentes de datos almacenadas en sitios con un ranking alto usando algoritmos de ranking web clásicos.

- *Fuentes relevantes*: Se prefieren fuentes que comparten términos poco habituales (URIs, IFPs, palabras clave) en vez de términos comunes que coincidan con los términos solicitados. Esta métrica de relevancia es comparable a TF/IDF en el campo de la recuperación de información.

### 1.4.3 Entrelazado de conjuntos de datos OD

Las tecnologías actuales para construir el Linked Open Data (LOD) se han materializado a día de hoy, en una Web de Datos abierta formada por varios billones de tripletes RDF. Entre los problemas no resueltos aún se encuentra el problema de que diferentes conjuntos LOD son unidades de información débilmente conectadas. Los enlaces entre diferentes conjuntos de datos existen casi exclusivamente a nivel de instancias (usando *owl:sameAs*) y la información a nivel de esquema, es decir, las taxonomías construidas usando *rdfs:subClassOf* (o enriquecidas con más axiomas de RDF o OWL que no involucren datos de instancias) es a menudo bastante pobre. Esto es, existe una carencia de enlaces entre los diferentes esquemas.

BLOOMS [9], un sistema para encontrar enlaces a nivel de esquema entre conjuntos de datos LOD para el alineamiento de ontologías se basa en la idea de enlazar información presente en la nube LOD con la ayuda de datos generados por la comunidad y disponibles en la Wikipedia. Utiliza su sistema jerárquico de categorías o clases, que no es una taxonomía ya que muchas relaciones de tipo “subcategoría” no son semánticamente relaciones de tipo *rdfs:subClassOf.*

Por lo tanto BLOOMS es un sistema para el alineamiento de ontologías es decir, se encarga de la generación de enlaces entre diferentes jerarquías de clases que se representaran como relaciones de tipo *rdfs:subClassOf*. Por ejemplo si “Humano” aparece en un conjunto de datos y “Mujer” aparece en otro distinto, se espera que BLOOMS (o cualquier otro sistema de alineamiento de ontologías) cree una relación entre estas dos clases en forma de triplete RDF: “Mujer rdfs:subClassOf Humano”. Dos clases A y B siempre estarán relacionadas por una las relaciones: A *rdfs:subClassOf* B, B *rdfs:subClassOf* A, A *owl:equivalentClass* B, o ninguna de las tres anteriores.

El funcionamiento de BLOOMS se basa en el sistema jerárquico de categorías de la Wikipedia, y construye un bosque (o conjunto de árboles) $T_c$ (llamado bosque BLOOMS para C) para cada candidato a asociarse con la clase C, que se corresponde con una selección de supercategorías del nombre de la clase. La comparación de los bosques $T_C$ y $T_B$ para emparejar clases candidatas C y B, devuelve la decisión de si C y B deben ser alineados y con cuál de las relaciones candidatas.

BLOOMS toma como candidatas a dos ontologías, para las que se asume que contienen información de esquema, y realiza los 4 pasos siguientes. Para ilustrar estos pasos con un ejemplo usaremos los nombres de clases "*Event*" "*JazzFestival*"tomados de los datasets LOD de DBPedia y Music Ontology respectivamente.

1) **Pre-proceso de las ontologías de entrada,** para eliminar restricciones de propiedades, individuos y separa nombres de clases compuestos, para obtener una lista de todas las palabras simples contenidas en ellas eliminando stopwords.

   Esta normalización de cada nombre de clase de entrada C, obtiene otro nombre C,' donde se reemplazan los caracteres de subrayado ("_") y guiones por espacios en blanco y se separan las palabras por los caracteres en letra mayúscula.

   En el ejemplo "*JazzFestival*" se transforma en "*Jazz Festival*" y "*Event*" permanece inalterado.

2) **Construcción del bosque BLOOMS $T_C$** por cada clase *C* usando información de la Wikipedia. Para ello se llama al servicio web de la Wikipedia usando *C'* como entrada. El servicio devuelve un conjunto de páginas de la Wikipedia $W_C$ que contiene los resultados de búsqueda en la Wikipedia de las palabras de esa cadena. Si el resultado devuelto es una página de desambiguación de la Wikipedia se elimina de $W_C$ y se reemplaza por todas las páginas de la Wikipedia mencionadas en la página de desambiguación. Denominaremos a cada uno de los elementos de $W_C$ significados para C.
   En el ejemplo para "*Event*" el web service nos devuelve "*Eventing*", "*Sport*", "*NFL Draft*", "*News*", "*Festival*", "*Event-driven programming*", "*Rodeo*", "*Athletics at the Summer Olympics*" y "*Extincion event*".

   Por cada significado *s* de $W_C$ se construye un árbol $T_S$ llamado el árbol BLOOMS para C con el significado s de la siguiente forma:

   - La ruta del árbol es *s*
   - Los hijos de *s* son todas las categorías de la Wikipedia en la cual se categoriza la página s.
   - Para cada categoría *c* que es un nodo del árbol, sus hijos son todas las categorías de la Wikipedia de las cuales *c* es una subcategoría.
   - $T_S$ es el árbol resultante que es cortado a nivel 4 (es decir que como máximo tendrá 5 nodos) para que no se incluyan categorías demasiado generales que no sería útiles para nuestros propósitos.

   La figura Ilustración 23 muestra los arboles BLOOMS para "*Event*" y "*JazzFestival*" con el significado "*Event*" y "*Jazz Festival*".

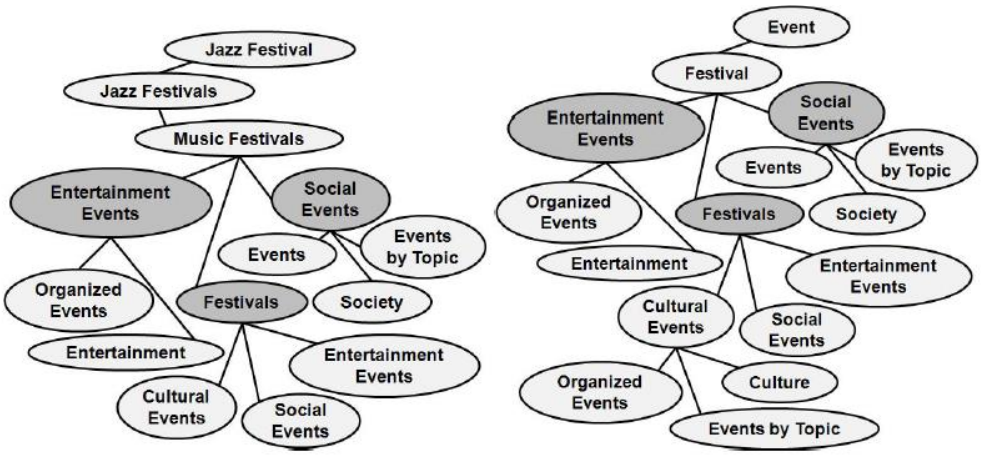



Ilustración 23: Árboles BLOOMS para JazzFestival y Event

3) **Análisis de los bosques BLOOMS construidos,** para decidir qué clases deben ser alineadas. Cualquier concepto *C* en una de las ontologías de entrada es emparejada contra cualquier concepto *D* de la otra ontología de entrada comparando cada $T_S \in T_C$ con cada $T_T \in T_D$. Se define una función o que asigna un número real a cada pareja de árboles BLOOMS. El valor $o(T_S.T_T)$ o solapamiento de esos dos árboles se calcula de la siguiente forma:

   *a)* Se eliminan de $T_S$ todos los nodos para los cuales existe un nodo padre que aparezca en $T_T$ obteniendo como resultado el árbol $T'_S$. Se eliminan porque no proporcionan información adicional al comparar dos árboles.
   *b)* $o(T_S.T_T) = n/k-1$ donde *n* es el número de nodos en $T'_S$ que aparecen también en $T_T$ y *k* es el número total de nodos en $T'_S$ sin contar el nodo raíz.

   En el ejemplo, los árboles de la Ilustración 23 se recortan bajo los nodos grises resultando: $o(T_{Event}, T_{Jazz\ Festival}) = 3/4$ y $o(T_{Jazz\ Festival}, T_{Event}) = 3/5$. Las decisiones de alineamiento que se toman son:

   - Si para cualquier $T_S \in T_C$ y $T_t \in T_D$ se obtiene que $T_S = T_T$ entonces se establece que C es equivalente a D: "C *owl:equivalentClass* D".
   - Si el mínimo { $o(T_S, T_T)$, $o(T_T, T_S)$} >= *x* para cualquier elección de $T_S$ y $T_T$ y para un umbral predefinido *x,* entonces se establece que C es una subclase de D: "C *rdfs:subClassOf* D".

   En el ejemplo, se concluye por tanto que $o(T_{Event}, T_{Jazz\ Festival}) > o(T_{Jazz\ Festival}, T_{Event})$ y por tanto "Jazz Festival rdfs:subClassOf Event".

4) **Post-procesamiento** de los resultados con la ayuda de una API de alineamiento y un razonador. Primero se invoca la API de alineamiento para encontrar alineamientos entre los valores originales de entrada que se añaden a los resultados obtenidos. De esta forma el razonador (basado en Jena) que encuentre que "*A es una subclase de B*" en una de las ontologías de entrada y que existe un alineamiento "*B rdfs:subClassOf C*" puede concluir que "*A rdfs:subClassOf C*".

Se realizó la evaluación de BLOOMS usando datasets de terceros y otros sistemas de alineamiento de ontologías. Esta evaluación se hizo de dos formas diferentes:

- Se examinó la capacidad de BLOOMS para servir como sistema de alineamiento de ontologías de propósito general y se comparó con otros sistemas similares (AROMA, RiMOM, S-Match) utilizando benchmarks de la Iniciativa de Evaluación de Alineamiento de Ontologías (OAEI). Los test realizados mostraron que BLOOMS mejora los resultados de la mayoría del resto de sistemas mostrando un rendimiento muy parejo con el mejor de los sistemas: RiMoM.
- Se evaluó BLOOMS para la tarea de integración de esquemas LOD y se comparó con otros sistemas similares. Al no existir benchmarks disponibles para las medidas de precisión y cobertura en alineamiento de esquemas LOD se realizó una evaluación manual llevada a cabo por expertos humanos. Los tests mostraron que BLOOMS se comporta significativamente mejor (40% mejor cobertura y al menos el doble de precisión) que el resto de sistemas a la hora de alinear ontologías de la nube LOD.

## 1.5 ¿Cómo evaluar la calidad del contenido enriquecido?

La evaluación de los resultados de un proceso de enriquecimiento de datos, sin usuarios, es un aspecto aún por resolver. Generalmente los trabajos pasan por alto este proceso de evaluación, aunque otros como [3] ofrecen al menos un estudio con el cálculo de diversas variables para mostrar que efectivamente el proceso de enriquecimiento se ha realizado correctamente.

En otros trabajos como [13] [9] se utilizan expertos humanos para llevar a cabo la evaluación. En [9] además se realiza una evaluación bastante exhaustiva de los resultados en el caso de alineamiento de ontologías, donde los resultados obtenidos se comparan con los de otros sistemas.

Como se verá en la parte segunda de esta memoria, el análisis de la viabilidad de la propuesta y la calidad del enriquecimiento de entidades del experimento realizado, se realiza en función de los resultados obtenidos respecto al número de entidades identificadas y enriquecidas con o sin errores provocados por no encontrar el identificador URI de la entidad o confundirlo, como resultado de una ambigüedad no resuelta.

## 1.6 ¿Qué aporta el proyecto Linked Open Data?

Este estado del arte se completa con una visión general del concepto de datos entrelazados ( [25] y [26]). Desde un punto de vista técnico, el término Linked Data se refiere a la publicación de datos en la Web de forma legible por máquinas, con su significado definido de forma explícita y con enlaces desde y hacia otros sistemas externos. De la misma forma que la unidad mínima de la Web hipertexto son los documentos HTML conectados por enlaces no tipados, Linked Data se fundamenta en documentos RDF que contienen sentencias tipadas que enlazan conceptos muy variados. El resultado es lo que se denomina la Web de Datos.

Berners-Lee acuñó un conjunto de *Principios de Linked Data*, para la publicación de datos en la Web: Utilizar URIs para nombrar los conceptos, usar URIs HTTP para que sea posible buscar y encontrar dichos conceptos, proporcionar información útil usando los estándares: RDF y SPARQL, cuando alguien solicite una URI, e incluir enlaces a otras URIs para que sea posible descubrir nueva información.

El ejemplo más visible de adopción y aplicación de los principios de Linked Data ha sido el proyecto *Linking Open Data* que trabaja identificando conjuntos de datos existentes que estén disponibles bajo licencias abiertas, convirtiendo dichos datasets a RDF de acuerdo a los principios de Linked Data y publicándolos en la Web.

El proyecto inicial ha salido de los laboratorios de investigación y ha crecido considerablemente ya que se han involucrado en el mismo grandes organizaciones que le han dado un fuerte impulso, fomentado por la naturaleza abierta del proyecto donde cualquiera puede participar simplemente al publicar un conjunto de datos de acuerdo a los principios de Linked Data y entrelazarlo con conjuntos de datos existentes.

Para la publicación de datos en la Web de acuerdo a los principios de Linked Data, deben seguirse los siguientes pasos:

1) **Elegir URIs y Vocabularios RDF**: Los proveedores de datos pueden utilizar dos tipos de URIs HTTP para identificar las diferentes entidades: URIs 303 que identifican entidades del mundo real o URIs que identifican un documento web que describen la entidad del mundo real. En un entorno abierto como la Web diferentes proveedores de información publican datos sobre una misma entidad (una persona, una localización,…). Cada uno de ellos puede utilizar diferentes URIs para identificar el mismo concepto. Por ejemplo la DBPedia utiliza la URI http://dbpedia.org/resource/Berlin para identificar Berlín, mientras Geonames la identifica como http://sws.geonames.org/2950159. Como ambas URIs se refieren a la misma entidad del mundo real esas URIs son alias y son muy habituales en la Web de Datos. Los alias de URIs permiten desrreferenciar diferentes descripciones o vistas de la misma entidad del mundo real. Es muy común que los proveedores incluyan enlaces de tipo *owl:sameAs* hacia alias de URIs que conozcan.

   Debido a su carácter abierto, la Web de Datos permite la utilización de cualquier vocabulario para representar la información, aunque se considera una buena práctica reutilizar vocabularios RDF conocidos como FOAF, SIOC, SKOS, Dublin Core, etc. El uso de

estos vocabularios facilitará a las aplicaciones cliente el procesamiento de los datos enlazados.

Un formato de serialización para Linked Data muy común es RDF/XML. Es común además la utilización Notation3 y su subconjunto Turtle cuando es necesaria la inspección de los datos por humanos debido a su mayor legibilidad. Otra posibilidad es serializar los datos en formato RDFa que permite la inclusión de tripletes RDF en HTML.

2) **Generación de enlaces RDF a otras fuentes de datos**: Los enlaces RDF permiten a las aplicaciones navegar entre diferentes fuentes de datos y descubrir datos adicionales. Es una práctica común la utilización de mecanismos automáticos o semiautomáticos para generar los enlaces RDF. En algunos dominios existen esquemas de nombrado aceptados universalmente (ISBN en el dominio de la publicaciones literarias, ISIN en el dominio financiero,…). Si los datasets a enlazar soportan alguno de estos esquemas generalmente aceptados es posible hacer explícitas las relaciones entre elementos mediante enlaces RDF de manera sencilla.

   Si no existe ningún esquema común de nombrado, los enlaces RDF pueden ser generados a partir de la similaridad de entidades entre datasets utilizando técnicas de detección de duplicados y encaje de ontologías. Existen diversos entornos de generación enlaces RDF que proporcionan lenguajes declarativos para especificar qué tipo de enlaces RDF deben ser creados y qué combinación de métricas de similaridad deben utilizarse para comparar entidades.

3) **Proporcionar Metadatos sobre los datos publicados**: La presencia de metadatos incrementa la utilidad de los datos de cara a los consumidores. La existencia de metadatos con información acerca de su creador, fecha de creación o forma de creación permiten calibrar la calidad de los datos y su origen. Para ello es posible proporcionar una información básica usando los términos de Dublin Core o el vocabulario de Publicación de Web Semántica.

Los temas de investigación de mayor interés actualmente pueden organizarse como sigue:

1. **Arquitectura de aplicaciones**: Los datos enlazados pueden ser obtenidos a través de un rastreo y cacheo previo o en el mismo momento de la ejecución mediante la recuperación de enlaces y consultas federadas. Motores de búsqueda como SWSE, Sindice o Falcons rastrean la Web de Datos y proporcionan a las aplicaciones acceso mediante APIs a los datos rastreados. Las arquitecturas de consultas federadas para datos enlazados, han demostrado que consultas complejas pueden ser resueltas a través de la recuperación de enlaces en tiempo de ejecución.
   a. **Mapeo de esquemas.** Una vez que los datos han sido recuperados de fuentes distribuidas deben ser integrados de una forma coherente antes de ser presentados al usuario. Para ello se requiere de un mapeo de términos de diferentes vocabularios al esquema de datos de la aplicación final además del fusionado de datos sobre la misma entidad recuperados desde diferentes fuentes.

Las fuentes de Linked Data suelen utilizar un esquema propio o una mezcla de términos a partir de vocabularios existentes y bien conocidos. Además pueden incluir correspondencias entre su propia terminología y la terminología utilizada por fuentes de datos relacionadas. Las recomendaciones del W3C *RDF Schema* y *OWL* definen una terminología básica como por ejemplo: *owl:equivalentClass, owl:equivalentProperty, rdfs:subClassOf, rdfs:subPropertyOf*para representar las correspondencias más simples. Sin embargo los lenguajes de mapeo de esquemas son necesarios para que permitan publicar relaciones más complejas.

b. **Fusión de datos** en aplicaciones de Linked Data. La fusión de datos consiste en la integración de múltiples elementos que representan el mismo elemento del mundo real en una representación única y consistente, solucionando la incertidumbre de los metadatos de calidad, necesarios para resolver inconsistencias.
c. **Mantenimiento de enlaces.** El contenido de las fuentes de datos enlazados cambia con la inserción, modificación o borrado de datos. Se hace necesario por tanto un mantenimiento de los mismos. Si no se actualizan convenientemente pueden existir numerosos enlaces rotos a URIs que ya no existen entre otros muchos problemas. Para solventar este inconveniente se han propuesto, sistemas de re-cálculo de enlaces a intervalos regulares, o nuevas fuentes de datos que publiquen las actualizaciones de enlaces o modelos de suscripción a registros centrales para obtener información de las actualizaciones de enlaces.
d. **Licencias de acceso.** Iniciativas como las de *CreativeCommons*[1] han proporcionado un marco para el licenciamiento abierto de trabajos creativos basados en la noción de derecho de copia (copyright). Sin embargo, la ley de derecho de copia no es aplicable a datos ya que tienen una perspectiva legal distinta.
   Marcos como el Open Data *CommonsPublicDomainDedication* and License[2] deberían ser adoptados por la comunidad para clarificar las cosas en este área.
e. **Confianza, calidad y relevancia.** Técnicas heurísticas basadas en un sistema de voto o equivalentes del algoritmo *PageRank* deberían ser utilizadas para obtener medidas de popularidad de una determinada fuente de datos.

Los enlaces actuales entre conjuntos de datos en la nube LOD son demasiado superficiales para llevar a cabo la mayoría de los beneficios prometidos. Si esta limitación no se mejora la nube LOD corre el riesgo de convertirse en simples datos que sufran del mismo tipo de problemas que abundan en la Web de Documentos. Luego la nube LOD debe ser transformada desde "*algo más que datos*" a "*datos enlazados semánticamente*" siguiendo las siguientes pautas:

## 2. Carencia de una Descripción Conceptual de los Conjuntos de Datos

La utilización de los conjuntos de datos LOD requiere que un ser humano identifique el dominio de los datos. Por ejemplo, no existe actualmente un mecanismo que describa que Jamendo[3] contiene información sobre música mientras que GeoNames[1] almacena información

[1] http://creativecommons.org
[2] http://opendatacommons.org/licenses/pddl/1.0
[3] http://dbtune.org/jamendo

geográfica. Este es un serio inconveniente para la creación de aplicaciones que puedan reutilizar de forma eficiente la ingente cantidad de conocimiento presente en la nube. Aunque se han llevado a cabo algunos esfuerzos en esta materia, se han centrado más en aspectos estadísticos de los conjuntos de datos, que en proporcionar los requerimientos para obtener información semántica. Una descripción conceptual ayudará a que el proceso de descubrimiento de conocimiento sea más fácil y sistemático.

**3. Ausencia de Enlaces a Nivel de Esquema**

Los conjuntos de datos de la nube LOD carecen de mapeos a nivel de esquema y no incluyen relaciones entre conceptos de diferentes conjuntos de datos a nivel de esquema. Por ejemplo, una característica en el esquema de Geonames puede servir como una sede de un evento: el modelo actual identifica "*Atlanta en Georgia fue la sede de las Olimpiadas de 1996*" a nivel de instancia. Esto crea limitaciones significativas con respecto al potencial de razonamiento que puede proporcionar el conocimiento a nivel de esquema.

**4. Carencia de expresividad**

La nube LOD posee una expresividad muy pobre como base de conocimiento por lo que es difícil hacer uso de su semántica formal subyacente mediante procesos de razonamiento. La nueve LOD consiste fundamentalmente en tripletes RDF sencillos y no utiliza las ricas expresiones proporcionadas por OWL o RDF Schema. Por ejemplo, existe una inconsistencia entre DBPedia y Geonames relativa a la población de Barcelona. Esta inconsistencia podría detectarse (y por tanto solucionarse) declarando las propiedades *dbpedia-owl:populationTotal* y *geonames:population* como propiedades funcionales. Ya que las instancias de Barcelona en ambos sistemas están enlazadas por una propiedad *owl:sameAs*, utilizando un razonador OWL sería posible detectar dicha inconsistencia ya que una instancia no puede tener múltiples valores para una propiedad funcional. La carencia de dichas características expresivas es un gran inconveniente para la utilización de la nube LOD por parte de la comunidad científica.

**5. Dificultades para realizar consultas**

SPARQL se ha convertido en el lenguaje de consultas de-facto para la Web Semántica. Las entidades u organizaciones responsables de los datos los publican en forma de extremos (endpoints) SPARQL. Sin embargo, la sintaxis de SPARQL requiere que los usuarios especifiquen detalles precisos de la estructura del grafo que va a ser consultado. Por ejemplo, para hacer una consulta que englobe varios conjuntos de datos como "*Seleccionar artistas de Jamendo que publicaron al menos un álbum etiquetado como 'punk' por un usuario de Jamendo, ordenado por el número de habitantes de los lugares de donde han nacido*", el usuario debe estar familiarizado con múltiples datasets y tiene que expresar de forma precisa las relaciones entre conceptos en el patrón de tripletes RDF, para lo cual, incluso en escenarios triviales, implica navegar por al menos dos o tres datasets.

Se han identificado los siguientes problemas o desafíos con respecto a la consulta sistemática de fuentes LOD:

[1] http://www.geonames.org

a. **Heterogeneidad de esquemas**: Los datasets de LOD se dirigen a diferentes dominios y por tanto han sido modelados de forma distinta. Por ejemplo, un usuario interesado en información relacionada con música debe consultar al menos 3 conjuntos de datos distintos: Jamendo, MusicBrainz y MySpace. Esto es perfectamente válido desde una perspectiva de ingeniería del conocimiento pero hace que las consultas sean más difíciles ya que los usuarios deben entender los distintos esquemas. Este problema es un derivado de la carencia de una descripción conceptual de los datasets.
b. **Desambiguación de entidades**: Los datasets de LOD a menudo presentan solapamientos entre dominios por lo que presentan información sobre la misma entidad. Por ejemplo, tanto DBPedia como GeoNames tienen información acerca de Barcelona. Aunque DBPedia referencia a GeoNames usando la propiedad *owl:sameAs*, es confuso para un usuario saber cuál es la mejor fuente para realizar una consulta. Este problema es más grave cuando se devuelven resultados contradictorios para la misma entidad desde distintos datasets. Por ejemplo, la DBPedia asegura que la población de Barcelona es de 1.615.908 mientras que para GeoNames es de 1.581.595. Se podría decir que esto es debido a diferentes nociones de la ciudad Barcelona en ambos sistemas, pero incluso así esto nos lleva a una pregunta interesante: ¿Está la propiedad *owl:sameAs* siendo mal utilizada en la nube LOD? Este problema está relacionado con la mencionada carencia de expresividad ya que no existen mecanismos para realizar verificaciones de hechos. Adicionalmente, la metodología LOD prohíbe la reificación de sentencias (es decir, tratar las sentencias como si fueran una entidad más) descartando la posibilidad de asignar un contexto a las sentencias. Varios investigadores han reconocido la severidad de esta prohibición y han propuesto técnicas para resolverla pero no está claro cómo afectará esto a la nube LOD.
c. **Ranking de resultados**: En escenarios donde los resultados de las consultas pueden ser calculados por múltiples conjuntos de datos la cuestión de cómo evaluar dichos resultados no está muy clara. El ejemplo sobre calcular *la población de Barcelona* puede ser realizado por múltiples datasets como DBPedia o GeoNames, pero cuál de ellos elegir como fuente más relevante para un escenario específico es una cuestión aún no resuelta. Este problema ha sido analizado bajo la perspectiva de la popularidad de los datasets considerando la cardinalidad y tipos de las relaciones pero no desde la perspectiva de requisitos respecto a una consulta específica.

Algunos de los problemas de la nube LOD pueden ser resueltos proporcionando una descripción formal y sistemática de la misma. Existe una carencia de una ontología que formalice la información contenida en los datasets de LOD. Esa ontología debería proporcionar descripciones sistemáticas de los dominios modelados por los datasets, enlazado a nivel de esquema de los conjuntos de datos, axiomas adicionales a nivel de esquema y por tanto mejores capacidades de razonamiento. Normalmente, tal integración debería hacer uso de una ontología de alto nivel. De hecho, en el pasado, la comunidad de Web Semántica se ha basado en ontologías de alto nivel como Cyc, SUMO o DOLCE para integrar bases de conocimiento heterogéneas. Utilizar una ontología de alto nivel produciría los siguientes beneficios:

- **Descripción formal y sistemática de los datasets LOD**, para crear un puente entre las abstracciones de las ontologías disponibles en la nube. Como resultado se crearía una descripción semántica de la nube que facilitaría la realización de razonamientos. La

Ilustración 24 refleja una posible integración de SUMO con la nube LOD. Esta cuestión ha sido tratada por otros investigadores se y han realizado trabajos para utilizar otra conocida ontología de alto nivel como Cyc para proporcionar una columna vertebral a la nube LOD mediante UMBEL. UMBEL contiene enlaces a nivel de esquema de 21 datasets LOD diferentes.

- **Facilidad de consultas**

Una ontología integrada de alto nivel facilitaría la realización de consultas ya que las ramas específicas de la ontología de alto nivel estarían enlazadas a la nube LOD por lo que el usuario sabría en qué secciones de la nube debería buscar. Además sería posible diseñar mecanismos automáticos que propagasen las consultas por la nube. Por ejemplo, si un usuario realizara una consulta SPARQL en términos de conceptos de la ontología de alto nivel, dichos mecanismos permitirían que la consulta se propagase por los diferentes datasets.

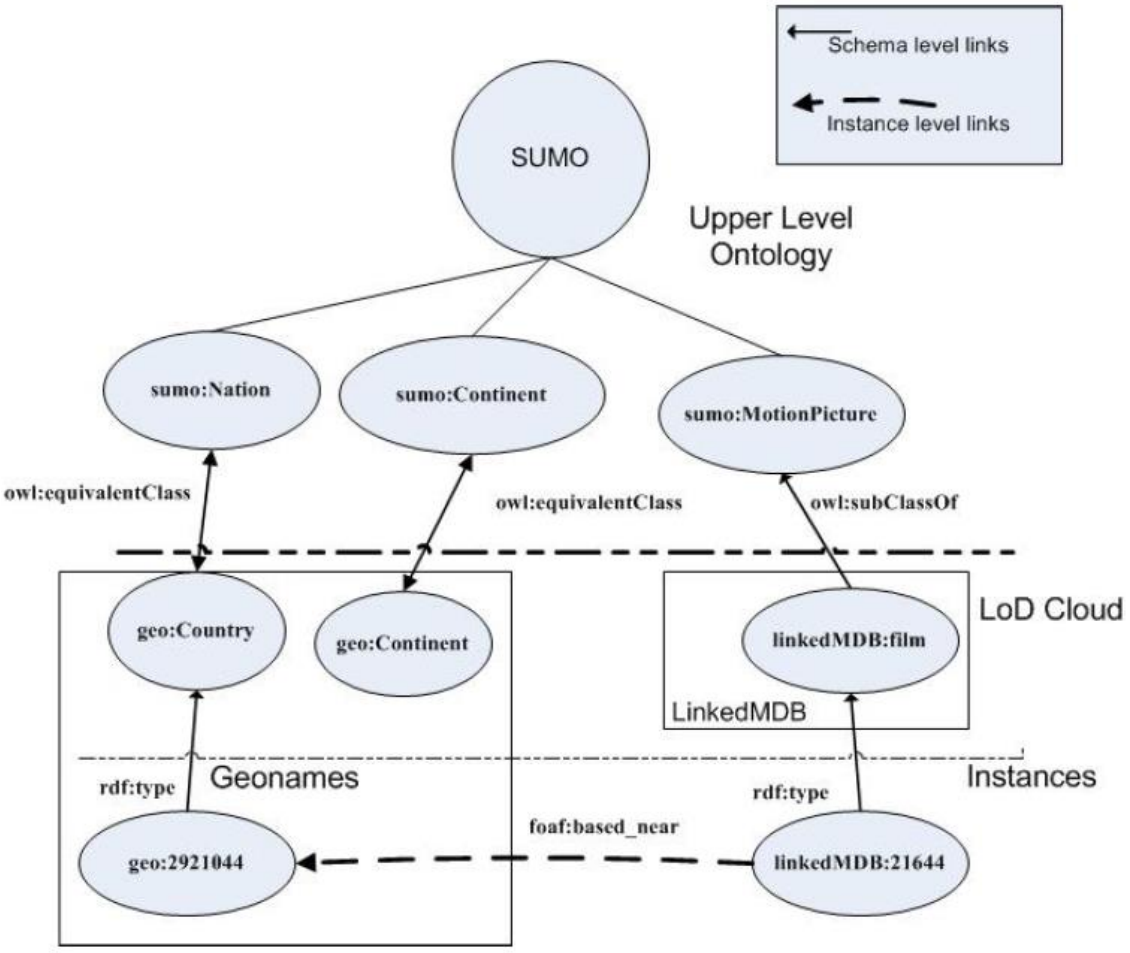



Ilustración 24: Posible Integración en LOD y SUMO

- **Comprobación de inconsistencias en la nube LOD**

Una ontología de alto nivel con axiomas puede ayudar en la detección de inconsistencias. Esto ayudaría en la verificación de información proveniente de LOD y permitiría identificar y filtrar dichas inconsistencias.

- **Facilidad de Mantenimiento y Extensibilidad**

Debido a que la nube LOD continúa creciendo y englobará más dominios en el futuro, debería ser más fácil de mantener para permitir modificaciones y soportar extensibilidad para incluir conceptos que no se soportan de forma nativa por la ontología.

Aunque se podría argumentar que el intento de enriquecer la nube LOD con más conocimiento a nivel de esquema sería un fracaso debido a la dificultad de tratar cantidades ingentes conocimiento por los razonadores de ontologías, creemos que esto no es necesariamente cierto. Recientes avances mostrados en los Billion Triple Challenges de las Conferencias Internacionales de Web Semántica [1] muestran que el razonamiento sobre bases de conocimiento de gran tamaño es posible.

[1] http://challenge.semanticweb.org

# Capítulo 2. Panorama tecnológico

La Web Semántica [27] es una extensión de la "Web de Documentos" a la que se incorpora información mejor definida y con más semántica para formar una verdadera "Web de Datos", entendible tanto por usuarios como por máquinas. Impulsada por el *World Wide Web Consortium* (W3C)[1] intenta dotar a la Web de más significado y más semántica para permitir obtener soluciones a problemas habituales en la búsqueda de información utilizando una infraestructura común.

A continuación se detallan algunos aspectos técnicos relacionados con los datos abiertos (Open Data, OD) y sus principios básicos, las recomendaciones sobre el entrelazado de datos (Linked Data, LD) y su actividad en España, los estándares relacionados con la representación de información y conocimiento (y las ontologías) que son relevantes para el LOD (Linked Open Data), los recursos léxicos y finalmente sobre los formatos de video y subtítulos.

## 2.1 Datos Abiertos (Open Data)

*Open Data* (OD) es un movimiento que promueve la liberación de datos, textuales o no y en formatos reutilizables. *Linked Data* se refiere a la exposición de esos datos en la web, descritos en RDF (*ResourceDescription Framework*) y con indicación de la relación que puede existir entre esos datos y otros. Los datos liberados pueden ser de cualquier temática: geográficos, meteorológicos, científicos, de tráfico…, y fue el gobierno británico quien comenzó a hacerlo con su iniciativa "*Opening up Government*"[2] en 2010 [28].

Los datos abiertos son aquellos que pueden ser utilizados, reutilizados y redistribuidos libremente por cualquier persona. Los datos abiertos y especialmente los datos abiertos gubernamentales constituyen un recurso inmensamente valioso con un valor potencial de utilización incalculable.

Existen muchas áreas donde se espera que los datos abiertos sean valiosos para diferentes grupos de personas y organizaciones, incluido el mismo gobierno. Al mismo tiempo es imposible predecir con exactitud cómo y donde será creado el valor en el futuro. Por la propia naturaleza de una innovación, su desarrollo proviene de los lugares más inverosímiles. Algunos de los beneficios que se pueden obtener con Datos Abiertos son (extracto de http://opendatahandbook.org/es/why-open-data/index.html):

- Transparencia y control democrático
- Productos privados y servicios nuevos o mejorados
- Innovación
- Mayor eficiencia de los servicios públicos
- Mayor eficacia de los servicios públicos
- Medición del impacto de políticas

[1] http://www.w3.org
[2] http://data.gov.uk

- Nuevos conocimientos a partir de fuentes de datos combinadas y patrones en grandes volúmenes de datos

Varios estudios estimaron el valor económico de los Datos Abiertos en varias decenas de billones de Euros al año, sólo en la Unión Europea. Existen ejemplos en la mayoría de las áreas anteriores:

- Proyectos como el finlandés "*taxtree*" y el británico "*Wheredoesmymoneygoes?*" muestran cómo el Gobierno está gastando el dinero de sus impuestos. Varios sitios web, como el danés www.folketsting.dk, siguen las actividades en el Parlamento y el proceso de formulación de leyes para que se pueda ver exactamente qué está pasando y que Miembros del Parlamento están involucrados.

- En los Países Bajos está disponible un servicio que te avisa con un mensaje si la calidad del aire en tus inmediaciones alcanzará, al día siguiente, el umbral que previamente definiste. En Nueva York puedes saber fácilmente dónde puedes pasear a tu perro, así como también encontrar otras personas que usan los mismos parques. Todos estos ejemplos usan Datos Abiertos.

- El sitio danés husetsweb.dk te ayuda a encontrar maneras de mejorar la eficiencia energética en tu casa, incluyendo planificación financiera e información sobre constructores que puedan hacer el trabajo. Está basado en información catastral y sobre subsidios gubernamentales, así como el registro de comercio local.

- El Traductor de Google usa el enorme volumen de documentos de la Unión Europea que aparecen en todos los idiomas europeos para entrenar sus algoritmos de traducción y así mejorar la calidad de su servicio.

- El Ministro de Educación holandés publicó en internet todos los datos relacionados con educación, para su reutilización. Desde entonces, el número de preguntas que reciben bajó, reduciendo el volumen de trabajo y los costos, y las preguntas restantes resultan más fáciles de responder porque es claro dónde puede encontrarse la información importante.

Este potencial exige que los datos sean verdaderamente abiertos. Toda restricción excluirá a personas de la posibilidad de reutilizar los datos públicos y hará más difícil que se encuentren maneras valiosas de hacerlo.

### 2.1.1 Principios básicos de Open Data

La idea de la iniciativa Open Data es que gran parte de la información no personal que poseen los gobiernos debe compartirse de forma libre para ser reutilizada de forma pública bajo las siguientes condiciones:[1]

[1] extracto de http://observatorio.cenatic.es/index.php?option=com_content&view=article&id=688:diez-consejos-open-data&catid=54:tecnologia&Itemid=62

- **Completitud**: Todo dato público debe ser accesible. Los datos públicos son datos no sometidos a límites derivados de protección de la intimidad personal, de seguridad u otros derechos preferentes.
- **Originalidad**: Los datos puestos a disposición son los obtenidos de las fuentes primarias, con el mayor nivel posible de granularidad, sin que se ofrezcan agregados o modificados.
- **Premura**: Los datos deben ponerse a disposición pública tan pronto como sea posible para preservar el valor de los mismos.
- **Accesibilidad**: Los datos han de ser accesibles para los mayores rangos de usuarios y propósitos.
- **Tratamiento automatizado**: Los datos deberán estar razonablemente estructurados para permitir su tratamiento automatizado.
- **No discriminación**: Los datos estarán accesibles universalmente, sin necesidad de registro para acceder a los mismos.
- **Formato no propietario**: Los datos han de hallarse accesibles en un formato del que ninguna entidad tenga un control exclusivo.
- **Licencia libre**: Los datos no se hallarán sujetos a derechos de autor, propiedad industrial o secreto comercial. Se podrá admitir una razonable limitación por cuestiones de intimidad personal, seguridad o derechos preferentes.
- **Revisables**: Cada institución o entidad que publique sus datos tendrá que contar con una persona designada para responder a las dudas o reclamaciones sobre dichos recursos, para lo cual su contacto deberá figurar entre la información ofrecida.
- **Reconocible:** La información se publicará de tal forma que cualquier persona pueda localizarla sin problemas; estará ubicada, para ello, en los catálogos de datos más apropiados y en sitios online accesibles a los buscadores de Internet.
- **Permanentes:** Habrá de garantizarse el archivo de los datos, con el paso del tiempo, según todos los criterios anteriores.

## 2.2 Datos Entrelazados (Linked Data)

Linked Data o Datos Enlazados es un término utilizado para describir una serie de recomendaciones sobre cómo publicar y conectar fuentes de datos estructuradas en la Web Semántica usando URIs y RDF. Estas buenas prácticas han sido adoptadas por un número creciente de proveedores de datos en los últimos años llegando a la creación de un espacio global de datos: La web de Datos.

La principal diferencia entre la web del hipertexto y la web semántica es que mientras la primera vincula páginas y documentos en html, la segunda aboga por ir más allá del concepto de documento y enlaza datos estructurados[1]. En 2006 Tim Berners-Lee definió cuatro reglas para la publicación de Linked Data:

1. Utilizar URIs (*Uniform Resource Identifiers*) identificando los recursos de forma unívoca.

[1] http://www.w3.org/DesignIssues/LinkedData.html

2. Utilizar URIs http para que la gente pueda acceder a la información del recurso.

3. Ofrecer información sobre los recursos usando los estándares (RDF y SPARQL)

4. Incluir enlaces a otros URIs, facilitando el vínculo entre distintos datos distribuidos en la web.

Estos principios están definidos como reglas, pero en realidad son más bien recomendaciones o buenas prácticas para el desarrollo de la web semántica. Es posible publicar datos que cumplan sólo los tres primeros principios, pero el hecho de no aplicar el cuarto los convierte en menos visibles y, como consecuencia, menos reutilizables.

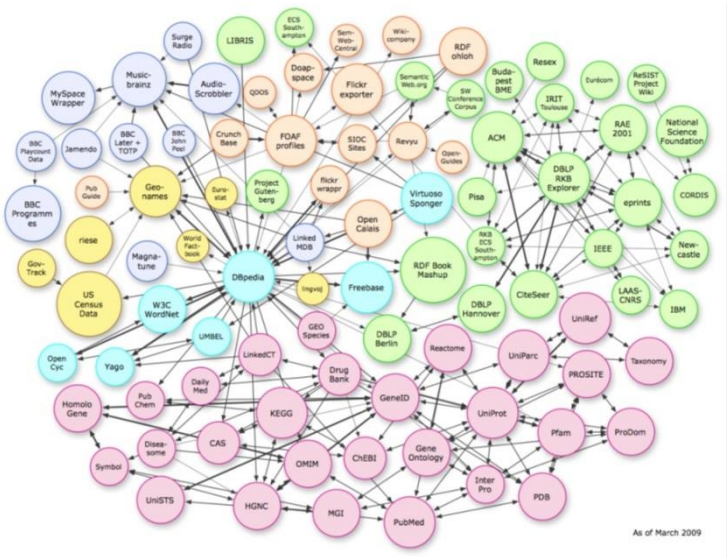



Ilustración 25: Diagrama de Datos Enlazados de Marzo 2009

Los recursos disponibles de Open Data están experimentando un gran crecimiento en diferentes sectores. Estados Unidos y el Reino Unido son pioneros es este campo tras sus iniciativas de *Open Government* (Gobierno Abierto) donde abogan por un fuerte impulso por la transparencia de datos públicos. En el caso de Estados Unidos, el proceso está siendo impulsado directamente por su presidente, Barack Obama, a través de su Iniciativa para el Gobierno Abierto[1]. Reino Unido comenzó su estrategia de apertura de la mano de Tim Berners-Lee, el creador de la Web.

Para conocer el avance de las fuentes de datos en todo el mundo podemos consultar el proyecto europeo *Public Sector Information*[2] (PSI) que recopila los paquetes de datos que liberan las administraciones. España aprueba en 2007 la Ley 37/2007, sobre reutilización de la información del sector público que regula y fomenta la reutilización de los datos elaborados o custodiados por las Administraciones y organismos del sector público. En el año 2009 nace el *Proyecto Aporta*[3] que persigue tres objetivos básicos:

[1] http://www.whitehouse.gov/open
[2] http://epsiplatform.eu
[3] http://www.aporta.es

- Fomentar una cultura favorable a la apertura de datos públicos
- Facilitar que las administraciones acometan dicha apertura
- Impulsar el mercado de la reutilización de la información pública.

En marzo de 2010, por impulso de los ministerios de Industria, Energía y Turismo y de Hacienda y Administraciones Públicas, Proyecto Aporta pone en marcha el Catálogo de Información Pública[1], el primer acceso único de España a las fuentes de datos disponibles del sector público estatal. En 2011 se lanza la iniciativa datos.gob.es, que toma el relevo de Proyecto Aporta y asume su compromiso con la apertura de la información del sector público.

Aunque España esté aún muy lejos de tener un Gobierno Abierto, a nivel autonómico hay diversas iniciativas pioneras, entre las que destaca la iniciativa vasca Open Data Euskadi[2]. Es importante también el esfuerzo de la Fundación CTIC[3] que lidera la iniciativa de *eGovernment* en el W3C y que ha colaborado en la implementación de las iniciativas de Open Data de País Vasco, Cataluña, Asturias, Andalucía…

Se puede decir por tanto que se están realizando esfuerzos en la apertura de datos y que existe una cierta concienciación sobre este tema. Se echa en falta sin embargo un mayor esfuerzo en la calidad de los datos publicados sobre todo en lo referente a utilización de estándares y seguimiento de buenas prácticas.

[1] http://datos.gob.es/datos/?q=catalogo
[2] http://opendata.euskadi.net
[3] http://datos.fundacionctic.org

## 2.3 Representación de la información

Para que la Web Semántica sea posible, son necesarios diversos metalenguajes y estándares de representación que permiten la especificación formal de la información, fomentando su descubrimiento y reutilización.

### 2.3.1 Estándares

#### 1. Xml (eXtensible Markup Language)

XML[1] o Lenguaje de Marcado Extensible aporta la sintaxis básica para los documentos estructurados pero sin dotarles de ninguna restricción sobre el significado. Se ha convertido en un estándar para el intercambio de información estructurada entre diferentes plataformas. Un ejemplo de documento XML sería:

```
<?xml version="1.0"?>
<note>
    <to>Tove</to>
    <from>Jani</from>
    <heading>Reminder</heading>
    <body>Don't forget me this weekend!</body>
</note>
```

Ejemplo 1: XML

#### 2. Json (JavaScript Object Notation)

Json es un formato ligero para el intercambio de datos. Su simplicidad ha propiciado la generalización de su uso, especialmente como alternativa a XML en tecnologías AJAX. Una de las ventajas de JSON sobre XML como formato de intercambio de datos en este contexto, es que es mucho más sencillo escribir un analizador sintáctico (parser) de JSON.

```
{
  "note": {
    "to": "Tove",
    "from": "Jani",
    "heading": "Reminder",
    "body": "Don't forget me this weekend!"
  }
}
```

Ejemplo 2: Json

#### 3. Xml Schema

XML Schema[2] es un lenguaje de esquema utilizado para describir la estructura y las restricciones de los contenidos de los documentos XML de una forma muy precisa, más allá de las normas sintácticas impuestas por el propio lenguaje XML. Estas restricciones se expresan utilizando reglas gramáticas que deciden el orden de los elementos, predicados booleanos que el contenido debe satisfacer, tipos de datos que a los que deben ajustarse el contenido de los

[1] http://www.w3.org/XML
[2] http://www.w3.org/XML/Schema

elementos y sus atributos y reglas más especializadas como restricciones de unicidad e integridad referencial. Un ejemplo de esquema XML sería el siguiente:

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="note">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="to" type="xs:string"/>
      <xs:element name="from" type="xs:string"/>
      <xs:element name="heading" type="xs:string"/>
      <xs:element name="body" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:schema>
```

**Ejemplo 3: XML Schema**

## 4. RDF (Resource Description Framework)

El Marco de Descripción de Recursos o RDF[1] es un modelo de datos para describir recursos y las relaciones que se pueden establecer entre ellos. Ha sido elaborado por el W3C y se basa en la idea de modelar un dominio declarando recursos usando expresiones con la forma: sujeto-predicado-objeto. Esta expresión es conocida en la terminología RDF como triplete. Un triplete RDF contiene tres componentes, todos con referencia en un URI:

- Sujeto, una referencia URI, una persona, o un nodo, es el ente al cual nos referimos.
- Predicado es la propiedad o relación que se desea establecer acerca del sujeto.
- Objeto es el valor de la propiedad o del otro recurso con el que se establece la relación.

Los recursos se identifican en RDF de forma inequívoca mediante una URI. El predicado de una sentencia RDF también se identifica mediante una URI ya que indica un recurso que representa a la relación. En la Ilustración 26 (tomada de la web del W3C[2]), podemos ver el grafo RDF que representa a la persona llamada Eric Miller. En él se identifica a la persona Eric Miller con la URI http://www.w3.org/People/EM/contact#me. La propiedad "correo electrónico" se identifica con la URI http://www.w3.org/2000/10/swap/pim/contact#mailbox .

Por tanto el siguiente triplete indica que el email de Eric Miller es em@w3.org:

```
http://www.w3.org/People/EM/contact#me, http://www.w3.org/2000/10/swap/pim/contact#mailbox,
em@w3.org
```

RDF es un estándar simple pero extremadamente efectivo, su serialización es adecuada tanto para procesamiento por parte de una máquina como para procesos de razonamiento automático. A pesar de su simplicidad, la mayoría de modelos de datos pueden ser mapeados a RDF. El hecho de utilizar URIs para enlazar los datos convierte la web semántica en una especie de gran base de datos que permite que las personas y las máquinas puedan explorar la

[1] http://www.w3.org/RDF
[2] http://www.w3.org/TR/rdf-primer

información referenciada e interconectada entre sí en la Web, lo que al mismo tiempo fomenta su crecimiento.

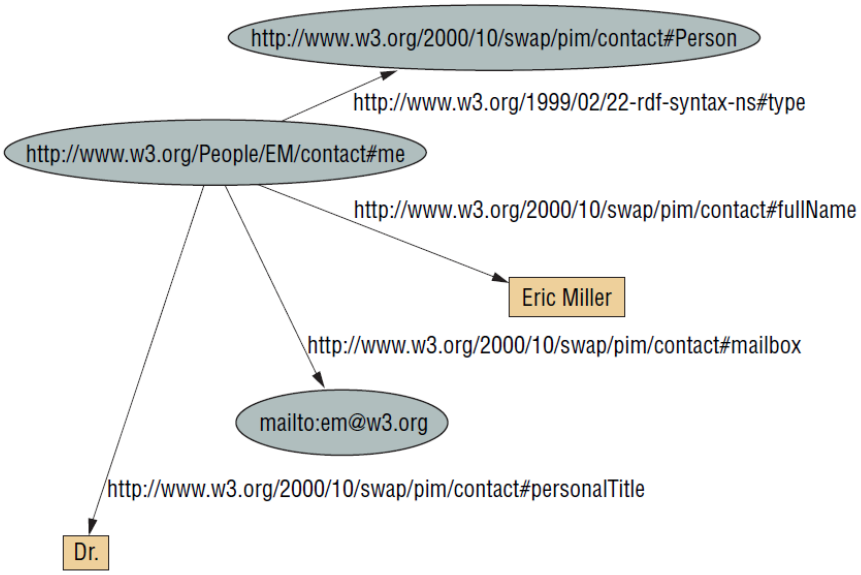



**Ilustración 26: Grafo RDF que representa a Eric Miller**

El vocabulario RDF incluye entre otras las siguientes clases y propiedades:

Clases RDF:

- *rdf:Property*: clase de las propiedades
- *rdf:Statement*: clase de las sentencias RDF
- *rdf:List*: clase de las listas RDF

Propiedades RDF:

- *rdf:type*: instancia de *rdf:Property* para indicar que un recurso es una instancia de una clase.
- *rdf:subject*: sujeto de una sentencia RDF
- *rdf:predicate*: predicado de una sentencia RDF
- *rdf:object*: objeto de una sentencia RDF

RDF es muy adecuado para trabajar con información distribuida ya que es sencillo juntar diversos archivos RDF publicados por diferentes autores en Internet y deducir nuevo conocimiento a partir de ellos. Esto es posible porque RDF permite enlazar documentos que utilizan vocabularios comunes y a la vez permite que cualquier documento utilice cualquier vocabulario. Como resultado RDF proporciona una gran flexibilidad para representar conocimiento sobre cualquier materia.

Debe recordarse que RDF no es un formato de datos sino un modelo de datos para describir recursos con la forma de tripletes *sujeto-predicado-objeto*. Para publicar un grafo RDF en la Web debe ser serializado usando una sintaxis RDF: RDF/XML o RDFa.

## 5. RDF/XML

RDF/XML[1] es una sintaxis definida por el W3C para representar grafos RDF basándose en XML. Como ejemplo, se muestra a continuación el grafo de la Ilustración 26 en su representación RDF/XML:

```
<?xml version="1.0"?>
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:contact="http://www.w3.org/2000/10/swap/pim/contact#">

<contact:Person rdf:about="http://www.w3.org/People/EM/contact#me">
  <contact:fullName>Eric Miller</contact:fullName>
  <contact:mailbox rdf:resource="mailto:em@w3.org"/>
  <contact:personalTitle>Dr.</contact:personalTitle>
</contact:Person>

</rdf:RDF>
```

Ejemplo 4: RDF/XML

La representación en formato RDF/XML carece de cierta transparencia y legibilidad por lo que a menudo se recurre a alternativas más fáciles de interpretar como Notation3. [29]

## 6. Notation3 (N3)

Notation3[2], también conocido como N3, es una forma abreviada de serialización no-XML para modelos RDF. Está diseñado para ser más legible por humanos que RDF/XML. Como ejemplo, se muestra a continuación el grafo de la Ilustración 26 en su representación N3:

```
@prefix contact: <http://www.w3.org/2000/10/swap/pim/contact#>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
<http://www.w3.org/People/EM/contact#me> a contact:Person;
        contact:fullName "Eric Miller";
        contact:mailbox <mailto:em@w3.org>;
        contact:personalTitle "Dr.".
```

Ejemplo 5: Notation3 (N3)

## 7. RDFa (Resource Description Framework in Attributes)

RDFa[3] es una recomendación del W3C consistente en un conjunto de extensiones a nivel de atributos XHTML para introducir semántica en documentos web. Permite incluir expresiones RDF del tipo *sujeto-predicado-objeto* en documentos XHTML, es por tanto un formato de serialización RDF que permite incluir tripletes RDF en documentos HTML.

Los atributos que proporciona son los siguientes:

- *typeof*: indica de qué tipo es la instancia descrita.
- *about*: una URI que indica el recurso al cual se hace referencia.
- *rel, rev*: atributos que establecen una relación o relación inversa con otro recurso.
- *property*: aporta una propiedad para el contenido de un elemento.
- *content*: atributo opcional que sobrescribe el contenido del elemento.

[1] http://www.w3.org/TR/REC-rdf-syntax
[2] http://www.w3.org/TeamSubmission/n3
[3] http://www.w3.org/TR/rdfa-syntax

- *datatype*: atributo opcional que indica el tipo de datos del contenido.

A modo de ejemplo, a continuación se representa con RDFa el grafo de la Ilustración 26:

```
<div xmlns="http://www.w3.org/1999/xhtml"
  prefix="
    rdf: http://www.w3.org/1999/02/22-rdf-syntax-ns#
    contact: http://www.w3.org/2000/10/swap/pim/contact#
    rdfs: http://www.w3.org/2000/01/rdf-schema#"
  >
  <div typeof="contact:Person" about="http://www.w3.org/People/EM/contact#me">
    <div property="contact:fullName" content="Eric Miller"></div>
    <div property="contact:personalTitle" content="Dr."></div>
    <div rel="contact:mailbox" resource="mailto:em@w3.org"></div>
  </div>
</div>
```

Ejemplo 6: RDFa

## 8. Dublin *Core*

*Dublin Core* es un modelo de metadatos elaborado por la DCMI (*Dublin Core Metadata Initiative*). Está formado por un conjunto de términos de vocabulario que se pueden utilizar para describir recursos muy diversos (video, imágenes, páginas web, objetos físicos,…), combinar vocabularios de metadatos diferentes o para proporcionar interoperabilidad entre vocabularios de metadatos en la nube LOD y en implementaciones de la Web Semántica. El estándar *Dublin Core* consiste en 15 metadatos:

| Metadato | Etiqueta | Descripción |
|---|---|---|
| Título | **DC. Title** | Nombre dado a un recurso |
| Autor | **DC. Creator** | Persona u organización responsable de la creación del contenido |
| Tema | **DC. Subject** | El tema sobre el que trata el contenido del recurso |
| Descripción | **DC. Description** | Descripción del contenido del recurso |
| Editor | **DC. Publisher** | Entidad responsable de hacer que el recurso se encuentre disponible |
| Colaborador | **DC. Contributor** | Entidad responsable de hacer colaboraciones al contenido del recurso |
| Fecha | **DC. Date** | Fecha en la cual el recurso se puso a disposición del usuario en su forma actual. |
| Tipo | **DC. Type** | Naturaleza o categoría del contenido del recurso |
| Formato | **DC. Format** | Manifestación física o digital del recurso |
| Identificación | **DC. Identifier** | Referencia no ambigua para el recurso dentro de un contexto dado |
| Fuente | **DC. Source** | Una referencia a un recurso del cual se deriva el recurso actual. |
| Lengua | **DC. Language** | La lengua del contenido intelectual del recurso |
| Relación | **DC. Relation** | Una referencia a un recurso relacionado |
| Cobertura | **DC. Coverage** | Extensión o ámbito del contenido del recurso |
| Derechos | **DC. Rights** | Información sobre los derechos de propiedad y sobre el recurso |

Tabla 5: Metadatos de Dublin Core

Los metadatos de *Dublin Core* se pueden utilizar para describir los recursos de un sistema de información. Se están utilizando en organizaciones educativas, bibliotecas, instituciones gubernamentales, sectores científicos y de investigación, páginas web, empresas privadas, etc.

El siguiente ejemplo muestra el uso de algunas propiedades de Dublin Core en un documento RDF:

```
<?xml version="1.0"?>

<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:dc= "http://purl.org/dc/elements/1.1/">

<rdf:Description rdf:about="http://www.ejemplos.com">
  <dc:description>Dublin Core example</dc:description>
  <dc:publisher>Jesus Rio</dc:publisher>
  <dc:date>2008-09-01</dc:date>
  <dc:type>Web Development</dc:type>
  <dc:format>text/html</dc:format>
  <dc:language>en</dc:language>
</rdf:Description>

</rdf:RDF>
```

Ejemplo 7: Dublin Core

La DCMI es una organización dedicada a fomentar la adopción extensa de los estándares interoperables de los metadatos y a promover el desarrollo de los vocabularios especializados de metadatos para describir recursos que permitan sistemas más inteligentes en el descubrimiento del recursos. Se puede encontrar más información sobre los metadatos de Dublin Core en la página de la DCMI[1].

## 9. SPARQL

SPARQL (*SPARQL Protocol and RDF QueryLanguage*) es un lenguaje estandarizado para la consulta de grafos RDF, normalizado por el RDF *Data Access WorkingGroup* (DAWG) del *Word Wide Web Consortium* (W3C). Es una tecnología clave en el desarrollo de la Web Semántica que se constituyó como recomendación oficial del W3C en Enero de 2008.

SPARQL permite realizar consultas no ambiguas sobre fuentes de datos RDF. Por ejemplo la consulta "*¿Cuáles son las capitales de países de África*?" se puede modelar en SPARQL como:

```
PREFIX abc: <http://example.com/exampleOntology#>
SELECT ?capital ?country
WHERE{
  ?x abc:cityname ?capital ;
     abc:isCapitalOf ?y .
  ?y abc:countryname ?country ;
     abc:isInContinent abc:Africa.
}
```

Ejemplo 8: Consulta SPARQL

Las consultas SPARQL se envían desde un cliente a un servicio conocido como "endpoint" de SPARQL usando el protocolo HTTP.

En un principio SPARQL únicamente incorpora funciones para la recuperación de sentencias RDF. Sin embargo, algunas propuestas también incluyen operaciones para el mantenimiento (creación, modificación y borrado) de datos.

[1] http://www.dublincore.org

Al igual que sucede con SQL, es necesario distinguir entre el lenguaje de consulta y el motor para el almacenamiento y recuperación de los datos. Por este motivo, existen múltiples implementaciones de SPARQL, generalmente ligadas a entornos de desarrollo y plataformas tecnológicas concretas.

### 2.3.2 Ontologías: Lenguajes y Vocabularios

RDF proporciona un modelo de datos abstracto y genérico para describir recursos usando tripletes *sujeto-predicado-objeto*. Sin embargo, no proporciona términos específicos de un dominio concreto para describir clases de objetos del mundo real y las relaciones que existen entre ellos. Esta función es realizada por taxonomías, vocabularios y ontologías expresados en RDFS, OWL y SKOS.

Una ontología es una representación formal del conocimiento acerca de un dominio. Se puede considerar como una especificación explícita de una conceptualización. Proporciona un vocabulario compartido que puede ser utilizado para modelar un dominio, identificando el tipo de los objetos o conceptos que existen y sus propiedades y relaciones.

Las ontologías se utilizan por tanto para modelar un dominio y a partir de él realizar razonamientos sobre conceptos de dicho dominio. Las ontologías constituyen por tanto una parte fundamental para el desarrollo de la Web Semántica [29]. Existen ontologías de dominios específicos y ontologías de alto nivel que modelan conceptos comunes que pueden ser usados por un gran número de ontologías. Ejemplos de ontologías de alto nivel son *OpenCyc*[1], SUMO[2], DOLCE[3].

Las ontologías incluyen a: individuos o instancias, clases (conceptos), atributos (características o propiedades de las clases), relaciones (entre clases o individuos), restricciones (descripciones formales que deben cumplirse), reglas (sentencias que describen inferencias lógicas que pueden realizarse), axiomas (definen las relaciones entre individuos), etc. Se codifican formalmente mediante lenguajes de ontologías, entre los cuales OWL (que está basado en RDFS) es el más utilizado. Ejemplos de ontologías son FOAF y YAGO.

#### 1. RDFS (RDF Schema)

RDF Schema[4] es una extensión semántica de RDF. Un vocabulario para describir las propiedades y clases de los recursos RDF, con una semántica para establecer jerarquías de generalización entre dichas propiedades y clases.

RDFS introduce el concepto de clase. Una clase es un tipo, por ejemplo un individuo es un miembro de la clase *Persona*. RDFS define las siguientes clases:

- *rdfs:Resource*: Es la clase más genérica. A ella pertenecen todos los recursos.
- *rdfs:Class*: Declara un recurso como una clase para otros recursos.

[1] http://www.cyc.com/platform/opencyc
[2] http://www.ontologyportal.org
[3] http://www.loa.istc.cnr.it/DOLCE.html
[4] http://www.w3.org/TR/rdf-schema

- *rdfs:Literal*: Representa valores literales como enteros o cadenas de texto.
- *rdfs:Datatype*: Es la clase que abarca los tipos de datos definidos en del modelo RDF.

En RDFS se define también el concepto de Propiedad. Las propiedades son instancias de la clase *rdf:Property* y describen la relación entre un sujeto y un objeto. RDFS define las siguientes propiedades:

- *rdfs:domain*: El dominio de un predicado declara la clase del sujeto en un triplete cuyo segundo componente es dicho predicado.
- *rdfs*:range: El rango de un predicado declara la clase o tipo de datos del objeto en un triplete cuyo segundo componente es dicho predicado.
- *rdfs:subClassOf*: Permite declarar jerarquías de clases indicando de una clase es subclase de otra.
- *rdfs:seeAlso*: Indica un recurso que puede que podría proporcionar información adicional.

## 2. OWL (Web Ontology Language)

El Lenguaje de Ontologías Web (OWL)[1] es una familia de lenguajes de representación de conocimiento para el trabajo con ontologías web. Está basado en RDFS por lo que se puede considerar una extensión semántica de RDFS.

OWL proporciona tres lenguajes, cada uno con un nivel de expresividad mayor que el anterior:

- OWL Lite: diseñado para usuarios que necesitan principalmente una clasificación jerárquica y restricciones simples. Por ejemplo, admite restricciones de cardinalidad pero sólo permite valores cardinales de 0 ó 1. Debería ser más sencillo proporcionar herramientas de soporte a OWL Lite que a sus parientes con mayor nivel de expresividad. OWL Lite proporciona una ruta rápida de migración para tesauros y otras taxonomías. Tiene también una menor complejidad formal que OWL DL.

- OWL DL: diseñado para aquellos usuarios que quieren la máxima expresividad conservando completitud computacional (se garantiza que todas las conclusiones sean computables), y resolubilidad (todos los cálculos se resolverán en un tiempo finito). OWL DL incluye todas las construcciones del lenguaje de OWL, pero sólo pueden ser usados bajo ciertas restricciones (por ejemplo, mientras una clase puede ser una subclase de otras muchas clases, una clase no puede ser una instancia de otra).

- OWL Full: dirigido a usuarios que quieren máxima expresividad y libertad sintáctica de RDF sin garantías computacionales. Por ejemplo, en OWL Full una clase puede ser considerada simultáneamente como una colección de clases individuales y como una clase individual propiamente dicha.

A continuación se muestra el aspecto de una ontología de ejemplo en formato RDF para la representación de plantas vegetales:

[1] http://www.w3.org/TR/owl-features

```
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:owl="http://www.w3.org/2002/07/owl#"
xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:plants="http://www.ejemplo.com/plantas#">

<!-- Cabecera OWL -->
<owl:Ontology rdf:about="http://www. ejemplo.com/plantas">
        <dc:title>Ontología de plantas</dc:title>
        <dc:description>Un ontología sobre plantas</dc:description>
</owl:Ontology>

<!-- Definicion de Clase OWL : tipoplanta -->
<owl:Class rdf:about="http://www.ejemplo.com/plants#tipoplanta">
        <rdfs:label>Tipo de planta</rdfs:label>
        <rdfs:comment>La clase de todos los tipos de plantas.</rdfs:comment>
</owl:Class>

<!-- Definición de Subclase OWL: Flores -->
<owl:Class rdf:about="http://www.ejemplo.com/plants#flores">
  <!-- flores es una sublclase de tipoplanta -->
  <rdfs:subClassOf rdf:resource=http://www.ejemplo.com/plants#tipoplanta />
  <rdfs:label>Plantas con flores</rdfs:label>
  <rdfs:comment>Plantas con flores o angioespermas.</rdfs:comment>
</owl:Class>

<!-- Individuo (Instancia) Ejemplo de sentencia RDF -->
<rdf:Description rdf:about="http://www.ejemplo.com/plants#magnolia">
  <!-- Magnolia es una instancia de flores -->
  <rdf:type rdf:resource="http://www.ejemplo.com/plants#flores"/>
</rdf:Description>

</rdf:RDF>
```

**Ejemplo 9: Ontología RDF**

En este ejemplo se puede observar la estructura de un documento que contiene a una ontología. Comienza con la especificación de los espacios de nombres a utilizar. Se ha especificado de esta forma que, entre otros, se va a hacer uso del vocabulario *Dublin Core* mediante el espacio de nombres dc:

```
xmlns:dc="http://purl.org/dc/elements/1.1/"
```

Se ha especificado además la cabecera de la ontología mediante *owl:Ontology*, las clases TipoPlanta y Flores utilizando *owl:Class* y la instancia Magnolia mediante *rdf:Description*. Se ha creado una pequeña taxonomía al especificar que Flores es una subclase de TipoPlanta utilizando *rdfs:subClassOf*.

Se puede observar que OWL es una extensión de RDFS y utiliza los mismos conceptos que utiliza RDFS, es decir: clases, propiedades, instancias. Sin embargo, comparado con RDFS, OWL añade más vocabulario para describir propiedades y clases. Se muestran a continuación algunos ejemplos [30]:

1. **Relaciones entre clases**: *owl:disjointWith* permite especificar que dos clases A y B sean disjuntas por lo que si una instancia pertenece a A entonces no puede pertenecer a B.
2. **Cardinalidad**: Mediante las restricciones de cardinalidad podemos especificar por ejemplo que la clase “Padre” tiene una cardinalidad mínima de 1 sobre la clase “*tieneDescendencia*”.

3. **Igualdad**: *owl:equivalentClass* permite definir dos clases como equivalentes. Las clases equivalentes tienen las mismas instancias. El valor de igualdad puede utilizarse para crear clases sinónimas. Por ejemplo Coche y Automóvil, y un razonador puede deducir que cualquier individuo que sea instancia de Coche, es también una instancia de Automóvil y viceversa. De forma similar *owl:sameAs* permite definir dos individuos como iguales.
4. **Características de propiedades**: Es posible definir propiedades *simétricas*. Por ejemplo "serAmigo" sería una propiedad simétrica: si A es amigo de B, B también es amigo de A. También es posible definir propiedades inversas, transitivas, funcionales, funcionales inversas, etc.
5. **Restricciones de propiedad**: OWL permite establecer restricciones sobre la forma en que las propiedades son utilizadas por las instancias de una clase. Por ejemplo mediante *owl:allValuesFrom* podríamos definir que la propiedad "*tieneHija*" de la clase "*Persona*" debe tener todos sus valores de la clase "*Mujer*".

Todo este conjunto de características aporta riqueza semántica que redunda en nuevas posibilidades para que un razonador pueda inferir nuevo conocimiento.

## 3. SKOS (Simple Knowledge Organization System)

SKOS[1] es una iniciativa del W3C en forma de aplicación de RDF que proporciona un modelo para representar la estructura básica y el contenido de esquemas conceptuales como son las listas de materias, taxonomías, esquemas de clasificación, tesauros y cualquier tipo de vocabulario controlado. SKOS es uno de los estándares de Web Semántica y su principal objetivo es la publicación de dichos vocabularios como datos enlazados. [31]

En SKOS los conceptos se identifican con referencias URI. Los conceptos pueden etiquetarse con cadenas de texto en uno o varios idiomas, documentarse y estructurarse a través de relaciones semánticas de diversa tipología. Este modelo permite mapear conceptos de diferentes esquemas, así como definir colecciones ordenadas y agrupaciones de conceptos.

El uso de RDF en el desarrollo de SKOS permite obtener documentos en un formato que permite su lectura por parte de aplicaciones informáticas, así como su intercambio y su publicación en la Web. SKOS se ha diseñado para crear nuevos sistemas de organización o migrar los ya existentes adaptándolos a su uso en la Web Semántica de forma fácil y rápida.

Proporciona un vocabulario muy sencillo y un modelo intuitivo que puede ser utilizado conjuntamente con OWL o de forma independiente. Por todo ello, SKOS se considera como un paso intermedio, un puente entre el caos resultante del bajo nivel de estructuración de la Web actual y el riguroso formalismo descriptivo de las ontologías definidas con OWL.

El modelo de datos SKOS es en realidad una ontología definida con OWL Full. Obviamente, al estar basado en RDF, SKOS estructura los datos en forma de tripletes que pueden ser codificados en cualquier sintaxis válida para RDF. SKOS puede ser utilizado conjuntamente con OWL para expresar formalmente estructuras de conocimiento sobre un dominio concreto ya que SKOS no puede realizar esta función al no tratarse de un lenguaje para la representación

[1] http://skos.um.es/TR/skos-primer

de conocimiento formal. El conocimiento descrito de manera explícita como una ontología formal se expresa mediante un conjunto de axiomas y hechos. Pero un tesauro o cualquier tipo de esquema de clasificación no incluye este tipo de afirmaciones, sino que identifica y describe (con el lenguaje natural o expresiones no formales) ideas o significados a los que nos referimos como conceptos. Estos conceptos pueden organizarse en estructuras que carecen de una semántica formal y que no pueden considerarse como axiomas o hechos. Es decir, un tesauro únicamente proporciona un mapa intuitivo de cómo están organizados los temas dentro de procesos de clasificación y búsqueda de objetos (generalmente documentos) relevantes a un dominio específico.

Los elementos del modelo SKOS son esencialmente **clases** y **propiedades**. La estructura e integridad del modelo de datos están definidas por las características lógicas y por las relaciones entre dichas clases y propiedades. Para SKOS, un sistema de organización del conocimiento se expresa en términos de conceptos que se estructuran en relaciones para conformar esquemas de conceptos. Tanto los conceptos como los esquemas de conceptos se identifican mediante URIs.

El modelo SKOS contempla el establecimiento de enlaces entre conceptos denominados **relaciones semánticas**. Estas relaciones pueden ser jerárquicas o asociativas, contemplándose la posibilidad de ampliar la tipología de relaciones. Los conceptos también pueden agruparse en colecciones que a su vez pueden etiquetarse y ordenarse. SKOS se complementa con la posibilidad de que conceptos de diferentes esquemas se puedan mapear entre sí empleando relaciones jerárquicas, asociativas o de equivalencia exacta.

Existen diversas propiedades definidas en SKOS, algunas de ellas nos permiten definir conceptos (*skos:Concept, skos:ConceptScheme, skos:inScheme*) y colecciones de conceptos (*skos:Collection, skos:OrderedCollection*), otras establecen etiquetas léxicas (*skos:prefLabel, skos:altLabel*) o relaciones semánticas (*skos:semanticRelation, skos:broader, skos:narrower, skos:related*), etc.

A continuación se muestra un ejemplo de una taxonomía RDF para la clasificación de entradas de blogs realizada con SKOS y extraído de [32].

```
<rdf:RDF
  xmlns:skos="http://www.w3.org/2004/02/skos/core#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xml:base="http://www.wasab.dk/morten/blog/archives/author/mortenf/skos.rdf">

<skos:ConceptScheme rdf:about="#scheme">
  <dc:title>Morten Frederiksen's Categories</dc:title>
  <dc:description>Concepts from the weblog "Binary Relations" based on
category usage by Morten Frederiksen.</dc:description>
  <dc:creator>Morten Frederiksen</dc:creator>
</skos:ConceptScheme>

<skos:Concept rdf:about="#c1">
  <skos:prefLabel>General</skos:prefLabel>
  <skos:narrower rdf:resource="#c23"/>
  <skos:inScheme rdf:resource="#scheme"/>
</skos:Concept>

<skos:Concept rdf:about="#c23">
  <skos:prefLabel>Travelling</skos:prefLabel>
```

```
  <skos:broader rdf:resource="#c1"/>
  <skos:inScheme rdf:resource="#scheme"/>
</skos:Concept>

</rdf:RDF>
```

Ejemplo 10: Taxonomía RDF realizada con SKOS

Un recurso se define como un concepto usando la etiqueta *skos:Concept* del vocabulario SKOS. Mediante *skos:prefLabel* se asigna una etiqueta léxica preferente al recurso. Los esquemas pueden crearse y utilizarse como entidades independientes pero generalmente se asocian mediante *skos:inScheme* a un vocabulario cuidadosamente recopilado utilizando *skos:ConceptScheme.* SKOS permite la representación de vínculos jerárquicos: para indicar que un concepto tiene un significado más amplio (es decir, más en general) que otro, se utiliza la propiedad *skos:broader*. La propiedad *skos:narrower* se utiliza para realizar la afirmación inversa, es decir, cuando un concepto tiene un significado más concreto (es decir, más específico) que otro.

## 4. FOAF (Friend Of A Friend)

FOAF[1] es una ontología que describe personas, sus actividades y sus relaciones con otras personas y objetos. Cualquier persona puede usar FOAF para describir a alguien o a sí mismo. Mediante FOAF es posible que grupos de personas describan redes sociales sin necesidad de una base de datos centralizada.

FOAF es un vocabulario descriptivo que utiliza RDF y OWL para definir relaciones entre personas por lo que se puede considerar como la primera aplicación de Web Semántica Social. Mediante FOAF es posible obtener por ejemplo el nombre de todas las personas que viven en Europa o listar el nombre de amigos comunes de dos personas.

Se muestra a continuación un documento FOAF serializado en RDF/XML

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
  <foaf:Person>
    <foaf:name>Jimmy Wales</foaf:name>
    <foaf:mbox rdf:resource="mailto:jwales@bomis.com" />
    <foaf:homepage rdf:resource="http://www.jimmywales.com/" />
    <foaf:nick>Jimbo</foaf:nick>
    <foaf:depiction rdf:resource="http://www.jimmywales.com/aus_img_small.jpg" />
    <foaf:interest>
      <rdf:Description rdf:about="http://www.wikimedia.org" rdfs:label="Wikipedia" />
    </foaf:interest>
    <foaf:knows>
      <foaf:Person>
        <foaf:name>Angela Beesley</foaf:name>
      </foaf:Person>
    </foaf:knows>
  </foaf:Person>
</rdf:RDF>
```

Ejemplo 11: Documento FOAF

[1] http://www.foaf-project.org

### 5. Yago

YAGO[1] es una ontología ligera y extensible de gran calidad y amplia cobertura que se construye a partir de entidades y relaciones entre ellas que constituyen hechos. Esta información se extrae automáticamente de la Wikipedia y WordNet utilizando una cuidadosa combinación de métodos heurísticos y sistemas basados en reglas. El resultado es una base de conocimiento que, dicen, supera a WordNet en calidad y cantidad, añadiendo conocimiento sobre personas, organizaciones, productos etc. Tiene una precisión del 95% y está basada en un modelo lógico extensible y compatible con RDFS. Se describirá con detalle en el Capítulo 2.

## 2.3.3 Bases de Conocimiento

A continuación de describen brevemente algunos aspectos de diferentes fuentes de datos o bases de conocimiento utilizadas a lo largo de este trabajo.

### 1. DBPedia

DBPedia[2] es un proyecto que extrae información estructurada a partir del contenido de la Wikipedia. Permite realizar consultas sofisticadas contra la Wikipedia y enlazar otros conjuntos de datos de la web con datos de la Wikipedia. La base de conocimiento DBPedia describe actualmente más de 3,64 millones de conceptos de los cuales 1,83 millones están clasificados según una ontología consistente que incluye 416.000 personas, 526,000 lugares, 60,000 películas, 169,000 organizaciones. Dada su importancia dentro de la nube *Linked Data* se describirá este recurso con más detalle en el Capítulo 1.

### 2. DBPedia Lookup Service (DLS)

*DBPedia Lookup Service*[3] es el servicio de búsquedas de la DBPedia. Se puede utilizar para encontrar URIs de la DBPedia a partir de términos relacionados con un concepto determinado. Se considera como "relacionados" a que el termino de búsqueda coincida con la etiqueta del recurso o con el texto de un enlace que se usa frecuentemente en la Wikipedia para referirse a dicho recurso. Por ejemplo el recurso http://dbpedia.org/resource/United_States se puede encontrar con el término de búsqueda "USA". Los resultados se ordenan según el número de enlaces entrantes desde otras páginas de la Wikipedia a la página de dicho término.

DLS tiene la forma de una API Web basada en REST. Para obtener por ejemplo, recursos relacionados con el término Berlín, se consultaría la url:

http://lookup.dbpedia.org/api/search.asmx/KeywordSearch?QueryString=Berlin

En la respuesta se incluye la URI de la ciudad de Berlín, así como su descripción, clases asociadas, categorías de la Wikipedia en las que se incluye, etc.

[1] http://www.mpi-inf.mpg.de/yago-naga/yago/
[2] http://dbpedia.org
[3] http://wiki.dbpedia.org/lookup

```
<ArrayOfResult xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://lookup.dbpedia.org/">
  <Result>
    <Label>Berlin</Label>
    <URI>http://dbpedia.org/resource/Berlin</URI>
    <Description>Berlin is the capital city of Germany and one of the 16 states of
Germany…</Description>
    <Classes>
      <Class>
        <Label>place</Label>
        <URI>http://dbpedia.org/ontology/Place</URI>
      </Class>
      <Class>
        <Label>city</Label>
        <URI>http://dbpedia.org/ontology/City</URI>
      </Class>
    </Classes>
    <Categories>
      <Category>
        <Label>States of Germany</Label>
        <URI>
          http://dbpedia.org/resource/Category:States_of_Germany
        </URI>
      </Category>
    </Categories>
  </Result>
…
</ArrayOfResult>
```

**Ejemplo 12: Información proporcionada por DLS**

En DLS se pueden establecer filtros para obtener resultados de un tipo determinado (LOCALIZACION, PERSONA,…) o el número máximo de elementos que se desea obtener.

Los resultados que devuelve DLS se refieren siempre a URIs dentro del *endpoint* inglés de la DBPedia, por lo que para encontrar URIs en los *endpoints* de otros idiomas será necesario obtener sus URIs consultando el predicado "sameAs" a partir de la URI inglesa.

## 3. Freebase

Freebase[1] es una base de conocimiento colaborativa que contiene una colección de datos estructurados obtenidos a partir de múltiples fuentes, incluyendo contribuciones individuales de sus usuarios. Aunque se trata de una herramienta de *Linked Data* muy eficaz y apoyada por Google, Freebase difiere en cierta medida de los estándares propios de Web Semántica al utilizar un lenguaje de consultas propio denominado *Metaweb Query Language* (MQL) en vez de SPARQL.

## 4. API de Wikipedia

La Wikipedia[2] ofrece una API Web que permite consultar sus contenidos de una forma automatizada. Proporciona un acceso de alto nivel a los datos contenidos en la Wikipedia. Mediante su utilización es posible, por ejemplo, consultar sus páginas, realizar búsquedas de términos etc. De esta forma podemos obtener la descripción de un término concreto, o buscar las entidades de la Wikipedia que se relacionan con un texto dado.

[1] http://www.freebase.com
[2] http://en.wikipedia.org/w/api.php

Por ejemplo para obtener las entidades de la Wikipedia que se refieren al término químico "Metilhexano", haciendo una petición HTTP GET a la dirección:

http://es.wikipedia.org/w/api.php?action=query&list=search&srprop=timestamp&srsearch=metilhexano obtendríamos como respuesta el siguiente xml que contiene los términos relacionados:

```
<?xml version="1.0"?>
<api>
  <query-continue>
    <search sroffset="10" />
  </query-continue>
  <query>
    <searchinfo totalhits="11" />
    <search>
      <p ns="0" title="3-metilhexano" timestamp="2013-03-07T20:24:04Z" />
      <p ns="0" title="4-etil-3-metilhexano" timestamp="2012-07-06T08:58:35Z" />
      <p ns="0" title="3-etil-3-metilhexano" timestamp="2012-07-06T08:46:21Z" />
      <p ns="0" title="4-etil-2-metilhexano" timestamp="2012-07-06T08:48:43Z" />
      <p ns="0" title="3-etil-2-metilhexano" timestamp="2012-07-06T08:27:19Z" />
      <p ns="0" title="Isoheptano" timestamp="2013-03-09T00:33:13Z" />
      <p ns="0" title="Nonano" timestamp="2013-05-02T11:27:19Z" />
      <p ns="0" title="Fórmula molecular" timestamp="2013-10-31T23:12:25Z" />
      <p ns="0" title="Reacción SN1" timestamp="2013-03-21T02:38:45Z" />
      <p ns="0" title="Heptano" timestamp="2013-05-16T01:12:35Z" />
    </search>
  </query>
</api>
```

Ejemplo 13: Resultados de búsqueda con la Wikipedia API

## 5. Otras Fuentes Linked Data Existentes

Se describen en este apartado algunas fuentes de datos adicionales que es posible utilizar para realizar procesos de enriquecimiento de datos:

- **CIA WorldFactbook**[1]: Es una publicación de la Agencia Central de Inteligencia (CIA) de los Estados Unidos con información básica tipo almanaque acerca de diversos países del mundo. Proporciona un resumen de la demografía, ubicación, capacidad de telecomunicaciones, gobierno, industria, capacidad militar, etc. de todos los países del mundo.
- **Google Maps**: Proporciona servicios de localización.
- **MusicBrainz**[2] : Base de datos musical de contenido abierto. Almacena información sobre artistas, sus grabaciones y la relación entre ellos. Los registros sobre las grabaciones contienen, al menos, el título del álbum, los nombres de las pistas, y la longitud de cada una de ellas.
- **GeoNames**[3]: Es una base de datos geográfica gratuita y accesible a través de Internet. Cada característica en GeoNames está representado como un sitio de recursos identificados por un identificador URI estable. Este identificador URI proporciona acceso, mediante la transferencia de información, a un Wiki en la página HTML o una descripción de los

[1] https://www.cia.gov/library/publications/the-world-factbook
[2] http://musicbrainz.org
[3] http://www.geonames.org

recursos RDF utilizando el dialecto GeoNames. Este dialecto describe las propiedades de las características de GeoNames utilizando OWL. A través de la URL de los artículos Wikipedia enlazados a la descripción RDF, los datos GeoNames se re-enlazan a los datos DBPedia y otros datos RDF.

- **RDF Book Mashup[1]:** Ofrece información sobre autores, libros, tiendas de libros online.
- **Facebook Linked Data Service[2] (LIDS)**: Servicio que implementa algunos de los métodos públicos de la API de Grafos de Facebook y los hace disponibles para su uso con datos enlazados.
- **Eurostat[3]:** Proporciona datos estadísticos sobre los distintos países que conforman la unión europea.

## 2.4 Recurso Léxico: Stilus

Para realizar un proceso de enriquecimiento de datos basado en LOD es imprescindible la utilización de un buen extractor de entidades con nombre. El Reconocimiento de Entidades con Nombre (NER) es una tarea que busca localizar y clasificar elementos atómicos en texto sobre categorías predefinidas como nombres de personas, organizaciones, localizaciones, expresiones de horas, cantidades, valores monetarios, porcentajes, etc.

STILUS Daedalus[4] es el NER utilizado en este trabajo. Se compone de una familia de herramientas para extraer significado de todo tipo de contenidos multimedia. STILUS Daedalus proporciona una API Web[5] que permite detectar las entidades existentes en un texto dado.

STILUS acepta textos de entrada en los idiomas inglés y español y devuelve las entidades encontradas en formato XML, JSON o HTML. Para cada entidad reconocida devuelve su tipo (Lugar, Persona, Organización,…), su enlace a la Wikipedia en su versión inglesa, las variantes de la entidad que aparecen en el texto junto a su posición así como otro tipo de información adicional.

A modo de ejemplo se muestra una respuesta de STILUS en formato XML donde se identifica la entidad “Madrid”:

```
<?xml version="1.0" encoding="utf-8"?>
<results>
  <status code="0">OK</status>
  <result>
    <entities>
      <entity>
        <form><![CDATA[Madrid]]></form>
        <semType>
          <type>LOCATION</type>
          <subtype>GEO_POLITICAL_ENTITY</subtype>
```

[1] http://www4.wiwiss.fu-berlin.de/bizer/bookmashup/index.html
[2] http://datahub.io/es/dataset/facebook-lids
[3] http://epp.eurostat.ec.europa.eu/portal/page/portal/eurostat/home
[4] http://www.daedalus.es
[5] http://api.daedalus.es

```
          <subsubtype>CITY</subsubtype>
        </semType>
        <semLinkedData><![CDATA[C_MADRID#CITY@en.wiki:Madrid]]></semLinkedData>
        <variants>
          <variant>
            <form><![CDATA[Madrid]]></form>
            <inip>0</inip><endp>5</endp>
          </variant>
        </variants>
        <relevance>100</relevance>
      </entity>
    </entities>
  </result>
</results>
```

Ejemplo 14: Respuesta de STILUS

En este ejemplo se puede observar cómo se ha reconocido la entidad Madrid en la Wikipedia (dentro del elemento *semLinkedData*) en las posiciones de 0 a 5 (*inip*, *endp*) del texto enviado como parámetro.

## 2.5 Formatos de vídeo y subtítulos

HTML5[1], la nueva versión de HTML que ya incorporan varios navegadores, presenta importantes novedades. Entre ellas destaca la introducción de la nueva etiqueta <video> que reemplaza parcialmente al elemento <object> y permite que un elemento de video sea reproducido de forma nativa por el navegador web sin necesidad de instalar *plugins* adicionales. HTML5 permite además el control de la reproducción del video y sus subtítulos mediante JavaScript lo que permite crear interfaces de usuario multimedia nunca vistos hasta ahora.

### 1. Ficheros de subtítulos WebVTT

*Web Video Text Tracks Format*[2] o WebVTT es un formato de fichero de subtítulos de texto plano diseñado por el W3C para ser utilizado en conjunto con la nueva especificación de HTML5 Video.

En WebVTT se incluyen diversas marcas temporales que indican el momento de la aparición del texto en un video. WebVTT permite la inclusión de recursos de estilo, listas de capítulos, y metadatos, es decir, información y contenido sobre el video que no se muestra por defecto en la reproducción del video por defecto pero que es posible mostrar y manipular mediante JavaScript. En este trabajo se utilizará este formato para mostrar los subtítulos con información enriquecida en forma de metadatos WebVTT. Se muestra a continuación un ejemplo de fichero WebVTT:

```
00:11.000 --> 00:13.000
<v Roger Bingham>We are in New York City
00:13.000 --> 00:16.000
<v Roger Bingham>We're actually at the Lucern Hotel, just down the street
00:16.000 --> 00:18.000
```

[1] http://www.w3.org/html/logo
[2] http://dev.w3.org/html5/webvtt

```
<v Roger Bingham>from the American Museum of Natural History
00:18.000 --> 00:20.000
<v Roger Bingham>And with me is Neil deGrasse Tyson
00:20.000 --> 00:22.000
<v Roger Bingham>Astrophysicist, Director of the Hayden Planetarium
00:22.000 --> 00:24.000
<v Roger Bingham>at the AMNH.
00:24.000 --> 00:26.000
<v Roger Bingham>Thank you for walking down here.
```

Ejemplo 15: Ejemplo de fichero WebVTT

## 2. Timed Text (TT) Authoring Format (TTAF)

*Timed Text Authoring Format*[1] proporciona una representación estandarizada para un tipo particular de información textual en el que se incluye información de estilo, distribución y semántica temporal por parte de un autor. Se trata, por tanto, de un formato adecuado para representar los subtítulos sincronizados de un elemento de video o audio.

TTAF es un formato basado en XML y permite la inclusión de metadatos arbitrarios. La estructura de un documento TTAF presenta el siguiente aspecto:

```
<tt xml:lang="" xmlns="http://www.w3.org/2006/10/ttaf1">
  <head>
    <metadata/>
    <styling/>
    <layout/>
  </head>
  <body/>
</tt>
```

Ejemplo 16: Estructura de un documento TTAF

La estructura consiste en un elemento “<tt>” que contiene una cabecera “<head>” y un cuerpo de documento “<body>”. En la cabecera se puede especificar la información de estilo, distribución y metadatos. En el cuerpo del documento se incluye el contenido en forma de etiquetas de párrafo que incluyen los atributos “begin” y “end” con la información de alineamiento temporal. A continuación se presenta un ejemplo de documento TTAF:

```
<?xml version="1.0" encoding="UTF-8"?>
<tt xmlns="http://www.w3.org/2006/04/ttaf1"
      xmlns:tts="http://www.w3.org/2006/04/ttaf1#styling" xml:lang="en">
<head> <styling>
     <style id="txtRight" tts:textAlign="right" tts:color="cyan"/>
     <style id="txtLeft" tts:textAlign="left" tts:color="#FCCA03"/>
     <style id="defaultSpeaker" tts:fontSize="12px" tts:fontFamily="Arial" />
    </styling>
    <metadata xmlns:ttm="http://www.w3.org/2006/10/ttaf1#metadata">
      <ttm:title>Video 8 AVIP</ttm:title>
      <ttm:copyright>UNED</ttm:copyright>
    </metadata>
</head>
<body id="ccbody" style="defaultSpeaker">
<div xml:lang="es">
 <p begin="00:00:00.29" end="00:00:05.53">Los sistemas de bases de datos… </p>
 <p begin="00:00:06.12" end="00:00:10.09">pasaron a convertirse en una parte fundamental … </p>
…
 <p begin="00:11:34.72" end="00:11:38.72">...y propósitos del sistema. Muchas gracias.</p>
</div> </body> </tt>
```

Ejemplo 17: Ejemplo de fichero TTAF

[1] http://www.w3.org/TR/2006/CR-ttaf1-dfxp-20061116

# Capítulo 3. Planteamiento, metodología y objetivos

En esta primera parte de la memoria se incluye, en el capítulo uno, una revisión de los trabajos relacionados más relevantes para enriquecimiento con datos abiertos y enlazados, y en el capítulo dos una panorámica de la tecnología actual , describiendo los recursos y estándares más relacionados con la propuesta y el experimento, que serán descritos en los capítulos cinco y seis.

En esta tesis fin de máster (TFM) se pretende mostrar las posibilidades de la tecnología actual para crear entornos enriquecidos de información. La hipótesis de trabajo es que un proceso de enriquecimiento de datos a partir de fuentes LOD es una tarea que requiere un buen conocimiento de la información disponible, estructurar los pasos a seguir para realizar dicha tarea, y seleccionar los patrones de diseño y herramientas que son estándares *de facto*. El hecho de tratar con videos multidominio hace imposible seleccionar a priori unos recursos (por ejemplo ontologías) de dominio concreto.

La metodología planteada, para la que se han desarrollado diferentes funcionalidades relacionadas con las etapas identificadas, como se explicará en la segunda parte de esta memoria, se basa en:

- La normalización de las transcripciones mediante el uso de algún formato que aporte marcas temporales.
- La identificación de unidades de información a enriquecer, básicamente entidades con relevancia para el contenido e interés para los usuarios. Conocer su tipo es importante para su enriquecimiento, aunque la ambigüedad puede persistir durante la asignación de su URI.
- El entrelazado de información proveniente de fuentes de datos (OD), a partir de las URIs de cada unidad de información.
- Presentación de la información enriquecida de forma sincronizada con el video.

En su vertiente práctica, se ha creado un entorno para enriquecer información transcrita de un catálogo de videos a partir de fuentes de datos Open Data. Para ello será necesario hacer uso de diferentes tecnologías:

- Open Data, Linked Data, SPARQL para acceso a la DBpedia, Wikipedia, y otras fuentes.
- Reconocimiento de entidades nombradas (NER)
- HTML5

Se ha experimentado con una colección de videos educativos AVIP, cuya transcripción ha sido realizada por un grupo de voluntarios durante el año 2012 y se han analizado los resultados, como se presenta con detalle en la segunda parte de esta memoria.

El objetivo final de este tipo de aplicaciones será contextualizar los videos mientras son reproducidos para proporcionar al usuario información relacionada interesante, entretenida e inesperada sobre los temas que se tratan en el video, aunque como no era el objetivo de este trabajo, no se ha realizado más que un interfaz simple que incluye las diferentes funcionalidades de enriquecimiento definidas y desarrolladas.

Finalmente indicar, que este trabajo de más de dos años de duración a tiempo parcial (el autor no se dedica a la investigación profesionalmente), es la respuesta realizada en este tiempo al siguiente conjunto de preguntas planteadas inicialmente:

1. **¿Cuáles son los objetivos básicos del trabajo?**

   - Analizar el estado del arte de las tecnologías relacionadas con Open Data y Linked Data.
   - Describir una metodología para el desarrollo de aplicaciones de enriquecimiento de datos a partir de fuentes basadas en Linked Open Data (LOD).
   - En su vertiente práctica, encontrar una colección de videos con metadatos o transcripciones, para identificar unidades de información en dichos videos y enriquecerlas a partir de fuentes de datos Open Data.

2. **¿Cuál es el contexto de investigación aplicada actualmente?**

   Las tecnologías de Web Semántica y Linked Open Data están muy de actualidad y están recibiendo gran atención por parte de la comunidad investigadora. Los trabajos de enriquecimiento de datos a partir de fuentes LOD conocidos son sobre documentos web, contenido multimedia, e incluso sobre contenido proveniente de las redes sociales (Facebook, Twitter,…). Muchos de estos trabajos presentan algunas características comunes como el uso de sistemas NER para la detección de entidades o la utilización de la DBPedia como concentrador principal del proceso de enriquecimiento. También es recurrente la utilización de la información existente en la DBPedia, el sistema de Categorías de la Wikipedia u ontologías como YAGO para realizar los procesos de enriquecimiento. Tras el planteamiento para enriquecimiento definido y la colección de videos en español disponible, se toma la decisión mostrada en esta memoria de los recursos y estándares a utilizar.

3. **¿Qué trabajos del estado del arte son los más importantes para desarrollar los objetivos?**

   La revisión del estado del arte es una tarea que dura todo el tiempo. Los trabajos más relevantes se han incluido en la parte primera de memoria, e incluso se encuentra uno de finales del 2013 que se corresponde casi totalmente con la contribución realizada. Cada uno de los trabajos incluidos en la memoria se describen en la primera parte y se justifican por su interés para el trabajo realizado.

**4. ¿Qué caso de estudio escoger?**

Para la realización de este TFM se ha realizado un proceso de enriquecimiento a partir de un corpus de videos provenientes de la Cadena Campus AVIP de la UNED. Los videos junto con sus transcripciones están disponibles en idioma español y tienen una temática variada, aunque predominan los contenidos sobre Química y se puede decir que en general los contenidos son muy técnicos.

La realización práctica del proceso de enriquecimiento ha hecho posible conocer de primera mano los problemas que surgen al afrontar este tipo de tareas e identificar trabajos futuros a realizar.

En el caso de estudio elegido se han aplicado la mayoría de las técnicas reflejadas en el estado del arte de este dominio como son: el empleo de sistemas NER para la detección de entidades, la utilización de la DBPedia como eje central del proceso de enriquecimiento y la utilización de las categorías de la Wikipedia para obtener información adicional.

**5. ¿Qué se ha alcanzado y qué sería interesante para continuar?**

Se han identificado las aproximaciones existentes y los problemas comunes que surgen en el desarrollo de este tipo de trabajo y se ha definido una metodología adecuada que guíe el proceso de enriquecimiento. Entre las contribuciones de este trabajo al estado del arte se encontrarían:

- La utilización combinada de diferentes técnicas (DBPedia Lookup Service, Wikipedia API, Sindice, Freebase) mostradas en el estado del arte para la recuperación de la URI de una entidad.
- La realización del proceso de enriquecimiento en idioma distinto del inglés, lo cual requiere de tareas adicionales al proceso general de enriquecimiento.
- La flexibilización de la definición de los procesos de enriquecimiento mediante consultas SPARQL.
- La utilización de HTML5 para la creación del interfaz de usuario.
- El código fuente de los prototipos desarrollados para este TFM se ha publicado en SourceForge[1], en el proyecto LinkedDataGenerator[2] .

Se echa en falta en este tipo de trabajos la posibilidad de realizar una evaluación basada en colecciones estudiadas y con juicios de relevancia sobre los resultados de enriquecimiento, ya que sin ella solo es posible una revisión manual llevada a cabo por usuarios expertos para el análisis de los resultados.

[1] http://sourceforge.net
[2] http://sourceforge.net/projects/linkeddatagenerator

# PARTE 2: EXPERIMENTACIÓN Y ANÁLISIS DE RESULTADOS

# Capítulo 4. Propuesta para el enriquecimiento de datos a partir de fuentes Open Data.

El proceso de enriquecimiento de datos planteado se basa en las siguientes etapas: etapa previa de definición concreta de la tarea y diseño de formatos, pre-procesado, proceso de enriquecimiento y presentación. En este capítulo se detalla la propuesta realizada para cada una de las etapas identificadas.

A continuación se muestra un diagrama de procesos del entorno que engloba a las funcionalidades desarrolladas, donde se puede observar la integración de las etapas y las funcionalidades que intervienen en cada una de ellas.

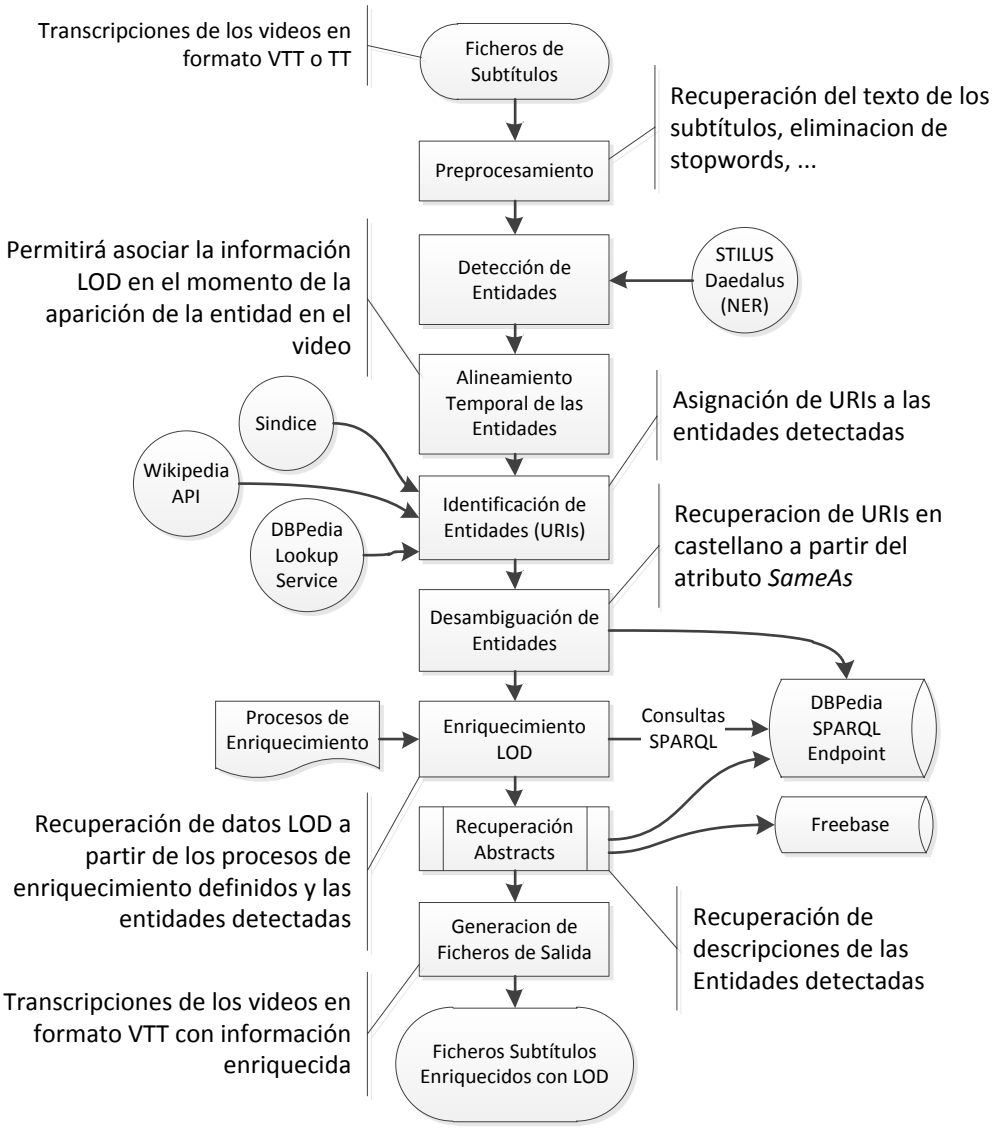



Ilustración 27: Diagrama de Procesos

## 4.1 Etapa previa al enriquecimiento de datos

Se describen a continuación las fases a realizar.

### 4.1.1 Definición del Corpus

Todo proceso de enriquecimiento de datos debe comenzar conociendo qué datos se desea enriquecer. Dichos datos constituirán el corpus sobre el que se realiza la tarea. Se deben tener en cuenta los siguientes aspectos:

1) **Naturaleza del corpus**: la naturaleza del corpus elegido puede ser muy variada: se podría decidir enriquecer datos que provienen de una base de datos, correos electrónicos, páginas web, documentos. Con el auge actual de las redes sociales también se están realizando trabajos de enriquecimiento sobre hilos de Facebook, Tweets de Twitter, etc. También se podrían utilizar datos multimedia como vídeos, audio, imágenes. En general se puede considerar que se pueden realizar tareas de enriquecimiento a partir de cualquier fuente de datos donde se disponga de contenido textual.

2) **Idioma del corpus**: Si es posible, es útil conocer de antemano el idioma de los textos que conformarán el corpus. El idioma determinará muchas tareas posteriores ya que las herramientas que se utilizarán deberán soportar el idioma elegido. La opción más sencilla es utilizar el idioma inglés, ya que es el idioma que soportan de forma nativa la mayoría de herramientas y tecnologías de Web Semántica. Es posible sin embargo, trabajar también con un corpus donde existan textos en diferentes idiomas, aunque implicaría aplicar técnicas de multilingüismo adecuadas a la tarea de enriquecer: traducir todo al inglés, utilizar recursos de cada idioma etc.

3) **Corpus Multidominio**: Normalmente los elementos del corpus generado presentarán una o varias temáticas más o menos relacionadas: documentos sobre cine, textos científicos, etc. Está probado que si es posible determinar la temática o dominio, y hay recursos disponibles para ellos, el escenario es muy diferente a si se trabaja en dominios abiertos. El multidominio supone un reto que se va a abordar en este TFM.

4) **Formato de los datos del corpus**: Los elementos que conforman el corpus presentarán un formato determinado: texto, XML, binario, formatos propietarios. Será necesario conocer dicho formato en detalle para procesar los elementos del corpus de forma adecuada (puede que sea necesario la utilización o generación de conversores de formato). Si la naturaleza del corpus son datos multimedia es necesaria la extracción del contenido textual a partir de los datos multimedia.

En este TFM se han utilizado los 20 videos en castellano provenientes de la Cadena Campus Plataforma AVIP[1] de la UNED (todos los transcritos disponibles). La temática de los videos es

[1] http://www.intecca.uned.es/portalavip/cadenaCampus.php

muy variada, pero todos pertenecen a clases online que imparten los profesores de la universidad: Química, Derecho, Informática, etc.

### 4.1.2 Selección de recursos: Detector de Entidades

El proceso de enriquecimiento de datos se realizará a partir de las entidades identificadas en los textos del corpus por el Detector de Entidades Nombradas (NER) por lo que la utilización de un buen NER determinará en gran medida la calidad del resultado final.

En la selección del detector NER se tendrán en cuenta los siguientes factores:

1) **Idioma**: El NER a utilizar debe reconocer entidades en el idioma o idiomas en que se encuentren los textos del corpus (además del inglés).
2) **Tipo de entidades reconocidas**: El sistema NER debe reconocer los tipos de las entidades que necesitemos procesar. Tipos habituales suelen ser: Persona, Lugar, Organización, Fecha,…
3) **Identificación de las entidades**: Algunos NER proporcionan una URI que identifica a la entidad en la nube LOD. En otro caso, el proceso es más complejo, por lo que se aborda en otro apartado.
4) **Formato** de intercambio de datos que utilizará el NER.
5) **Rendimiento**.
6) **Instalación**: Algunos detectores NER funcionan como un programa independiente y requieren instalación y otros funcionan como un servicio web.
7) **Precio**: Existen detectores NER comerciales y de libre distribución.

En nuestro caso utilizaremos el software *Social Media Analytics* de STILUS DAEDALUS[1] con licencia para investigación. Es capaz de detectar, entre otros, los tipos de entidad Persona, Organización, Localización, y Producto en textos en inglés y castellano. En algunas ocasiones ofrece el enlace a la página de la DBPedia que identifica a la entidad. No requiere instalación ya que funciona como una API Web[2] y soporta los formatos XML y JSON.

### 4.1.3 Identificación de las fuentes de datos Open Data a utilizar

Es necesario definir la/las fuentes de datos Open Data a utilizar por el proceso de enriquecimiento, es decir, seleccionar los endpoint SPARQL sobre los que realizar las consultas de enriquecimiento de información.

A la hora de consultar un endpoint SPARQL debe tenerse en cuenta que el procesamiento de las consultas y el viaje de los datos por la red generará cierta latencia. Si el volumen de datos a procesar es muy grande o las consultas muy costosas debe tenerse en cuenta que existe la posibilidad de descargar un volcado del conjunto de datos e instalar el endpoint de forma local lo cual mejorará el rendimiento general del sistema.

[1] http://api.daedalus.es/sma-info
[2] http://api.daedalus.es/sma

En el caso de estudio de este TFM, dado que se pretende realizar el enriquecimiento de información en castellano se usará el endpoint SPARQL de la DBPedia en castellano[1].

### 4.1.4 Definición del formato de salida resultado del proceso de enriquecimiento

Como resultado del proceso de enriquecimiento de datos se generará un fichero resultado. En nuestro caso particular, al tratarse de un enriquecimiento de subtítulos de videos se ha optado por generar un fichero en formato WebVTT[2], el formato estándar de HTML5 que puede ser reproducido de forma nativa por un navegador con soporte de HTML5 sin necesidad de *plugins* adicionales. Cada video dispondrá por tanto de un fichero asociado de subtítulos en formato WebVTT cuyo nombre debe ser idéntico al del video asociado pero con la extensión ".vtt".

A continuación se detalla el formato de fichero de subtítulos enriquecidos utilizado en el experimento. El fichero comienza con una línea que indica el formato de fichero

```
WEBVTT FILE
```

*Cabecera del fichero vtt*

A continuación se incluyen las diferentes líneas de subtítulos. Cada línea incluye la siguiente información: número de línea, marca temporal (incluye inicio y fin del subtítulo) y el texto del subtítulo.

```
1
00:00:00.204 --> 00:00:02.990
{"texto":"Mi primera clase en la <b>UNED</b>…",
"accion":"DBPedia",
"entidades":[
{"nombre":"UNED", "semType":"FACILITY",
"source":"National_University_of_Distance_Education",
"abstract": "La Universidad Nacional de Educación a Distancia (UNED), es una
universidad pública española de ámbito estatal… ",
"lod": "Juan A. Gimeno Ullastres@es http://www.uned.es ",
"lodDescription": "Nombre del rector y pagina web",
"sparql": "
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dbpedia-owl:<http://dbpedia.org/ontology/>
SELECT ?abstract WHERE {
<http://dbpedia.org/resource/National_University_of_Distance_Education>
dbpedia-owl:abstract ?abstract .
FILTER langMatches( lang(?abstract), 'es') }"},
"}]}
```

*Fichero vtt etiquetado con metadatos*

El texto del subtítulo incluye la siguiente información:

- **texto**: texto del subtítulo a mostrar. Puede contener marcado HTML para destacar ciertos aspectos del texto, como las entidades identificadas.
- **acción**: posible acción que el reproductor del video puede utilizar para enriquecer el texto mostrado. Las acciones posibles son: DBPedia o Maps.

[1] http://es.dbpedia.org/sparql
[2] http://dev.w3.org/html5/webvtt

- **entidades**: Datos de las entidades identificadas en el texto. Las entidades se codifican en formato JSON para facilitar su reutilización por un reproductor de videos basados en JavaScript. Cada entidad contiene el conjunto de datos:
    - **nombre**: nombre de la entidad identificada
    - **semType**: etiqueta semántica de la entidad, junto con sus subetiquetas tal como se codifican en Stilus. Algunos de sus posibles valores son:
        - PERSON, FULL_NAME
        - LOCATION, CONTINENT
        - LOCATION,GEO_POLITICAL_ENTITY,CITY
        - …
    - **source**: URI a la DBPedia donde se describe la entidad.
    - **abstract:** Descripción de la entidad obtenida de la DBPedia mediante una consulta SPARQL
    - **lod:** Información de datos enlazados recuperada.
    - **lodDescription**: Descripción de la información LOD recuperada.
    - **sparql**: Consulta SPARQL que permitió obtener los datos LOD.

## 4.2 Pre-proceso

La primera etapa del Pre-proceso consistirá en la extracción del texto del objeto multimedia a enriquecer. Adicionalmente se pueden aplicar, o no, a los textos algoritmos de eliminación de palabras vacías (*stopwords*) o lematización (*stemming*) para obtener las raíces léxicas de los términos.

### 4.2.1 Identificación de las Entidades Nombradas (NE) y alineamiento temporal

Una vez que se dispone del contenido textual se procederá a la identificación de entidades nombradas en dichos textos. Esta es una tarea que será realizada por un extractor de entidades con nombre (Sistema NER).

Un detector NER, toma como entrada un texto y devuelve la lista de entidades que ha detectado en dicho texto. Se adjuntará además para cada entidad, la posición física donde aparece dicha entidad (número de línea y carácter), ya que esta información será necesaria para mantener un alineamiento temporal de la aparición de las entidades en el texto, es decir, en todo momento debemos mantener una asociación entre el texto y la aparición de la entidad. En el experimento realizado para enriquecimiento de videos, esto nos permitirá hacer visible la información enriquecida de una entidad en el momento exacto en que dicha entidad se menciona en el video, creando una experiencia de usuario más dinámica.

### 4.2.2 Identificación de las entidades detectadas en la nube LOD y asignación de URIs

El sistema NER detecta entidades en un texto, pero además, para cada entidad reconocida será necesario disponer de una URI que identifique dicha entidad dentro de la nube LOD y que posteriormente permita enriquecer esa información con información adicional mediante consultas SPARQL.

La DBPedia se ha establecido como un estándar de facto para actuar como *hub* en la LOD y es habitual utilizar las URIs de la DBPedia como identificadores universales de entidades. De esta forma la entidad "España" está identificada con la URI http://es.dbpedia.org/resource/España en el endpoint de la DBPedia en castellano y como http://dbpedia.org/resource/Spain en el endpoint de la DBPedia en inglés[1].

Una vez seleccionado un detector NER y definido un sistema universal de identificación de las entidades es necesario asignar a cada entidad detectada su identificador. En algunos casos, es posible que el NER proporcione dicha información asociada a la entidad, pero no siempre sucederá esto. Será necesario por tanto, una funcionalidad que, partir de una entidad, nos de su URI en la DBPedia. Cuando el NER no proporciona la URI, algunas opciones disponibles son Sindice[2] y el DBPedia Lookup Service[3].

[1] http://dbpedia.org/sparql
[2] http://www.sindice.com
[3] http://wiki.dbpedia.org/lookup

En el experimento realizado, se utiliza una combinación de estas estrategias ya que STILUS nos proporciona, para algunas entidades, su URI en el endpoint ingles de la DBPedia. Por lo que se tratan varios casos posibles:

- STILUS nos proporciona la URI de una entidad en el endpoint inglés de la DBPedia. Por ejemplo para la entidad "España" obtendríamos la URI http://dbpedia.org/resource/Spain.

- STILUS no proporciona la URI de la entidad. En este caso se utilizará el DBPedia Lookup Service para buscar la entidad en la DBPedia. Dado el texto que denota a una entidad, dicho servicio devolverá las URIs que la identifican en el dataset inglés de la DBPedia. Una opción es no realizar ninguna tarea de desambiguación y utilizar el primer resultado devuelto por el servicio.

- Si no obtenemos una URI ni con STILUS ni con el DBPedia Lookup Service recurriremos a la API de la Wikipedia.

- Si no obtenemos una URI con los métodos anteriores recurriremos a Sindice para buscar la URI indicando que deseamos resultados en formato RDF y en el dominio dbpedia.org.

Cuando el resultado sea una URI a la DBPedia en inglés, será necesario realizar la traducción al castellano de la misma, ej: http://es.dbpedia.org/resource/España realizando una labor de desambiguación.

### 4.2.3 Desambiguación de Entidades

El éxito de la iniciativa de Linked Data ha propiciado el surgimiento de repositorios de datos que cubren un amplio rango de temáticas: personas, lugares, organizaciones... Dichos repositorios se han entrelazado entre sí siguiendo los principios de Linked Data y esto ha originado problemas de duplicación y correferencia. Los datasets existentes a menudo provienen de fuentes que pueden ser incompletas o poco exactas. El proceso de entrelazado entre datasets produce inconsistencias que generan un efecto "bola de nieve" a medida que surgen nuevos conjuntos de datos.

El problema de la correferencia existe en multitud de disciplinas. En el ámbito de la Web Semántica la correferencia se produce de dos formas: Una URI puede identificar a más de un recurso o varias URIs pueden identificar al mismo recurso. Por ejemplo "España" tiene diferentes URIs en diferentes datasets:

http://dbpedia.org/resource/Spain
http://es.dbpedia.org/page/España
http://www4.wiwiss.fu-berlin.de/factbook/resource/Spain
http://sws.geonames.org/2510769

Este hecho no es un problema en sí mismo, pero el caso se agrava cuando estas URIs se enlazan a otras mediante enlaces de tipo *owl:sameAs* y surgen inconsistencias. La Wikipedia

trata de mitigar el problema de correferencia mediante las "*páginas de desambiguación*" que se crean cuando existe más de una entrada con el mismo nombre y distinto significado. Dichas páginas de desambiguación también se han arrastrado a la DBPedia donde no había una verdadera necesidad de ellas ya que dichas páginas añaden más ambigüedad a la DBPedia. Por ejemplo en la URI del político Robert Williams (http://dbpedia.org/resource/Robert_Williams ) se han mezclado las propiedades del cantante Robert Williams y del actor Robert Williams.

Las fuertes implicaciones asociadas al atributo *owl:sameAs* provocan también serios problemas. Al enlazar dos URIs por dicha propiedad se establece que las dos URIs identifican al mismo recurso, pero existen numerosos ejemplos donde esto no es correcto (los enlaces se han realizado a la ligera).

En este trabajo ha sido necesario realizar tareas de desambiguación para recuperar la URI en el dataset castellano de la DBPedia de una entidad a partir de la URI inglesa de dicha entidad. Por ejemplo, dada la URI http://dbpedia.org/resource/Spain sería necesario obtener la URI http://es.dbpedia.org/resource/España

Esta traducción se realizará consultando mediante SPARQL la URI inglesa en el endpoint inglés y obteniendo los valores de la propiedad *owl:sameAs* que nos devolverá URIs equivalentes en otros datasets. Mediante una operación de filtrado se obtiene el valor deseado.

```
SELECT ?p WHERE
{
 <http://dbpedia.org/resource/Spain> owl:sameAs ?p
 FILTER regex(str(?p), "^http://es.dbpedia.org/resource") .
}
```

**Ejemplo 18: Traducción de una URI al endpoint castellano de la DBPedia**

Hay trabajos [33] [34] que muestran los errores provocados por la utilización de los enlaces *owl:sameAs* de la DBPedia para representar relaciones de identidad y similaridad en Datos Enlazados. La resolución de la ambigüedad en la nube Linked Data es un problema abierto que se sale del ámbito de estudio de este trabajo, y se plantea como trabajo futuro por su interés.

## 4.3 Etapa de enriquecimiento

Una vez detectadas e identificadas las entidades, se procederá a realizar los procesos de enriquecimiento de datos basados en LOD. Las posibilidades de enriquecimiento son muy amplias y diversas por lo que se deben definir con precisión.

### 4.3.1 Procesos de enriquecimiento de datos

En este trabajo se ha decidido especificar los procesos de enriquecimiento mediante consultas SPARQL (parametrizadas por el nombre de entidad) que se ejecutarán o no dependiendo de que la entidad presente determinadas propiedades con determinados valores. Para permitir una mayor flexibilidad todas las consultas se almacenarán en un fichero de configuración independiente. Para ello se definirán los siguientes pasos:

1. Entre los tipos de entidades devueltos por el NER, se deben definir los tipos de entidades que se procesarán. En nuestro caso los tipos de entidades a tratar serán:

   - PERSON (Persona)
   - ORGANIZATION (Organizaciones, asociaciones, …)
   - FACILITY (Infraestructuras)
   - LOCATION (Lugares: Ciudades, Países, Regiones,…)
   - PRODUCT (Productos)
   - UNKNOWN (Tipo Desconocido)

   Las entidades que pertenezcan a cualquier otro tipo se ignorarán.

2. Definir para cada tipo de entidad (PERSON, LOCATION, ORGANIZATION) las propiedades a utilizar en el endpoint SPARQL seleccionado (DBPEDIA), y la consulta SPARQL a realizar dependiendo de cada valor encontrado. Por ejemplo si se pretende obtener información acerca de actores de cine, películas y directores se podrían definir las siguientes consultas:

   Si la entidad es PERSON, se analiza la información obtenida de la DBPEDIA usando el atributo *occupation* de esa persona. Si la propiedad tiene el valor "ACTOR" se podría definir la consulta SPARQL que devuelve las películas del actor detectado.

```
SELECT DISTINCT ?titulo ?resumen
WHERE {
?titulo rdf:type <http://dbpedia.org/ontology/Film> .
?titulo rdfs:comment ?resumen .
?titulo dbpedia-owl:starring dbpedia:Daniel_Day-Lewis
FILTER(lang(?resumen) = "es" ).
} LIMIT 10
```

   De forma similar, si la propiedad "*occupation*" tiene el valor "DIRECTOR DE CINE" se podría definir la consulta SPARQL que devuelve las películas que ha dirigido ese director:

```
SELECT DISTINCT ?titulo ?resumen
WHERE {
?titulo rdf:type <http://dbpedia.org/ontology/Film> .
?titulo rdfs:comment ?resumen.
?titulo dbpedia-owl:director dbpedia:Steven_Spielberg
```

```
FILTER(lang(?resumen) = "es" ).
} LIMIT 10
```

De forma análoga se podrían definir procesos como:

- Si la entidad es una persona y su ocupación es “Filósofo” obtener las escuelas filosóficas donde se engloba.
- Si la entidad es una Universidad obtener el nombre del rector y su página web.
- Si la entidad es un producto comercial obtener su eslogan.
- Si la entidad es una ciudad obtener sus coordenadas de localización, el nombre de su alcalde y sus lugares de interés.
- …

Esta información se definirá formalmente en un fichero de configuración XML.

3. Fichero de configuración de los procesos de enriquecimiento: definirá formalmente los procesos de enriquecimiento de datos basados en LOD a realizar para cada tipo de entidad detectada. Mediante la etiqueta <tipoEntidad> se define el tipo de entidades a considerar en el proceso de enriquecimiento.

```
<tipoEntidad tipo=" FACILITY ">
```

Mediante la etiqueta <propiedad> se definen las propiedades y valores a tener en cuenta dentro de un tipo de entidad determinado y una descripción textual del proceso de enriquecimiento.

```
<propiedad nombre="rdf:type"
 valor="http://dbpedia.org/ontology/University"
 descripcion="Nombre del rector y pagina web">
```

Mediante la etiqueta <consultaSPARQL> definiremos la consulta SPARQL a realizar sobre la entidad de detectada. La consulta incluirá el texto “ENTIDAD” que en tiempo de ejecución se sustituirá por el identificador de la entidad detectada en la DBPedia.

```
<consultaSPARQL>
     <![CDATA[
     PREFIX dbpprop: <http://es.dbpedia.org/property/>
     PREFIX dbres: <http://es.dbpedia.org/resource/>
     SELECT ?a ?b
     WHERE {
              dbres:ENTIDAD dbpprop:rector ?a .
              dbres:ENTIDAD dbpprop:sitioWeb ?b .
     }]]>
</consultaSPARQL>
```

**Ejemplo 19: Consulta SPARQL que devuelve el nombre del rector y sitio web de una Universidad**

El fichero presentará un aspecto similar al siguiente:

```
<?xml version="1.0" encoding="UTF-8"?>
<configuracion>

<tipoEntidad tipo="FACILITY">
<propiedad
  nombre="rdf:type"
  valor="http://dbpedia.org/ontology/University"
  descripcion="Nombre del rector y pagina web">
```

```
  <consultaSPARQL>
    <![CDATA[
    PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
    PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
    PREFIX dbpediaowl: <http://dbpedia.org/ontology/>
    PREFIX dbpprop: <http://es.dbpedia.org/property/>
    PREFIX dbres: <http://es.dbpedia.org/resource/>

    SELECT ?a ?b
    WHERE {
            dbres:ENTIDAD dbpprop:rector ?a .
            dbres:ENTIDAD dbpprop:sitioWeb ?b .
    }]]>
</consultaSPARQL>
</propiedad>
</tipoEntidad>

< tipoEntidad tipo="LOCATION">
…
</ tipoEntidad>

< tipoEntidad tipo="PERSON">
…
</ tipoEntidad>

</configuracion>
```

Ejemplo 20: Fichero de configuración de los procesos de enriquecimiento

### 4.3.2 Detalles del enriquecimiento de datos

Una vez tenidos en cuenta los pasos anteriores ya es posible enfrentarse al proceso de enriquecimiento de datos. Se describe a continuación, a modo de ejemplo, un proceso completo:

1. El proceso de etiquetado de videos comienza con la lectura del fichero de subtítulos de un video en formato TT. El fichero VTT presentará un aspecto similar al siguiente:

```
<?xml version="1.0" encoding="UTF-8"?>
<tt xmlns="http://www.w3.org/2006/04/ttaf1"
      xmlns:tts="http://www.w3.org/2006/04/ttaf1#styling" xml:lang="en">
  <head>
    <styling>
    </styling>
  </head>
  <body id="ccbody" style="defaultSpeaker">
  <div xml:lang="en">
    <p begin="00:00:00.68" end="00:00:03.22">En esta clase en la UNED ...
</p>
    <p begin="00:00:04.03" end="00:00:07.22">hablaremos sobre ... </p>
    …
  </div>
  </body>
</tt>
```

   Tras la lectura del fichero se generará una representación en memoria del contenido del fichero. Esto facilitará su posterior manipulación y la inclusión de las diferentes entidades reconocidas en las diferentes líneas de subtítulos.

2. A continuación se extraen las diferentes líneas de subtítulos y se realiza un proceso de reconocimiento de entidades nombradas (NER) para detectar las entidades presentes en

dicho subtitulo. La detección de entidades se realiza mediante el software STILUS a través de un servicio web. Dicho servicio web devuelve un conjunto de datos con un formato similar al siguiente:

```
<?xmlversion="1.0"encoding="utf-8"?>
<results>
  <status code="0">OK</status>
<result>
<entities>
  <entity>
    <form><![CDATA[ UNED ]]></form>
    <dictionary>s</dictionary>
    <semType>
      <type>FACILITY</type>
   …
    </semType>
    <semLinkedData><![CDATA[ C_UNED@ ]]></semLinkedData>
<variants>
  <variant>
    <form><![CDATA[ UNED ]]></form>
    <inip>20</inip><endp>23</endp>
  </variant>
</variants>
</entity>
</entities>
</result>
</results>
```

En este ejemplo se puede observar como se ha identificado la entidad "*UNED*" en la posiciones entre 20 y 23 de la primera línea de subtítulos y se ha identificado la entidad como de tipo FACILITY. Para sincronizar la aparición de cada entidad en cada línea de subtítulo, nuestra representación en memoria del fichero debe permitir asociar cada entidad al momento temporal de su aparición en el video.

3. El detector NER ha detectado la entidad UNED, pero no ha sido capaz de proporcionar la URI de la entidad en la DBPedia, por lo que será necesario buscar dicha URI mediante el DBPedia Lookup Service. Como resultado obtenemos que la URI de la UNED en el endpoint de DBPedia inglés es:

```
http://dbpedia.org/resource/National_University_of_Distance_Education
```

Dado que queremos trabajar con información en castellano será necesario realizar una consulta SPARQL para encontrar la URI equivalente en castellano, que será:

```
http://es.dbpedia.org/resource/Universidad_Nacional_de_Educación_a_Distancia
```

4. Una vez que se dispone de las URIs de las entidades, es posible aplicar diferentes tareas de enriquecimiento que dependerán del objetivo perseguido y que se definirán en un fichero de configuración. Para cada entidad detectada se revisarán los distintos procesos de enriquecimiento (consultas SPARQL) y si la entidad encaja (por su tipo y valores de sus propiedades) con alguno de ellos se ejecutará ese proceso para dicha entidad. A modo de ejemplo, se aplican los siguientes tareas de enriquecimiento de información:

   - Si la entidad detectada es una persona, se registra la entidad con el metadato "PERSON, FULL_NAME". Posteriormente se realizará una consulta SPARQL para

recuperar información sobre esa persona desde el endpoint SPARQL de la DBPedia. Se registrará esa información obtenida en el fichero de subtítulos, así como la URI a dicha entidad en la DBPedia y la consulta SPARQL que se utilizó. Esto permitiría a un reproductor de videos enriquecidos mostrar la información obtenida o recuperar información más reciente utilizando la propia consulta SPARQL.

- Si la entidad detectada es una localización (País, Ciudad, etc.), se registra la entidad con el metadato LOCATION y se registra la posición de dicha localización. Esto permitirá a un reproductor de videos enriquecidos buscar información sobre esa localización, mostrando un mapa de dicho lugar.

- Si la entidad detectada es una Universidad se obtiene el nombre de su rector y página web y un resumen de su descripción. Para el ejemplo que nos ocupa (UNED) la consulta SPARQL que recupera la información LOD de dicha universidad será:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dbpediaowl: <http://dbpedia.org/ontology/>
PREFIX dbpprop: <http://es.dbpedia.org/property/>
PREFIX dbres: <http://es.dbpedia.org/resource/>
SELECT ?a ?b
WHERE {
OPTIONAL { dbres:UNED dbpprop:rector ?a . }
OPTIONAL { dbres:UNED dbpprop:sitioWeb ?b . }
}          LIMIT 1
```

5. Como resultado de la ejecución se crea un fichero de subtítulos enriquecidos, que tendrá una estructura similar al fichero original pero al que se le ha añadido la información semántica apropiada. Un ejemplo sería el siguiente:

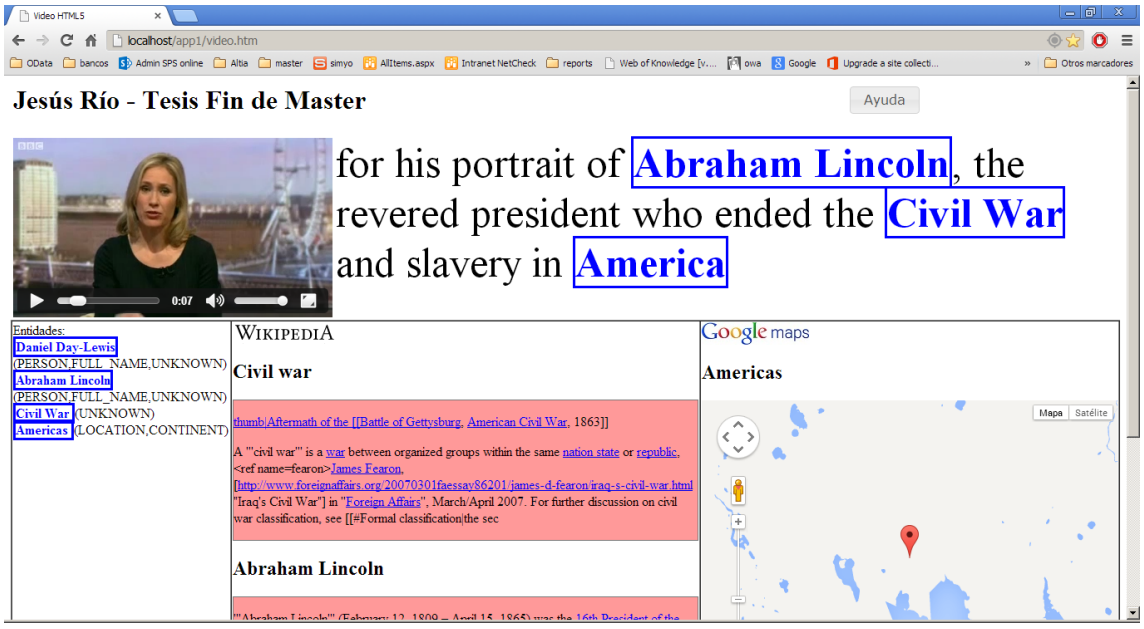
```
WEBVTT FILE

1
00:00:00.68 --> 00:00:03.22
{"texto":"En esta clase en la <b>UNED</b>   ",
"accion":"DBPedia",
"entidades":[
{"nombre":"UNED", "semType":"FACILITY",
"source":"National_University_of_Distance_Education",
"abstract": "La Universidad Nacional de Educación a Distancia (UNED), es
una universidad pública española de ámbito estatal… ",
"lod": "Juan A. Gimeno Ullastres http://www.uned.es ",
"lodDescription": "Nombre del rector y pagina web",
"sparql": "
   PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
   PREFIX dbpedia-owl:<http://dbpedia.org/ontology/>
   SELECT ?abstract WHERE {

http://dbpedia.org/resource/National_University_of_Distance_Education
      dbpedia-owl:abstract ?abstract .
   FILTER langMatches( lang(?abstract), 'es') }"}]}
```

Dicho fichero de salida ya incluye la información enriquecida (en este caso se ha detectado el nombre del rector y la página web de la entidad UNED, así como un resumen

explicando el origen de la entidad). Dicho fichero de subtítulos podría ser reproducido por un reproductor de videos.

## 4.4 Presentación de videos con contenido enriquecido

Una vez generados los ficheros de subtítulos enriquecidos, es necesario construir un reproductor de videos enriquecidos que muestre la información obtenida a partir de diversas fuentes Open Data. A modo de ejemplo se ha construido un portal web capaz de reproducir los videos procesados con sus correspondientes subtítulos. Se puede consultar dicho portal en http://jesusrio2.0fees.net/video.htm.




El portal construido se basa en las características de reproducción sincronizada de videos y subtítulos ofrecidas por HTML5. Mediante dichas características es posible reproducir un video y desencadenar de forma sincronizada un evento JavaScript que carga el subtítulo adecuado en cada momento junto con sus metadatos. Dicho evento JavaScript se encargará de leer los metadatos del subtítulo:

- Para una entidad de tipo PERSON se mostrará la descripción (*abstract*) incluida en el fichero de subtítulos enriquecido. Otra posibilidad sería que el reproductor utilizase la consulta SPARQL incluida en el fichero de subtítulos para obtener información más reciente que la incluida en el propio fichero.

- Para una entidad de tipo LOCATION se mostrará un mapa de Google Maps con dicha localización.

Finalmente indicar a nivel técnico, que el proceso de etiquetado de videos en el experimento, se ha realizado mediante un desarrollo Java que utiliza ARQ: el motor de consultas SPARQL de

JENA[1]. JENA es un proyecto de Apache que incorpora diversas herramientas y librerías para desarrollar aplicaciones basadas en Web Semántica y Datos Enlazados.

La presentación de videos con contenido enriquecido se ha desarrollado con HTML5 y JavaScript para el navegador Google Chrome 31. Tanto los videos como ficheros de subtítulos son reconocidos por HTML5 de forma nativa por lo que no es necesaria la utilización de *plugins* adicionales. Esta presentación, podría mejorarse, pero no ha sido considerado un objetivo principal del trabajo TFM.

El código fuente de los prototipos desarrollados para este TFM se ha publicado en SourceForge[2], en el proyecto LinkedDataGenerator[3] y están disponibles para cualquier persona interesada. El código fuente incluye tanto el generador que enriquece las transcripciones de videos como el propio reproductor de videos.

[1] http://jena.apache.org
[2] http://sourceforge.net
[3] http://sourceforge.net/projects/linkeddatagenerator

## 4.5 La publicación de datos

A la hora de afrontar un proceso de enriquecimiento de datos basado en Linked Open Data es fundamental conocer las diferentes herramientas, los distintos estándares utilizados y los diferentes patrones de diseño que dan solución a problemas muy comunes.

Existen una serie de patrones de diseño de Linked Data bien documentados [35] y una serie de buenas prácticas [36] que es conveniente seguir. Se exponen a continuación algunos patrones útiles para la creación de datos LOD aunque es conveniente conocerlos todos para poder navegar por los distintos datasets con cierta soltura.

### 1. Patrones de Identificación

Una de las características más importantes del enfoque de Linked Data es la utilización de identificadores universales en la web: URIs. Cualquier cosa que se publique en la Web debe tener una URI que permita referenciarla, navegar a través de ella o enlazarla desde otras URIs. En un proyecto de Linked Data es fundamental establecer un esquema de identificadores apropiado, esto determinará como se asignan las URIs a los recursos.

Se exponen a continuación algunos patrones de identificación.

#### 1.1 URIs Patrón (Patterned URIs)

| **Contexto** |
|---|
| *¿Cómo crear URIs más predecibles y legibles por humanos?* |
| **Descripción** |
| URIs claras y predecibles son más fáciles de reutilizar, especialmente si siguen un patrón común. |
| **Solución** |
| Deben crearse URIs que sigan patrones de nombrado sencillos. En Linked Data es habitual usar nombres en plural como parte de la URI. Por ejemplo, una aplicación que publique datos sobre libros podría utilizar URIs con la forma: "/libros/12345" donde "/libros" es la base de la URI que indica una colección de libros y "12345" sería el identificador de un libro individual. |

#### 1.2 URIs Jerárquicas (Hierarchical URIs)

| **Contexto** |
|---|
| ¿Cómo deberían asignarse las URIs a un grupo de recursos que forman una jerarquía natural? |
| **Descripción** |
| Frecuentemente colecciones de recursos forman una jerarquía natural. Ej: Capítulos de un libro, departamentos de una organización,… Al reflejar esa jerarquía en la URI permitimos a los usuarios o desarrolladores navegar por dicha jerarquía. |
| **Solución** |
| En un sistema que publique datos sobre libros y sus capítulos podríamos usar la siguiente URI para identificar el capítulo 1 de un libro: "libros/12345/capítulos/1". La URI "/capítulos" refleja de forma natural la colección de capítulos de un libro concreto. |

## 2. Patrones de Modelado

Para la creación de un modelo de dominio para publicarlo como Linked Data es necesario conocer las distintas entidades con sus atributos y las relaciones entre ellas.

- Utilizando RDF Schema y OWL es posible compartir vocabularios (ontologías o esquemas) en la Web de la misma forma que se comparten datos.
- No existe una separación entre el modelo físico y el lógico de los datos. El modelo lógico define cómo se almacenan los datos. Un repositorio de tripletes RDF está "libre de esquema" en el sentido de que no tiene que definir su esquema físico (tablas, columnas…). Cualquier repositorio RDF puede almacenar cualquier dato RDF en cualquier vocabulario RDF.

Los patrones de modelado ilustran el uso de determinadas características de RDF para el modelado de datos.

### 2.1 Etiqueta Todo (Label Everything)

| **Contexto** |
|---|
| *¿Cómo podemos asegurar que cada recurso posee un nombre legible por humanos?* |
| **Descripción** |
| En un dataset pueden existir diferentes entidades. Algunas serán muy simples y obvias (personas, organizaciones) mientras que otras pueden ser más complejas (por ejemplo una observación hecha en un momento concreto bajo determinadas condiciones). No siempre es claro para un usuario que navega por un grafo RDF lo que representa un recurso. |
| **Solución** |
| Asegurar que cada recurso del datase posee una propiedad *rdfs:label* |

### 2.2 Enlaza, no etiquetes (Link Not Label)

| **Contexto** |
|---|
| *¿Cómo modelar un dataset para maximizar los beneficios de un modelo basado en grafos?* |
| **Descripción** |
| La mayoría de los datasets se diseñan sobre unos tipos de recursos principales. Por ejemplo un dataset sobre redes sociales se puede centrar en personas, grupos y organizaciones. Suelen existir además otro tipo de recursos que a menudo son subestimados en la fase de modelado y se representan como simples valores literales. Esto hace que las consultas sean menos eficientes y limita su capacidad de anotación, por ejemplo al hacer links de equivalencia. |
| **Solución** |
| Asegurar que todos los recursos de un dataset son modelados como recursos en vez de como valores literales. Relacionar los recursos entre sí para crear un modelo de grafo más rico. Usar los valores literales como etiquetas de los nuevos recursos creados. Ejemplo de recursos que se suelen subestimar son: Idiomas, Géneros, Formatos, Códigos de País,… |

## 3. Patrones de Publicación

Existe un gran número de organizaciones que publican LOD de formas muy variadas. Independientemente del origen de los datos y los medios de publicación existe un gran número de patrones o buenas prácticas de publicación que hacen que los datos sean más fáciles de descubrir, enriquecer y entrelazar.

### 3.1 Enlaces de equivalencia (Equivalence Links)

| **Contexto** |
|---|
| *¿Cómo indicar que diferentes URIs se refieren al mismo recurso o concepto?* |
| **Descripción** |
| En la nube Linked Data los datos se publican de una forma descentralizada donde muchas personas y organizaciones publican datos sobre los mismos recursos. Aunque es recomendable reutilizar los identificadores, no siempre es posible. |
| **Solución** |
| Utilizar las propiedades *owl:sameAs* o *skos:exactMatch* para indicar que dos URIs son equivalentes. |

### 3.2 Autodescubrimiento de asunto principal (Primary Topic Autodiscovery)

| **Contexto** |
|---|
| *¿Cómo conocer el tema principal de una determinada página web?* |
| **Descripción** |
| A menudo una página web se refiere a un determinado objeto físico o recurso. El autor que la publica puede incluir la URI de dicho recurso de forma explícita. |
| **Solución** |
| Al publicar una página web incluir un elemento link en la cabecera de la página web apuntando a la URI del recurso que identifica el asunto principal de la página. Se debe utilizar el valor *primarytopic* para el atributo *rel*:<br><link rel="primarytopic" href="http://dbpedia.org/resource/London"/> |

### 3.3 Ver además (See Also)

| **Contexto** |
|---|
| *¿Cómo enlazar los documentos RDF para permitir a los crawlers navegar entre ellos?* |
| **Descripción** |
| Los datos enlazados se descubren mediante la desrreferenciación de las URIs. Comenzando por una URI, un agente software puede descubrir otras URIs. En algunas ocasiones es útil incluir links adicionales a otros recursos de forma explícita. Estos enlaces no son relaciones semánticas propiamente dichas sino enlaces a otros recursos. |
| **Solución** |
| Utilizar la propiedad *rdfs:seeAlso* para enlazar a otros documentos RDF |

## 4. Patrones de Aplicación

En este apartado se incluyen algunos patrones de buenas prácticas a la hora de crear aplicaciones basadas en Linked Data.

### 4.1 Sigue tu instinto (Follow your nose)

| **Contexto** |
|---|
| *¿Cómo encontrar datos adicionales relevantes en la web?* |
| **Descripción** |
| Al recuperar datos de una URI es probable que exista contenido adicional en esa fuente de datos o en otras de la web. |
| **Solución** |
| Identificar enlaces adicionales útiles en los datos disponibles y desrreferenciar esas URIs una a una para encontrar datos adicionales. |

### 4.2 Recuperador de URIs (URI Resolver)

| **Contexto** |
|---|
| *¿Cómo personalizar el comportamiento de las aplicaciones asociado a la recuperación o desrreferenciación de una URI en sentencias RDF?* |
| **Descripción** |
| Las aplicaciones de Linked Data suelen adquirir datos relevantes adicionales mediante el comportamiento "Sigue tu instinto": asumir que cualquier URI en un grafo puede ser desrreferenciable para obtener nuevos datos. Sin embargo, en la práctica, la desrreferenciación simple, es decir, realizar peticiones GET sobre una URI no siempre es deseable, por ejemplo:<br>- Una aplicación móvil podría necesitar funcionar en modo off-line cuando los datos remotos no están disponibles.<br>- Podría existir un mirror local que proporcione una mejor calidad de servicio.<br>- Un servicio remoto podría estar no disponible de forma intermitente. |
| **Solución** |
| Las aplicaciones deberían dirigir sus peticiones de desrreferenciación a un Recuperador de URIs que no es más que una función que mapea URIs a ficheros RDF. De esta forma en la recuperación de URIs se permitiría:<br>- Cacheo de documentos RDF tanto en memoria como en sistema de ficheros.<br>- Redirección a mirrors locales para mejorar el rendimiento o cuando el servicio remoto no esté disponible.<br>- Proporcionar tareas de razonamiento sobre los datos recuperados |

### 4.3 Que no esté no quiere decir que no exista (Missing Isn't Broken)

| **Contexto** |
|---|
| *¿Cómo tratar con los datos desorganizados o incompletos que existen en la Web?* |
| **Descripción** |
| En RDF cualquiera puede decir cualquier cosa, en cualquier lugar. Cualquiera puede añadir información (afirmaciones) sobre un recurso y publicarla en la web para que los demás lo utilicen. No hay requisitos sobre cuantos datos deben ser publicados: no hay reglas de validación que requieran una cantidad de datos mínima. Esto significa que los datos pueden variar mucho en calidad y detalle, como consecuencia de la flexibilidad del modelo. |
| **Solución** |
| Admite que "algo que no esté no quiere decir que no exista". Las aplicaciones deben ser tolerantes a datos inválidos o no existentes y hacer más esfuerzo en procesar los datos disponibles.<br>Por supuesto, una aplicación debe requerir un mínimo de datos para poder hacer algo útil con ellos. Aunque si el publicador de los datos ha seguido el patrón "Etiqueta Todo" al menos será posible procesar el nombre del recurso. |

### 4.4 Consulta de Aseveración (Assertion Query)

| **Contexto** |
|---|
| *¿Cómo probar un dataset para detectar un patrón?* |
| **Descripción** |
| En ocasiones es útil probar patrones en un dataset. Por ejemplo para saber si hay datos disponibles sobre un recurso determinado o de un vocabulario específico. |
| **Solución** |
| Usar la sentencia ASK o CONSTRUCT de SPARQL para probar el dataset. Por ejemplo, para saber si existen datos sobre un recurso concreto:<br><br>ASK WHERE { <http://www.example.org/person/bob> ?p ?o.} |

# Capítulo 5. Experimentación y pruebas

Los experimentos consisten en realizar un proceso de enriquecimiento a transcripciones de videos en castellano de la Cadena Campus Plataforma AVIP[1] de la UNED, disponibles con transcripción (20 en octubre 2013). La temática de los videos es variada pero todos pertenecen a clases online que imparten los profesores de la UNED, y por lo tanto son videos educativos mutidominio (de temática diversa), la mayoría contienen textos de Química y en menor medida Derecho, Informática, etc. El formato de los videos es mp4. Se dispone de las transcripciones textuales de los mismos en formato TT (Timed Text Authoring Format).

| Fichero | Nombre | Número de palabras |
|---|---|---|
| 1 | 01LuisVazquezLopezcaptions.dfxp.xml | 1577 |
| 2 | 02BegonaSanchezRouraLugo.mp4captions.dfxp.xml | 3400 |
| 3 | 03JoseLopezRiopedrecaptions.dfxp.xml | 2526 |
| 4 | 04LugoRosaMRicoyCasascaptions.dfxp.xml | 3166 |
| 5 | 05TransformadadeLaplacecaptions.dfxp.xml | 1898 |
| 6 | 06LosAlcanosteoriasubtitulos.dfxp.xml | 4000 |
| 7 | 07CicloAlcanosCiclohexanos.dfxp.xml | 6541 |
| 8 | 08LosAlcanosejerciciossubtitulos.dfxp.xml | 2398 |
| 9 | 09IntroduccionQuimicaOrg.dfxp.xml | 8256 |
| 10 | 10Halurosdealquilo.dfxp.xml | 8510 |
| 11 | 11QuimicaOrganicaIJpalop.dfxp.xml | 8296 |
| 12 | 12QuimicaIHalogenurosDeAlquilo.dfxp.xml | 3315 |
| 13 | 13QuimicaOrganicaCarlaIsabelNietoTema6Ejercicios.dfxp.xml | 3097 |
| 14 | 14QuimicaMiguelAngelVazquez.dfxp.xml | 5249 |
| 15 | 15QuimicaJPalop.dfxp.xml | 7027 |
| 16 | 16QuimicaCarlaIsabel.dfxp.xml | 2239 |
| 17 | 17Dienos.dfxp.xml | 1899 |
| 18 | 18QuimicaOrganicaAlquinos.dfxp.xml | 4406 |
| 19 | 19QuimicaJpalop.dfxp.xml | 6407 |
| 20 | 20Alquenos.dfxp.xml | 7785 |

Tabla 6: Ficheros que componen el Corpus

[1] http://www.intecca.uned.es/portalavip/cadenaCampus.php

## 5.1 Procesos de enriquecimiento basados en LOD

Para la realización de los experimentos se ha definido el proceso de enriquecimiento de datos basado en LOD con varias reglas. Este proceso está codificado mediante SPARQL en el fichero de configuración de la aplicación de enriquecimiento que se puede consultar en la sección de Anexos. Los procesos a realizar dependen del tipo de la entidad y realizan las siguientes tareas:

- Para entidades de tipo PERSONA:
    - Para cualquier persona se obtiene su fecha de nacimiento y la actividad a la cual se dedica. Si, por ejemplo, su profesión es “Filósofo” se obtienen los libros de los cuales es autor.

- Para entidades de tipo LOCALIZACION, se ha probado:
    - Si se trata de un país se obtienen los científicos famosos de dicho país.
    - Si se trata de una comunidad autónoma se obtiene el nombre de su capital.
    - Para cualquier localización se obtiene su número de habitantes.

- Para entidades de tipo ORGANIZACIÓN se recupera su nombre, sede y página web.

- Para entidades de tipo SERVICIOS o INFRAESTRUCTURAS (FACILITY)
    - Si el tipo de servicio es una UNIVERSIDAD se recupera el nombre de su rector y su página web.

- Para entidades de tipo PRODUCTO: Si el producto es una página web se obtiene el nombre de su creador y el eslogan del producto.

- Para entidades de tipo DESCONOCIDO se obtiene la lista de categorías de la DBPedia a las cuales pertenece dicha entidad.

- Para entidades sin ningún tipo, se recuperan las categorías de la Wikipedia a las que pertenece la entidad.

- Además, para todas las entidades se obtiene su descripción de la DBPedia, y si no es posible se obtiene su descripción en Freebase.

Este enriquecimiento es útil para la prueba de concepto de la propuesta, pero uno de los trabajos futuros realizar es encontrar reglas genéricas que se adapten al dominio identificado del vídeo, la sección etc. Una vez decididos los procesos de enriquecimiento se describen a continuación algunos casos concretos de éxito y algunos donde se producen errores.

### 5.1.1 Ejemplos de enriquecimientos de datos con éxito

Cuando se detecta una entidad de tipo LOCALIZACION y se corresponde con un País, se define el proceso de enriquecimiento siguiente para recuperar el nombre de científicos relevantes de ese país:

| | |
|---|---|
| **Proceso de enriquecimiento:** | Obtención de los científicos relevantes de un país. |
| **Tipo de Entidad:** | LOCATION |
| **Propiedad a considerar:** | dbpprop:país = * |
| **Consulta SPARQL:** PREFIX dbpprop: <http://es.dbpedia.org/property/> | |
| **Ejemplo de Resultados para:** ENTIDAD = España | |
| | http://es.dbpedia.org/resource/Santiago_Ramón_y_Cajal<br>http://es.dbpedia.org/resource/Severo_Ochoa<br>http://es.dbpedia.org/resource/Josep_Trueta<br>http://es.dbpedia.org/resource/Margarita_Salas<br>http://es.dbpedia.org/resource/Bernardo_Rodríguez_Largo |

Tabla 7: Ejemplo de un proceso de enriquecimiento LOD

La consulta SPARQL asociada utiliza el concepto de "Categorías" de la DBPedia. Tras la exploración de dicho sistema de categorías se conoce que existen categorías que contienen listas de científicos de cada país y que presentan el formato "*Científicos_de_PAIS*" de esta forma existen las categorías: "*Científicos_de_España*[1]", "*Científicos_de_Francia*[2]", etc. Podemos por tanto asumir que cuando nos encontremos un nombre de país es muy probable que exista en la DBPedia una categoría de ese estilo para ese país. Se construye por tanto, una consulta SPARQL que toma como parámetro el nombre de un país y a través de un reemplazo del texto ENTIDAD por el nombre del país devolverá los datos deseados.

Otros ejemplos de enriquecimiento con éxito son los siguientes:

| **Entidad (Tipo)** | **LOD recuperado** | **Descripción LOD** |
|---|---|---|
| Julian_Assange (PERSON) | [Australia, Periodista ] | [Lugar de nacimiento, profesión] |
| WikiLeaks (PRODUCT) | [Julian_Assange, Abrimos a los gobiernos] | [Autor, eslogan] |
| UNED (FACILITY) | [Alejandro Tiana Ferrer, http://www.uned.es] | [Nombre del rector, página web] |
| Transformada de Laplace (UNKNOWN) | http://es.dbpedia.org/page/Categoría:Transformaciones_integrales | Categorías a las que pertenece |
| Dialquilcuprato (UNKNOWN) | http://es.dbpedia.org/page/Categoría:Compuestos_de_litio<br>http://es.dbpedia.org/resource/Categoría:Compuestos_de_cobre | Categorías a las que pertenece |
| Propilciclohexano (UNKNOWN) | http://es.dbpedia.org/resource/Categoría:Alcanos<br>http://es.dbpedia.org/resource/Categoría:Hidrocarburos | Categorías a las que pertenece |
| Francia (LOCATION) | http://es.dbpedia.org/resource/André_Lwoff<br>http://es.dbpedia.org/resource/Jean_Antoine_ | Científicos del país |

[1] http://es.dbpedia.org/resource/Categoría:Científicos_de_España
[2] http://es.dbpedia.org/resource/Categoría:Científicos_de_Francia

| | | |
|---|---|---|
| | Nollet<br>http://es.dbpedia.org/resource/Jean_Picard<br>http://es.dbpedia.org/resource/Henri_Pitot<br>http://es.dbpedia.org/resource/Henry_Darcy | |

En algunos procesos de enriquecimiento se define la recuperación de unos datos concretos sobre una entidad, por ejemplo, para una entidad de tipo PRODUCT se recupera el nombre de su autor y su eslogan. De esta forma para la entidad "*WikiLeaks*" se han recuperado los datos enlazados "*Julian Assange*" y "*Abrimos los gobiernos*". Para las entidades que no tienen un tipo definido en este trabajo, se ha recuperado las categorías de la Wikipedia a la cual pertenece la entidad. Por ejemplo para la entidad "*Transformada de Laplace*" se ha detectado que pertenece a la categoría "*Transformaciones Integrales*" lo cual permitiría seguir el enlace de dicha categoría para obtener otros términos que pertenecen a la misma y se puede considerar relacionados por dicha categoría: "*Transformada de Hilbert*", "*Transformada de Radón*", etc.

### 5.1.2 Clases de enriquecimiento erróneo de datos con LOD

Existen ciertas situaciones en las que un proceso de enriquecimiento de datos puede no obtener información relacionada o la información que recupere no sea adecuada, algunos de los más habituales son:

- **El NER no detecta la entidad**. La detección de entidades realizada por el NER es el punto de partida de todo proceso de enriquecimiento. Si una entidad no es detectada por el NER será imposible su enriquecimiento. Más adelante se muestra la precisión alcanzada con el NER utilizado de Stilus.

- **El NER detecta la entidad pero no es posible asignar una URI a la entidad**. Sin una URI válida es imposible consultar información adicional. Ej: El compuesto químico "*Ciclododecano*" (aparece en el video 7) es detectado por el NER pero ninguno de los sistemas de asignación de URIs es capaz de encontrar una URI válida para el mismo. Esta situación puede ocurrir cuando el NER identifica entidades con muy poca relevancia o desconocidas en LOD, pero sucede en un porcentaje muy bajo de casos.

- **El NER detecta la entidad y se le asigna una URI que guarda poca relación con la misma (ambigüedad)**. Ej: En el video 11 el isómero "CIS" no es capaz de resolverse correctamente y en vez de identificarse como un compuesto químico se identifica como el "*Centro de Investigaciones Sociológicas*". Como consecuencia el proceso de enriquecimiento recuperará información con muy poca relación con la entidad original. Esta situación ocurre con algunos términos químicos muy específicos que se identifican con abreviaturas o acrónimos: RR, E2, SP2, PCC. Este caso está reflejado uno de los trabajos futuros, en el que se plantea la identificación del dominio temático, para apoyar la desambiguación.

Se muestra a continuación un listado de algunas entidades para las que no ha sido posible detectar información o desambiguar:

| Entidad | Tipo | Motivo |
|---|---|---|
| Ciclododecano | UNKNOWN | Ha sido imposible obtener su URI en la nube LOD |
| Pepe Riopedre | (PERSON) | Ha sido imposible obtener su URI en la nube LOD |
| San Clemente | LOCATION | El término hace referencia a una localidad italiana, en cambio el servicio DBPedia Lookup le ha asignado la URI de una ciudad de California. |
| Newman | UNKNOWN | El término hace referencia a las "Proyecciones de Newman", un tipo de representación química. La detección de la entidad por el NER no ha sido adecuada y se le ha asignado la URI del actor "Paul Newman". |
| London | UNKNOWN | El término London hace referencia a un tipo de fuerzas moleculares. La detección de la entidad por el NER no ha sido adecuada y en su lugar se le ha asignado la URI de la ciudad de Londres. |
| Teoría del Estado I | UNKNOWN | El texto hace referencia al nombre de una asignatura. La Wikipedia le asigna la URI de una política chilena (Evelyn_Matthei) que guarda muy poca relación con el término original. |
| ilo | UNKNOWN | El término original hace referencia al sufijo "ilo" utilizado en muchos términos químicos. La detección de la entidad por el NER no ha sido adecuada y en su lugar se le ha asignado la URI de la ciudad de la ciudad peruana Ilo. |
| Octa | UNKNOWN | El término original hace referencia al prefijo "octa" utilizado en muchos términos químicos. La detección de la entidad por el NER no ha sido adecuada y en su lugar se le ha asignado la URI del término musical "Octava". |
| ciano | UNKNOWN | El término original hace referencia a un grupo químico. La detección de la entidad por el NER no ha sido adecuada y en su lugar se le ha asignado la URI del político italiano Galeazzo_Ciano. |
| Py | UNKNOWN | El término original hace referencia a un nivel químico. La detección de la entidad por el NER no ha sido adecuada y en su lugar se le ha asignado la URI de una localidad francesa. |
| Lewis | PERSON | El término original hace referencia a un término químico "Estructuras de Lewis". La detección de la entidad por el NER no ha sido adecuada y en su lugar se le ha asignado la URI del escritor "Lewis Carrol". |
| NaSH | UNKNOWN | El término original hace referencia a una formula química. La detección de la entidad por el NER no ha sido adecuada y en su lugar se le ha asignado la URI de una localidad americana: Nash_(Dakota_del_Norte). |

## 5.2 Análisis de resultados

Tras la ejecución de los procesos se ha realizado el análisis siguiente. Se muestra en la Tabla 8 el número de entidades detectadas de cada tipo por Stilus. Se observa que predominan las entidades de tipo Localización, Persona y Organización. Son muy escasas las entidades de tipo Infraestructura (Facility) y hay bastantes entidades de tipo Desconocido (87).

En la Tabla 8 también se puede observar que para algunas entidades reconocidas, Stilus no ha sido capaz de asignar un tipo, aunque sí detectarlas. Solo se asigna un tipo a un 28.6% de las entidades reconocidas. Este dato también refleja que el reconocimiento de entidades es muy dependiente del dominio y uno de los trabajos futuros planteados es cómo añadir contexto que permita identificar entidades ya existentes en recursos LOD.

| Fichero | Location | Organization | Facility | Person | Product | Unknown |
|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 1 | 0 | 0 | 2 |
| 2 | 7 | 3 | 1 | 3 | 0 | 10 |
| 3 | 7 | 1 | 0 | 4 | 1 | 7 |
| 4 | 10 | 3 | 1 | 6 | 0 | 5 |
| 5 | 1 | 0 | 0 | 5 | 0 | 4 |
| 6 | 5 | 3 | 0 | 5 | 1 | 9 |
| 7 | 3 | 3 | 0 | 0 | 2 | 7 |
| 8 | 1 | 0 | 0 | 1 | 2 | 2 |
| 9 | 1 | 2 | 0 | 5 | 0 | 5 |
| 10 | 2 | 5 | 0 | 4 | 1 | 5 |
| 11 | 2 | 2 | 0 | 4 | 1 | 5 |
| 12 | 1 | 4 | 0 | 0 | 1 | 1 |
| 13 | 1 | 6 | 0 | 0 | 0 | 0 |
| 14 | 5 | 13 | 0 | 2 | 3 | 4 |
| 15 | 2 | 1 | 0 | 6 | 1 | 4 |
| 16 | 2 | 2 | 0 | 3 | 2 | 5 |
| 17 | 1 | 1 | 0 | 5 | 0 | 0 |
| 18 | 3 | 8 | 0 | 1 | 7 | 5 |
| 19 | 1 | 2 | 0 | 6 | 2 | 2 |
| 20 | 3 | 5 | 0 | 7 | 0 | 5 |
| **Fichero** | **Location** | **Organization** | **Facility** | **Person** | **Product** | **Unknown** |
| **Total** | **59** | **64** | **3** | **67** | **24** | **87** |

**Tabla 8: Número de entidades detectadas de cada tipo de entidad**

Se muestra en la Tabla 9 el número de entidades detectadas por STILUS en cada fichero, el número de entidades a las cuales ha sido posible asignar una URI y se detalla cuál de los mecanismos de asignación de URI (Stilus, DBPedia Lookup Service, Sindice, Wikipedia) la asignó y el número de entidades con descripción. El orden en el que se aplica cada método de asignación de URIs viene determinado por la calidad esperada de la asignación de URI:

- Se considera que las URIs que asigna Stilus tienen una gran precisión ya que el NER dispone información de contexto (al recibir el texto completo de la transcripción) por lo que es el método utilizado en primer lugar.

- Las entidades para las que no se ha podido identificar por Stilus su URI, se envían al DBPedia Lookup Service (DLS). Se trata de un servicio de asignación de URIs a términos

de búsqueda "relacionados". Se consideran términos relacionados aquellos que coinciden con la etiqueta del recurso o con el texto de un enlace que se use frecuentemente en la Wikipedia para referirse al recurso. Por ejemplo el recurso http://dbpedia.org/resource/United_States puede ser encontrado al buscar el término "USA". Los resultados se evalúan por el número de enlaces que apuntan desde otras páginas de la Wikipedia a la página resultado. Se ha elegido este método en segundo lugar porque es un servicio específico de búsqueda de URIs, que aunque no recibe información del contexto de la entidad, consigue buenos resultados.

- Las entidades que no es posible identificar con los métodos anteriores se envían a la API de la Wikipedia. Se trata de un servicio genérico de búsqueda que puede devolver las URIs de la Wikipedia donde se referencian los términos buscados.

- El resto de entidades se envían al buscador semántico Sindice indicando como parámetros la búsqueda de resultados en formato RDF y en el dominio "dbpedia.org", ya que no se observa que el dominio español de la DBPedia esté indexado por Sindice. Las pruebas realizadas no han obtenido buenos resultados en español (aunque sí que se obtienen muy buenos en inglés) por lo que se ha decidido dejar este método en último lugar.

| | | Total de URIs obtenidas con | | | | | Descripción de entidad obtenida con | | |
|---|---|---|---|---|---|---|---|---|---|
| **Fich. Nro.** | **1. Entidades Detecta-das** | **2. TOTAL con URI** | **3. STILUS** | **4. DLS** | **5. Wikipedia API** | **6. Sindice** | **7. DBPedia** | **8. Freebase** | **9. Entidades con LOD** |
| **1** | 4 | 4 | 1 | 1 | 2 | 0 | 3 | 0 | 3 |
| **2** | 24 | 24 | 3 | 6 | 15 | 0 | 18 | 0 | 16 |
| **3** | 20 | 19 | 1 | 10 | 8 | 0 | 13 | 0 | 14 |
| **4** | 25 | 25 | 2 | 13 | 10 | 0 | 16 | 0 | 18 |
| **5** | 10 | 9 | 0 | 3 | 6 | 0 | 8 | 0 | 6 |
| **6** | 23 | 23 | 1 | 7 | 15 | 0 | 20 | 0 | 12 |
| **7** | 15 | 14 | 0 | 3 | 10 | 0 | 13 | 0 | 8 |
| **8** | 6 | 6 | 0 | 2 | 4 | 0 | 5 | 0 | 2 |
| **9** | 13 | 12 | 0 | 2 | 10 | 0 | 11 | 0 | 5 |
| **10** | 17 | 16 | 0 | 6 | 10 | 0 | 16 | 0 | 7 |
| **11** | 14 | 14 | 0 | 2 | 12 | 0 | 10 | 0 | 8 |
| **12** | 7 | 6 | 0 | 1 | 5 | 0 | 6 | 0 | 4 |
| **13** | 7 | 7 | 0 | 2 | 5 | 0 | 4 | 0 | 5 |
| **14** | 27 | 27 | 0 | 8 | 19 | 0 | 27 | 0 | 6 |
| **15** | 14 | 14 | 0 | 4 | 10 | 0 | 9 | 0 | 6 |
| **16** | 14 | 14 | 0 | 3 | 11 | 0 | 11 | 0 | 5 |
| **17** | 7 | 7 | 0 | 4 | 3 | 0 | 6 | 0 | 0 |
| **18** | 24 | 21 | 0 | 6 | 16 | 0 | 18 | 0 | 7 |
| **19** | 13 | 13 | 0 | 2 | 11 | 0 | 9 | 0 | 2 |
| **20** | 20 | 20 | 0 | 9 | 11 | 0 | 9 | 0 | 10 |
| **Fich. Nro.** | **1. Entidades Detecta-das** | **2. TOTAL con URI** | **3. STILUS** | **4. DLS** | **5. Wikipedia API** | **6. Sindice** | **7. DBPedia** | **8. Freebase** | **9. Entidades con LOD** |
| **Total** | **304** | **295** | **8** | **94** | **193** | **0** | **232** | **0** | **144** |

Tabla 9: Detalles sobre las entidades detectadas en cada fichero.

Se describe a continuación con más detalle, el significado de las columnas de la Tabla 9:

1. Entidades Detectadas por NER: Número de entidades detectadas por el NER Stilus. El NER proporcionará URIs para algunas de ellas. Nuestro sistema deberá buscar las URIs de aquellas entidades que el NER no las proporcione.

2. Entidades Con URI: Numero de entidades a las cuales se ha conseguido asignar una URI por alguno de los métodos disponibles: STILUS, DLS, Wikipedia API, o Sindice.

3. URIs STILUS: Número de entidades a las cuales el NER STILUS ha asignado una URI. En los experimentos se constata que STILUS asigna URIs a un porcentaje muy bajo de las entidades detectadas (2,7%) por lo que es necesario recurrir a otros métodos de asignación de URIs.

4. URIs DLS: Número de entidades a las cuales se les ha asignado una URI utilizando el DBPedia Lookup Service (DLS). Solo se envían a la DLS aquellas entidades a las que el NER ha identificado, pero no ha podido asignar su URI.

5. URIs Wikipedia API: Número de entidades a las cuales se les ha asignado una URI utilizando la API de la Wikipedia.

6 URIs Sindice: Número de entidades a las cuales se les ha asignado una URI utilizando Sindice (recurso en inglés).

7. Desc DBPedia: Número de entidades para las que se ha encontrado la descripción o *abstract* con la DBPedia.

8. Desc Freebase: Número de entidades para las que se ha encontrado la descripción o *abstract* de Freebase (recurso en inglés).

9. Entidades con LOD: Número de entidades a las que ha sido posible asignar correctamente información enriquecida proveniente de la nube Linked Open Data. Es posible que aunque una entidad esté perfectamente identificada por su URI no se pueda recuperar información LOD porque los procesos de enriquecimiento no estén definidos para ese tipo de entidad.

Algunas conclusiones a partir de los resultados anteriores son:

- La Tabla 9 muestra que se ha asignado una URI a la mayoría de las entidades detectadas (295 de las 304). Stilus ha proporcionado pocas URIs (sólo 8) lo cual refleja que los documentos que forman el corpus presentan entidades difíciles de detectar lexicalmente, o que son de dominios muy concretos.
- La mayoría de las URIs han sido asignadas por DLS (94) y Wikipedia (193).
- Finalmente ha sido posible enriquecer con información LOD a un 47,3% de las entidades detectadas y asociar una descripción a un 76,3% de las entidades.

Como ya se ha comentado en este tipo de trabajos de enriquecimiento de datos es muy difícil hacer una evaluación de laboratorio, con medidas estándar como la precisión y la cobertura, ya que al tratarse de procesos donde se descubre información nueva, no existe un mecanismo

claro para determinar la calidad final del proceso y debe de realizarse manualmente. Por ello sería necesario disponer de un corpus de pruebas que incluyan los resultados esperados para poder hacer una comparación de resultados. Desafortunadamente no se ha encontrado ninguno de estas características y en castellano utilizable para este trabajo. Por lo tanto, no se han obtenido datos de precisión y cobertura ya que los datos seleccionados en el corpus requerirían de personal cualificado que revisara manualmente los textos.

En la bibliografía, parece consensuado que la precisión varía mucho dependiendo de la calidad de asignación de la URI a cada entidad. En nuestro caso, la precisión es bastante buena cuando la entidad ha sido identificada por el sistema NER de Stilus. Stilus recibe el texto completo para realizar la detección de entidades por lo que dispone de información de contexto para asignar la URI. La asignación ha sido peor en las pruebas realizadas con Sindice que no ofrece resultados para el español. En general se ha constatado las dificultades existentes para realizar los procesos de enriquecimiento al carecer las distintas herramientas disponibles como DLS o Sindice de versiones adaptadas al castellano. Aunque es posible realizar las búsquedas y posteriores traducciones mediante enlaces *rdf:sameAs* es muy probable que los resultados fueran mejores si dispusiéramos de versiones localizadas de las herramientas en nuestro idioma.

## 5.3 Calidad del enriquecimiento de los datos

Sobre la calidad de la información que enriquece a los datos indicar que:

- La calidad de los datos enriquecidos es dispar y depende de la correcta detección de la URI de la entidad. Si la URI asignada a una entidad es correcta, la recuperación de datos LOD es muy probable que tenga éxito. Si la URI asignada a un término es errónea por cuestiones de ambigüedad, los datos enlazados no serán apropiados.

- Estos resultados no son una comparativa de la precisión de los diferentes mecanismos de obtener una URI a partir de un concepto, ya que se debe tener en cuenta que DLS sólo intenta detectar las URIs que Stilus no ha podido y la Wikipedia API sólo intenta detectar las URIs que los otros dos sistemas no han sido capaces de resolver. Un trabajo futuro sería identificar las URIs de todas las entidades con los tres recursos y comparar o agregar la información.

- La calidad de los resultados finales también puede verse afectada por la calidad de la transcripción generada a partir de los objetos multimedia. Las transcripciones de usuarios humanos pueden incluir errores o erratas y las generadas automáticamente por sistemas ASR también pueden presentar deficiencias. Utilizar otra información de los metadatos para identificar el contexto (sin saber muy bien cómo definirlo en este momento) es otro de los posibles trabajos planteados para un futuro inmediato.

- Se generan asignaciones erróneas de URIs a partir de abreviaturas. Gran parte de los textos pertenecen a clases de Química y contienen multitud de nombres abreviados de fórmulas químicas: RR, E2, SP2, PCC. Se trata de términos para los que un diccionario terminológico podría ser de ayuda, si previamente se conoce el dominio concreto de las siglas. Como ya se ha indicado, se podría profundizar en este aspecto en un trabajo futuro.

- En estos experimentos no se busca que los procesos de enriquecimiento sean perfectos, sino estudiar el comportamiento de dichos procesos y aprender de los éxitos y errores que se producen al explorar fuentes Linked Data. En una aplicación comercial probablemente fuera deseable una precisión máxima para no recuperar datos erróneos. Para aumentar la precisión bastaría con descartar las entidades de tipo DESCONOCIDO o todas aquellas sobre las cuales Stilus no proporciona una URI, pero en este trabajo no se ha procesado de esta forma, dando prioridad al enriquecimiento.

- Toda la información obtenida a partir del enriquecimiento realizado se ha compilado en un conjunto de ficheros que quedan a disposición de nuevas investigaciones, con licencia de la UNED (AVIP).

# Capítulo 6. Comentarios finales y trabajos futuros

## 6.1 Comentarios finales

- La posibilidad encontrar contenido enriquecido relevante y de calidad depende en gran medida de la naturaleza de la aplicación que consumirá los datos. Por ejemplo un portal de noticias como la BBC una vez detectada una entidad en una noticia (Ej: *Barack Obama*) puede decidir mostrar otras noticias referentes a esa misma entidad. En otro tipo de aplicaciones puede ser más difícil establecer qué se considera contenido relevante y definir qué contenido enriquecido es de interés. Las posibilidades que ofrece Linked Data son tan amplias que a menudo es difícil determinar qué datos se desea obtener.

- La calidad final de un proceso de enriquecimiento depende de varios factores pero sobre todo de la calidad del sistema NER utilizado. El reconocimiento de entidades nombradas es un problema que aún dista de ser resuelto. A medida que la asignación de las URIs mejore, los procesos de enriquecimiento de datos basados en ellos mejorarán igualmente.

- Los resultados finales de un proceso de este tipo también dependen mucho del dominio al que pertenecen los datos. En general, la calidad de los resultados de los sistemas NER son muy dependientes de los diferentes dominios, por lo que es mayor la complejidad en los procesos en una aproximación multidominio. La decisión de las fuentes de datos de las que extraer información influye directamente en los resultados en el caso de aproximaciones multidominio.

- Un problema adicional en los procesos de enriquecimiento de datos es el tratamiento de los idiomas. Las diferentes herramientas y aplicaciones de la bibliografía funcionan muy bien en inglés, pero recuperar información en un idioma distinto del inglés es más complejo: el endpoint español de la DBPedia es más reducido que el inglés, se hace necesaria una traducción adicional de las URIs, el DBPedia Lookup Service solo devuelve resultados en inglés, herramientas como Sindice no ofrecen filtros de idioma, etc.

- El enriquecimiento de datos a partir de fuentes LOD es una tarea compleja. Exige un buen conocimiento de diferentes tecnologías, estándares y fuentes de datos que intervienen en el proceso. Sin embargo es una realidad, hoy en día, y presenta un enorme potencial para ser aplicada en múltiples ámbitos, aunque todavía está de lleno en el ámbito de investigación: ¿cómo facilitar el tratamiento de la ambigüedad?, ¿cómo hacer una adecuada presentación de los resultados a los usuarios?, ¿son viables aplicaciones que continuamente descubran información en la web?

## 6.2 Trabajos futuros

Las posibilidades que ofrece el Linked Open Data son enormes y existen grandes posibilidades de ampliar la investigación en este campo. Se proponen las siguientes:

- Utilizar la información semántica que incorpora el sistema NER para filtrar significados con poca relación con el dominio del texto a enriquecer, y así mejorar la precisión del detector NER aunque se disminuyera la cobertura. Hay que decidir si en el proceso de enriquecimiento prevalece obtener datos de gran precisión aunque no exista una gran cobertura (no se identifiquen todas las entidades) o viceversa.

- Un sistema NER además de recuperar las entidades que encuentra en un texto, también es capaz de recuperar ciertos "conceptos" del mismo que aunque no llegan a tener el rango de "Entidad" sí que pueden proporcionar cierta información útil para los procesos de enriquecimiento. Quizá pudieran ayudar a identificar el contexto o el dominio. Una vez identificado el dominio, el tratamiento de siglas y acrónimos tendría posiblemente mejores resultados.

- En este trabajo se ha utilizado el NER del paquete Stilus y se ha verificado que la calidad del sistema NER determina en gran medida el trabajo final. Sería interesante analizar los resultados obtenidos al utilizar distintos sistemas NER sobre el mismo corpus de prueba para comparar los resultados. Habría que estudiar en futuros trabajos el comportamiento de Spotlight y otros.

- El proceso de desambiguación de URIs encontradas para una misma entidad también es un trabajo futuro de gran interés. Además, lo largo del desarrollo de este trabajo se ha echado en falta la existencia de un mecanismo de asignación de URIs provenientes del endpoint castellano de la DBPedia. Sería interesante disponer de algún servicio equivalente al DBPedia Lookup Service para el endpoint castellano de la DBPedia y utilizarlo en el futuro para desambiguar.

- En la recuperación de información desde fuentes LOD suele plantearse el problema de qué información concreta debemos recuperar de una cierta entidad. Es obvio que toda entidad puede presentar unas propiedades "*más importantes*" que otras. Sería interesante disponer de un mecanismo que permitiera distinguir las diferentes propiedades por orden de relevancia.

- Sería asimismo muy útil determinar un método efectivo de evaluación en procesos de enriquecimiento basados en LOD o disponer de un conjunto de pruebas con resultados esperados de calidad.

- Finalmente indicar que un proceso de enriquecimiento podría tener en cuenta las preferencias personales de los usuarios (intereses, aficiones, campos profesionales afines, localización geográfica,…) a la hora de recuperar datos enlazados relevantes para el usuario final.

# Bibliografía

[1] J. Blom, "Open Images videos enriched with Open Data http://www.openimages.eu/blog/2012/01/13/open-images-videos-enriched-with-open-data," 2012.

[2] P. Oehme, M. Krug, F. Wiedemann and M. Gaedke, "The Chrooma+ Approach to Enrich Video Content using HTML5," in *Proceedings of the 22nd international conference on World Wide Web companion (pp. 479-480). International World Wide Web Conferences Steering Committee.*, 2013.

[3] Y. Li, G. Rizzo, R. e. Troncy, M. Wald and G. Wills, "Creating Enriched YouTube Media Fragments With NERD Using Timed-Text," 2012.

[4] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak and Z. Ives, "DBpedia: A Nucleus for a Web of Open Data," in *The semantic web (pp. 722-735). Springer Berlin Heidelberg*, 2007.

[5] F. M. Suchanek, G. Kasneci and G. Weikum, "Yago: A Core of Semantic Knowledge," in *In Proceedings of the 16th international conference on World Wide Web (pp. 697-706). ACM.*, 2007.

[6] G. Kobilarov, T. Scott, Y. Raimond, S. Oliver, C. Sizemore, M. Smethurst, C. Bizer and R. Lee., "Media Meets Semantic Web – How the BBC Uses DBpedia and Linked Data to Make Connections," in *The semantic web: research and applications (pp. 723-737). Springer Berlin Heidelberg.*, 2009.

[7] D.-P. Deng, G.-S. Mai, C.-H. Hsu, C.-L. Chang, T.-R. Chuang and K.-T. Shao., "Linking Open Data Resources for Semantic Enhancement of User–Generated Content," in *Semantic Technology (pp. 362-367). Springer Berlin Heidelberg.*, 2013.

[8] M. Yoshioka and N. Kando., "Issues for Linking Geographical Open Data of GeoNames and Wikipedia," in *Semantic Technology (pp. 375-381). Springer Berlin Heidelberg.*, 2013.

[9] P. Jain, P. Hitzler, A. P. Sheth, K. Verma and P. Z. Yeh, "Ontology Alignment for Linked Open Data," in *The Semantic Web–ISWC 2010 (pp. 402-417). Springer Berlin Heidelberg*, 2010.

[10] A. Latif, M. TanvirAfzal, A. U. Saeed, P. Hoefler and K. Tochtermann, "CAF-SIAL: Concept Aggregation Framework for Structuring Informational Aspects of Linked Open Data," in *Networked Digital Technologies. NDT'09. First International Conference on (pp. 100-105). IEEE.*, 2009.

[11] G. Tummarello, R. Delbru and E. Oren, "Sindice.com: Weaving the open linked data," in *The Semantic Web (pp. 552-565). Springer Berlin Heidelberg.*, 2007.

[12] P. N. Mendes, A. Passant, P. Kapanipathi and A. P. Sheth., "Linked Open Social Signals," in *IEEE/WIC/ACM International Conference on Web Intelligence, WI 2010, page 224-231.*, 2010.

[13] Y. Li, E. Draffan, H. Glaser, I. Millard, R. Newman, M. Wald, G. Wills and M. White, "RailGB: using open accessibility data to help people with disabilities," 2012.

[14] M. Wald, "Synote: Accessible and Assistive Technology Enhancing Learning for All Students," in *Computers Helping People with Special Needs (pp. 177-184). Springer Berlin Heidelberg*, 2010.

[15] Y. Li, M. Wald, G. Wills, S. Khoja, D. Millard, J. Kajaba, P. Singh and E. Gilbert, "Synote: development of a Web-based tool for synchronized annotations," in *New Review of Hypermedia and Multimedia, 17(3), 295-312.*, 2011.

[16] Y. Li, M. Wald, S. Khoja, G. Wills, D. Millard, J. Kajaba and P. Singh, "Synote: Enhancing Multimedia E-Learning with Synchronised Annotation," in *Proceedings of the first ACM international workshop on Multimedia technologies for distance learning (pp. 9-18). ACM*, 2009.

[17] Y. Li, M. Wald, T. Omitola, N. Shadbolt and G. Wills, "Synote: weaving media fragments and linked data," in *5th International Workshop on Linked Data on the Web (LDOW'12)*, 2012.

[18] Y. Li, M. Wald and G. Wills, "Applying Linked Data to Media Fragments and Annotations," 2011.

[19] P. Mendes, M. Jakob and C. Bizer, "DBpedia for NLP: A Multilingual Cross-domain Knowledge Base.," in *Proceedings of the International Conference on Language Resources and Evaluation, LREC 2012*, Istanbul, Turkey, 2012.

[20] T. Schandl and A. Blumauer, "PoolParty: SKOS Thesaurus Management utilizing Linked Data," in *In The Semantic Web: Research and Applications (pp. 421-425). Springer Berlin Heidelberg*, 2010.

[21] F. Abel, N. Henze, R. Kawase and D. Krause, "The Impact of Multifaceted Tagging on Learning Tag Relations and Search," in *The Semantic Web: Research and Applications (pp. 90-105). Springer Berlin Heidelberg*, 2010.

[22] P. N. Mendes, M. Jakob, A. García-Silva and C. Bizer, "DBpedia Spotlight: Shedding Light on the Web of Documents," in *In Proceedings of the 7th International Conference on Semantic Systems (pp. 1-8). ACM.*, 2011.

[23] P. N. Mendes, J. Daiber, R. Rajapakse, F. Sasaki and C. Bizer, "Evaluating the Impact of Phrase Recognition on Concept Tagging," in *LREC (pp. 1277-1280).*, 2012.

[24] P. N. Mendes, J. Daiber, M. Jakob and C. Bizer, "Evaluating DBpedia Spotlight for the TAC-KBP Entity Linking Task," in *Text Analysis Conference*, 2011.

[25] C. Bizer, T. Heath and T. Berners-Lee, "Linked Data – The Story so Far," in *International Journal on Semantic Web and Information Systems (IJSWIS), 5(3), 1-22.*, 2009.

[26] P. Jain, P. Hitzler, P. Z. Yehy, K. Vermay and A. P. Sheth, "Linked Data is Merely More Data," in *AAAI Spring Symposium: Linked Data Meets Artificial Intelligence.*, 2010.

[27] T. Berners-Lee, J. Hendler and O. Lassila, "The Semantic Web," 2001. [Online]. Available: http://isel2918929391.googlecode.com/svn-history/r347/trunk/RPC/Slides/p01_theSemanticWeb.pdf.

[28] F. Peset, A. Ferrer-Sapena and I. Subirats-Coll, "Open Data Y Linked Open Data: Su impacto en el área de bibliotecas y documentación," 2011. [Online]. Available: http://riunet.upv.es/bitstream/handle/10251/29837/Peset-Ferrer-Subirats2.pdf?sequence=1.

[29] N. Shadbolt, W. Hall and T. Berners-Lee, "The Semantic Web Revisited," *Intelligent Systems, IEEE (Volume:21 , Issue: 3 ),* 2006.

[30] W3C, "Lenguaje de Ontologías Web (OWL)," 2004. [Online]. Available: http://www.w3.org/2007/09/OWL-Overview-es.html.

[31] Wikipedia, "Simple Knowledge Organization System," 2003. [Online]. Available: http://es.wikipedia.org/wiki/Simple_Knowledge_Organization_System.

[32] A. Miles, B. Matthews, M. Wilson and D. Brickley, "SKOS Core: Simple knowledge organisation for the Web," in *DCMI International Conference on Dublin Core and Metadata Applications*, 2005.

[33] H. Halpin, P. J. Hayes, J. P. McCusker, D. L. McGuinness and H. S. Thompson, "When owl:sameAs isn't the Same: An Analysis," in *The Semantic Web–ISWC 2010 (pp. 305-320). Springer Berlin Heidelberg*, 2010.

[34] A. Jaffri, H. Glaser and I. Millard, "URI Disambiguation in the Context of Linked Data," 2008.

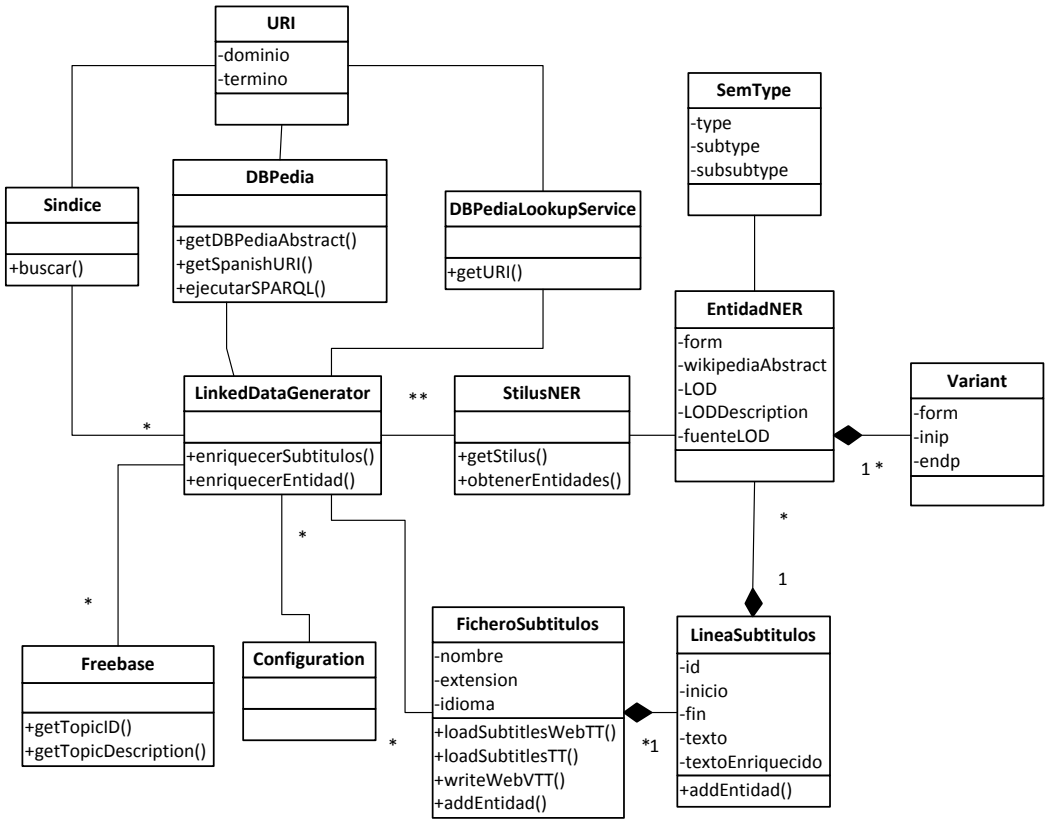
[35] L. Dodds and I. Davis, Linked data patterns. A pattern catalogue for modelling, publishing, and consuming Linked Data., 2011.

[36] T. Heath and C. Bizer, Linked data: Evolving the web into a global data space, Synthesis lectures on the semantic web: theory and technology, 2011.

[37] G. Rizzo and R. Troncy, "NERD: A Framework for Unifying Named Entity Recognition and Disambiguation Extraction Tools," in *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics (pp. 73-76). Association for Computational Linguistics*, 2012.

# PARTE III: ANEXOS

# ANEXO 1: Diagrama de clases



# ANEXO 2: Manual de Usuario: Generador de Linked Data

En la vertiente práctica de este TFM se ha desarrollado un generador de datos enlazados, que a partir de un conjunto de fichero de subtítulos que conforman el corpus y un fichero de configuración donde se encuentren codificados los procesos de enriquecimiento a utilizar, genera un fichero de subtítulos enriquecidos con información proveniente de la nube Linked Data.

**Ficheros de entrada**

- **Ficheros de subtítulos**: En la ruta donde se encuentre el fichero de la aplicación existirá una carpeta denominada "videos" que contendrá los ficheros de subtítulos a procesar. Estos ficheros se encontrarán en el formato TT (Timed Text Authoring Format) y contendrán las transcripciones de los videos. A continuación se muestra un extracto de un fichero con dicho formato:

```
<?xml version="1.0" encoding="UTF-8"?>
<tt xmlns="http://www.w3.org/2006/04/ttaf1"
      xmlns:tts="http://www.w3.org/2006/04/ttaf1#styling" xml:lang="en">
  <head>
    <styling>
      <style id="txtRight" tts:textAlign="right" tts:color="cyan"/>
      <style id="txtLeft" tts:textAlign="left" tts:color="#FCCA03"/>
      <style id="defaultSpeaker" tts:fontSize="12px" tts:fontFamily="Arial"
tts:fontWeight="normal" tts:fontStyle="normal" tts:textDecoration="none"
tts:color="white" tts:backgroundColor="black" tts:textAlign="center" />
      <style id="defaultCaption" tts:fontSize="14px" tts:fontFamily="SansSerif"
tts:fontWeight="normal" tts:fontStyle="normal" tts:textDecoration="none"
tts:color="white" tts:backgroundColor="black" tts:textAlign="left" />
    </styling>
  </head>
  <body id="ccbody" style="defaultSpeaker">
    <div xml:lang="en">
        <p begin="00:00:02.43" end="00:00:08.45">Buenas tardes. <br />Mi nombre es
Pepe Riopedre...</p>
        <p begin="00:00:09.99" end="00:00:16.09">...y soy profesor-tutor en el Centro
Asociado de la UNED en Lugo.</p>
        <p begin="00:00:16.97" end="00:00:25.27">...y profesor-tutor de la asignatura
Antropología de la Sexualidad, de la licenciatura,</p>
        <p begin="00:00:25.84" end="00:00:33.40">...asignatura que cuenta con una de
las mayores matrículas, </p>
        <p begin="00:00:33.75" end="00:00:37.75">...ha sido de las optativas más
exitosas durante la licenciatura...</p>
        <p begin="00:00:37.96" end="00:00:44.83">...aunque ahora, en cierta forma, ha
desaparecido en la nueva titulación de grado.</p>
        <p begin="00:00:45.58" end="00:00:52.81">Esperemos que en el futuro se vuelva
a tener en cuenta esta materia... </p>
        <p begin="00:23:24.02" end="00:23:29.84">Gracias por vuestra atención y un
saludo desde el centro de Lugo. </p>
        <p begin="00:23:30.54" end="00:23:32.43">Adiós, buenas tardes.</p>
    </div>
  </body>
</tt>
```

- **Fichero de configuración**: El proceso de enriquecimiento de datos será guiado por un fichero de configuración (configuración.xml) donde se definen los procesos de enriquecimiento a realizar en forma de consultas SPARQL. Dicho fichero puede ser modificado por el usuario para personalizar o afinar los procesos de enriquecimiento. En esta configuración es posible definir para cada posible tipo de entidad (PERSON, LOCATION, ORGANIZATION) las propiedades a utilizar en el endpoint SPARQL seleccionado (DBPEDIA), y la consulta SPARQL a realizar dependiendo de cada valor encontrado. Se puede consultar un ejemplo de fichero de configuración en el Anexo 3.

**Ejecución**

El generador de Linked Data ha sido desarrollado en tecnología *JAVA* por lo que para su ejecución debe tener instalado en su equipo la *Máquina Virtual de Java* [1]. Para ejecutar la aplicación debe ejecutarse en línea de comandos el comando:

```
java –jar LinkedDataGenerator.jar
```

**Ficheros de salida**

- **Ficheros NER Stilus**: Por cada fichero de entrada se generará un fichero con la salida del sistema NER. Se trata de un fichero XML que sigue el formato definido por STILUS Daedalus[2] y contiene la respuesta de STILUS al texto enviado como entrada. En el fichero se describen las entidades identificadas en el texto, su posición en el texto y sus URIs.

```
<?xml version="1.0" encoding="utf-8"?>
<results>
  <status code="0">OK</status>
  <result>
    <entities>
      <entity>
        <form><![CDATA[Suecia]]></form>
        <normalizedForm>
          <![CDATA[Suecia#ISO3166-1-a2:SE#ISO3166-1-a3:SWE]]>
        </normalizedForm>
        <semType><type>LOCATION</type>
          <subtype>GEO_POLITICAL_ENTITY</subtype>
          <subsubtype>COUNTRY</subsubtype>
        </semType>
        <semGeo><continent>Europa</continent></semGeo>
        <semLinkedData>
          <![CDATA[C_SUECIA#1@en.wiki:Sweden]]>
        </semLinkedData>
        <variants>
          <variant>
           <form><![CDATA[Suecia]]></form>
           <inip>10734</inip><endp>10739</endp>
        </variant>
        </variants>
      ...
```

[1] http://www.java.com/es/download
[2] http://api.daedalus.es

```
  </entity>
    </entities>
  </result>
</results>
```

- **Fichero enriquecido de subtítulos**: Como resultado de la ejecución, se generará un fichero de subtítulos enriquecido con información LOD por cada fichero de entrada. Dichos ficheros resultantes se generarán en formato *WebVTT*[1] con la extensión ".VTT". *WebTT* es un formato estándar de *HTML5* que puede ser reproducido de forma nativa por un navegador con soporte de *HTML5* sin necesidad de plugins adicionales. A continuación se muestra un extracto de un fichero VTT:

```
WEBVTT FILE

1
00:00:02.43 --> 00:00:08.45
{"texto":"Buenas tardes. Mi nombre es <b>Pepe Riopedre</b> ",
"accion":"DBPedia",
"entidades":[
{"nombre":"Pepe Riopedre", "semType":"PERSON",
"source":"",
"abstract": "",
"lod": "",
"lodDescription": "",
"sparql": ""}]}

. . .

105
00:10:32.03 --> 00:10:35.88
{"texto":"el caso famoso de Wikileaks, de <b>Julian Assange</b>,  ",
"accion":"DBPedia",
"entidades":[
{"nombre":"Julian Assange", "semType":"PERSON",
"source":"Julian_Assange",
"abstract": "Julian Paul Assange, conocido como Julian Assange, es un programador,
periodista y activista de Internet australiano, conocido por ser el fundador, editor
y portavoz del sitio web WikiLeaks.@es",
"lod": "http://es.dbpedia.org/resource/Australia
http://es.dbpedia.org/resource/Periodista ",
"lodDescription": "Lugar de nacimiento y profesión",
"sparql": "PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#> PREFIX dbpedia-
owl:<http://dbpedia.org/ontology/> SELECT ?abstract WHERE {
<http://dbpedia.org/resource/Julian_Assange> dbpedia-owl:abstract ?abstract . FILTER
langMatches( lang(?abstract), 'es') }"},
{"nombre":"WikiLeaks", "semType":"PRODUCT",
"source":"WikiLeaks",
"abstract": "WikiLeaks (del inglés leak, «fuga», «goteo», «filtración ») es una
organización mediática internacional sin ánimo de lucro que publica a través de su
sitio web informes anónimos y documentos filtrados con contenido sensible en materia
de interés público, preservando el anonimato de sus fuentes ",
"lod": "http://es.dbpedia.org/resource/Julian_Assange Abrimos a los gobiernos@es ",
"lodDescription": "Autor y eslogan",
"sparql": "PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#> PREFIX dbpedia-
owl:<http://dbpedia.org/ontology/> SELECT ?abstract WHERE {
<http://dbpedia.org/resource/WikiLeaks> dbpedia-owl:abstract ?abstract . FILTER
langMatches( lang(?abstract), 'es') }"}]}
```

[1] http://dev.w3.org/html5/webvtt

El texto del subtítulo incluye la siguiente información:

- **texto**: texto del subtítulo a mostrar. Puede contener marcado HTML para destacar ciertos aspectos del texto, como las entidades identificadas.

- **acción**: posible acción que el reproductor del video puede utilizar para enriquecer el texto mostrado. Las acciones posibles son: DBPedia o Google Maps. En el primer caso el reproductor recuperaría información de la DBPedia. En el segundo caso el reproductor identifica en un mapa de Google Maps una localización (país, ciudad, etc). Sólo se exponen estas dos opciones a modo de ejemplo pero se podría desarrollar el trabajo contemplando información proveniente de Facebook, Twitter, Flickr, …

- **entidades**: Datos de las entidades identificadas en el texto. Las entidades se codifican en formato JSON para facilitar su reutilización por un reproductor de videos basado en JavaScript. Cada entidad contiene el conjunto de datos:

    - **nombre**: nombre de la entidad identificada
    - **semType**: etiqueta semántica de la entidad, junto con sus subetiquetas tal como se codifican en Stilus. Algunos de sus posibles valores son:
        - PERSON, FULL_NAME
        - LOCATION, CONTINENT
        - LOCATION,GEO_POLITICAL_ENTITY,CITY
        - …
    - **source**: URI a la DBPedia donde se describe la entidad.
    - **abstract:** Descripción de la entidad obtenida de la DBPedia mediante una consulta SPARQL
    - **lod:** Información de datos enlazados recuperada.
    - **lodDescription**: Descripción de la información LOD recuperada.
    - **sparql** : Consulta SPARQL que permitió obtener los datos LOD.

Por cada fichero de entrada se genera un fichero de entidades donde se listan las entidades detectadas en cada fichero. Las entidades han sido procesadas por Stilus. Por cada entidad se lista el nombre de la entidad, se indica si se ha obtenido información LOD acerca de la misma, el tipo de entidad (Localización, Persona, Producto,…), y la URI asociada a cada entidad. A continuación se muestra, como ejemplo, un fichero de entidades:

| **Entidad** | **LOD** | **Tipo** | **URI** | **Fuente URI** |
|---|---|---|---|---|
| Lugo | | LOCATION | Lugo:es | DBPedia |
| Don Kulick | LOD | UNKNOWN | Tercer_sexo:es | Wikipedia |
| Antropología de la Sexualidad | | UNKNOWN | Roberto_Novoa_Santos:es | Wikipedia |
| Países del Este | LOD | UNKNOWN | Bloque_del_Este:es | Wikipedia |
| Suecia | LOD | LOCATION | Suecia:es | DBPedia |
| Finlandia | LOD | LOCATION | Finlandia:es | DBPedia |
| Ministerio de Igualdad | LOD | UNKNOWN | Ministerio_de_Igualdad:es | Wikipedia |

| | | | | |
|---|---|---|---|---|
| Asuntos Sociales de la Comunidad de Madrid | LOD | UNKNOWN | Comunidad_de_Madrid:es | Wikipedia |
| Madrid | LOD | LOCATION | Comunidad_de_Madrid:es | DBPedia |
| UE | | LOCATION | CEN_1789:en | Stilus |
| Europa | | LOCATION | Europa:es | DBPedia |
| Julian Assange | LOD | PERSON | Julian_Assange:es | DBPedia |
| WikiLeaks | LOD | PRODUCT | WikiLeaks:es | DBPedia |
| España | LOD | LOCATION | España:es | DBPedia |
| Parlamento | | ORGANIZATION | Parlamento:es | Wikipedia |
| Anthropological Theory | LOD | UNKNOWN | Evolución_humana:es | DBPedia |
| Michel Foucault | LOD | PERSON | Michel_Foucault:es | DBPedia |
| José Antonio Nieto | | PERSON | José_Antonio_Nieto:es | Wikipedia |

**Fichero de estadísticas**: Como resultado final se genera un fichero de estadísticas donde se hace un resumen estadístico de la ejecución contabilizando los siguientes factores:

| Fich | Ent | Con URI | Con LOD | Stilus | DBPedia | Sindice | Loc | Org | Fac | Per | Pro | Unk | Abs DBP | Abs Free |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 4 | 4 | 2 | 1 | 1 | 2 | 1 | 0 | 1 | 0 | 0 | 2 | 1 | 0 |
| 2 | 24 | 23 | 13 | 3 | 5 | 15 | 7 | 3 | 1 | 3 | 0 | 10 | 2 | 3 |
| 3 | 20 | 19 | 12 | 1 | 11 | 7 | 7 | 1 | 0 | 4 | 1 | 7 | 7 | 2 |
| 4 | 25 | 25 | 9 | 2 | 11 | 12 | 10 | 3 | 1 | 6 | 0 | 5 | 5 | 2 |
| 5 | 10 | 8 | 3 | 0 | 4 | 4 | 1 | 0 | 0 | 5 | 0 | 4 | 4 | 0 |
| 6 | 23 | 20 | 5 | 1 | 9 | 10 | 5 | 3 | 0 | 5 | 1 | 9 | 9 | 6 |
| 7 | 15 | 9 | 3 | 0 | 3 | 6 | 3 | 3 | 0 | 0 | 2 | 7 | 2 | 3 |
| 8 | 6 | 4 | 0 | 0 | 1 | 3 | 1 | 0 | 0 | 1 | 2 | 2 | 1 | 2 |
| 9 | 13 | 12 | 5 | 0 | 4 | 8 | 1 | 2 | 0 | 5 | 0 | 5 | 3 | 5 |
| 10 | 17 | 16 | 11 | 0 | 13 | 3 | 2 | 5 | 0 | 4 | 1 | 5 | 12 | 1 |
| 11 | 14 | 12 | 5 | 0 | 5 | 7 | 2 | 2 | 0 | 4 | 1 | 5 | 3 | 3 |
| 12 | 7 | 4 | 1 | 0 | 1 | 3 | 1 | 4 | 0 | 0 | 1 | 1 | 0 | 2 |
| 13 | 7 | 4 | 3 | 0 | 3 | 1 | 1 | 6 | 0 | 0 | 0 | 0 | 0 | 0 |
| 14 | 27 | 23 | 11 | 0 | 13 | 10 | 5 | 13 | 0 | 2 | 3 | 4 | 12 | 7 |
| 15 | 14 | 11 | 3 | 0 | 6 | 5 | 2 | 1 | 0 | 6 | 1 | 4 | 5 | 1 |
| 16 | 14 | 13 | 4 | 0 | 6 | 7 | 2 | 2 | 0 | 3 | 2 | 5 | 5 | 4 |
| 17 | 7 | 7 | 3 | 0 | 5 | 2 | 1 | 1 | 0 | 5 | 0 | 0 | 4 | 1 |
| 18 | 24 | 16 | 9 | 0 | 9 | 7 | 3 | 8 | 0 | 1 | 7 | 5 | 7 | 3 |
| 19 | 13 | 11 | 3 | 0 | 7 | 4 | 1 | 2 | 0 | 6 | 2 | 2 | 6 | 1 |
| 20 | 20 | 20 | 12 | 0 | 14 | 6 | 3 | 5 | 0 | 7 | 0 | 5 | 10 | 1 |

**Fich**: Fichero de entrada.
**Ent**: Número de entidades detectadas en cada fichero.
**ConURI**: Número de entidades a las que ha sido posible asignarles una URI.
**ConLOD**: Número de entidades de las que ha sido posible recuperar información LOD.
**Stilus**: Número de entidades a las cuales Stilus ha asignado la URI.
**DBPedia**: Número de entidades a las cuales la DBPedia ha asignado la URI.
**Sindice**: Número de entidades a las cuales Sindice ha asignado la URI.
**Loc**: Número de entidades de tipo LOCATION.
**Org**: Número de entidades de tipo ORGANIZATION.
**Fac**: Número de entidades de tipo FACILITY.
**Per**: Número de entidades de tipo PERSON.
**Pro**: Número de entidades de tipo PRODUCT.
**Unk**: Número de entidades de tipo UNKNOWN.
**AbsDBP**: Número de entidades de las que se ha recuperado su descripción DBPedia.
**AbsFree**: Número de entidades de las que se ha recuperado su descripción Freebase.

# ANEXO 3: Fichero de Configuración con procesos de enriquecimiento.

```
<?xml version="1.0" encoding="UTF-8"?>
<configuracion>

  <tipoEntidad tipo="PRODUCT">
    <propiedad nombre="rdf:type"
               valor="http://dbpedia.org/ontology/Website"
               descripcion="Autor y eslogan">
      <consultaSPARQL>
      <![CDATA[
PREFIX dbpediaowl: <http://dbpedia.org/ontology/>
PREFIX dbpprop: <http://es.dbpedia.org/property/>
PREFIX dbres: <http://es.dbpedia.org/resource/>
PREFIX dcterms: <http://purl.org/dc/terms/>
SELECT DISTINCT ?a ?b
WHERE {
 OPTIONAL { dbres:ENTIDAD dbpprop:creador ?a. }
 OPTIONAL { dbres:ENTIDAD dbpprop:eslogan ?b. }
} LIMIT 5
]]>
      </consultaSPARQL>
    </propiedad>
  </tipoEntidad>

  <tipoEntidad tipo="PERSON">
    <propiedad nombre="dbpediaowl:occupation"
               valor="http://es.dbpedia.org/resource/Filósofo"
               descripcion="Obras">
      <consultaSPARQL>
<![CDATA[
PREFIX dbpediaowl: <http://dbpedia.org/ontology/>
PREFIX dbpprop: <http://es.dbpedia.org/property/>
PREFIX dbres: <http://es.dbpedia.org/resource/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT DISTINCT ?a
WHERE {
 ?a  dbpprop:autor dbres:ENTIDAD
} LIMIT 5
]]>
      </consultaSPARQL>
    </propiedad>

    <propiedad nombre="*" valor="*" descripcion="Lugar de nacimiento y profesión">
      <consultaSPARQL>
        <![CDATA[
PREFIX dbpediaowl: <http://dbpedia.org/ontology/>
PREFIX dbpprop: <http://es.dbpedia.org/property/>
PREFIX dbres: <http://es.dbpedia.org/resource/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT DISTINCT ?a ?b
WHERE {
OPTIONAL { <http://es.dbpedia.org/resource/ENTIDAD> dbpediaowl:birthPlace ?a }
OPTIONAL { <http://es.dbpedia.org/resource/ENTIDAD> dbpediaowl:occupation ?b }
} LIMIT 1
]]>
      </consultaSPARQL>
    </propiedad>

  </tipoEntidad>
```

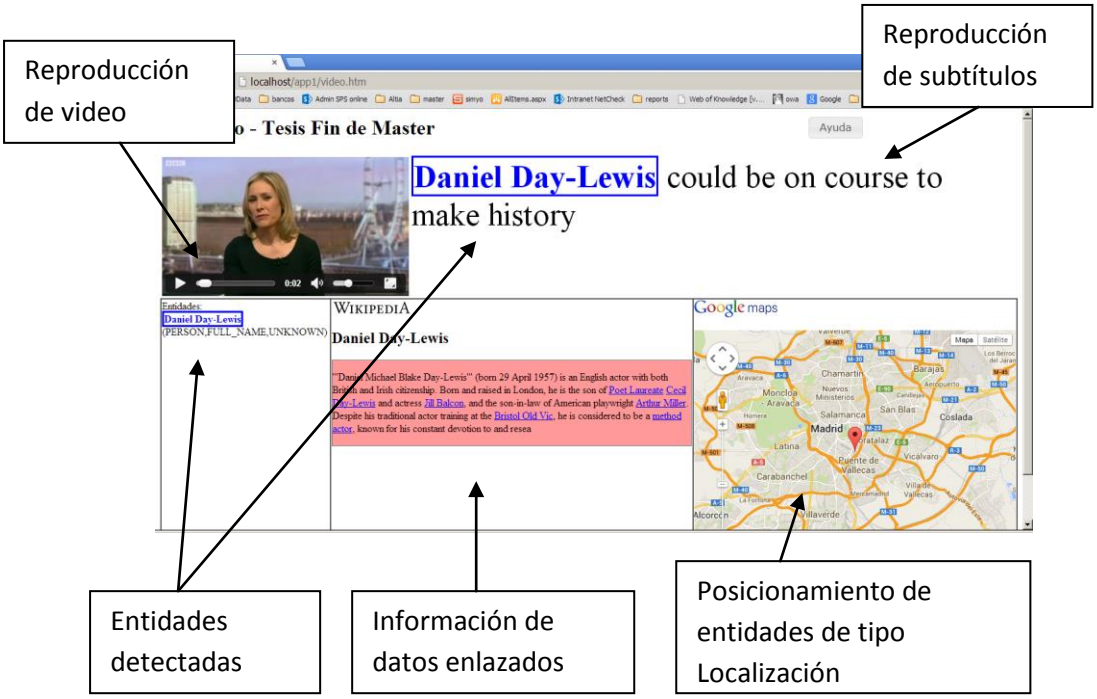
```
  <tipoEntidad tipo="FACILITY">
    <propiedad nombre="rdf:type" valor="http://dbpedia.org/ontology/University"
descripcion="Nombre del rector y pagina web">
      <consultaSPARQL>
        <![CDATA[
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dbpediaowl: <http://dbpedia.org/ontology/>
PREFIX dbpprop: <http://es.dbpedia.org/property/>
PREFIX dbres: <http://es.dbpedia.org/resource/>
SELECT ?a ?b
WHERE {
OPTIONAL { dbres:ENTIDAD dbpprop:rector ?a . }
OPTIONAL { dbres:ENTIDAD dbpprop:sitioWeb ?b . }
}      LIMIT 1
]]>
      </consultaSPARQL>
    </propiedad>
  </tipoEntidad>

  <tipoEntidad tipo="ORGANIZATION">

    <propiedad nombre="*" valor="*" descripcion="nombre, sede y pagina web">
      <consultaSPARQL>
        <![CDATA[
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dbpediaowl: <http://dbpedia.org/ontology/>
PREFIX dbpprop: <http://es.dbpedia.org/property/>
PREFIX dbres: <http://es.dbpedia.org/resource/>
SELECT ?a ?b ?c
WHERE {
OPTIONAL {<http://es.dbpedia.org/resource/ENTIDAD> dbpprop:nombre ?a . }
OPTIONAL {<http://es.dbpedia.org/resource/ENTIDAD> dbpprop:sede ?b . }
OPTIONAL {<http://es.dbpedia.org/resource/ENTIDAD> dbpediaowl:wikiPageExternalLink ?c
.}
}      LIMIT 1
]]>
      </consultaSPARQL>
    </propiedad>


  </tipoEntidad>

  <tipoEntidad tipo="LOCATION">

    <propiedad nombre="dbpprop:país" valor="*"
               descripcion="Científicos del país">
      <consultaSPARQL>
        <![CDATA[
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dbpediaowl: <http://dbpedia.org/ontology/>
PREFIX dbpprop: <http://es.dbpedia.org/property/>
PREFIX dbres: <http://es.dbpedia.org/resource/>
PREFIX dcterms:       <http://purl.org/dc/terms/>
select ?a where {
 ?a dcterms:subject <http://es.dbpedia.org/resource/Categoría:Científicos_de_ENTIDAD>
.
} LIMIT 5
]]>
      </consultaSPARQL>
    </propiedad>
```

```
    <propiedad nombre="dbpediaowl:type"
               valor="http://es.dbpedia.org/resource/Comunidad_autónoma"
               descripcion="capital">
      <consultaSPARQL>
        <![CDATA[
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dbpediaowl: <http://dbpedia.org/ontology/>
PREFIX dbpprop: <http://es.dbpedia.org/property/>
PREFIX dbres: <http://es.dbpedia.org/resource/>
SELECT ?a
WHERE {
dbres:ENTIDAD dbpediaowl:capital ?a .
}      LIMIT 1
]]>
      </consultaSPARQL>
    </propiedad>


    <propiedad nombre="*" valor="*"
               descripcion="Número de habitantes, web">
      <consultaSPARQL>
        <![CDATA[
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dbpediaowl: <http://dbpedia.org/ontology/>
PREFIX dbpprop: <http://es.dbpedia.org/property/>
PREFIX dbres: <http://es.dbpedia.org/resource/>
SELECT ?a ?b
WHERE {
OPTIONAL {<http://es.dbpedia.org/resource/ENTIDAD> dbpediaowl:populationTotal ?a .}
OPTIONAL {<http://es.dbpedia.org/resource/ENTIDAD> dbpprop:sitioWeb ?b . }
OPTIONAL {<http://es.dbpedia.org/resource/ENTIDAD> dbpprop:web ?c . }
}      LIMIT 1
]]>
      </consultaSPARQL>
    </propiedad>
  </tipoEntidad>

  <tipoEntidad tipo="UNKNOWN">
    <propiedad nombre="*" valor="*"
               descripcion="Categorias a las que pertence">
      <consultaSPARQL>
        <![CDATA[
PREFIX dcterms:      <http://purl.org/dc/terms/>
select ?a where {
  <http://es.dbpedia.org/resource/ENTIDAD> dcterms:subject ?a .
} LIMIT 5
]]>
      </consultaSPARQL>
    </propiedad>
  </tipoEntidad>

</configuracion>
```

# ANEXO 4: Manual de Usuario del Visualizador de videos con contenido enriquecido.

Para visualizar los ficheros de subtítulos enriquecidos se ha construido una aplicación web que permite visualizar los videos junto con dichos subtítulos enriquecidos con información LOD.

El reproductor muestra de forma sincronizada con la reproducción del video tanto los subtítulos como las entidades que se reconocen en los mismos, creando una experiencia de usuario novedosa donde la información enlazada aparece a medida que avanza el video en cuestión.




**Instalación**

Para la instalación del reproductor de videos en un entorno Windows se debe disponer del servidor web *IIS (Internet Information Services)* y crear en él un sitio web donde alojar las carpetas y ficheros que componen la aplicación:

- **corpusVideos**: Carpeta de videos y ficheros de subtítulos enriquecidos.
- **scripts**: Carpeta de ficheros de *JavaScript* necesarios para la ejecución.
- **css**: Ficheros de hojas de estilo *CSS*
- **images**: Ficheros de imágenes
- **Video.htm**: Página principal del reproductor de videos. Se establecerá como página principal del sitio web.

**Configuración del navegador**

Esta aplicación utiliza nuevas características de HTML5 que aún no son soportadas por todos los navegadores. Para ejecutarla correctamente, es necesario utilizar una versión reciente del navegador Google Chrome y tener activado el flag "*Habilitar elemento <track>*" en los flags del navegador. Para ello debe teclear "*about:flags*" en la barra de direcciones del navegador, activar el flag mencionado y reiniciar el navegador para que el cambio del flag tenga efecto.

Además debería tener activadas las características de Geolocalización de *HTML5*.

**Ejecución**

Una vez instalado el servidor web y configurado el navegador se debe teclear en la barra de direcciones del navegador la url del sitio web creado http://localhost/app1/video.htm y presionar el botón "play" en la parte inferior del video. El video comenzará su reproducción y los subtítulos e información enlazada se reproducirán automáticamente y de forma sincronizada con su aparición en el video.

Adicionalmente, es posible también acceder a una versión online del reproductor de videos en Internet, en la dirección: http://jesusrio2.0fees.net/video.htm