# Clasificación de las recomendaciones obtenidas del BlueFinder para la propiedad semántica *birthPlace*

Lic. Andrea Noemí Alende

Directora:
Dra. Alicia Díaz

**Trabajo Final Integrador para obtener el grado de Especialista en Ingeniería de Software**

Facultad de Informática

Universidad Nacional de La Plata

Mayo, 2015

# Resumen

Wikipedia es una gran enciclopedia editada colaborativamente por usuarios de todo el mundo. DBpedia es un proyecto desarrollado para extraer información estructurada de Wikipedia. La información semántica extraída de facilita la búsqueda de información que en algunos casos no es posible obtener navegando por Wikipedia. Para resolver este gap de información entre la web semántica y la web social, podemos aplicar el algoritmo BlueFinder que devuelve un conjunto de recomendaciones en forma de caminos navegacionales para una propiedad semántica de DBpedia en Wikipedia. En este artículo se analizará el nivel de precisión de los resultados de la aplicación de dicho algoritmo. Se propondrá entonces una taxonomía que permita clasificar un conjunto de recomendaciones para luego determinar la validez de las mismas.

# Índice

# Índice de Figuras

# 1 INTRODUCCIÓN

En el presente capítulo se introduce al lector en el contexto de los temas a desarrollar, se define el objetivo del trabajo y se detalla la estructura del documento.

## 1.1 Contexto

Wikipedia[1] se ha convertido en la enciclopedia web más ampliamente usada, siendo el sexto sitio web mas consultado (Alexa, 2015), resultando uno de los mejores ejemplos de creación de contenidos en forma colaborativa (Lehmann, Isele, Jakob, Jentzsch, & Kontokostas, 2013) .

Además de texto libre, los artículos de Wikipedia consisten en diferentes tipos de datos estructurados como infoBoxes, tablas, listas y categorías. Este último resulta un aspecto especialmente interesante de Wikipedia en cuanto a la vinculación de su contenido.

Los artículos en la Wikipedia son asignados explícitamente a una o más categorías. Estas categorías deberían representar los temas principales y de su uso depende que la búsqueda de información resulte útil (Chernov, Iofciu, Nejdl, & Zhou).

De la información disponible en Wikipedia, el proyecto comunitario DBpedia[2] extrae conocimiento estructurado, multilingüe para dejarlo disponible en la web gratuitamente, usando los estándares de la Web Semántica.

La estructura de la base de conocimiento DBpedia es mantenida por la comunidad de usuarios de DBpedia. Esta comunidad genera reglas de mapeos de información entre Wikipedia y las estructuras de representación de las ontologías de DBpedia (DBpedia, 2015).

Sin embargo, muchas de las relaciones existentes entre recursos en DBpedia, no tienen en Wikipedia una vinculación equivalente entre los correspondientes artículos, provocando un gap de información entre la web semántica y la web social. En algunos casos, el agregado de estos vínculos en Wikipedia enriquecerá su contenido, por lo que permitirá una mejor navegación (Torres, Molli, Skaf-Molli, & Diaz, 2012).

Existe un algoritmo definido para resolver el problema, BlueFinder (Torres, Skaf-Molli, Molli, & Diaz, 2013), cuyo objetivo es obtener la mejor representación para una propiedad semántica de DBpedia en Wikipedia. Como resultado de la aplicación del algoritmo, se obtiene un conjunto de caminos navegacionales recomendados para representar esa propiedad semántica. Este algoritmo fue definido siguiendo una estrategia de filtrado colaborativo.

[1] http://www.wikipedia.org/
[2] http://dbpedia.org/

Como en todo algoritmo de recomendación, se busca mejorar la adecuación de los resultados obtenidos. A pesar que las recomendaciones obtenidas por el BlueFinder son un aporte importante para colaborar en la mejora de la navegabilidad de Wikipedia, es necesario analizar los casos erróneos, como por ejemplo:

- recomendaciones semánticamente incorrectas[3]
- relaciones no recomendadas por el algoritmo que sin embargo resultan óptimas como mecanismo de categorización en Wikipedia

## 1.2 Objetivo

El presente trabajo tiene como objetivo analizar las causas de las fallas detectadas en las recomendaciones obtenidas del algoritmo BlueFinder para la propiedad semántica de DBpedia *birthPlace*, realizando una clasificación con los resultados obtenidos que pueda ser utilizada para mejorar el nivel de precisión de dichas recomendaciones.

En el análisis de los resultados se encontraron diferentes tipos de fallas, casos en donde las recomendaciones no son semánticamente correctas o sea no representan la propiedad analizada *birthPlace*, u otros tipos de fallas como relaciones no recomendadas por BlueFinder que representan la propiedad analizada mejor que la recomendada por el algoritmo.

Con el análisis realizado y la categorización propuesta se pretende mostrar cómo trabaja el algoritmo, lo que puede servir en un futuro para una modificación o mejora del mismo teniendo en cuenta los datos recabados en el presente trabajo.

## 1.3 Estructura

En el Capítulo 2 se desarrollan los conceptos principales de las tecnologías relacionadas con el tema de investigación, como son la Web Semántica, la enciclopedia Wikipedia, el proyecto DBpedia.

En el Capitulo 3 se desarrolla el Algoritmo BlueFinder. Se presenta una introducción de los algoritmos de filtrado, se describe el funcionamiento del algoritmo y se muestran ejemplos de debilidades detectadas.

En el Capitulo 4 se presenta la taxonomía propuesta para clasificar las recomendaciones y se describe la metodología utilizada para catalogar dichas recomendaciones en la clasificación definida anteriormente. Posteriormente se detalla la evaluación y análisis estadístico de los resultados obtenidos.

En el Capitulo 5 se presentan los conocimientos adquiridos y temáticas abordadas, las conclusiones obtenidas y posibles trabajos futuros.

Por último se detallan las referencias bibliográficas consultadas para la realización del presente artículo.

[3] Consideramos semánticamente incorrecta toda recomendación que no represente la propiedad semántica analizada birthplace que corresponde estrictamente al lugar de nacimiento de una persona.

# 2. TECNOLOGÍAS RELACIONADAS

En este capítulo se desarrollaran los principales aspectos de las técnicas relacionadas con el presente trabajo:

- Los principios de la Web Semántica, los lenguajes RDF, RDFS, OWL, y los lenguajes de consulta SPARQL.

- Wikipedia: recursos para estructurar información en Wikidata. Infoboxes. Categorías. Estructura de Categorías, formas de consultas.

- DBpedia: ontología de DBpedia, SPARQL endpoint, la importancia de DBpedia en enfoque global de Linked Open Data

## 2.1. Web Semántica

El termino Web Semántica fue concebido originalmente por Sir Tim Berners-Lee [4], director del World Wide Web Consortium (W3C) [5], y creador de la WWW en los años 80. Fue presentado formalmente al mundo en su artículo Scientific American "The Semantic Web" en Mayo de 2001 (Berners-Lee, Hendler, & Lassila, The Semantic Web, 2001)

Tim Berners-Lee definió a la Web Semántica como una extensión de la Web actual, en la que la información es provista de un significado bien definido, permitiendo a las computadoras y a las personas trabajar cooperativamente. También expresa en su artículo que el objetivo de la Web Semántica es desarrollar estándares y tecnologías diseñadas para ayudar a las computadoras a entender más información en la web y con esto lograr que aporten conocimiento, integración de los datos, mejor navegación y automatización de tareas.

En el W3C Consortium existe un grupo dedicado al desarrollo de la Web Semántica, denominado hasta diciembre de 2013 W3C Semantic Web Activity (W3C, 2013), y reemplazado en la actualidad por W3C Data Activity [6]. Según su definición la Web Semántica provee un framework que permite que la información sea compartida y reutilizada por las aplicaciones, las empresas y la comunidad. Este grupo es un esfuerzo colaborativo dirigido por el W3C con la participación de un gran número de investigadores y socios de la

[4] http://www.w3.org/People/Berners-Lee/
[5] http://www.w3.org/
[6] http://www.w3.org/2013/data/

industria.

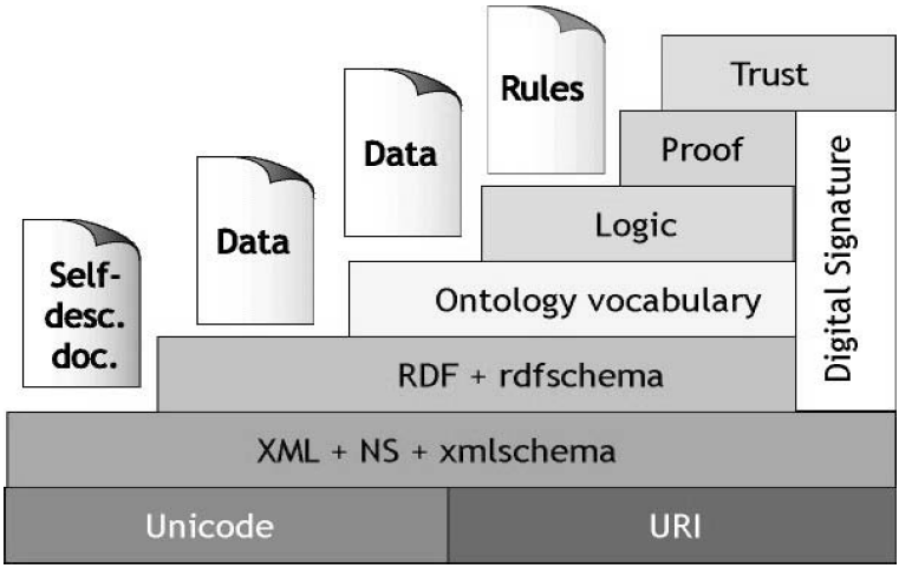
*Logo de la Web Semántica y W3C*[7]

El desarrollo de la Web Semántica se realiza en forma escalonada. Cada escalón construye una capa sobre el anterior. La justificación de este enfoque es que resulta más simple consensuar pequeñas partes entre los grupos de investigación, que esperar a lograr acuerdo sobre todos los aspectos involucrados en su evolución y desarrollo.

En la actualidad existen muchos grupos de investigación que se encuentran trabajando en diferentes aspectos de la Web Semántica. Esta competencia de ideas es un impulso importante para el progreso científico, sin embargo, es necesario estandarizarlas. Para lograrlo se buscan los puntos de acuerdo entre los diferentes grupos y se establece un estándar. Una vez establecido, otros grupos y empresas de desarrollo pueden adoptarlo y no deben esperar a ver cual línea de investigación será exitosa para elegirla.

La idea de la Web Semántica es hacer que los usuarios y las empresas construyan herramientas, agreguen contenido y luego lo utilicen. No es necesario esperar a que la visión completa de la Web Semántica se materialice. Va a llevar algunos años más para que este desarrollada en toda su extensión (Antoniou & van Harmelen, 2008).

Tim Berners-Lee, propuso el modelo original en capas, "layer cake", en 2001, el cual se ha ido modificando con el advenimiento de las nuevas tecnologías desarrolladas para la Web Semántica, llegando en la actualidad al modelo propuesto por el W3C.




*Modelo de la Web Semántica propuesto por Tim Berners-Lee (Berners-Lee, Hendler, & Lassila, The Semantic Web, 2001)*

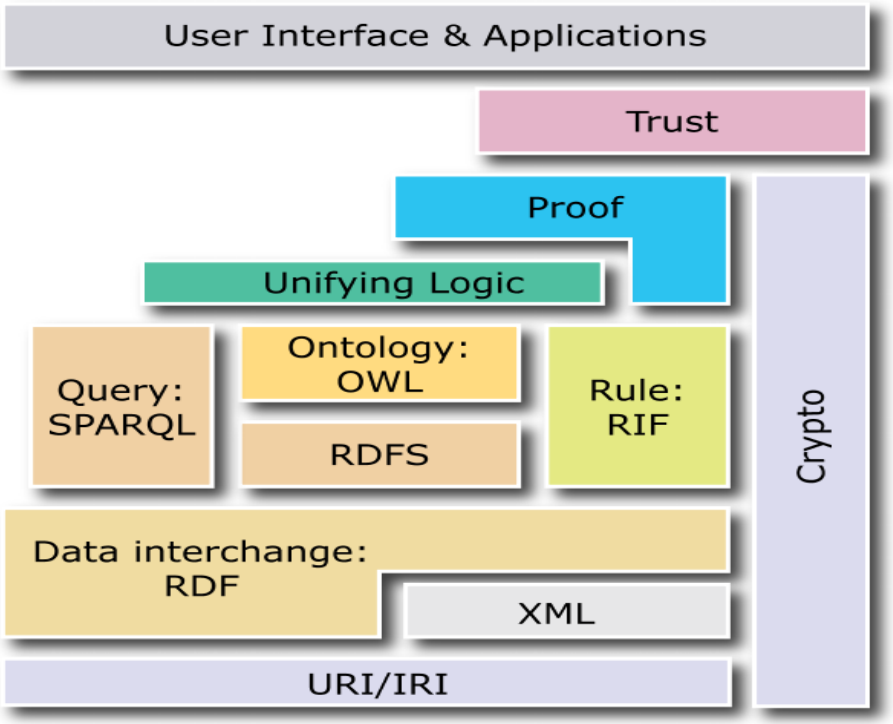
En la construcción de la Web Semántica en capas se deben seguir dos principios (Antoniou & van Harmelen, 2008):

-Compatibilidad descendente. Los agentes desarrollados para una capa, deben ser capaces de interpretar y

[7] http://www.w3.org/2007/10/sw-logos.html

utilizar la información de las capas inferiores. Por ejemplo los agentes de la capa OWL, son capaces de utilizar todas las ventajas de la información escrita en RDF y del esquema RDF.

-Entendimiento parcial ascendente: Los agentes desarrollados para una capa, deben ser capaces de interpretar, al menos parcialmente, la información de los niveles superiores. Por ejemplo un agente capaz de interpretar semánticas RDF y el esquema RDF, pude interpretar parcialmente información escrita en OWL, no teniendo en cuenta los elementos que van más allá del RDF. No existe ningún requerimiento para que todas las herramientas posean esta funcionalidad, sino que sería interesante que fueran capaces de hacerlo.




*Modelo de la Web Semántica propuesto por el W3C* [8]

La arquitectura de la Web Semántica es un tema constantemente debatido y aún no ha finalizado su desarrollo, por lo tanto es muy probable que continúe refinándose en el futuro.

### 2.1.1. Lenguajes de la Web Semántica

#### 2.1.1.1. RDF - Resource Descripcion Framework

RDF ha sido descripto como la piedra fundamental de la Web Semántica. Fue creado por el W3C y formalmente presentado en 1999 en el documento que especifica el estándar original (W3C, 1999). En 2004 el RDF Core Working Group como parte de la actividad del W3C presentó una actualización que consta de seis documentos (Primer, Concepts, Syntax, Semantics, Vocabulary, and Test Cases) los que en conjunto, reemplazan al original.

RDF fue propuesto como un estándar para representar metadatos en forma estructurada. Como venimos diciendo desde un principio, la Web fue concebida para consumo humano, y no es comprensible para las

[8] http://www.w3.org/2007/03/layerCake.png

máquinas, lo cual hace muy dificultoso poder automatizar algo en la Web, por lo menos a gran escala. La solución propuesta por el W3C es utilizar metadatos para describir los datos que se encuentran en la Web, y el hecho de que esta información pueda ser procesada por las computadoras permite el procesamiento automatizado de la misma.

El objetivo de RDF es definir un mecanismo para describir recursos que no hagan referencia a ningún dominio en particular (o sea independientes del dominio), y que puedan luego ser utilizados para describir información de cualquier dominio.

El resultado final del concepto y el modelo RDF es que diferentes aplicaciones intercambien información en la web de forma automatizada, el basamento fundamental de la Web Semántica.

RDF tiene numerosas aéreas de aplicación, por ejemplo, en descubrimiento de recursos, para proporcionar mejores capacidades a los motores de búsqueda, en catalogación, para describir el contenido y las relaciones entre los contenidos de un sitio Web, una página o una biblioteca digital, en los agentes de software inteligentes, para facilitar el intercambio de información, en clasificación de contenidos, en la descripción de colecciones de páginas que representan un único documento lógico, para describir los derechos de propiedad intelectual de las páginas Web, y para expresar las preferencias de privacidad de un usuario como así también las políticas de privacidad de un sitio Web. RDF junto con la firma digital, será la clave para construir una "Web de Confianza" para el comercio electrónico y aplicaciones de uso colaborativo. (W3C, 1999)

El concepto principal de RDF es descomponer la información/conocimiento en pequeñas partes, con reglas simples sobre la semántica de cada una de esas partes. El objetivo es proveer un método general que sea simple y flexible y que permita expresar cualquier hecho, pero suficientemente estructurado para que una aplicación de software pueda operar con esta información.

Los componentes principales de este modelo son:

**-sentencias o declaraciones**

**-recursos (sujeto y objeto)**

**-predicado o propiedades**

**Sentencias**: Cada sentencia representa una de las pequeñas partes de conocimiento en que se descompone la información, como mencionamos antes. Cada sentencia toma la forma de sujeto – predicado – objeto y este orden debe respetarse siempre.

El sujeto y el objeto representan dos "cosas" que existen en el mundo, reales o abstractas, denominadas recursos, y el predicado es la relación que los une, también denominada propiedad.

Las sentencias pueden ser representadas por un gráfico RDF, donde el sujeto y el objeto ser representan en óvalos y el predicado son las flechas que los unen. La dirección de la flecha es importante. El arco siempre empieza en el sujeto y apunta hacia el objeto de la sentencia.

Este modelo es suficientemente flexible para representar cualquier tipo de información, un conjunto de sentencias puede representar conocimiento sobre un tema en particular.

**Recursos:** El objeto y el sujeto representan algo del mundo real, que puede ser visible o abstracto. Estos recursos son identificados por URIs, definidas anteriormente, lo que permite su identificación inequívoca. Cualquier cosa puede tener un URI; la extensibilidad de URIs permite la introducción de identificadores para cualquier entidad imaginable. El objeto también puede ser considerado como el valor de la propiedad.

**Predicado:** El predicado o propiedad es un aspecto específico, característica, atributo, o relación utilizada para describir un recurso. Cada propiedad tiene un significado específico, define sus valores permitidos, los tipos de recursos que puede describir, y sus relaciones con otras propiedades. Al igual que los recursos, también se identifican con URIs. La forma de expresar estas propiedades esta descripta en el RDFS (Esquema RDF).

Esta tripleta que forma una sentencia RDF es muy potente y capaz de describir cualquier situación del mundo real. Además al ser entendible por un procesador, permite desarrollar aplicaciones capaces de ejecutar consultas y obtener resultados sobre la información representada.

Como hemos visto hasta ahora RDF proporciona un modelo conceptual y abstracto para describir recursos de forma que puedan ser procesados por una computadora. El próximo paso era definir una sintaxis para crear y leer modelos RDF concretos, de forma tal que las aplicaciones puedan escribir y compartir sus documentos RDF. Las especificaciones del W3C definen una sintaxis basada en XML, denominada RDF/XML que se utiliza para representar un grafo RDF en un documento XML. No es la única serializarían que se utiliza, también existen otras como por ejemplo Notation 3 [9].

Las tres reglas fundamentales de RDF están fuertemente relacionadas a los aspectos fundacionales de la Web Semántica. (Yu, 2011)

**Regla 1:** El conocimiento o información es expresado como una lista de declaraciones, y cada declaración tiene la forma Sujeto-Predicado-Objeto, y este orden es fijo y no debe ser modificado.

**Regla 2**: El nombre de un recurso debe ser global y podría ser identificado por un URI. El nombre del predicado también debe ser global y también puede ser identificado por un URI.

**Regla 3**: Puedo hablar de cualquier recurso como quiera, y si decido usar un URI ya existente para identificar el recurso del que estoy hablando, luego lo siguiente debe cumplirse:

El recurso que estoy haciendo referencia y el recurso identificado anteriormente con el URI ya existente son exactamente lo mismo y representan el mismo concepto.

Todo lo dicho sobre el recurso es considerado una ampliación adicional del conocimiento del recurso original.

La primera regla está relacionada con el aspecto de la Web Semántica que establece que la información y el conocimiento debe ser expresado de forma tal que sea entendible por una computadora.

La segunda y tercera regla son importantes para el concepto de ampliación de la información distribuida. Estas dos reglas son el principio fundamental del proyecto Open Linked Data[10] y son el punto de partida para el descubrimiento de nuevo conocimiento.

#### 2.1.1.2. RDFS – RDFS Schema.

RDF por sí solo no proporciona un vocabulario tal capaz de definir tipos o clases específicas de recursos, por lo cual estas propiedades y clases se definen en el esquema RDF. El esquema RDF también fue definido por el W3C (W3C, 2004)

RDFS es un lenguaje de representación que extiende el conocimiento de RDF, es una extensión semántica de RDFS. Puede ser usado para crear un vocabulario para la descripción de clases, subclases y propiedades de recursos RDF. Por ser un estándar provee construcciones que permiten definir clases y propiedades de un dominio específico. Las construcciones del lenguaje RDFS son también clases y propiedades que son utilizadas para describir las clases y propiedades de un dominio en particular. (Yu, 2011)

RDFS no proporciona un vocabulario sobre aplicaciones orientadas a clases, sino que provee de mecanismos para especificar que tales clases y propiedades son parte de un vocabulario y de cómo se espera su relación. También permite definir a los recursos como instancias de una o más clases. Además permite que las clases puedan ser organizadas en forma jerárquica. (Garcia Garcia, 2003)

El vocabulario RDFS para la descripción de clases y propiedades es similar a los lenguajes orientados a objetos como Java. RDFS se diferencia de estos lenguajes en que en lugar de definir una clase en los términos de las propiedades que pueden tener sus instancias, describe las propiedades en función de las clases de los recursos a los que se aplican. Este es el rol de los mecanismos dominio y rango.

El dominio de una propiedad se utiliza para especificar con que clase se puede utilizar la propiedad que se está definiendo. El rango especifica los valores que puede tomar dicha propiedad. Se pueden definir varios dominios y rangos para una misma propiedad.

[9] http://www.w3.org/TeamSubmission/n3/
[10] http://linkeddata.org/home

Por ejemplo, podemos definir la propiedad `eg:autor` con un dominio `eg:Documento` y un rango `eg:Persona`. Esto se puede traducir como: el recurso autor puede ser una propiedad de un recurso de la clase Documento y los valores que puede tomar son de la clase Persona.

En una definición típica de un lenguaje orientado a objetos clásico se especifica la clase `eg:Libro` con un atributo llamado `eg:autor` del tipo `eg:Persona`. Al definir de esta forma, las propiedades encapsuladas en la clase, no las hacemos visibles, y solo pueden ser ampliadas por quien define la clase. Esto va en contra de la regla 3 de RDF que definimos anteriormente, donde cualquiera puede extender el conocimiento de un recurso existente.

Con el enfoque RDFS, es simple para otros definir subsecuentemente propiedades adicionales con un dominio `eg:Documento` y un rango `eg:Persona`. Esto es posible sin necesidad de redefinir la descripción original de estas clases. El beneficio del enfoque centrado en propiedades de RDFS es que permite que cualquiera extienda la descripción de los recursos existentes, este es uno de los principios arquitectónicos fundamentales de la Web Semántica propuesta por Berners Lee. (Berners-Lee, Hendler, & Lassila, The Semantic Web, 2001) (Yu, 2011) (W3C, 2004)

Con RDFS se puede definir una ontología sobre un dominio específico. Se definen las clases, la jerarquía a la que pertenecen y sus propiedades, pero aun así no es suficientemente expresivo, es necesario poder refinar el conocimiento embebido para lograr una mejor inferencia de conocimiento por parte de la aplicación que va a interpretar la ontología subyacente. OWL es esencialmente una versión más avanzada de RDFS que se utiliza para escribir ontologías y lo describiremos a continuación.

#### 2.1.1.3. OWL Ontology Web Language

OWL es el lenguaje más popular utilizado para escribir ontologías. Su propósito es básicamente el mismo de RDFS: definir ontologías que incluyen clases, propiedades y sus relaciones en un dominio de aplicación específico. Lo que diferencia a OWL de RDFS es su capacidad para expresar relaciones más complejas y ricas en conocimiento. Esto permite construir aplicaciones con un mayor poder de razonamiento. Como OWL se basa en RDFS todos los términos del vocabulario RDFS pueden ser utilizados cuando se crean documentos OWL.

OWL fue desarrollado por el W3C y la presentación formal de su primera versión aparece en el documento W3C Recommendation en febrero de 2004 (W3C, 2004). Desde su estandarización, OWL ha sido utilizado para escribir ontologías en diferentes áreas de conocimiento como medicina, biología, geografía, astronomía, defensa e industrias aeroespaciales. Se ha convertido en el estándar de facto para desarrollo de ontologías e intercambio de datos en la comunidad científica.

Con el uso intensivo, surgieron algunas deficiencias para expresar ciertos contextos específicos y muchos desarrolladores hicieron sus reclamos. En respuesta a ello, el grupo de trabajo del W3C en 2009 publica un nuevo estándar, una versión mejorada del original, conocido como OWL 2 (W3C, 2009) y una segunda edición en 2012 (W3C, 2012), quedando así el estándar del 2004, todavía vigente, ya que muchas aplicaciones están utilizándolo todavía como el OWL 1.

La definición formal de OWL2 según el W3C es la siguiente: es un lenguaje de la Web Semántica diseñado para representar conocimiento rico y complejo sobre cosas, grupos de cosas y relaciones entre ellas. OWL es un lenguaje computacional con una base lógica, tal que el conocimiento expresado en OWL puede ser razonado por un programa de computadora tanto para verificar la consistencia del conocimiento como para hacer explícito el conocimiento implícito.

Cuando se dice que una computadora puede entender una ontología dada, lo que se está queriendo expresar es que la aplicación puede parsear la ontología y crear una lista de axiomas basados en dicha ontología y todos los hechos expresados en las sentencias RDF. A su vez, cuando se dice que se hace explícito el conocimiento implícito significa que la aplicación puede inferir nuevas sentencias RDF a partir de las que están explícitas en el documento, utilizando la lógica del lenguaje. Estas nuevas sentencias no están mencionadas en ninguna parte en la ontología o en el documento original.

En RDFS, las propiedades se definen por su dominio y su rango. En OWL a esto se le agrega la posibilidad de

agregar restricciones de valor y cardinalidad. Las restricciones de valor acotan el rango de la propiedad, y las de cardinalidad la cantidad de valores que puede tomar.

Las restricciones de valor utilizadas son:

`owl:allValuesFrom`: El valor que puede tomar la propiedad puede ser solamente de la clase especificada o del rango especificado. También admite valor nulo.

`owl:someValuesFrom`: Algunos valores de la propiedad deben ser de la clase o valor definido, por lo menos uno. Puede también tomar otros valores

`owl:hasValue`: No importa cuantos valores tenga una clase para una propiedad en particular, pero al menos uno de ellos debe ser el especificado.

Las restricciones de cardinalidad son:

`owl:cardinality`: restringe la cantidad de valores de una clase que puede tener una propiedad.

`owl:min(max)` define un rango de valores que puede tomar una propiedad.

Además, OWL permite agregar características a la definición de las propiedades, una propiedad puede ser: simétrica, transitiva, funcional, inversamente funcional, o la inversa de otra propiedad.

A la definición de clases de RDFS, OWL le agrega la capacidad de definirlas a través de operaciones de conjuntos (`owl:intersectionOf` , `owl:unionOf` , `owl:complementOf`, por enumeración de sus instancias (`owl:oneOf`), diciendo que una clase es equivalente a una ya existente (`owl:equivalentClass`) o su inversa (`owl:disjointWith`).

Todo lo mencionado es válido para el estándar original, y por supuesto para OWL2. Las mejoras introducidas en OWL 2 se pueden resumir en:

1. "Syntactic sugar", lo podemos traducir como condimento sintáctico, que hace que la construcción de las sentencias comunes sea más sencilla. La razón por la que estas nuevas características son conocidas con este nombre, es porque no alteran el proceso de razonamiento de la ontología que usa dichas construcciones, su único propósito es hacer más fácil la utilización del lenguaje.
2. Nuevas construcciones que mejoran la expresividad. Una colección de nuevas propiedades como la reflexiva, irreflexiva, y asimétrica. También se agregaron nuevos calificadores para las restricciones de cardinalidad que mejoran ampliamente la expresividad del lenguaje.
3. Soporte extendido para tipos de datos. Incluye más tipos de datos ya definidos. También permite a los usuarios definir sus propios tipos de datos cuando crean la ontología.
4. La capacidad de meta modelaje incluye una nueva característica llamada "punning" (juego de palabras). Las anotaciones son bastante más poderosas que la versión anterior. Se pueden agregar anotaciones a los axiomas, agregar anotaciones a las propiedades en el dominio y rango, y anotaciones informativas de las mismas anotaciones.
5. Nuevos sublenguajes: los perfiles. OWL2 EL, OW2 QL y OWL2 PL. Estos sublenguajes ofrecen diferentes niveles de compromiso entre expresividad y eficiencia, por lo tanto ofrecen más posibilidades de elección a los usuarios.

En resumen, OWL es un lenguaje para escribir ontologías, basado en RDFS, pero con mucho mayor poder de expresividad. Sirve para definir la ontología subyacente, y permite a una aplicación inferir nuevo conocimiento, a través de la lógica del lenguaje, que no está explicito en el documento de la definición original.

### 2.1.2. Lenguaje de Consulta – SPARQL

SPARQL es un lenguaje de consulta RDF y un protocolo para acceso a datos en la Web Semántica. Fue estandarizado por el W3C SPARQL Working Group (antes denominado RDF Data Access Working Group) en

Enero de 2008 como SPARQL Query Language for RDF [11] y su última recomendación SPARQL 1.1 de marzo del 2013[12]

La recomendación consiste en tres especificaciones por separado: la primera *SPARQL Query Language specification* documento al que hacemos referencia arriba, siendo el documento principal. Junto con estas especificaciones del lenguaje esta el documento *SPARQL Query Results XML Format* [13] que define un formato de documento XML para la representación de los resultados de las consultas SELECT Y ASK del lenguaje SPARQL. La tercera especificación *SPARQL Protocol for RDF*[14] define el protocolo remoto para ejecutar consultas a un base remota RDF e interpretar los resultados.

En resumen la recomendación del W3C consiste en un lenguaje de consultas, un formato XML en el que deben ser respondidas las consultas, y un protocolo que permite realizar consultas a un servidor remoto.

Los beneficios de tener un lenguaje de consultas como SPARQL son principalmente:

- Consultar grafos RDF para obtener información específica;
- Realizar consultas a un servidor RDF remoto y obtener resultados en línea;
- Automatizar consultas sobre una base de datos RDF para generar reportes periódicos.
- Permitir el desarrollo de aplicaciones de mayor nivel que trabajen sobre los resultados de las consultas SPARQL y no directamente con sentencias RDF. (Yu, 2011)

El lenguaje de consulta SPARQL está basado en comparación de patrones gráficos. Los patrones gráficos contienen patrones triples. Los patrones triples son como las tripletas RDF, pero con la opción de una variable consulta en lugar de un término RDF en las posiciones del sujeto, predicado u objeto. Combinando los patrones triples obtenemos un patrón grafico básico, donde es necesario una comparación exacta entre gráficos.

| **Sintaxis básica de una consulta SPARQL** | |
|---|---|
| **Prologue (optional)** | BASE <iri><br>PREFIX prefix: <iri> (repeatable) |
| **Query Result forms (required, pick 1)** | SELECT (DISTINCT)sequence of ?variable<br>SELECT (DISTINCT)*<br>DESCRIBE sequence of ?variable or <iri><br>DESCRIBE *<br>CONSTRUCT { graph pattern }<br>ASK |
| **Query Dataset Sources (optional)** | Add triples to the background graph (repeatable):<br>FROM <iri><br>Add a named graph (repeatable):<br>FROM NAMED <iri> |
| **Graph Pattern (optional, required for ASK)** | WHERE { graph pattern z} |
| **Query Results Ordering (optional)** | ORDER BY ... |
| **Query Results Selection (optional)** | LIMIT n, OFFSET m |

La función de la palabra clave PREFIX es equivalente a la declaración de namespaces en XML, es decir asocia

[11] http://www.w3.org/TR/rdf-sparql-query/
[12] http://www.w3.org/TR/sparql11-overview/
[13] http://www.w3.org/TR/rdf-sparql-XMLres/
[14] http://www.w3.org/TR/rdf-sparql-protocol/

una URI a una etiqueta, que se usará más adelante para describir el namespace. Pueden incluirse varias de estas etiquetas en una misma consulta.

Todo comienzo de una consulta SPARQL queda marcada por la palabra clave SELECT, similar a su uso en SQL[15], sirve para definir los datos que deben ser devueltos en la respuesta. La palabra clave FROM identifica los datos sobre los que se ejecutará la consulta, es necesario indicar que una consulta puede incluir varios FROM. La palabra clave WHERE indica el patrón sobre el que se filtrarán los tripletes del RDF. (Valencia Castillo, 2007)

El ejemplo (W3C, 2008) mostrado a continuación ha sido tomado de W3C, y se trata de encontrar el titulo de un libro de un grafico de datos dado. La consulta consiste de dos partes:

- La cláusula SELECT que identifica las variables que aparecen en los resultados de la consulta y
- La clausura WHERE que proporciona el patrón grafico básico a comparar con el grafico de datos.

El patrón grafico básico en este ejemplo consiste de un simple patrón triple con una simple variable (¿title) en la posición del objeto.

**Datos:**

```
<http://example.org/book/book1>
<http://purl.org/dc/elements/1.1/title> "SPARQL Tutorial".
```

**Consulta:**

```
SELECT ?title
WHERE
{
  <http://example.org/book/book1>
 <http://purl.org/dc/elements/1.1/title> ?title.
}
```

**Resultado de la consulta:**

```
title
"SPARQL Tutorial"
```

SPARQL es un lenguaje muy parecido a SQL, con una serie de diferencias propias de cada modelo. Es hoy en día el principal lenguaje de consulta de la Web Semántica. Las consultas sobre DBpedia se hacen es SPARQL.

Muchas plataformas de RDF ya lo implementaron como por ejemplo **protege**[16], un editor de ontologías open-source y framework para la construcción de sistemas inteligentes, desarrollado por la Univesidad de Stanford**, Jena Semantic Web Toolkit**[17], un framework Java, open-source, para construir aplicaciones para la Web Semántica y Linked Data, de la fundación Apache, y muchos otros.

La última recomendación del W3C es SPARQL 1.1 Update[18]

## 2.2. Wikipedia

Wikipedia[19] es una enciclopedia libre, políglota y editada colaborativamente. Es administrada por la Fundación Wikimedia[20], una organización sin fines de lucro. Sus más de 24 millones de artículos en 284

[15] SQL: Structured Query Language
[16] http://protege.stanford.edu/
[17] https://jena.apache.org/
[18] http://www.w3.org/TR/sparql11-update/ http://www.w3.org/TR/sparql11-update/
[19] http://en.wikipedia.org/wiki/Main_Page

idiomas han sido redactados conjuntamente por voluntarios de todo el mundo, y prácticamente cualquier persona con acceso al proyecto puede editarlos. La versión de Wikipedia mas grande es la versión en Inglés con más cuatro millones de artículos, siendo la que se utilizó para el presente trabajo. Fue iniciada en enero de 2001 por Jimmy Wales y Larry Sanger, siendo en la actualidad la mayor y más popular obra de consulta en Internet (Foundation, 2015).


***Wikipedia logo 2.0 - Wikimedia Foundation***

Wikipedia se rige por muchas políticas, guías y convenciones, que afectan tanto a la redacción de artículos como a la convivencia con otros editores.
Estas convenciones son definidas por la comunidad siguiendo el espíritu colectivo de la obra, aunque están sujetas a debates y cambios. Cubren distintos tópicos: estructura del artículo, estilo de escritura, contexto y relación entre artículos, entre otros, aunque no siempre es evidente la aplicación de estas convenciones (Wikipedia, Category:Wikipedia guidelines, 2015).

### 2.2.1. Recursos para estructurar información en Wikidata

Wikidata[21] es un proyecto de la Fundación Wikimedia: se trata de una base de datos secundaria libre, colaborativa y multilingüe, que recopila datos estructurados para dar soporte a Wikipedia, Wikimedia Commons, a los otros proyectos Wikimedia y mucho más.

El contenido de Wikidata está disponible bajo una licencia libre, se exporta en formatos estándares y puede interrelacionarse a otros conjuntos de datos abiertos en la web como Open Linked Data. (Wikidata, 2015)


***"Wikidata-logo-en" by Planemad - Own work. Licensed under Public Domain via Wikimedia Commons*** [22]

Wikidata fue lanzada el 30 de octubre de 2012 y fue el primer proyecto nuevo de la Fundación Wikimedia desde 2006.

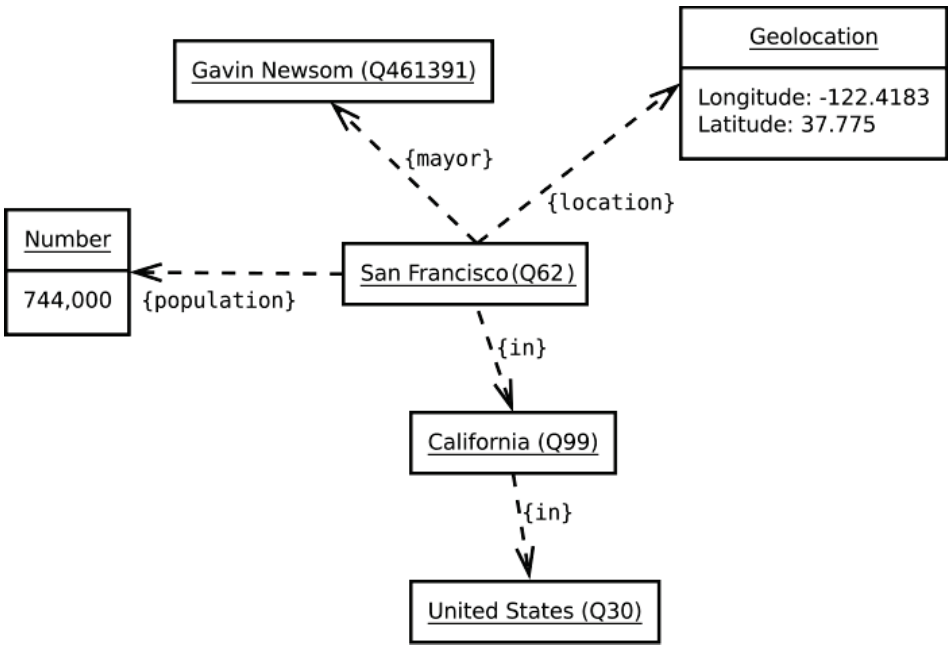
La creación del proyecto fue financiada por donaciones del Instituto Allen para la Inteligencia Artificial

[20] http://wikimediafoundation.org/wiki/Home
[21] https://www.wikidata.org/wiki/Wikidata:Main_Page
[22] http://commons.wikimedia.org/wiki/File:Wikidata-logo-en.svg#/media/File:Wikidata-logo-en.svg

[23], la Fundación Gordon y Betty Moore [24] y Google Inc.[25], sumando en total 1,3 millones de euros El desarrollo inicial del proyecto está siendo supervisado por Wikimedia Deutschland[26], y se ha dividido en tres fases (Wikipedia, Wikidata - Historia del desarrollo, 2015):

- Centralizar los enlaces interwiki
- Proporcionar un lugar central para los datos de infobox en todas las Wikipedias
- Crear y actualizar una lista de artículos basados en datos de Wikidata

Wikidata funciona como repositorio centralizado de datos estructurados, de forma que una sola lista o conjunto de datos que se repite entre artículos, se estructura y rellena una sola vez para poder ser reutilizado a posteriori y puede ser accedido por clientes Wikis. La gran ventaja es que si se debe corregir un error, se hace una sola vez, y este cambio se actualiza automáticamente en el resto de versiones idiomáticas, y no tiene que mantenerse al día en cada cliente individual. (Mastermagazine, 2014).




***Los elementos y sus datos están interconectados. (Wikidata, 2015)***

Para Wikipedia, la ventaja es simple y poderosa - si hay una fuente central, de lectura mecánica de los sucesos, como por ejemplo la población de una ciudad, cualquier actualización de esos datos se puede reflejar al instante a través de todos los artículos en los que se incluyen estos hechos.

Para plantear un ejemplo: un cantante puede tener decenas o incluso cientos de versiones lingüísticas de la entrada de la Wikipedia y, si fuera a morir, la adición de la fecha de la muerte a la base de datos de Wikidata se propagaría inmediatamente a través de todas esas versiones, sin que tenga que actualizarse manualmente cada uno. En efecto, Wikidata ahora está siendo utilizado como una fuente de datos común para todas las

[23] http://allenai.org/

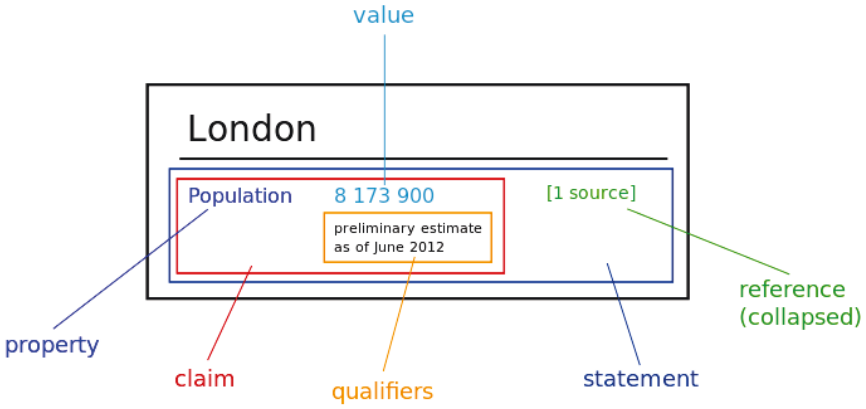
[24] http://www.moore.org/

[25] http://www.google.com/about/company/

[26] http://es.wikipedia.org/wiki/Wikipedia_en_alemán#Wikimedia_Deutschland

versiones idiomáticas de Wikipedia. (IndustriaIT, 2014)

Pero lo realmente interesante con Wikidata es que no se trata sólo de Wikipedia. Se puede acceder a los datos mediante clientes Wikis usando una interfaz Lua Scribunto[27]. Todos los datos pueden ser obtenidos mediante el uso de la API [28] . La base de datos puede ser utilizado por cualquiera, ya que los datos en Wikidata están publicados bajo Creative Commons 1.0 [29] como dominio público, lo que permite la reutilización de la información en distintos escenarios. Se puede copiar, modificar, distribuir y presentar la información, incluso para propósitos comerciales, sin necesidad de pedir permiso. (Wikidata, 2015)




***Este diagrama sugiere los términos más importantes que resultan interesantes en torno a Wikidata. (Wikidata, 2015)***

El director Wikidata, Denny Vrandecic [30] expresó en un comunicado el espíritu del proyecto: " El objetivo de Wikidata es recopilar conocimiento complejo del mundo de una manera estructurada así cualquiera puede beneficiarse de ella, ya sean lectores de Wikipedia que son capaces de estar al día acerca de ciertos hechos o ingenieros que pueden utilizar estos datos para crear nuevos productos que mejoren la forma de acceder al conocimiento" (IndustriaIT, 2014).

### 2.2.2. Infoboxes

Un infobox (Wikipedia, Help:Infobox, 2015) es una tabla, con un formato preestablecido, diseñada para ser colocada en la esquina derecha superior de los artículos de Wikipedia. Su objetivo es mostrar un resumen actualizado y conciso de algunos aspectos comunes que comparten los artículos y también para mejorar la navegación hacia otros artículos relacionados.

Muchos infoboxes contienen metadatos estructurados, que son utilizados por terceros, como por ejemplo DBpedia.

El uso de infoboxes en los artículos no es un requisito obligatorio, ni tampoco está prohibido. Como incluirlo, cual infobox y que partes del mismo utilizar esta determinado por la discusión y el consenso[31] entre los editores de cada artículo.

---

[27] http://www.mediawiki.org/wiki/Extension:Wikibase_Client/Lua

[28] http://www.mediawiki.org/wiki/Wikibase/API

[29] http://creativecommons.org/publicdomain/zero/1.0/

[30] http://simia.net/wiki/Denny

[31] http://en.wikipedia.org/wiki/Wikipedia:Consensus

La plantilla infobox contiene datos importantes y estadísticas sobre artículos que tienen un tema en común. Por ejemplo, todos los animales tienen una clasificación científica, un estado de conservación, un tipo de especie, etc. Agregar un {{taxobox}}[32] en artículos relacionados con animales hace mas fácil encontrar rápidamente dicha información y compararla con otros artículos.

Los infoboxes no son tablas estadísticas, solamente resumen material del artículo, la cual debe estar presente en el texto principal, en parte porque puede no ser posible para algunos lectores acceder al contenido del infobox. En particular, para los lectores que utilizan herramientas de asistencia, como las aplicaciones que leen el contenido de la pantalla, el infobox pasa totalmente desapercibido.


*Ejemplo de Infobox del articulo Norwegian Lundehund en la Wikipedia en Ingles* [33]

La información en el infobox debería ser:

- **Comparable**: Si muchos temas diferentes comparten un atributo en común (por ejemplo, todas las personas tienen un nombre y una fecha de nacimiento), entonces es útil poder comparar estos datos entre diferentes paginas. Esto también implica, que cuando sea posible, el material debe ser presentado en un formato estándar.
- **Concisa**: Los infoboxes tienen que ser comprendidos a simple vista, para poder chequear datos rápidamente.
- **Contener información relevante.**
- **Citadas anteriormente en alguna parte del artículo**: Los infoboxes, como la introducción a un

[32] http://en.wikipedia.org/wiki/Template:Taxobox

[33] http://en.wikipedia.org/wiki/Norwegian_Lundehund

artículo[34], deben contener principalmente datos que puedan ser ampliados con referencias de fuentes confiables dentro del mismo artículo. Sin embargo, si es necesario, porque por ejemplo el artículo no está completo, es posible incluir notas al pie[35]

La información en el infobox no debe ser:

- **Extensa**: Largos textos o estadísticas muy detalladas, corresponden al cuerpo del artículo.
- **Detalles triviales**: Un problema frecuente es la inclusión de material en el infobox que es trivial y no sería incluido de otra forma en el cuerpo del artículo. Por ejemplo, el tipo de sangre de un personaje puede ser mencionado en el trabajo, pero no es relevante para entender el artículo. Los infoboxes no deben ser utilizados para incluir detalles que son poco relevantes como para ser incluidos en el artículo. Existen algunas excepciones como las propiedades químicas.
- **Banderas**: Los iconos de banderas generalmente no deben ser utilizados en los infoboxes, incluso cuando sea un campo de país o nacionalidad o un equivalente. Provocan una distracción innecesaria y le dan una importancia indebida a ese campo entre los otros.

### 2.2.3. Categorías

En Wikipedia, una categoría es una agrupación de páginas que comparten algún tema en común. En una analogía con el sistema de archivos de una computadora, las categorías cumplen la misma función que un directorio.

Las categorías tienen a su vez subcategorías (más específicas) y supercategorías (más generales), permitiendo navegar de lo general a lo concreto y viceversa, a través de una estructura de árbol.

Una categoría es un tipo especial de página, que utiliza el espacio de nombres[36] *Categoría* y que se divide en una cabecera, subcategorías y páginas. Todo artículo de Wikipedia debe pertenecer por lo menos a una categoría.

Las categorías se encuentran normalmente al final de la página de cada artículo. Para navegar en la Wikipedia de un artículo a otro podemos entonces utilizar la estructura de categorías, que son enlaces que permiten encontrar información relacionada entre estos artículos.

El proceso de insertar los artículos en la correspondiente categoría se denomina Categorización y debe cumplir con las políticas y convenciones establecidas en Wikipedia. (Wikipedia, Wikipedia:Categorization, 2015)

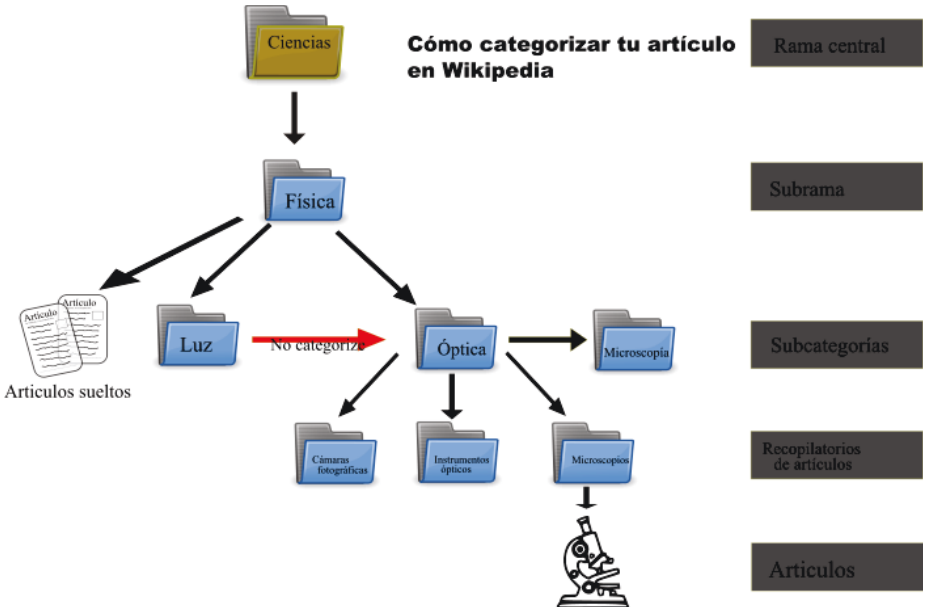
Toda página de Wikipedia puede en principio categorizarse. Una página está correctamente categorizada si en el pie de ésta aparecen las categorías de las cuales debe formar parte. Si en el pie de página no aparece ninguna categoría o alguna de las categorías que aparecen no existe —lo cual se muestra con un enlace en rojo—, es porque la página no está correctamente categorizada. (Wikipedia, Ayuda:Categoría, 2015)

En el presente trabajo se utilizo la estructura de categorías para analizar las distancias de un artículo a otro.

[34] http://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style/Lead_section

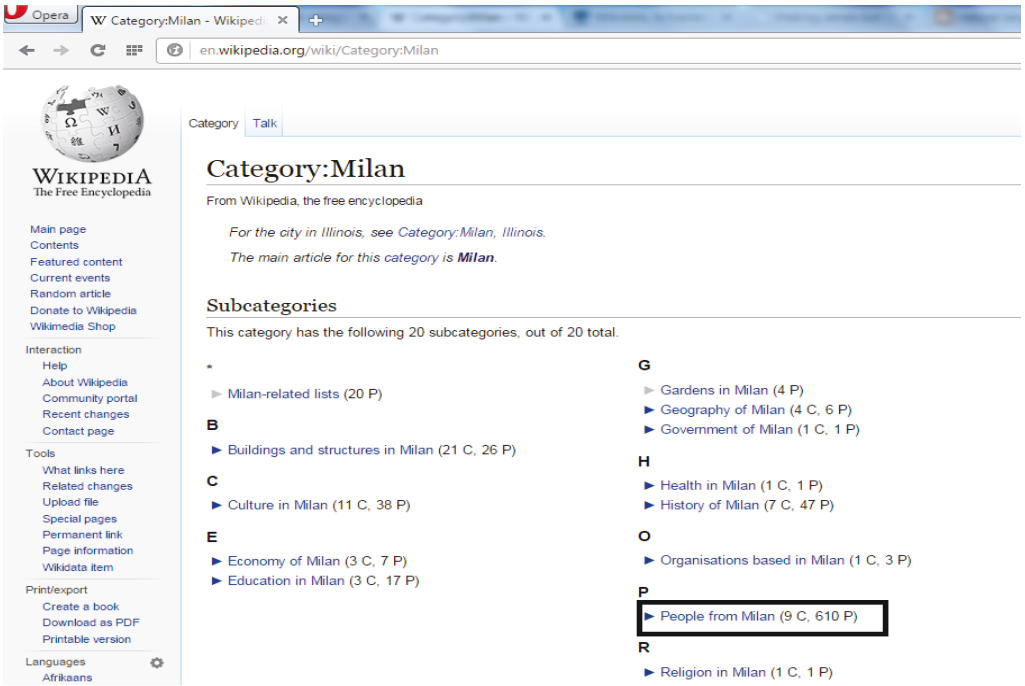
[35] http://en.wikipedia.org/wiki/Help:Footnotes

[36] http://en.wikipedia.org/wiki/Wikipedia:Namespace




*La Categorización de artículos en Wikipedia y su modo de usarlo*[37]

Por ejemplo, para navegar desde el artículo de la ciudad de Milán[38] hasta el artículo del deportista Giancarlo Brusati[39], se puede seguir el siguiente camino a través de las categorías: Milan, People from Milan, People from Milan by occupation, Sportspeople from Milan.




*Category:Milán* [40]

---

37 http://commons.wikimedia.org/wiki/File:Esquema_de_categorizaci%C3%B3n_de_art%C3%ADculos.svg#/media/File:Esquema_de_categorizaci%C3%B3n_de_art%C3%ADculos.svg

38 http://en.wikipedia.org/wiki/Milan

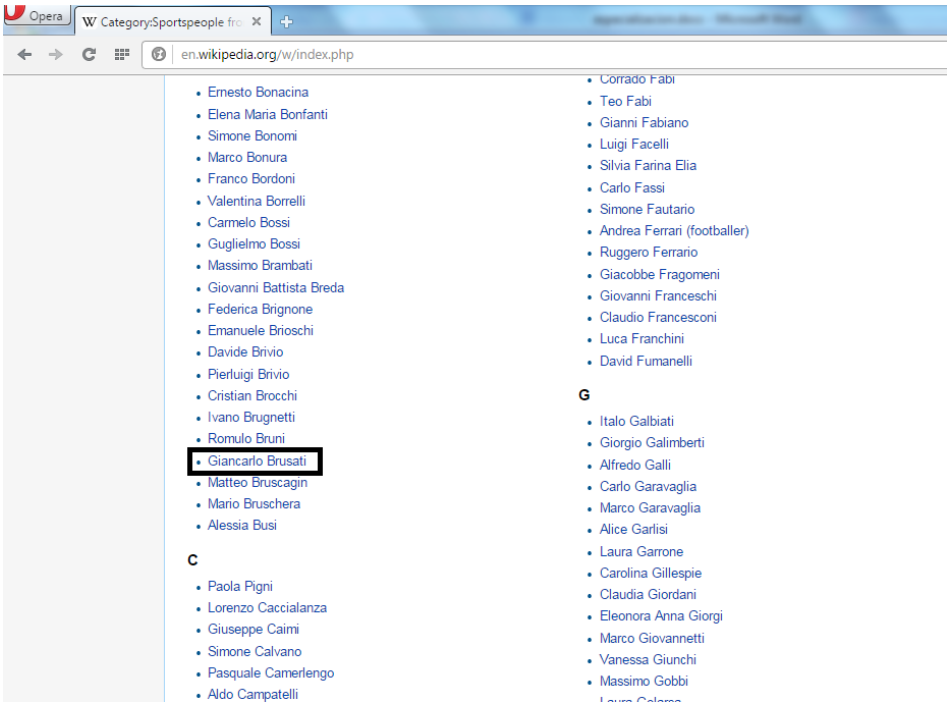
39 http://en.wikipedia.org/wiki/Giancarlo_Brusati

40 http://en.wikipedia.org/wiki/Category:Milan


*Category:People from Milan*[41]


*Category:People from Milan by occupation* [42]

[41] http://en.wikipedia.org/wiki/Category:People_from_Milan
[42] http://en.wikipedia.org/wiki/Category:People_from_Milan_by_occupation




*Category: SportPeople from Milan* [43]

## 2.3. Dbpedia

La web semántica ha ido tomando mayor importancia durante los últimos años, ya que permite mejoras en la navegación y la búsqueda de contenido en la web. La información de la Web Semántica es extraída, en gran medida, de la información producida por la Web Social. Un buen ejemplo es el caso de DBpedia (Bizer, et al., 2009), una base semántica de conocimiento que obtiene el contenido de los infoBoxes y markups disponibles en Wikipedia. La base semántica de DBpedia está conformada por recursos relacionados entre sí por propiedades semánticas. De esta forma, la cualidad semántica de DBpedia permite deducir información que no está presente en Wikipedia. (Kaminose, Blanco, Parmisano, Torres, & Díaz, 2012)

Por ejemplo, en DBpedia podemos ejecutar una consulta de este estilo: "Dame todos los deportistas nacidos en Milán en el siglo 20". Esta consulta da como resultado nombres de personas que no pueden obtenerse navegando en Wikipedia desde la página de la ciudad de Milán. Estas diferencias generan una brecha semántica entre Wikipedia y DBpedia.


*Logo del Proyecto DBpedia (DBpedia, 2015)*

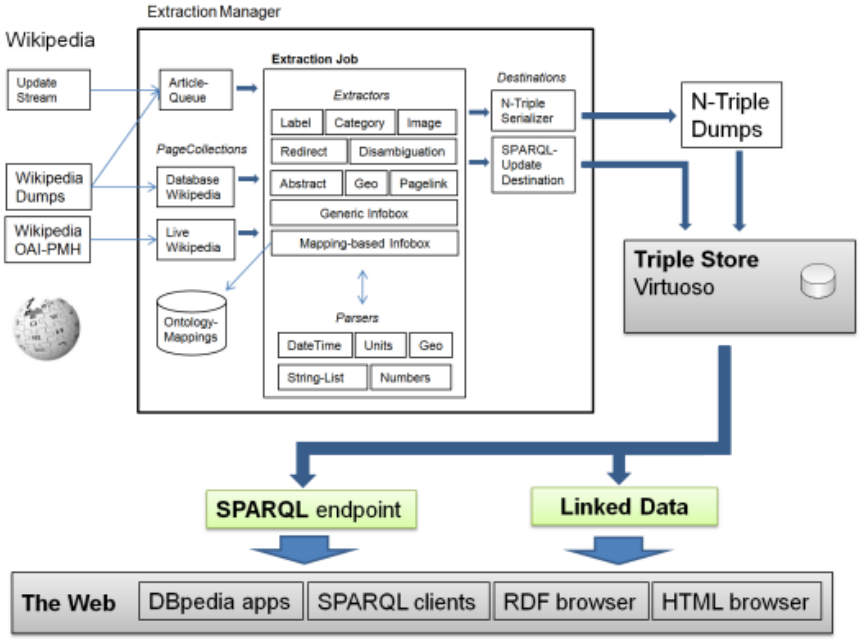
La versión en inglés de la base de conocimiento DBpedia actualmente describe 4.58 millones de conceptos, de los cuales 4.22 millones están clasificados en ontologías consistentes, incluyendo 1.445.000 personas, 735000

[43] http://en.wikipedia.org/wiki/Category:Sportspeople_from_Milan

lugares (incluyendo 478000 lugares populares), 411000 trabajos creativos (incluyendo 123000 álbumes musicales, 87000 películas y 19000 video juegos), 241000 organizaciones (incluyendo 58000 compañías y 49000 instituciones educativas), 251000 especies y 6000 enfermedades. (DBpedia, 2015)

La base de conocimiento de DBpedia tiene muchas ventajas sobre las bases existentes: abarca varios dominios; representa un acuerdo real en la comunidad; evoluciona automáticamente cuando cambia Wikipedia y es realmente multilingüe, en la actualidad está disponible en 125 idiomas.

En resumen, las aplicaciones de la base de conocimiento DBpedia son generalizadas y van desde la gestión del conocimiento empresarial, hasta la búsqueda en la web para mejorar las consultas en Wikipedia.




*Los componentes de DBpedia. (Bizer, et al., 2009)*

### 2.3.1. Ontología de DBpedia

La ontología DBpedia Ontology (DBpedia, 2014) tiene como principal característica que es superficial y extendida sobre múltiples dominios, y ha sido creada utilizando la información de los infoBoxes mas utilizados en Wikipedia. La ontología actualmente cubre sobre 685 clases que forman una jerarquía en donde conceptos se incluyen dentro de otros, y sobre ellas están descriptas más de 2795 propiedades diferentes.

Con la release 3.2, se introdujo un nuevo método de extracción de infoboxes basado en mapeos generados manualmente de los infoboxes de Wikipedia a la ontología DBpedia. Estos mapeos definen reglas refinadas sobre como analizar los valores de un infobox. Estos mapeos también abordan las debilidades del sistema de infoboxes de Wikipedia, como el hecho de tener diferentes infoboxes para una misma clase, usando diferentes nombres para una misma propiedad, y no tener claramente definidos tipos de datos para los valores de las propiedades. Por lo tanto, la instancia de un dato en la ontología del infobox, con la nueva release, es más clara y mejor estructurada que los datos de los infoboxes generados con el método de extracción anterior de infoboxes en DBpedia.

Con la release 3.5, se introduce una wiki pública, Mapping Wiki, [44] para escribir los mapeos de infoboxes, editar los existentes, como así también editar la ontología DBpedia. Esto permite a colaboradores externos definir mapeos de los infoboxes que necesiten y extender la ontología DBpedia con clases y propiedades

[44] http://mappings.dbpedia.org/index.php/Main_Page

adicionales.

Desde la release 3.7, la ontología es un grafo a cíclico dirigido, y no un árbol. Las clases pueden tener múltiples superclases, lo que es importante para los mapeos de schema.org [45]. Una taxonomía puede ser todavía construida ignorando todas las superclases, excepto aquella que esta especificada en la lista y es considerada la más importante.

La ontología contiene en la actualidad aproximadamente 4.233.000 instancias. La versión actual de la ontología se puede ver en http://mappings.dbpedia.org/server/ontology/classes/ , tal como está definida en Mapping Wiki.

La ontología DBpedia puede ser consultada a través de DBpedia SPARQL endpoint [46] y puede ser explorada vía DBpedia Linked Data interface.[47]

### 2.3.2. SPARQL endpoint

Un SPARQL endpoint es un servicio del lenguaje de consultas SPARQL descripto en la sección 2.1.3. Permite a los usuarios (humanos u otros) realizar consultas a una base de datos utilizando este lenguaje.

Los resultados son típicamente retornados en uno o más formatos. Se puede decir que un SPARQL endpoint es una interface que permite consultar una base de datos RDF.

El termino endpoint tiene un significado más amplio. En el documento que define los requerimientos de los Web Servicies [48] se lo define como una asociación entre una interface con un protocolo bien definido y una dirección de internet especificada por una URI, que puede ser utilizada para comunicarse con una instancia de un Web Service. Un endpoint tiene una localización específica para acceder a un Web Service utilizando un protocolo y formato específicos. También define a un servicio como una colección de endpoints. (SemanticWeb.org, 2011)

Resumiendo se puede decir que un SPARQL endpoint es una URI donde se puede ejecutar un requerimiento (Consulta, Update, etc.) al cual el endpoint responde apropiadamente.

La recomendación del W3C SPARQL 1.1 Protocol describe un medio para transmitir consultas (queries) y actualizaciones (updates) SPARQL a un servicio que procese con este lenguaje, retornando los resultados vía HTTP a la entidad que lo requiera. (W3C, 2013)

Para las consultas que se ejecutan sobre una base de datos, el lenguaje especifica cuatro variantes diferentes, cada una con un propósito distinto.

**SELECT query**

Se utiliza para extraer valores en bruto de un SPARQL enpoint. Los resultados son retornados en forma de tabla.

**CONSTRUCT query**

Se utiliza para extraer información de un SPARQL enpoint y transformar los resultados en un formato RDF válido.

**ASK query**

Se utiliza para proveer un resultado Verdadero/Falso a consulta realizada en un SPARQL endpoint.

**DESCRIBE query**

[45] http://schema.org/
[46] http://dbpedia.org/sparql
[47] http://wiki.dbpedia.org/OnlineAccess
[48] http://www.w3.org/TR/ws-desc-reqs/

Es utilizado para extraer un grafo RDF de un SPARQL endpoint, el contenido se deja a consideración de quien realizo la consulta para que decida qué información le es de utilidad.

Cada una de estas consultas tiene una clausula where para restringir los resultados, salvo en el caso de DESCRIBE donde el WHERE es opcional.

SPARQL 1.1 (W3C, 2013) amplía el lenguaje permitiendo actualizaciones sobre la base de datos y algunos otros formatos nuevos. (Wikipedia, 2015)

En la wiki del W3C se puede consultar el listado completo de los SPARQL enpoint actualmente en uso [49]

DBpedia provee un SPARQL endpoint para realizar consultas sobre su base de datos. Aplicaciones cliente, pueden enviar consultas utilizando el protocolo SPARQL al endpoint en http://dbpedia.org/sparql. Además del SPARQL standart el endpoint soporta varias extensiones del lenguaje que han sido de utilidad para muchos clientes, como la búsqueda de texto completa en predicados RDF seleccionados, y funciones de agregado, notablemente COUNT().

Para proteger el servicio de una sobrecarga, se han puesto limites en la complejidad y tamaño de los resultados. El endpoint esta hosteado en Virtuoso Universal Server[50] (Bizer, et al., 2009)

A continuación un ejemplo de consulta que se puede ejecutar en Virtuoso sobre la base de DBpedia.

"Dame 50 ejemplos de *concepts* in la base DBPedia"

```
SELECT DISTINCT ?concept
WHERE {
    ?s a ?concept .
} LIMIT 50
```

- `LIMIT` es un *modificador de la solución* que limita el número de filas devueltas por la consulta. SPARQL tiene otros dos modificadores de solución:
    - `ORDER BY` sirve para agrupar los resultados con los valores de una o más variables.
    - `OFFSET` se utiliza en conjunción con `LIMIT` y `ORDER BY` para tomar una parte del conjunto solución, por ejemplo paginarlo.
- La palabra reservada *a* es una abreviatura del predicado común rdf:type, definiendo cual es la clase del recurso.
- El modificador `DISTINCT` elimina las filas duplicadas del resultado de la consulta. (W3C - Cambridge Semantics, 2009)

[49] http://www.w3.org/wiki/SparqlEndpoints
[50] http://docs.openlinksw.com/virtuoso/index.html


*DBpedia SPARQL endpoint*

### 2.3.3. La importancia de DBpedia en el enfoque global de Linked Open Data

El término Linked Data (datos vinculados) hace referencia a un conjunto de las mejores prácticas para publicar datos estructurados en la Web. Estos principios han sido acuñados por Tim Berners-Lee en el documento de diseño Linked Data. [51]

Dichos principios son:

1. Utilizar URIs como nombres de las cosas.
2. Utilizar HTTP URIs de forma que las personas puedan buscar esos nombres.
3. Cuando alguien busque una URI, proveerle con información útil.
4. Incluir links a otras URIs, de esta forma pueden descubrir más cosas.

La idea detrás de estos principios en por un lado, utilizar estándares para la representación y el acceso a los datos en la Web. Por el otro lado, los principios propagan insertar links entre datos de diferentes recursos. Estos link conectan todo Linked Data en un solo grafico global, similar a los links en la Web tradicional que conectan todos los documentos HTML en un solo espacio de información global. Se puede decir que LinkedData es para las hojas de cálculo y las bases de datos, lo que los documentos de la Web tradicional son para los procesadores de texto. (W3C, 2014)

El proyecto W3C SWEO Linking Open Data[52] fue creado en enero de 2007 por el W3C. El objetivo de la comunidad que lleva adelante este proyecto es extender la Web con bienes de datos comunes a partir de la publicación de variados conjuntos de datos abiertos en formato RDF en la Web y agregar links RDF entre los ítems de las diferentes fuentes de datos.

[51] http://www.w3.org/DesignIssues/LinkedData.html
[52] http://www.w3.org/wiki/SweoIG/TaskForces/CommunityProjects/LinkingOpenData


*Logo del Proyecto Linking Open Data* [53]

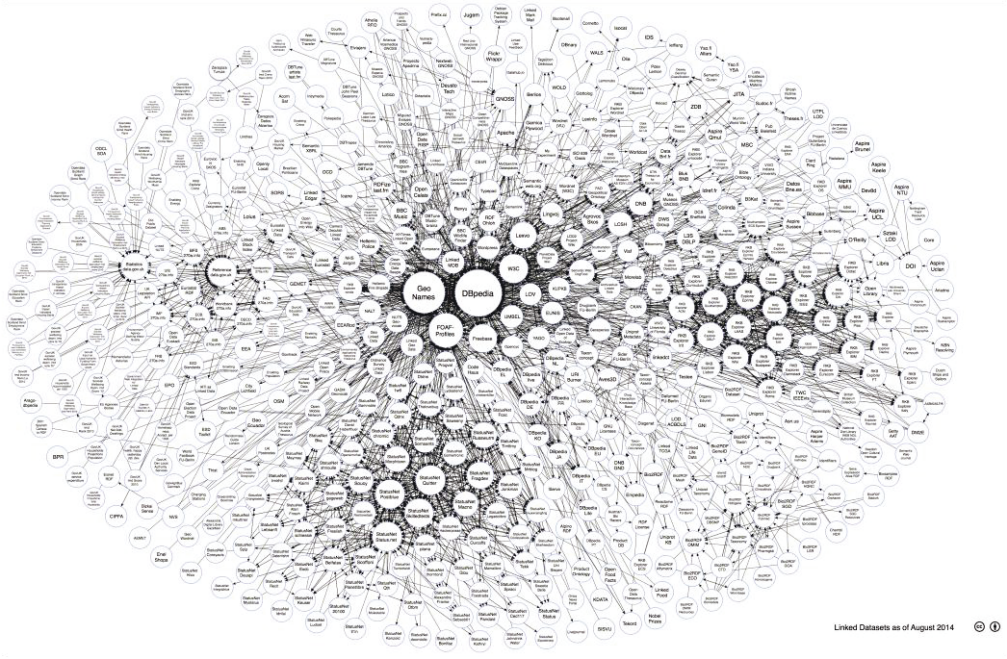
Los links RDF permiten navegar desde un ítem en un origen de datos hacia otro ítem de otro origen de datos diferente utilizando un navegador de la Web Semántica. Los links RDF pueden ser seguidos por los rastreadores de los motores de búsqueda de la Web Semántica, que proveen resultados para búsquedas sofisticadas y tienen capacidad de consultar sobre los rastreadores de datos. Como los resultados de las consultas son datos estructurados no simplemente links a paginas HTML, pueden ser utilizados por otras aplicaciones.

En la Web existen actualmente varias fuentes de datos libres interesantes. Algunos ejemplos son Wikipedia, Wikibooks [54], Geonames [55], MusicBrainz [56], WordNet [57] , la DBLP bibliography [58] y muchos más que son publicados bajo licencias Creative Commos [59] .

Existen ya varios esfuerzos para la publicación de datos, como son el proyecto DBpedia, Geonames Ontology [60], el D2R Server publishing the DBLP bibliography [61] y el servidor de música DBTune [62]. También se está trabajando en la conexión entre estas fuentes de datos. Por ejemplo, las descripciones RDF de DBpedia para las ciudades incluyen `owl:sameAs` links hacia los datos que contiene Geonames sobre las ciudades. Otro ejemplo es el de RDF Book Mashup [63] que conecta los autores de libros con los autores de papers de la DBLP bibliography [64]. (W3C, 2014)

El diagrama del proyecto Linked Open Data da un panorama de los conjuntos de datos linkeados disponibles en la Web.

[53] http://www.w3.org/wiki/images/1/1d/SweoIG%24%24TaskForces%24%24CommunityProjects%24%24LinkingOpenData%24LoDLogo.gif
[54] http://www.wikibooks.org/
[55] http://www.geonames.org/
[56] https://musicbrainz.org/
[57] http://wordnet.princeton.edu/wordnet/download/current-version/
[58] http://dblp.uni-trier.de/db/
[59] http://creativecommons.org/licenses/
[60] http://www.geonames.org/ontology/documentation.html
[61] http://dblp.l3s.de/d2r/
[62] http://dbtune.org/
[63] http://wifo5-03.informatik.uni-mannheim.de/bizer/bookmashup/
[64] http://lists.w3.org/Archives/Public/semantic-web/2006Dec/0022




*The Linking Open Data cloud diagram* [65]

Ya vimos en la sección 2.3.2 como acceder a los datos de DBpedia a través de SPARQL endpoint. Otra modalidad provista por Dbpedia para acceder a sus datos es a través de Linked Open Data.

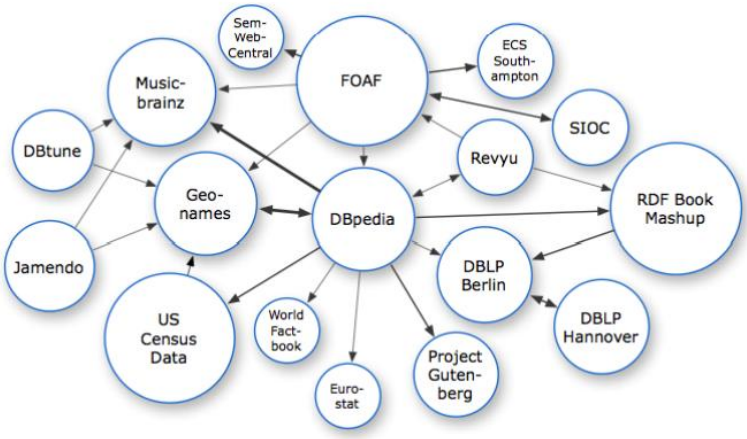
Los identificadores de recursos de DBpedia (por ejemplo http://dbpedia.org/page/Berlin ) están configurados para retornar:

- Descripciones RDF cuando son accedidos a través de agentes de la Web Semántica, como navegadores de datos o rastreadores de los motores de búsqueda de la Web Semántica.
- Una vista HTML simple de la misma información de los navegadores de la Web tradicional. Protocolos de negociación de contenido HTTP son utilizados para retornar los datos en un formato apropiado. (Bizer, et al., 2009)

Agentes de la Web Semántica que pueden acceder a Linked Data son:

- Navegadores de Web Semántica como Disco [66], Tabulator (Berners-Lee, et al., 2006), o OpenLink Data Web Browser [67].
- Rastreadores de la Web Semántica como SWSE[68] y Swoogle[69]
- Agentes de consulta de Web Semántica como Semantic Web Client library [70] y el cliente SemWeb para SWI prolog [71] (Auer, et al.)

[65] http://lod-cloud.net/
[66] http://wifo5-03.informatik.uni-mannheim.de/bizer/ng4j/disco/
[67] http://demo.openlinksw.com/DAV/JS/rdfbrowser/index.html
[68] http://www.w3.org/2001/sw/wiki/SWSE
[69] http://swoogle.umbc.edu/
[70] http://wifo5-03.informatik.uni-mannheim.de/bizer/ng4j/semwebclient/
[71] http://www.swi-prolog.org/web/




***Conjuntos de datos que están linkeados con DBpedia (Auer, et al.)***

# 3. BLUEFINDER

En el presente capítulo se hace una descripción de los sistemas de recomendación basados en filtrado colaborativo, se desarrolla la aplicación del algoritmo BlueFinder y se ejemplifican algunas de las debilidades encontradas en los resultados del mismo.

## 3.1. Sistemas de recomendación basados en filtrado colaborativo

Los Sistemas de Recomendación son herramientas y técnicas de software que proporcionan una serie de sugerencias personalizadas (recomendaciones) a los usuarios sobre ítems que pueden ser de su interés. Dichas sugerencias se relacionan con diferentes procesos de decisión como que ítems comprar, que música escuchar o que noticias leer.

Ítem es el término usado generalmente para referirse al tema específico que el sistema recomienda al usuario. Un sistema de recomendación normalmente se enfoca en un tipo específico de ítem, (por ejemplo, libros, noticias, música, etc.), y de acuerdo con su diseño, su interfaz grafica y las técnicas de recomendación utilizadas para generar dichas recomendaciones, son personalizados para ofrecer sugerencias útiles y eficaces para ese tipo específico de ítem. (Ricci, Rokach, & Shapira, 2011)

Es decir, se encargan de guiar a un usuario mediante recomendaciones en la búsqueda de aquellos servicios o productos que puedan ser más atractivos para él, modificando el proceso de navegación y búsqueda. Esto es sin duda una gran ventaja para los usuarios, que encontrarán lo que necesitan de una forma más rápida, cómoda y fácil dentro de las enormes bases de datos que ofertan las tiendas electrónicas en Internet y además descubrirán nuevos productos o servicios que le puedan ser atractivos, que de otra manera les hubiese sido mucho más difícil o incluso imposible de encontrar.

Existen diversos tipos de sistemas de recomendación, siendo los dos más importantes y utilizados los siguientes:

- **Sistemas de recomendación basados en contenido** .Se basan en la similitud entre objetos, es decir, predicen que para un usuario serán de interés aquellos objetos muy parecidos en su contenido con aquellos que ya sabemos que han sido de su agrado. Dos ejemplos muy populares son Amazon[72] y Youtube[73], que registran las páginas visitadas por los usuarios para luego recomendarles otras similares.
- **Sistemas de recomendación colaborativos** .Son más cercanos a la forma de pensar de los seres humanos que los basados en contenido. Son aquellos en los que las recomendaciones se realizan basándose solamente en los términos de similitud entre los gustos de los usuarios. (Albín Rodríguez)

Los sistemas que recomiendan elementos del gusto de los usuarios, cuando aplican la técnica de filtrado basado en contenido, cuentan con algunas limitaciones. Por un lado, los elementos que sugieren deben tener atributos que permitan describirlos.

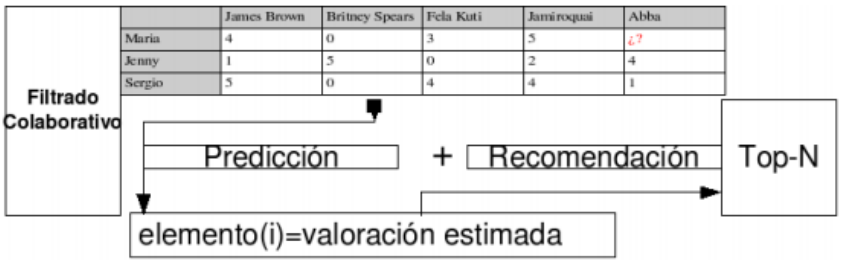
[72] http://www.amazon.com/
[73] https://www.youtube.com/

Con la tecnología actual y medios como música, fotografía, arte, video o elementos físicos, se dificulta esta tarea, ya que generalmente estos ítems no pueden ser analizados automáticamente debido a la dificultad de obtener un conjunto de atributos que permitan describirlos realmente. Las predilecciones de los usuarios pueden deberse a las sensaciones que provocan tales elementos y no a sus atributos. Por ejemplo, el agrado por una canción puede ser ocasionado por la satisfacción de escucharla, no influyendo en ello su género, autor, intérprete o frecuencia musical.

Por otro lado, dadas las características de la técnica de filtrado basado en contenido, estos sistemas no permiten que el usuario descubra ítems de contenido totalmente distinto a los ya conocidos por él aunque puedan ser de su interés. Por ejemplo, en un sistema en que se recomienda películas teniendo en cuenta el género preferido por el usuario, si éste manifestó interés por películas de suspenso, el sistema omitirá realizarle recomendaciones de películas de otros estilos distintos a suspenso, sin sugerirle que experimente otros géneros.

Con el fin de paliar estas limitaciones surgen los Sistemas de Recomendación basados en filtrado colaborativo. En estos sistemas, el filtrado de información no se realiza analizando las características de los elementos a recomendar, sino que para ello se tiene en cuenta únicamente las valoraciones de todos los usuarios sobre los ítems del sistema.

El mecanismo general de esta técnica consiste en calcular un puntaje de predicción, o sea, el puntaje que se estima que un usuario daría a un determinado elemento del sistema no conocido hasta el momento por él, basándose en los puntajes que las personas han ofrecido sobre los distintos elementos del sistema. Con ese valor de predicción se concluye si ese ítem podría ser de interés para el usuario y en cuyo caso se le recomienda.




***Modelo de un sistema de filtrado colaborativo (Galán Nieto, 2007)***

El primer sistema de recomendación basado en filtrado colaborativo, Tapestry (Goldberg, Nichols, Oki, & Douglas, 1992), fue desarrollado por XeroX PARC en el año 1992, adoptando la frase "Filtrado Colaborativo" que a partir de ese momento fue el término utilizado. Este sistema permitía a los usuarios realizar anotaciones sobre documentos electrónicos publicados en un grupo de noticias, para que luego fueran utilizadas por los restantes usuarios. Tapestry utilizaba la idea de filtrado colaborativo pero no lo hacía de forma automática. En este caso el sistema no era quien realizaba la sugerencia sino que permitía a los usuarios, a partir de las anotaciones que se realizaban sobre los artículos, concluir si un determinado elemento era de su interés. En 1997, Resnick y Varian (Resnick & Varian, 1997) proponen llamarles a estas aplicaciones "Sistemas de Recomendación" por dos razones: en primer lugar porque puede ocurrir que los usuarios no colaboren explícitamente entre ellos y en segundo lugar porque el sistema puede sugerir elementos no conocidos hasta el momento por el usuario y en ese caso se estaría realizando una "recomendación".

Dado que los algoritmos de filtrado colaborativo tienen la particularidad de no analizar el contenido de los ítems en el cálculo de predicción, pueden aplicarse a una gran variedad de tipos de elementos como libros, películas, música, comidas, etc.

Algunos enfoques se basan en calcular la similitud entre usuarios, y a partir de sus preferencias se obtienen los elementos a recomendar. Para cada usuario se crea un conjunto de "vecinos cercanos": personas cuyas evaluaciones tienen grandes semejanzas a las del usuario que solicita la recomendación. Los resultados para

los elementos no calificados por él, se predicen en base a la combinación de puntajes conocidos de los vecinos cercanos.

Entonces el procedimiento general aplicado en estos sistemas se resume de la siguiente forma:

1. Los usuarios expresan sus valoraciones sobre elementos del sistema, generalmente mediante una escala numérica.
2. A partir de esa información, se intenta predecir el puntaje que daría el usuario que solicita la recomendación (usuario activo), a los elementos del sistema no conocidos hasta el momento por él.
3. De las predicciones calculadas se seleccionan los elementos con valores más altos para realizar la recomendación. Existen muchas maneras de generar las predicciones mencionadas, desde diferentes enfoques y basando los cálculos en diferentes algoritmos. (Betarte, Machado, & Molina, 2006)

Se pueden distinguir dos tipos generales de algoritmos de Filtrado Colaborativo:

- **Algoritmos de filtrado colaborativo basados en memoria, o algoritmos de vecinos cercanos (Nearest Neighbour)** (Zhang & Zhou, 2005)**.** Utilizan toda la base de datos de elementos y usuarios para generar predicciones. Primeramente emplean técnicas estadísticas para encontrar a vecinos, es decir usuarios con un historial de valoraciones sobre los elementos similar al usuario actual. Una vez que se ha construido una lista de vecinos se combinan sus preferencias para generar una lista con los N elementos más recomendables para el usuario actual.

  Entre sus inconvenientes se encuentra la necesidad de disponer de un número mínimo de usuarios con un número mínimo de predicciones cada uno, incluido el usuario para el que se pretende realizar la recomendación.

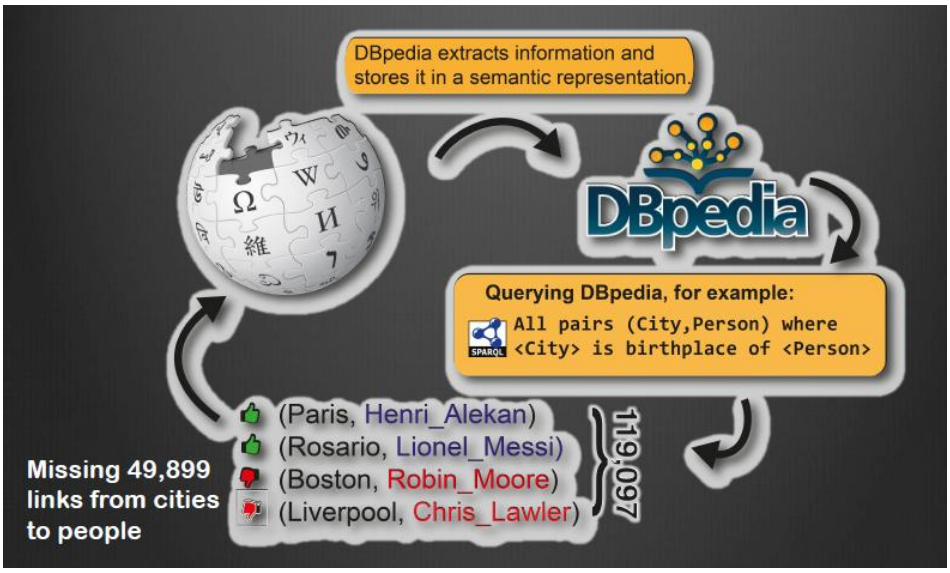
- **Algoritmos de filtrado colaborativo basados en Modelo**. Desarrollan primero un modelo de los ratings del usuario. Tratan el problema como un problema de predicción estadística y calculan el valor esperado para cada ítem en función de los ratings anteriores.

  Para ello se utilizan distintos algoritmos de aprendizaje Clustering o redes neuronales como las Redes de Funciones de Base Radial (RBFN). Por ejemplo utilizando clustering se trata de clasificar a un usuario en particular dentro de una clase de usuarios y a partir de ahí se estiman las probabilidades condicionadas de esa clase hacia los elementos a evaluar. En general, ante las consultas responden más rápido que los basados en memoria, pero por contra necesitan de un proceso de aprendizaje intensivo. (Galán Nieto, 2007)

El algoritmo BlueFinder, desarrollado en la siguiente sección es un algoritmo de filtrado colaborativo.

## 3.2. BlueFinder

Muchas de las relaciones existentes entre recursos en DBpedia no existen entre los correspondientes artículos de Wikipedia, por lo tanto se produce una pérdida de información en la web social que sí está disponible en la web semántica. Realizando una consulta sobre una propiedad semántica en la DBpedia se obtienen muchos más resultados que navegando en la Wikipedia a través del sistema de categorías que representa la misma relación, e incluso a veces no se encuentra dicha relación.




*Gap de información entre la Web Semántica y la Web Social*

La figura representa el gap de información entre la web semántica y la web social. DBpedia extrae información de Wikipedia y la almacena en una representación semántica. Esto permite realizar consultas con SPARQL en la base de datos de DBpedia, por ejemplo:

Todos los pares (Ciudad, Persona) donde <Ciudad> es el lugar de nacimiento (propiedad semántica *birthPlace*) de <Persona>

Esta consulta arroja en el ejemplo 119097 resultados.

Si se busca en Wikipedia, navegando a través de los links se obtienen muchos menos resultados, se pierden alrededor de 50000 relaciones entre personas y su lugar de nacimiento, siendo que la información está en Wikipedia, pero no es accesible ya que no está vinculada.

En la figura se puede observar que se puede navegar en la Wikipedia desde Paris hasta Henri Alekan, de igual forma desde Rosario a Lionel Messi, pero no se puede navegar desde Boston hasta Robin Moore o de Liverpool a Chris Lawler. Estos dos últimos son los que vamos a denominar links perdidos

Para resolver este gap de información entre la web semántica y la web social, podemos aplicar el algoritmo BlueFinder (Torres, Skaf-Molli, Molli, & Diaz, 2013).

BlueFinder es un sistema de recomendación de filtrado colaborativo. Utiliza un algoritmo basado en memoria (Breese, Heckerman, & Kadie., 1998) que realiza un ranking de predicciones sobre una colección completa de caminos navegacionales previamente calificados. Como consecuencia, el valor de clasificación desconocido para un par de recursos y un camino navegacional será computado y agregado al ranking de otros pares similares para el mismo camino navegacional.

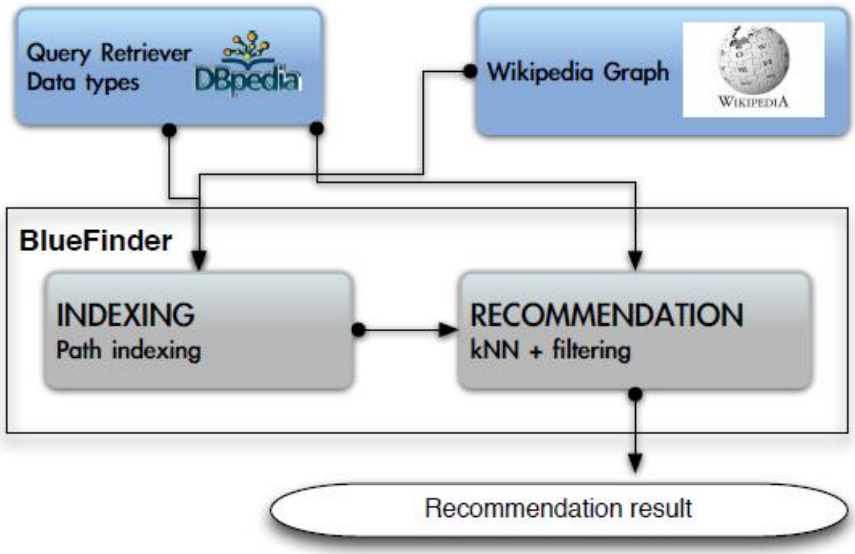
El sistema de recomendación retorna un conjunto de caminos navegacionales que puede ser utilizados para representar la propiedad semántica analizada. Las recomendaciones deben incluir al menos un camino navegacional que represente la relación semántica siguiendo las convenciones de la comunidad Wikipedia (Wikipedia, 2015)

BlueFinder se basa en el algoritmo popular de K-Nearest Neighbors(kNN) y el algoritmo Multi laber KNN (Zhang & Zhou, 2005) adaptados al contexto de DBpedia y Wikipedia.

Básicamente, el algoritmo BlueFinder identifica una cantidad (k) pasada como parámetro de vecinos conectados para los pares desconectados (from,to), y luego selecciona caminos navegacionales de los k vecinos más cercanos. Finalmente los caminos navegacionales seleccionados son el conjunto de recomendaciones.

El algoritmo BlueFinder está organizado en dos pasos principales:

- **Indexación**: Dada una propiedad semántica de DBpedia, este paso genera el conjunto de caminos navegacionales para la propiedad semántica dada.
- **Recomendación**: para un par de artículos de Wikipedia desconectados, el algoritmo genera un conjunto de caminos navegacionales que mejor representan la convención para conectar dichos artículos. El algoritmo estima los caminos navegacionales que maximizan la utilidad funcional para el par desconectado basándose en la utilidad de los caminos navegacionales de pares conectados similares. Implementa el algoritmo KNN utilizando el conjunto generado en el paso anterior. (Torres, Co-evolución entre la Web Social y la Web Semántica, 2014)




*Esquema que representa el funcionamiento de BlueFinder*

Como resultado de su aplicación obtenemos un conjunto de recomendaciones de una relación entre un par dado de artículos de Wikipedia y una propiedad semántica dada, representadas en forma de caminos navegacionales, ordenados desde el más relevante al menos relevante, y que a partir de ahora llamaremos path queries. En el presente trabajo se analizó la propiedad semántica *birthPlace*

Ejemplo del resultado de la ejecución del algoritmo aplicado al par (London, Ben_Cross) con los siguientes parámetros

- K (vecino) =10
- número máximo de recomendaciones 10
- propiedad analizada *birthPlace*

```
resources:
        to: London
        from: Ben_Cross

related_resources:
        (0.0) London , Charlie_Cox
        (0.038461538461538436) London , Joseph_A._Bennett
        (0.038461538461538436) London , Ritchie_Coster
        (0.041666666666666685) London , Christopher_Neame
        (0.041666666666666685) London , Danny_Webb_(actor)
        (0.041666666666666685) London , Elyes_Gabel
        (0.041666666666666685) London , Katherine_Woodville_(actress)
        (0.041666666666666685) London , Michael_Byrne_(actor)
        (0.041666666666666685) London , Nicholas_Clay
        (0.041666666666666685) London , Peter_Bowles
```

**Paths:**

```
Path1: {#from / * / Cat:Actors_from_#from / #to=1001}

Path2: {#from / * / Cat:Actors_from_#from / #to=1002}

Path3: {#from / * / Cat:Actors_from_#from / #to=1003, #from / * /
Cat:People_from_Edmonton,_#from / #to=1}

Path4: {#from / * / Cat:Actors_from_#from / #to=1004, #from / * /
Cat:People_from_Edmonton,_#from / #to=1}

Path5: {#from / * / Cat:Actors_from_#from / #to=1005, #from / * /
Cat:People_from_Edmonton,_#from / #to=1}

Path6: {#from / * / Cat:Actors_from_#from / #to=1006, #from / * /
Cat:People_from_Edmonton,_#from / #to=1}

Path7: {#from / * / Cat:Actors_from_#from / #to=1007, #from / * /
Cat:People_from_Edmonton,_#from / #to=1}

Path8: {#from / * / Cat:Actors_from_#from / #to=1008, #from / * /
Cat:People_from_Edmonton,_#from / #to=1}

Path9: {#from / * / Cat:Actors_from_#from / #to=1009, #from / * /
Cat:People_from_Edmonton,_#from / #to=1}

Path10: {#from / * / Cat:Actors_from_#from / #to=1010, #from / * /
Cat:People_from_Edmonton,_#from / #to=1}
time: 51063
```

**relevantPaths:**

```
#from / Cat:#from / Cat:People_from_#from / Cat:People_from_#from_by_occupation /
Cat:Actors_from_#from / #to
```

Cada resultado contiene un par de recursos (lugar de nacimiento, persona), los vecinos analizados, los path recomendados, y los relevant path.

- **resource**: es el par a analizar
- **related_resources**: son los k vecinos obtenidos por la aplicación del algoritmo kNN (Zhang & Zhou, 2005) que utiliza BlueFinder.
- **Path1** representa a cualquier path query que comience con #from y finalice con Cat:Actors_from_#from / #to
- El símbolo * representa cualquier secuencia.
- Los path recomendados (path queries) son los caminos navegacionales resultantes de la aplicación del algoritmo BlueFinder, para relacionar el par de recursos.
- Los relevant path, representan los caminos existentes en Wikipedia para navegar de un recurso al otro.

1path: {#from / * / Cat:Actors_from_#from / #to=1001}

Se traduce como:

Cat:#from / Cat:People_from_#from /

Cat:People_from_#from_by_occupation / Cat:Actors_from_#from / #to

En este ejemplo:

Para el par (London, Ben_Cross) el camino real aplicado a Wikipedia seria reemplazando los #**from** con London y los #**to** con Ben_Cross

- London / Cat:London / Cat: People_from_London / Cat:

People_from_London_by_occupation / Cat:Actors_from_London / Ben_Cross

## 3.3. Debilidades detectadas en BlueFinder

A continuación se muestran varios ejemplos de los resultados arrojados por BlueFinder del dataset utilizado y el análisis realizado sobre los mismos.

Cabe recordar cómo se mencionó anteriormente que se considera para el presente trabajo semánticamente incorrecta toda recomendación que no represente la propiedad semántica analizada *birthPlace* que corresponde estrictamente al lugar de nacimiento de una persona.

Cada ejemplo, tomado del dataset analizado arroja 10 path, algunos de los cuales se repiten, por lo tanto se analizaron solamente los que son diferentes.

### 3.3.1. Ejemplo 1

```
resources:
         to: Milan
         from: Giancarlo_Brusati

related_resources:

         (0.13333333333333336) Milan , Renzo_Minoli
         (0.1764705882352941) Milan , Carlo_Agostoni
         (0.20833333333333331) Milan , Guido_Monzino
         (0.2142857142857143) Milan , Piero_Trapanelli
         (0.22692307692307695) Turin , Alfredo_Pezzana
         (0.23076923076923078) Milan , Anna_Sforza
         (0.23076923076923078) Milan , Nino_Rota
         (0.23333333333333334) Milan , Andrea_Vaturi
         (0.23333333333333334) Milan , Camilla_Spelta
         (0.23333333333333334) Milan , Marco_Fabbri

Paths:

Path1: {#from / * / Cat:Sportspeople_from_#from / #to=1002}

Path2: {#from / * / Cat:Sportspeople_from_#from / #to=1004}

Path3: {#from / * / Cat:Sportspeople_from_#from / #to=4, #from / * / Cat:People_from_#from /
#to=3}

Path4: {#from / * / Cat:Sportspeople_from_#from / #to=6, #from / * / Cat:People_from_#from /
#to=3, #from / * / Cat:A.C._#from_players / #to=1}

Path5: {#from / * / Cat:Sportspeople_from_#from / #to=8, #from / * / Cat:People_from_#from /
#to=3, #from / * / Cat:A.C._#from_players / #to=1}

Path6: {#from / * / Cat:Sportspeople_from_#from / #to=8, #from / * / Cat:People_from_#from /
#to=6, #from / * / Cat:A.C._#from_players / #to=1, #from / * / Cat:House_of_Sforza / #to=1}

Path7: {#from / * / Cat:People_from_#from / #to=9, #from / * / Cat:Sportspeople_from_#from /
#to=8, #from / * / Cat:A.C._#from_players / #to=1, #from / * / Cat:House_of_Sforza / #to=1}

Path8: {#from / * / Cat:Sportspeople_from_#from / #to=10, #from / * / Cat:People_from_#from /
#to=9, #from / * / Cat:A.C._#from_players / #to=1, #from / * / Cat:House_of_Sforza / #to=1}

Path9: {#from / * / Cat:Sportspeople_from_#from / #to=12, #from / * / Cat:People_from_#from /
#to=9, #from / * / Cat:A.C._#from_players / #to=1, #from / * / Cat:House_of_Sforza / #to=1}

Path10: {#from / * / Cat:Sportspeople_from_#from / #to=14, #from / * / Cat:People_from_#from /
#to=9, #from / * / Cat:A.C._#from_players / #to=1, #from / * / Cat:House_of_Sforza / #to=1}
              time: 28580
```

```
relevantPaths:
        #from / Cat:Communes_of_the_Province_of_#from / Cat:#from / Cat:Sport_in_#from /
Cat:Sportspeople_from_#from / #to ,
        #from / Cat:#from / Cat:Sport_in_#from / Cat:Sportspeople_from_#from / #to
```

Para este ejemplo solo arroja 4 paths diferentes. A continuación analizamos su nivel de relevancia y precisión.

**#from / * / Cat:Sportspeople_from_#from / #to**

- *Es semánticamente correcto.*
- *Es relevant path.*
- *Está en la mejor sucategoria, aunque se recomienda subcategorizar por deporte por la cantidad de artículos que posee la misma*

Como podemos ver la primera recomendación es una recomendación destacada y correcta.

**#from / * / Cat:People_from_#from / #to**

- *Es semánticamente correcto.*
- *No es relevant path.*
- *Existe mejor subcategoria (**Cat:Sportspeople_from_#from)***

Esta es una recomendación semánticamente correcta, pero existe una subcategoría que representa mejor la propiedad semántica, que es el caso anterior.

**#from / * / Cat:A.C._#from_players / #to**

- *No es semánticamente correcto.*
- *No es relevant path.*
- *Existe mejor subcategoria (**Cat:Sportspeople_from_#from)***

Esta recomendación más allá de no ser semánticamente correcta, tampoco es correcto decir que Giancarlo Brusati pertenece a la Asociación de futbolistas de Milán, cuando en realidad es esgrimista y no futbolista.

**#from / * / Cat:House_of_Sforza / #to**

- *No es semánticamente correcto.*
- *No es relevant path.*
- *No hay información que perteneciera a esta familia*

Esta recomendación más allá de no ser semánticamente correcta, es errónea, ya que no hay evidencia en Wikipedia para decir que Giancarlo Brusati pertenece a House of Sforza.

Este tipo de errores se puede dar cuando el algoritmo toma como válidos los path vinculados a los vecinos relacionados en los parámetros de ejecución del mismo, en este caso Piero Trapanelli pertenece a la Asociación de futbolistas de Milán y Anna_Sforza pertence a House of Sforza.

### 3.3.2. Ejemplo 2

```
resources:
```

```
          to Milan
          from Emilio_Gattoronchieri

related_resources:
          (0.1111111111111111) Milan, Oliviero_Mascheroni
          (0.13333333333333336) Milan, Giuseppe_Mettica
          (0.1388888888888889) Milan, Giuseppe_Castelli_(footballer)
          (0.15625) Milan, Mario_Ciminaghi
          (0.16666666666666669) Milan, Elpidio_Coppa
          (0.16666666666666669) Milan, Giovanni_Bolzoni (footballer)
          (0.16666666666666669) Milan , Giuseppe_Ballerio
          (0.16666666666666669) Milan , Umberto_Scotti
        (0.1764705882352941) Milan , Edgardo_Rebosio
        (0.1764705882352941) Milan , Luigi_Rizzi_(footballer)

Paths:

Path1: {#from / * / Cat:A.C._#from_players / #to=1001, #from / * / Cat:Inter_#from_players /
#to=1001}

Path2: {#from / * / Cat:Inter_#from_players / #to=1002, #from / * / Cat:A.C._#from_players /
#to=1}

Path3: {#from / * / Cat:Inter_#from_players / #to=4, #from / * / Cat:A.C._#from_players / #to=1}

Path4: {#from / * / Cat:Inter_#from_players / #to=6, #from / * / Cat:A.C._#from_players / #to=1}

Path5: {#from / * / Cat:Inter_#from_players / #to=8, #from / * / Cat:A.C._#from_players / #to=3}

Path6: {#from / * / Cat:Inter_#from_players / #to=9, #from / * / Cat:A.C._#from_players / #to=4}

Path7: {#from / * / Cat:Inter_#from_players / #to=11, #from / * / Cat:A.C._#from_players / #to=4}

Path8: {#from / * / Cat:Inter_#from_players / #to=11, #from / * / Cat:A.C._#from_players / #to=5}

Path9: {#from / * / Cat:Inter_#from_players / #to=13, #from / * / Cat:A.C._#from_players / #to=5}

Path10: {#from / * / Cat:Inter_#from_players / #to=14, #from / * / Cat:A.C._#from_players /
#to=5}

relevantPaths:
        #from / Cat:#from / Cat:Sport_in_#from / Cat:A.C._#from / Cat:A.C._#from_players / #to ,
        #from / Cat:#from / Cat:Sport_in_#from / Cat:Inter_#from / Cat:Inter_#from_players / #to
```

Para este ejemplo solo arroja 2 paths diferentes. A continuación analizamos su nivel de relevancia y precisión.

### #from / * / Cat:A.C._#from_players / #to

- *No es semánticamente correcto.*
- *Es relevant path.*
- *No Es la mejor categoría.*
- *Se recomienda la subcategoria Sportspeople from Milan*

Esta recomendación no es semánticamente correcta, así mismo es un relevant path en Wikipedia. No es la mejor categorización para la propiedad semántica analizada ya que existe la subcategoría Sportspeople from Milán que representa su lugar de nacimiento, y no aparece en los relevant path, ni en las recomendaciones de BlueFinder.

### #from / * / Cat:Inter_#from_players / #to

- *No es semánticamente correcto.*
- *Es relevant path.*
- *No Es la mejor categoría.*
- *Se recomienda la subcategoria Sportspeople from Milan*

Ídem la recomendación anterior.

En estos casos las personas son categorizadas por su profesión en lugar de su lugar de nacimiento, siendo que existe una subcategoría para los deportistas nacidos en Milán. Incluso en el infobox de, Emilio_Gattoronchieri aparece que es nacido en Milán. Sería interesante que el BlueFinder hiciera la recomendación con el vínculo a la categoría mencionada.

**Emilio Gattoronchieri**

| Personal information | |
|---|---|
| Date of birth | February 7, 1912 |
| Place of birth | Milan, Italy |
| Height | 1.74 m (5 ft 8½ in) |
| Playing position | Midfielder |

| Senior career* | | | |
|---|---|---|---|
| Years | Team | Apps† | (Gls)† |
| 1931–1934 | Mottese | | |
| 1934–1936 | Milan | 7 | (0) |
| 1936–1938 | Ambrosiana-Inter | 15 | (0) |
| 1938–1940 | Venezia | 45 | (0) |
| 1940–1942 | Liguria | 38 | (0) |
| 1942–1943 | Pro Patria | 11 | (0) |
| 1943–1944 | Tresoldi Cassano | | |

* Senior club appearances and goals counted for the domestic league only.
† Appearances (Goals).

***Inbobox Emilio Gattoronchieri Wikipedia*** [74]

### 3.3.3. Ejemplo 3

```
resources:
          to Hungary
          from Bence_Gyurján

related_resources:
          (0.07692307692307693) Hungary , Gábor_Rajos
          (0.07692307692307693) Hungary , István_Kovács_(footballer)
          (0.10714285714285715) Hungary , Dániel_Rózsa
          (0.10714285714285715) Hungary , Ferenc_Rácz
          (0.10714285714285715) Hungary , Gábor_Nagy_(footballer)
          (0.10714285714285715) Hungary , Ignác_Irhás
          (0.10714285714285715) Hungary , Máté_Skriba
          (0.10714285714285715) Hungary , Richárd_Guzmics
          (0.10714285714285715) Hungary , Roland_Ugrai
          (0.10714285714285715) Hungary , Szabolcs_Schimmer

Paths:

Path1: {#from / * / Cat:People_from_Szombathely / #to=1001}

Path2: {#from / * / Cat:People_from_Szombathely / #to=1002}

Path3: {#from / * / Cat:People_from_Szombathely / #to=1003}

Path4: {#from / * / Cat:People_from_Szombathely / #to=1004}

Path5: {#from / * / Cat:People_from_Szombathely / #to=1005}

Path6: {#from / * / Cat:People_from_Szombathely / #to=5, #from / * / Cat:People_from_Miskolc /
#to=1}
```

[74] http://en.wikipedia.org/wiki/Emilio_Gattoronchieri

```
Path7: {#from / * / Cat:People_from_Szombathely / #to=5, #from / * / Cat:People_from_Celldömölk /
#to=1, #from / * / Cat:People_from_Miskolc / #to=1}

Path8: {#from / * / Cat:People_from_Szombathely / #to=6, #from / * / Cat:People_from_Celldömölk /
#to=1, #from / * / Cat:People_from_Miskolc / #to=1}

Path9: {#from / * / Cat:People_from_Szombathely / #to=6, #from / * / Cat:People_from_Celldömölk /
#to=1, #from / * / Cat:People_from_Miskolc / #to=1, #from / * / Cat:People_from_Békéscsaba /
#to=1}

Path10: {#from / * / Cat:People_from_Szombathely / #to=7, #from / * / Cat:People_from_Celldömölk
/ #to=1, #from / * / Cat:People_from_Miskolc / #to=1, #from / * / Cat:People_from_Békéscsaba /
#to=1}

relevantPaths:
            #from / Cat:#from / Cat:Hungarian_people /  Cat:People_by_city_in_#from /
Cat:People_from_Nyíregyháza / #to
```

Para este ejemplo solo arroja 4 paths diferentes. A continuación analizamos su nivel de relevancia y precisión.

### #from / * / Cat:People_from_Szombathely / #to

- *No es semánticamente correcto.*
- *No es relevant path. Nació en Nyíregyháza otra ciudad de Hungría*

### #from / * / Cat:People_from_Miskolc / #to

- *Ídem anterior*

### #from / * / Cat:People_from_Celldömölk / #to

- *Ídem anterior*

### #from / * / Cat:People_from_Békéscsaba / #to

- *Ídem anterior*

En este caso, ninguna de las recomendaciones de BlueFinder es semánticamente correcta, sin embargo el relevant path de Wikipedia es correcto, por lo tanto tendría que aparecer en las recomendaciones. Incluso en el infobox de Wikipedia está dicha información.

**Bence Gyurján**

| Personal information | | | |
|---|---|---|---|
| **Full name** | Bence Gyurján | | |
| **Date of birth** | 21 February 1992 (age 23) | | |
| **Place of birth** | Nyíregyháza, Hungary | | |
| **Height** | 1.70 m (5 ft 7 in) | | |
| **Playing position** | Midfielder | | |
| **Club information** | | | |
| **Current team** | Gyirmót | | |
| **Youth career** | | | |
| 2002–2007 | Nyíregyháza | | |
| 2007–2010 | Haladás | | |
| **Senior career*** | | | |
| **Years** | **Team** | **Apps†** | **(Gls)†** |
| 2010–2015 | Haladás | 44 | (2) |
| 2015- | Gyirmót | 0 | (0) |
| **National team‡** | | | |
| 2012 | Hungary U-20 | 2 | (0) |
| 2012– | Hungary U-21 | 7 | (1) |

* Senior club appearances and goals counted for the domestic league only and correct as of 26 October 2014.

† Appearances (Goals).

‡ National team caps and goals correct as of 5 March 2014

***Infobox Bence Gyurjan en Wikipedia*[75]**

[75] http://en.wikipedia.org/wiki/Bence_Gyurj%C3%A1n

### 3.3.4. Ejemplo 4

```
resources:
         to Middlesboro,_Kentucky
         from Trish_Suhr

related_resources:
         (0.08333333333333331) Brooklyn , David_Frye
         (0.08333333333333331) Queens , Steve_Hofstetter
         (0.16666666666666669) Brooklyn , Bill_Benulis
         (0.16666666666666669) Brooklyn , Chris_Rush
         (0.16666666666666669) Fremantle , JJ_DeCeglie
         (0.16666666666666669) Guangzhou , Yu_Haoming
         (0.16666666666666669) Manila , Tony_DeZuniga
         (0.16666666666666669) Queens , Adam_Ferrara
         (0.16666666666666669) Queens , Dave_Attell
         (0.16666666666666669) The_Bronx , Robert_Schimmel

Paths:

Path1: {#from / * / Cat:People_from_#from / #to=1002}

Path2: {#from / * / Cat:People_from_#from / #to=4}

Path3: {#from / * / Cat:People_from_#from / #to=4, #from / * /
List_of_people_from_#from,_New_York / #to=1}

Path4: {#from / * / Cat:People_from_#from / #to=6, #from / * /
List_of_people_from_#from,_New_York / #to=2}

Path5: {#from / * / Cat:People_from_#from / #to=8, #from / * /
List_of_people_from_#from,_New_York / #to=2}

Path6: {#from / * / Cat:People_from_#from / #to=8, #from / * /
List_of_people_from_#from,_New_York / #to=2, #from / * / Cat:Musicians_from_#from / #to=1}

Path7: {#from / * / Cat:People_from_#from / #to=11, #from / * /
List_of_people_from_#from,_New_York / #to=2, #from / * / Cat:Musicians_from_#from / #to=1, #from
/ * / Cat:University_of_Santo_Tomas_alumni / #to=1}

Path8: {#from / * / Cat:People_from_#from / #to=13, #from / * /
List_of_people_from_#from,_New_York / #to=2, #from / * / Cat:Musicians_from_#from / #to=1, #from
/ * / Cat:University_of_Santo_Tomas_alumni / #to=1}

Path9: {#from / * / Cat:People_from_#from / #to=15, #from / * /
List_of_people_from_#from,_New_York / #to=2, #from / * / Cat:Musicians_from_#from / #to=1, #from
/ * / Cat:University_of_Santo_Tomas_alumni / #to=1, #from / * / List_of_Long_Islanders / #to=1}

Path10: {#from / * / Cat:People_from_#from / #to=15, #from / * /
List_of_people_from_#from,_New_York / #to=2, #from / * / Cat:People_from_the_Bronx / #to=2, #from
/ * / Cat:Musicians_from_#from / #to=1, #from / * / Cat:University_of_Santo_Tomas_alumni / #to=1,
#from / * / List_of_people_from_the_Bronx / #to=1, #from / * / List_of_Long_Islanders / #to=1}

relevantPaths:
        #from / #to
        #from / Cat:Cities_in_Kentucky / Cat:People_by_city_in_Kentucky / Cat:People_from_#from /
        #to
```

Para este ejemplo arroja 7 paths diferentes. A continuación analizamos su nivel de relevancia y precisión.

**#from / * / Cat:People_from_#from / #to**

- *Es semánticamente correcto.*
- *Es un relevant path.*
- *No existe una mejor subcategoria. No se recomienda crearla, solo hay 11 personas notables en esta ciudad*

Esta es la única recomendación semánticamente correcta.

**#from / * / List_of_people_from_#from,_New_York / #to**

- *No es semánticamente correcto*
- *No es relevant path.*
- *No existe esta lista porque Middlesboro está en Kentucky no en New York son lugares diferentes*

En este caso se recomienda un path a una lista inexistente y errónea

**#from / * / Cat:Musicians_from_#from / #to**

- *No es semánticamente correcto.*
- *No es relevant path.*
- *No es Músico, es actriz y comediante.*
- *No existe la categoría.*

En este caso se recomienda un path a una categoría de músicos inexistente, cuando en realidad es actriz y comediante.

**#from / * / Cat:University_of_Santo_Tomas_alumni / #to**

- *No es semánticamente correcto.*
- *No es relevant path.*
- *No hay información en Wikipedia que haya asistido a esta Universidad.*

En este caso se recomienda un path a una categoría de una Universidad de la que no hay información en Wikipedia que haya asistido

**#from / * / List_of_Long_Islanders / #to**

- *No es semánticamente correcto.*
- *No es relevant path*
- *No hay información en Wikipedia que haya vivido en Long Island*

En este caso se recomienda un path a una lista donde no hay información en Wikipedia que haya vivido en ese lugar.

**#from / * / Cat:People_from_the_Bronx / #to**

- *No es semánticamente correcto.*
- *No es relevant path.*
- *No nació en el Bronx*

En este caso se recomienda un path a una categoría donde no nació, siendo que la primera recomienda el lugar correcto de nacimiento.

**#from / * / List_of_people_from_the_Bronx / #to**

- *No es semánticamente correcto.*
- *No es relevant path.*
- *No hay información en Wikipedia que haya vivido en el Bronx*

En este caso se recomienda un path a una categoría donde no hay información en Wikipedia que haya vivido.

Como mencionamos en el ejemplo 1 se considera que estos path con recomendaciones erróneas se deben a los vecinos elegidos por el algoritmo.

En el presente trabajo se analiza el nivel de precisión de estas recomendaciones, proponiendo una taxonomía de acuerdo a diferentes niveles de correctitud, ya que algunas pueden ser semánticamente incorrectas, como vimos en los ejemplos anteriores, y hasta se han encontrado relaciones no recomendadas por el BlueFinder que resultan óptimas como mecanismo de categorización.

# 4. CLASIFICACIÓN DE LAS RECOMENDACIONES

En este capítulo se describe la taxonomía propuesta para clasificar las recomendaciones del algoritmo BlueFinder, luego se describe la metodología utilizada para realizar dicha clasificación y por último se presenta una evaluación y análisis de los resultados obtenidos.

## 4.1. Taxonomía propuesta

En esta sección se presenta la taxonomía propuesta para clasificar las recomendaciones obtenidas de la aplicación del algoritmo BlueFinder, distinguiendo entre clases positivas y negativas, con un signo + o - delante de su nombre.

Que una clase sea positiva significa que se considera una recomendación correcta ya sea por su semántica, porque es un relevant path de Wikipedia o porque es la mejor categoría donde puede estar ubicada la relación analizada.

Son clases negativas todas aquellas en las que se detecta algún error o aspecto negativo del path recomendado.

Un path recomendado puede ser catalogado como perteneciente a clases positivas y también pertenecer a clases negativas, sin establecer ningún límite para esta clasificación.

### 4.1.1. Clases Positivas

**+(SemanticaOK)** : En el análisis realizado se considera semánticamente correcto todo path que esté directamente relacionado con la propiedad semántica analizada, en este caso *birthPlace*, entendiendo como *birthPlace* estrictamente lugar de nacimiento de la persona que se está observando.

Se consideran como correctas todas las categorías **People_from_#from** de Wikipedia (Personas del lugar #from) si la persona nació en el lugar expresado en #from, o las subcategorías que se derivan de ellas por ser más específicas de la persona que se está analizando.

Una excepción a esta regla sucede en artículos que representan poblaciones muy grandes, como países por ejemplo. En muchos de estos casos la categoría **People_from_#from** se divide en **People_by_ occupation** (Personas por ocupación) y **People_by_location** (Personas por ubicación). Para este análisis se considera semánticamente correcto el segundo camino (**People_by_location**), ya que la alternativa People_by_ occupation no indica necesariamente que la persona haya nacido en ese lugar, y se busca un subcategoría dentro de **People_by_location** que represente el lugar de nacimiento de la persona.

Otro caso que se presenta son las categorías por nacionalidad, por ejemplo **#from_writers** (French_writers).

Estas categorías se consideran semánticamente incorrectas por referirse a la nacionalidad de la persona, la que puede ser diferente al lugar de nacimiento. Típico es el caso de personas nacidas en un lugar y que desarrollaron su actividad en otra ciudad diferente, o personas con doble nacionalidad.

**+(EsRelevantPath)** : Esta clasificación es directa. Significa que el path recomendado es un relevant path.

**+(NoExisteMejorCategoría)** : Esta clase representa los path que son semánticamente correctos, son relevant path y son la mejor categoría recomendada para el par analizado. Para esto se buscó previamente si no existían mejores categorías disponibles en Wikipedia en donde se pudiera agregar el artículo, dando como resultado que la recomendada era la mejor.

**+(AgregarALaLista)** : Esta clase está compuesta por los path recomendados que son Listas de Wikipedia que representan el lugar de nacimiento de la persona del path analizado, pero como la persona no aparece como un artículo de la lista, se recomienda agregarla.

### 4.1.2. Clases Negativas

**-(SemanticaNo) :** es la opuesta a **+(SemanticaOK)**. Todo lo que no se considera semánticamente correcto, forma parte de esta clasificación.

**-(NoRelevantPath):** es la opuesta a **+(EsRelevantPath)**. Todos los path recomendados que no son relevant path de Wikipedia se encuadran en esta clase.

**-(MejorSubcategoría(parámetro)):** Esta clase significa que existe una mejor categorización dentro del mismo árbol de categorías de Wikipedia. El parámetro se compone de un número y la palabra "arriba" o "abajo". El número representa la cantidad de links que hay que recorrer desde el path recomendado hasta la categoría o subcategoría considerada semánticamente correcta. La palabra arriba o abajo significa si hay que navegar a las categorías superiores (arriba) o a las subcategorías (abajo) dentro del mismo árbol de categorías.

**-(MejorCategoría(parámetro1:parámetro2))** : Esta clase significa que existe una mejor categorización dentro de las categorías de Wikipedia para el par analizado, pero no es categoría o subcategoría del mismo árbol. Tiene dos parámetros. Ambos se forman con un número y la palabra "arriba" o "abajo" . El número representa los links que hay que navegar dentro de un mismo árbol desde el path recomendado hasta la categoría o subcategoría que permite luego navegar al árbol de la categoría semánticamente correcta. El primer parámetro representa la cantidad de links hacia una categoría más general (arriba) y el segundo representa la cantidad de links hacia la categoría más específica (abajo) o sea la que se considera semánticamente correcta.

**-(CategoríaIncorrecta):** Esta clase agrupa los casos en los que la recomendación es una categoría semánticamente incorrecta y no se puede medir la cercanía con la categoría semánticamente correcta.

**-(CategorizadoPorOtraPropiedadSemántica):** Esta clase representa los pares en donde el o los relevant path de Wikipedia categorizan a la persona del path analizado con otra propiedad semántica, como puede ser

la nacionalidad y ocupación, o algún lugar en donde se destacó la persona analizada, y no con la propiedad que se está analizando (*birthPlace*).

**-(SeRecomiendaRecategorizar(parámetro)):** Es subclase de la anterior. Como el par está en los relevant path categorizado por otra propiedad semántica se propone la categoría más representativa de la propiedad semántica analizada (*birthPlace*). El parámetro es la categoría propuesta.

**-(SeRecomiendaSubcategorizar):** En esta clase se clasifican los path recomendados en donde la categoría subcategoría o Lista de Wikipedia contiene muchos artículos, por lo que se recomienda subcategorizarla para una mejor navegación y búsqueda de las páginas que agrupa.

**-(CategoríaCorrectaNoRelevantPath):** Esta clase agrupa los path recomendados en los el artículo se encuentra clasificado en la categoría que mejor representa la propiedad semántica, pero no aparece entre los relevant path de Wikipedia.

**-(No existe):** Esta clase agrupa los casos en donde el path recomendado no existe en Wikipedia.

**-(EsCorrecto):** Es subclase de **-(No existe)**. El path recomendado no existe en Wikipedia pero sería semánticamente correcto si existiera.

**-(NoSeRecomiendaCrear):** Es subclase de **-(EsCorrecto).** El path recomendado no existe en Wikipedia, es semánticamente correcto, pero no se recomienda crearlo por ser un artículo muy pequeño o porque hay mejores categorías para clasificar el par analizado.

**-(ArtículoPequeño):** Es subclase de **-(NoSeRecomiendaCrear)** Es un path que no existe en Wikipedia, es semánticamente correcto, pero no se recomienda su creación por ser un artículo muy pequeño para subcategorizarlo.

**-(ExisteMejorSubcategoría):** Es subclase de **-(NoSeRecomiendaCrear)** Es un path que no existe en Wikipedia, es semánticamente correcto, pero no se recomienda su creación ya que existen mejores categorías para clasificar el par analizado

**-(NoEsCorrecto):** Es subclase de **-(No existe).** El path analizado no existe en Wikipedia, y sería incorrecto que existiera.

**-(MejorRelevantPahtNoRecomendado):** Esta clase se utiliza para los pares analizados donde uno de los relevant path lo representa mejor que todas las recomendaciones y no aparece entre ellas.

## 4.2. Descripción de la Metodología

A continuación se detalla la metodología aplicada para la clasificación de las recomendaciones en la taxonomía presentada previamente. Como resultado de esta clasificación se busca obtener una cantidad de datos que nos permita realizar un análisis del nivel de precisión de las recomendaciones.

Para la evaluación, se comenzó trabajando a partir del dataset proporcionado con los resultados obtenidos de la ejecución del BlueFinder para la propiedad semántica de DBpedia *birthPlace*

En la sección 3.2 se explicó cómo interpretar los resultados arrojados por el algoritmo BlueFinder. Luego en la sección 3.3 se mostraron varios ejemplos tomados textualmente del dataset utilizado para realizar el análisis de los mismos.

Los pasos seguidos fueron:

1. Se eligió al azar un par del dataset para analizar los path recomendados.
2. Se verifico en Wikipedia que los path recomendados por BlueFinder cumplan con la propiedad semántica *birthPlace*
3. Para cada path analizado se realizóó una breve descripción donde se detallo en principio los aspectos positivos:
    a. La correctitud semántica
    b. Si pertenecía a los relevant path de Wikipedia
    c. Si estaba categorizado en la mejor categoría que representa la propiedad semántica analizada.
4. Cuando los aspectos positivos no se cumplían se realizaba otros tipos de análisis:
    a. Primero se buscaba la distancia contando a cuantos links estaba de la categoría que se consideró como la correcta para la propiedad semántica. Esto se realizó navegando por los árboles de categorías[76] de Wikipedia.
    b. Hubo casos donde no se pudo llegar a la categoría correcta navegando por Wikipedia.
    c. En muchos casos se encontró que la persona estaba categorizada por otra propiedad semántica, como ocupación o nacionalidad, y se recomendó una mejor categoría existente en Wikipedia que represente la propiedad semántica
    d. En otros casos se encontró que la persona estaba categorizada en la categoría correcta, pero esta no aparecía en los relevant path de Wikipedia.
    e. En algunos casos se encontró que el path recomendado por BlueFinder no existe en Wikipedia, dentro de estos casos se analizo si de existir hubiera sido correcto y si valía la pena crearlo y otros casos donde no era correcto que existiera el path, ya que no tenia lógica.
    f. Otros casos que se detectaron fueron relevant path de Wikipedia que representaban correctamente la propiedad semántica, pero no aparecen en las recomendaciones propuestas por BlueFinder.
5. Por último se clasificaron los path recomendados en las clases correspondientes a la taxonomía definida.

Siguiendo con los ejemplos, de la sección 3.3 se obtuvieron los siguientes resultados:

Para el par (*Milan , Giancarlo_Brusati***)** se realizó la siguiente clasificación:

**#from / * / Cat:Sportspeople_from_#from / #to**

- *+(SemanticaOK)*
- *+(EsRelevantPath)*
- *+(NoExisteMejorCategoria)*

[76] http://en.wikipedia.org/wiki/Special:Categories

**#from / * / Cat:People_from_#from / #to**

- *+(SemanticaOK)*
- *-(NoRelevantPath)*
- *-(MejorSubcategoria(2))*

**#from / * / Cat:A.C._#from_players / #to**

- *-(SemanticaNo)*
- *-(NoRelevantPath)*
- *-(MejorCategoria(2arriba,1abajo))*

**#from / * / Cat:House_of_Sforza / #to**

- *-(SemanticaNo)*
- *-(NoRelevantPath)*
- *-(MejorCategoria(3arriba,3abajo))*

Para el par *(Milan , Emilio_Gattoronchieri)* se realizó la siguiente clasificación:

**#from / * / Cat:A.C._#from_players / #to**

- *-(SemanticaNo)*
- *-(MejorCategoria (3arriba1abajo))*
- *-(CategorizadoPorOtraPropiedadSemantica)*
- *-(SeRecomiendaRecategorizar(Sportspeople from#from))*

**#from / * / Cat:Inter_#from_players / #to**

- *Ídem anterior*

Para el par *(Hungary , Bence_Gyurján)* se realizó la siguiente clasificación:

**#from / * / Cat:People_from_Szombathely / #to**

- *-(SemanticaNo)*
- *-(NoRelevantPath)*
- *-(MejorCategoria(1arriba1abajo))*
- *-(MejorRelevantPahtNoRecomendado*)

**#from / * / Cat:People_from_Miskolc / #to**

- *Ídem anterior*

**#from / * / Cat:People_from_Celldömölk / #to**

- *Ídem anterior*

**#from / * / Cat:People_from_Békéscsaba / #to**

- *Ídem anterior*

Para el par *(Middlesboro,_Kentucky , Trish_Suhr)* se realizó la siguiente clasificación:

**#from / * / Cat:People_from_#from / #to**

- *+(EsRelevantPath)*
- *+(NoExisteMejorCategoria)*

**#from / * / List_of_people_from_#from,_New_York / #to**

- *-(SemanticaNo)*
- *-(NoRelevantPath)*
- *-(No existe) / -(NoEsCorrecto)*

**#from / * / Cat:Musicians_from_#from / #to**

- *Ídem anterior*

**#from / * / Cat:University_of_Santo_Tomas_alumni / #to**

- *-(SemanticaNo)*
- *-(NoRelevantPath)*
- *-(CategoriaIncorrecta)*

**#from / * / List_of_Long_Islanders / #to**

- *Ídem anterior*

**#from / * / Cat:People_from_the_Bronx / #to**

- *Ídem anterior*

**#from / * / List_of_people_from_the_Bronx / #to**

- *Ídem anterior*

Una vez finalizada la evaluación y clasificación de los casos seleccionados a partir del conjunto de datos, se obtuvo una muestra representativa para poder realizar un análisis del nivel de precisión de las recomendaciones.

## 4.3. Evaluación y análisis estadístico

En esta sección se evaluaron algunas de las cantidades de clases determinadas en la taxonomía propuesta.

Se analizaron 33 pares del tipo (lugar de nacimiento, persona) relacionados por la propiedad semántica *birthPlace*, y 173 path recomendados por el algoritmo BlueFinder para estos 33 pares.

De los 173 path recomendados, 65 representan a la propiedad semántica *birthPlace*, por lo tanto son

recomendaciones válidas del BlueFinder clasificados en **+(SemanticaOK)** .


***Cantidad de Path Analizados***

De estas 65, 30 no son relevant path de Wikipedia **-(NoRelevantPath),** por lo tanto, son recomendaciones de vínculos inexistentes en Wikipedia que se podrían recomendar a la comunidad para su evaluación. Los restantes 35 ya figuraban como vínculos en Wikipedia, por lo que la recomendación es válida y están clasificados en **+(EsRelevantPath).**

De los 108 que no representan a la propiedad semántica y están clasificados en -(SemanticaNo), 101 no son relevant path **-(NoRelevantPath),** por lo tanto, estas recomendaciones no son válidas a nuestro criterio.

De las restantes 7, que son relevant path en Wikipedia **+(EsRelevantPath),** todos están categorizados por otra propiedad semántica en la Wikipedia **-(CategorizadoPorOtraPropiedadSemántica),** como por ejemplo nacionalidad, club al que pertenecen, actividad que desarrollaron, etc. Para 4 de estas recomendaciones proponemos una categoría ya existente (parámetro) que representa mejor la propiedad semántica analizada **-(SeRecomiendaRecategorizar(parámetro)),** los 3 restantes, están categorizados en Wikipedia representando correctamente a la propiedad semántica, pero no aparecen entre los relevant path de Wikipedia resultantes del análisis del BlueFinder **-(MejorRelevantPahtNoRecomendado).**

Siguiendo con los 108 casos que consideramos semánticamente no válidos **-(SemanticaNo)**, se puede determinar que 23 pertenecen a la clasificación de **-(CategoríaIncorrecta)**, esto significa que no se puede medir la cercanía con la categoría semánticamente correcta.

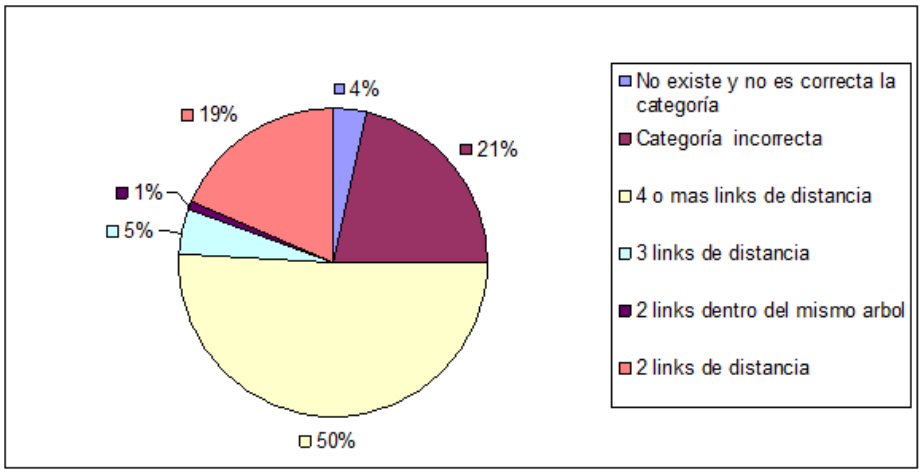
Hay 4 casos clasificados como **-(No existe)** y **-(NoEsCorrecto),** esto significa que la categoría no existe en WIkipedia y no debería ser creada porque no es correcta.

Hay 20 casos clasificados como **-(MejorCategoria(1arriba1abajo))**, lo que representa una cercanía con la categoría semánticamente correcta (2 links de distancia). Uno de estos casos está clasificado dentro de **-(MejorRelevantPahtNoRecomendado)**

Hay 5 casos clasificados como **-(MejorCategoria(1arriba2abajo))** y **-(MejorCategoria(2arriba1abajo))**, lo que representa una cercanía de 3 link de distancia.

Hay 55 casos clasificados a 4 o más links de distancia.

Un caso pertenece a la clasificación **-(MejorSubcategoria(2arriba))** significa que está dentro del mismo árbol a dos links de distancia navegando hacia las categorías más generales.




***Clasificación de los casos que no representan la propiedad birthPlace***

# 5. CONCLUSIONES Y TRABAJOS FUTUROS

En el presente capítulo se desarrollan los conocimientos adquiridos por la autora del trabajo, las conclusiones arribadas y la propuesta de trabajos futuros.

## 5.1. Conocimientos adquiridos

El presente trabajo, resultó personalmente en un aprendizaje de principio a fin, ya que nunca había abordado ninguno de los temas relacionados con el mismo.

Lo primero fue introducirme en la Web Semántica, con la bibliografía recomendada por la directora, e ir incorporando los distintos conceptos como son los lenguajes RDF, OWL, los lenguajes de consultas SPARQL y la filosofía que rodea a la evolución de la Web Semántica en capas, sus creadores y desarrolladores el consorcio W3C, los distintos aspectos que abarca como Linked Open Data, las Wikis Semánticas, los agentes inteligentes, etc., algunos de estos se abordaron en el capítulo 2, sobre todo los relacionados con la temática del trabajo en cuestión.

A continuación me introduje en la temática de la Wikipedia, la cual solo la conocía como usuario de consulta, pero no sabía nada de categorizaciones, infoboxes, etc., y de la misma forma con la DBpedia la cual no había utilizado nunca, por lo que también resultó en una amplia adquisición de conocimientos anteriores a la temática principal del trabajo, pero indispensables para poder llevarlo a cabo.

El siguiente paso fue entender el funcionamiento del algoritmo BlueFinder, como se relaciona con la Wikipedia, la DBpedia y la Web Semántica. Para luego poder interpretar los resultados que arroja y realizar el trabajo propuesto.

## 5.2. Conclusiones

Con los conocimientos adquiridos, y el estudio realizado sobre los resultados del BlueFinder y los links de Wikipedia, es que se propone una clasificación de las recomendaciones obtenidas del algoritmo para la propiedad semántica, *birthPlace*.

De esta clasificación surge la evaluación de la validez de las mismas, encontrando recomendaciones válidas y otras que no representan a la propiedad semántica analizada.

A partir de las primeras, algunas podrían ser puestas a consideración por la comunidad Wikipedia, ya que a pesar de ser válidas semánticamente, no se puede navegar entre el par de recursos analizado a través de los vínculos existentes en Wikipedia. Para las segundas, las semánticamente no válidas, se realizó un análisis para determinar el grado de incorrectitud, derivando en la clasificación mostrada anteriormente. En algunos casos se recomienda una categorización que represente con mayor exactitud la propiedad semántica.

## 5.3. Trabajos futuros

Como trabajo posterior, se podría extender este análisis a otras propiedades semánticas, a fin de obtener más resultados que puedan ser utilizados por los creadores de BlueFinder para mejorar la exactitud de las recomendaciones del algoritmo.

# REFERENCIAS


Albín Rodríguez, A. P. (s.f.). *Sistema de Recomendacion Colaborativo basado en algoritmos de filtrado mejorados.* Recuperado el 3 de 2015, de http://sinbad2.ujaen.es/cod/archivosPublicos/pfc/pfc_antonio_pedro.pdf

Alexa. (2015). *Alexa an Amazon.com Company*. Recuperado el 3 de 2015, de http://www.alexa.com/topsites

Antoniou, G., & van Harmelen, F. (2008). *A Semantic Web Primer, second edition.* London, England: The MIT Press.

Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., & Ives, Z. (s.f.). *DBpedia: A Nucleus for a Web of Open Data*. Recuperado el 3 de 2015, de http://richard.cyganiak.de/2008/papers/dbpedia-iswc2007.pdf

Aulet, J. (8 de 2011). *Capas de la web Semantica*. Recuperado el 2 de 2013, de Sindikos: http://www.sindikos.com/2011/08/capas-de-la-web-semantica/

Berners-Lee, T., Chen, Y., Chilton, C., Connolly, D., Ruth, D., Hollenbach, J., . . . Sheets, D. (2006). *Tabulator: Exploring and Analyzing linked data on the Semantic Web.* Obtenido de in Proceedings of the 3rd International Semantic Web User Interaction Workshop: http://swui.semanticweb.org/swui06/papers/Berners-Lee/Berners-Lee.pdf

Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The Semantic Web.

Betarte, L., Machado, R., & Molina, V. (2006). *PGMÚSICA - Sistema de recomendacion de Musica.* Recuperado el 3 de 2015, de Universidad de la Republica: http://www.fing.edu.uy/inco/grupos/pln/prygrado/InformePGMusica.pdf

Bizer, C. (3 de 2007). *Quality-Driven Information Filtering in the Context of Web-Based.* Recuperado el 3 de 2015, de http://wifo5-03.informatik.uni-mannheim.de/bizer/pub/DisertationChrisBizer.pdf

Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., & Hellmann, S. (2009). Dbpedia - a crystallization point for the web of data. En *Web Semantics: Science, Services and Agents on the World Wide Web* (Vol. 7(3), págs. 154-165). Elsevier B.V.

Breese, J., Heckerman, D., & Kadie., C. (1998). Empirical analysis of predictive algorithms for collaborative filtering. *In Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence* (págs. 43-52). Morgan Kaufmann Publishers Inc.

Burleson, C. (10 de 2007). *Introduction to the Semantic Web Vision and Technologies - Part 2 - Foundations*. Recuperado el 2013, de http://www.semanticfocus.com/blog/entry/title/introduction-to-the-semantic-web-vision-and-technologies-part-2-foundations/

*Capas de la Web Semantica*. (5 de 2007). Recuperado el 2 de 2013, de Web Semantica y Agentes: http://websemanticayagentes.blogspot.com.ar/2007/05/capas-de-la-web-semntica.html

Chernov, S., Iofciu, T., Nejdl, W., & Zhou, X. (s.f.). *Extracting Semantic Relationships between Wikipedia Categories.* Recuperado el 3 de 2015, de http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.73.5507&rep=rep1&type=pdf

Codina, L., & Rovira, C. (2006). *La Web Semantica*. Recuperado el 2 de 2013, de http://eprints.rclis.org/8899/1/web_semantica__.pdf

DBpedia. (9 de 2014). *The DBpedia Ontology (2014)*. Recuperado el 3 de 2015, de http://wiki.dbpedia.org/Ontology2014

DBpedia. (2015). *DBpedia*. Recuperado el 25 de 3 de 2015, de http://dbpedia.org/About

Foundation, W. (2015). *Our projects*. Recuperado el 3 de 2015, de http://wikimediafoundation.org/wiki/Our_projects#Wikipedia

Galán Nieto, S. M. (1 de 2007). *Filtrado Colaborativo y Sistemas de Recomendación.* Recuperado el 3 de 2015, de Universidad Calos III de Madrid: http://www.it.uc3m.es/jvillena/irc/practicas/06-07/31.pdf

Garcia Garcia, G. (6 de 2003). *RDF y RDF schema*. Recuperado el 3 de 2013, de http://www.matem.unam.mx/~grecia/semantic_web/rdf.html

Goldberg, D., Nichols, D., Oki, B. M., & Douglas, T. (12 de 1992). *Using collaborative filtering to weave an information Tapestry.* Recuperado el 3 de 2015, de https://www.ischool.utexas.edu/~i385d/readings/Goldberg_UsingCollaborative_92.pdf

IndustriaIT. (2014). *Wikidata, la propagación de la información*. Recuperado el 3 de 2015, de http://industriait.com.ar/blog/wikidata-la-propagacion-de-la-informacion

Kaminose, M., Blanco, F., Parmisano, A., Torres, D., & Díaz, A. (2012). *ASPIA, un Repositorio Federado como Base Semántica de Wikipedia.* Obtenido de http://41jaiio.sadio.org.ar/sites/default/files/3_ASAI_2012.pdf

Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., & Kontokostas, D. (2013). *DBpedia – A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia.* (K. Janowicz, Ed.) Recuperado el 3 de 2015, de http://svn.aksw.org/papers/2013/SWJ_DBpedia/public.pdf

Mastermagazine. (2014). *Wikidata, la fuente de datos estructurados para Wikipedia*. Recuperado el 2015, de http://www.mastermagazine.info/articulo/wikidata-fuente-datos.php

Moya, R. (7 de 2013). *¿Que son los Sistemas de Recomendación?* Recuperado el 3 de 2015, de http://jarroba.com/que-son-los-sistemas-de-recomendacion/

Organismo Metahumano. (2012). *Wikidata busca traducir su interfaz*. Recuperado el 3 de 2015, de http://metahumano.org/log/tag/wikidata/

Pedraza-Jiménez, R., Codina, L., & Rovira, C. (12 de 2007). *Web semántica y ontologías en el procesamiento de la información documental*. Recuperado el 2 de 2013, de http://eprints.rclis.org/14298/1/webSemanticaOntologias2007.pdf

Peis, E., Herrera-Viedma, E., Hassan, Y., & Herrera, J. C. (10 de 2003). *Análisis de la web semántica: estado actual y requisitos futuros*. Recuperado el 2 de 2013, de http://eprints.rclis.org/11446/1/peis.pdf

Pérez Valdés, D. (6 de 2007). *Web Semántica y sus principales características*. Recuperado el 2013, de Maestros del Web by Platzi: http://www.maestrosdelweb.com/web-semantica-y-sus-principales-caracteristicas/

Resnick, P., & Varian, H. R. (3 de 1997). *Recommender systems.(Special Section: Recommender Systems)(CoverStory).* Recuperado el 3 de 2015, de https://www.ischool.utexas.edu/~i385d/readings/Resnick_Recommender_97.pdf

Ricci, F., Rokach, L., & Shapira, B. (2011). Introduction to Recommender Systems Handbook. En *Recommender Systems Handbook* (págs. 1-35). Springer. Obtenido de http://www.inf.unibz.it/~ricci/papers/intro-rec-sys-handbook.pdf

SemanticWeb.org. (11 de 2011). *SPARQL endpoint*. Recuperado el 3 de 2015, de http://semanticweb.org/wiki/SPARQL_endpoint

Simon, E., Madsen, P., & Adams, C. (8 de 2001). *An Introduction to XML Digital Signatures*. Recuperado el 2 de 2013, de O' REILLY xml.com: http://www.xml.com/pub/a/2001/08/08/xmldsig.html#sigs

Torres, D. (10 de 2014). *Co-evolución entre la Web Social y la Web Semántica.* Obtenido de Tesis de Doctorado: http://hdl.handle.net/10915/41223

Torres, D., Molli, P., Skaf-Molli, H., & Diaz, A. (12 de 2012). *From DBpedia to Wikipedia: Filling the Gap by Discovering Wikipedia Conventions.* Recuperado el 3 de 2015, de 2012 IEEE/WIC/ACM International Conference on Web Intelligence (WI'12) , Macau, China: https://hal.inria.fr/hal-00741160/document

Torres, D., Skaf-Molli, H., Molli, P., & Diaz, A. (5 de 2013). *BlueFinder: RecommendingWikipedia Links Using DBpedia Properties.* Recuperado el 3 de 2015, de Web Science, Paris, France. ACM.: https://hal.inria.fr/hal-00822696/document

Valencia Castillo, E. (8 de 2007). *Recuperación y organización de la información a través de RDF usando SPARQL.* Recuperado el 3 de 2015, de https://ggomez.files.wordpress.com/2008/09/informe-sparql.doc

W3C - Cambridge Semantics. (6 de 2009). *SPARQL By Example a Tutorial*. (L. Feigenbaum, Editor) Recuperado el 3 de 2015, de Query #4: Exploring DBPedia: http://www.w3.org/2009/Talks/0615-qbe/

W3C. (2 de 1999). *Resource Description Framework (RDF) Model and Syntax Specification*. (O. Lassila, & R. R. Swick, Editores) Recuperado el 3 de 2013, de http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/

W3C. (2 de 2004). *OWL Web Ontology Language Overview*. (D. McGuinness, & F. van Harmelen, Editores) Recuperado el 3 de 2013, de http://www.w3.org/TR/owl-features/

W3C. (2 de 2004). *RDF Vocabulary Description Language 1.0: RDF Schema*. (D. Brickley, R. Guha, & B. McBride, Editores) Recuperado el 3 de 2013, de http://www.w3.org/TR/2004/REC-rdf-schema-20040210/

W3C. (1 de 2008). *SPARQL Query Language for RDF*. (E. Prud'hommeaux, & A. Seaborne, Editores) Recuperado el 3 de 2015, de Making Simple Queries (Informative): http://www.w3.org/TR/rdf-sparql-query/#basicpatterns

W3C. (10 de 2009). *OWL 2 Web Ontology Language Document Overview*. Obtenido de http://www.w3.org/TR/2009/REC-owl2-overview-20091027/

W3C. (12 de 2012). *OWL 2 Web Ontology Language Document Overview (Second Edition)*. Obtenido de http://www.w3.org/TR/owl-overview/

W3C. (3 de 2013). *SPARQL 1.1 Protocol*. (L. Feigenbaum, G. T. Williams, K. G. Clark, & E. Torres, Editores) Recuperado el 3 de 2015

W3C. (3 de 2013). *SPARQL 1.1 Query Language*. (S. Harris, & A. Seaborne, Editores) Recuperado el 3 de 2015, de http://www.w3.org/TR/2013/REC-sparql11-query-20130321/

W3C. (2013). *W3C Semantic Web Activity*. Recuperado el 2 de 2013, de http://www.w3.org/2001/sw/

W3C. (9 de 2014). *LinkedData*. Recuperado el 3 de 2015, de http://www.w3.org/wiki/LinkedData

W3C. (12 de 2014). *SweoIG/TaskForces/CommunityProjects/LinkingOpenData*. Recuperado el 3 de 2015, de http://www.w3.org/wiki/SweoIG/TaskForces/CommunityProjects/LinkingOpenData

Wikidata. (2015). *Wikidata:Introducción*. Recuperado el 2015, de https://www.wikidata.org/wiki/Wikidata:Introduction/es

Wikipedia. (2015). *Ayuda:Categoría*. Recuperado el 2015, de http://es.wikipedia.org/wiki/Ayuda:Categor%C3%ADa

Wikipedia. (2015). *Category:Wikipedia guidelines*. Obtenido de http://en.wikipedia.org/wiki/Category:Wikipedia_guidelines

Wikipedia. (2015). *Help:Infobox*. Recuperado el 3 de 2015, de http://en.wikipedia.org/wiki/Help:Infobox

Wikipedia. (2 de 2015). *SPARQL*. Recuperado el 3 de 2015, de http://en.wikipedia.org/wiki/SPARQL

Wikipedia. (2015). *Wikidata - Historia del desarrollo*. Recuperado el 2015, de http://es.wikipedia.org/wiki/Wikidata

Wikipedia. (2015). *Wikipedia:Categorization*. Recuperado el 2015, de http://en.wikipedia.org/wiki/Wikipedia:Categorization

Wikipedia. (4 de 2015). *Wikipedia:Categorization.* Obtenido de http://en.wikipedia.org/wiki/Wikipedia:Categorization

Yu, L. (2011). *A Developers Guide to the Semantic Web* (Primera ed.). Springer.

Zhang, M., & Zhou, Z. (2005). *A k-nearest neighbor based algorithm for multi-label classification.* Beijing China: In Granular Computing, 2005 IEEE International Conference.