

La Web Semántica como plataforma para sistemas de recomendación

Guido Zuccarelli

31 de octubre de 2014

Índice general

1. Introducción	3
1.1. Evaluaciones entrecruzadas	3
1.2. Sistemas de Recomendación	5
1.3. La web semántica	5
1.4. Reviews en la web semántica	6
1.5. Organización	7
2. Caso de Estudio	9
3. Enfoque General	10
3.1. Aplicaciones en la Web Semantica	10
3.2. Principios	10
3.2.1. rdf	10
3.2.2. rdfs	11
3.2.3. sparql	12
3.2.4. semantichtml	12
3.2.5. extractores	13
3.3. Tecnologías	13
3.4. Estrategia propuesta	13
3.5. Selección de vocabularios	14
3.5.1. lov	15
3.5.2. vocabcc	15
3.6. Recolección y extracción de los datos	16
3.6.1. Búsqueda	16
3.7. Evaluación de Calidad de los Datos	17
3.8. Curado de los Datos	18
3.9. Integración de los datos	18
3.10. Publicación del Dataset Curado	18
3.11. Explotación del Dataset	19
4. Selección de Vocabularios	20
4.0.1. Review Ontology	20
4.0.2. hReview	21
4.0.3. dataVocabulary	23

4.0.4. Schema.org	23
5. Recolección de Datos	26
6. Evaluación de Calidad de los Datos	27
7. Curado de los Datos	28
8. Integración de los Datos	29
9. Publicación del Dataset Curado	30
10.Explotación del Dataset	31
11.Conclusiones y trabajo futuro	32
12.Bibliografía	33
A. Publicaciones	35

Capítulo 1

Introducción

1.1. Evaluaciones entrecruzadas

Llamaremos ítem, a un elemento del mundo real que es aprovechado por las personas, dicho ítem puede ser, un objeto, un servicio, una idea, un programa, etc. Además, debería poder ser abstraído a un modelo representado por una computadora, de manera tal, que esa representación describa con precisamente a qué se refiere ante la interpretación del lector.

Para ello, el modelo del ítem contendrá un conjunto de atributos que lo describen. Algunos más importantes que otros.

Por ejemplo, imaginemos que se debe representar una bicicleta mediante un conjunto de pares atributo;valor, tendríamos algo como esto.

Tipo: Bicicleta

Sexo: Unisex

Talle: 51

Cuadro-Material: Aluminio

Cuadro-color: Rojo

Cuadro-Tipo: Ruta

Horquilla-Material: Fibra de carbono

Horquilla-Color: Rojo

Asiento-tipo: Adamo

Asiento-Color: Blanco

La cantidad de atributos que pueden utilizarse para describir el ítem puede ampliarse prácticamente hasta el cansancio.

Se pueden optar también por una representación como ésta

Tipo: Bicicleta

Talle: 51

Marca: Merida

Modelo: Reacto 500

Puede aprovecharse el hecho de que es muy probable que ya existan definiciones del ítem que intentamos describir, por lo que si se utilizan sólo los atrib-



utos que lo identifican, será suficiente para que el lector tenga la interpretación correcta del mismo.

Cada ítem es de alguna manera utilizado por uno o más usuarios. Y cada usuario puede de la misma manera ser representado en la computadora mediante pares de atributo;valor. Y al igual que los ítems algunos atributos servirán para identificar al usuario.

Esta relación de uso entre usuario e ítem, contiene un grado de conformidad entre el primero y el segundo, si el ítem cumplió o no con las expectativas del usuario debería poder modelarse también para ser representado computacionalmente.

Definiremos a una reseña (de aquí en adelante review) representación mediante la computadora, de la medida de conformidad con respecto a dicha relación entre usuario e ítem.

El objetivo de un review, es que un usuario pueda reflejar el sentimiento que le generó utilizar el ítem e informarlo a otros usuarios.

También los reviews dispondrán de un conjunto de atributos de los cuales algunos serán indispensables y otros que enriquecerán no sólo su valor intrínseco, sino también su utilidad para el contexto que será planteado.

Imaginemos un usuario que realiza un review sobre la bicicleta, podría generar el siguiente review:

Puntuación: 4

Título: "Buena opción"

Texto: "Liviana y cómoda, pero un poco rígida para doblar."

Fecha: 15/10/2012

Al terminar de leer esta sección, el lector debe entender a que te referís con reviews (en términos generales), que muchas de ellas son publicadas en la web en redes sociales, en sitios de productos etc, y que sirven para ???. Podés agregar algunos ejemplos e incluso imágenes. No queda claro por que en el titulo de la sección dice "entrecruzadas", ¿es importante o fue solo una elección al azar?. Para que este se conecte con el que sigue, podés dejar picando algo como "muchos han intentado procesar automáticamente las opiniones de los usuarios para ... pero eso es muy difícil porque..."

1.2. Sistemas de Recomendación

Los sistemas de recomendación son herramientas de software que, en base a un conjunto de ítems (películas, libros, productos, hoteles, etc) e información sobre estos, y un conjunto de usuarios, intentan sugerir ítems apropiados a dichos usuarios. Los sistemas de recomendación se han vuelto una de las herramientas más poderosas para múltiples tipos de aplicaciones web, como comercio electrónico o páginas de noticias.

El desarrollo de estas herramientas, involucra conocimiento en múltiples áreas, como inteligencia artificial, minería de datos, estadística, etc.

Los sistemas de recomendación poseen dos enfoques, el de filtrado colaborativo, y el basado en contenido.

Los sistemas de recomendación basados en contenido utilizan un conjunto de informaciones y descripciones de los ítems previamente valorados por un usuario para poder construir un perfil del mismo de manera tal de poder determinar, cuales son sus intereses.

Una vez construido el perfil, pueden procesarse las características de distintos ítems que potencialmente pueden ser recomendados a dicho usuario para determinar si alguno de ellos va acorde al perfil.

Los sistemas de recomendación de filtrado colaborativo utilizan la información histórica de cada usuario para intentar encontrar y generar grupos de usuarios con gustos similares. Para lograrlo se compara cada usuario con otro observando qué ítems evaluaron y qué puntajes fueron otorgados.

De esta forma, si se requiere predecir el interés de un usuario en un ítem, se podrá buscar en dicho grupo de usuarios con gustos similares e inspeccionar aquellos usuarios que hayan realizado un review sobre el ítem.

Los sistemas de recomendación de filtrado colaborativo tienen una eficacia superior a los basados en contenido, pero cuentan con una gran desventaja, no permiten predecir intereses para aquellos ítems que aún no poseen evaluaciones, o también para aquellos usuarios que no han evaluado ningún ítem. A esto se lo llama “arranque en frío”.

1.3. La web semántica

En sus comienzos en los 90, la web podía verse como un conjunto de sitios web que ofrecían una colección de documentos web interconectados mediante la hipermedia, con el objetivo de comunicar información a los usuarios. El contenido de esos documentos sólo era generado por el mismo creador y publicador del documento, y los usuarios se limitaban a consumirlo. Por otro lado, era a su vez estático, es decir, se publicaba en la misma forma que se almacenaba y no cambiaba.

Hacia fines de los 90 los sitios web comenzaron a implementar una serie de herramientas (que si bien ya se encontraban disponibles anteriormente no se utilizaban por un problema de performance) que permitieron a los usuarios finalmente participar de la producción del contenido web. Lo que produjo notorios

cambios en cuanto a la cantidad de información disponible y proveyó diferentes formas de uso de la web (blogs, redes sociales, canales rss, etc). Más adelante ante la apreciación del pasaje de web estática a una web dinámica, se acuñó a esa actual web como web 2.0 y retrónimamente 1.0 a la anterior.

Ese cambio provocó un cambio en el tamaño de la web, que se volvió inmensamente grande, y llevó a la necesidad de implementar tecnologías que ayuden al aprovechamiento de esa cantidad de información. Se comenzó entonces a utilizar una serie de frameworks y estándares que permitieron enriquecer (mediante metadatos semánticos y ontológicos dentro de los estándares de la W3C) los datos contenidos en los documentos de manera tal que éstos puedan ser consumidos, interpretados y utilizados directamente no sólo por las personas, sino también por las computadoras. Esto generó que los datos también puedan ser relacionados entre sí, de la misma manera que los documentos son interconectados formando una web de documentos, los datos interconectados forman una web de datos paralela. Todo este conjunto de actividades frameworks y herramientas forman la “Web semántica” que es el puntapié inicial para una web mucho más interoperable, lo que permite facilidades para el uso de la web por parte de las aplicaciones y da lugar a otro paso en la evolución de la web, la web 3.0. La Web Semántica promete facilitar el desarrollo de la web social y la inteligencia colectiva, a través de mecanismos de clasificación, relación y descripción del contenido de la información publicada mejorando su interoperabilidad. Promete también mejorar la recolección, agregación e integración de datos específicos mediante el uso de agentes de búsqueda automáticos que aprovechan las ventajas.

Al terminar de leer esta sección, el lector debe entender que es la web semántica y a que apunta. Si dejaste picando el tema de automatización en el párrafo anterior, acá se va a imaginar porque hablaste de ws. Habla muy brevemente de triplets y menciónala RDF. También mencioná linked open data para darle una idea de que la web semántica es una web de datos interconectada. Podés tomar lo que ya escribiste en la propuesta. Ejemplifica con dbpedia o algo así... En un capítulo mas adelante vas a entrar en detalle en web semántica, rdf, etc. Acá contás solo lo suficiente para que se entienda lo que vas a proponer en la próxima sección

1.4. Reviews en la web semántica

La web 2.0 dio la posibilidad a los usuarios consumidores de la web de generar y publicar contenido en la misma, lo cual cumple con los requerimientos de una plataforma para reviews. Proporciona un entorno para crearlos y publicarlos, lo cual en el caso es una tarea muy sencilla. Pero la dificultad de encontrarlos y explotarlos parece ser inversamente proporcional a generarlos. Dado que a mayor facilidad de publicar contenido en la web, mayor se vuelve la inmensidad de la misma y mayor se vuelve la dificultad de encontrar algo dentro de ella.

Veamos dos ejemplos que requieren encontrar reviews en la web:

Imagínense que son ingenieros de Mérida y lanzaron al mercado la bicicleta Reacto 500. Como buenos ingenieros necesitan conocer qué opinan sus clientes, por lo que buscarán reviews en algún motor de búsqueda y luego leerán uno por uno cada review con el objetivo de resumir las opiniones. Humanamente esta tarea demandará mucho tiempo y con un límite corto de cantidad de reviews. Ahora peor aún, imagínense que necesitan saber que opina la gente de determinada región (por ejemplo el norte de europa), la tarea de buscar los reviews necesarios se volvería aún más complicada. Con la posibilidad de contar con una web en la cual los datos son interpretados por las aplicaciones de software y estos a su vez están interconectados, podría crearse una aplicación que pueda automáticamente buscar, clasificar y procesar los reviews para generar automáticamente el resumen.

Ahora bien, si en lugar de requerir reviews de un ítem en particular, se necesitan reviews para una aplicación que recomiende ítems a usuarios, ya no sería una tarea realizable humanamente. Para ello haría falta una aplicación que haga crawling en la web y de alguna manera identifique reviews y a su vez identifique el ítem al cual el review hace referencia. Parece algo muy difícil de lograr, salvo que se trabaje sobre sitios conocidos con documentos estructuralmente dominados los cuales se pueda recorrer el DOM automáticamente y acceder a los reviews. De nuevo, la Web Semántica promueve solucionar este problema, con el uso de metadatos que dan información sobre los datos, haciendo que una aplicación pueda fácilmente identificar reviews y navegar por la hipermedia de los datos para conseguir información sobre el ítem y usuario referenciados.

Acá es donde presentas el problema a resolver. Ya contaste que son los reviews y por que alguien querría integrarlos. Ya contaste que es la web semántica y linked open data. Ahora tenés que contar que ha habido iniciativas para darle semántica a los reviews y que existen datos; ahora hay una posibilidad de aprovechar esa información. Y ahí decís algo como lo que dijiste al final de la propuesta: *El objetivo principal de esta tesis es evaluar la viabilidad, y entender los desafíos de la utilización de la información contenida en la web semántica en la construcción de sistemas de recomendación. Para eso, y con foco en el caso particular de opiniones de usuarios sobre distintos tipos de recursos se buscará: capturar, extraer, validar calidad, curar, integrar, publicar y explotar los datos disponibles.*

1.5. Organización

ya saben lo que vas a contar; acá les das una idea de como te vas a organizar para contarlo - puede ser algo como lo que está a continuación. Para el caso de estudio podés tomar el texto que escribimos en el artículo

El capítulo 2 presenta una aplicación de ejemplo que muestra claramente el problema que se quiere resolver y que servirá como referencia a lo largo de esta tesis. El capítulo 3 introduce los conceptos de Web Semantica, sus principios y tecnologías y presenta la estrategia de solución en términos generales. Los capítulos 4 a 10 discuten en detalle cada uno de pasos de la estrategia elegida. Finalmente, el capítulo 11 presenta los resultados observados, saca conclusiones al respecto, y plantea trabajo futuro. El anexo 12 presenta una publicación que fue resultado del trabajo efectuado en este trabajo de tesis.

Capítulo 2

Caso de Estudio

Capítulo 3

Enfoque General

Lo que vamos a construir es un caso de aplicación de la web semantica....

3.1. Aplicaciones en la Web Semantica

3.2. Principios

Dar historia (muy breve), definiciones (RDF, tripletas, owl, inferencia, ..)
Escribir a medida que lo necesitas-.

3.2.1. rdf

Resource Description Framework (RDF) Es una familia de especificaciones de WC3 diseñada para modelar datos intercambiables en la web. Consta de expresiones hechas por tripletas. Cada triplete es considerada una sentencia para este lenguaje. Una triplete es un conjunto de un valor de cada uno de los siguientes tipos:

Recurso - Todo aquello descrito por RDF se lo denomina recurso, este último podría ser por ejemplo una página web entera (por ejemplo un documento HTML "http://www.w3.org/Overview.html") o una parte de ella. Un recurso puede ser también un objeto que no se puede acceder directamente a través de la Web; por ejemplo un libro impreso. Los recursos son siempre nombrados por las URIs más identificadores de anclaje opcionales (ver [URI]). Cualquier cosa puede tener un URI; la extensibilidad de URIs permite la introducción de identificadores para cualquier entidad imaginable.

Propiedad - Una propiedad es un aspecto específico, característica, atributo o relación se utiliza para describir un recurso. Cada propiedad tiene un significado específico, define sus valores permitidos, los tipos de recursos que se puede describir, y su relación con otras propiedades.

Objeto - puede ser otro recurso o puede ser un literal; es decir, un recurso (especificado por un URI) o una cadena simple o de otro tipo de datos primitivo

definido por XML. En términos RDF, un literal puede tener contenido que es el formato XML, pero no se evalúa aún más por el procesador de RDF.

Esta estructura forma una vinculación dirigida de un gráfico etiquetado, donde las aristas representan el enlace entre dos recursos (propiedad), representados por los nodos del grafo.

Su representación puede ser asociada tanto a un modelo entidad-relación como a un diagrama de clases de objetos. Las Propiedades RDF pueden ser considerados como atributos de los recursos y en este sentido corresponden a pares atributo-valor tradicionales. Las Propiedades RDF también representan las relaciones entre los recursos y un modelo RDF, por tanto, pueden parecerse a los de un diagrama entidad-relación. En la terminología de diseño orientado a objetos, los recursos corresponden a los objetos y propiedades corresponden a las variables de instancia.

3.2.2. rdfs

RDF Schema (RDFS)

Es una extensión semántica de RDF, que provee elementos básicos para la descripción de ontologías (también llamadas vocabularios RDF) con el objetivo de estructurar los recursos RDF. Proporciona la manera de definir udefinir las clases y las propiedades específicas de la aplicación en RDF. RDFS no proporciona las clases sino que proporciona el marco necesario para poder definir las.

Clases en RDFS se parecen mucho a las clases en lenguajes orientados a objetos. Esto permite que los recursos se definan como instancias de clases y subclases de las clases.

Elementos de RDFS

Básicos

`rdfs:Class` permite declarar recursos como clases para otros recursos.

Atravez de la propiedad `rdf:type`, se puede asignar un tipo al recurso, un recurso de tipo `rdfs:Class` puede entonces ser tipo de otro recurso. Por ejemplo, podemos definir el recurso `Auto` como un `rdfs:Class`, y a su vez podemos definir el recurso `Volkswagen Gol` con la propiedad `rdf:type` y siendo el objeto de esa propiedad el recurso `Auto`, entonces el tipo de `Volkswagen Gol` será `auto`.

La defición de `rdfs:Class` es recursiva: `rdfs:Class` es la `rdfs:Class` de cualquier `rdfs:Class`.

`rdfs:Resource` es la clase a la que pertenecen todos los recursos.

`rdfs:Literal` es la clase de todos los valores literales, cadenas y enteros.

`rdfs:Datatype` es la clase que abarca los tipos de datos definidos en el modelo RDF.

`rdfs:Property` es la clase que abarca las propiedades.

Definen relaciones

`rdfs:subClassOf` es una instancia de `rdf:Property` (osea una propiedad) que permite definir jerarquías. Relaciona una clase con sus superclases.

`rdfs:subPropertyOf` es una instancia de `rdf:Property` que permite definir jerarquías de propiedades.

Restricciones de Propiedades

`rdfs:domain` es una instancia de `rdf:Property` que especifica el dominio de una propiedad `P`. Esto es, la clase de los recursos que aparecen como sujetos en las tripletas donde `P` es predicado.

`rdfs:range` es una instancia de `rdf:Property` que especifica el rango de una propiedad `P`. Esto es, la clase de los recursos que aparecen como objetos en las tripletas donde `P` es predicado.

3.2.3. sparql

SPARQL

Se trata de un lenguaje estandarizado de consulta de grafos RDF. SPARQL se puede utilizar para expresar consultas que permiten interrogar diversas fuentes de datos (si es que los datos fueron almacenados de forma nativa como RDF o fueron definidos mediante vistas a RDF a travez de algún middleware). Tiene capacidades para la consulta de los patrones obligatorios y opciones de grafo, junto con sus conjunciones y disyunciones. SPARQL también soporta la ampliación o restricción del ámbito de las consultas indicando los grafos sobre los que opera (grafos con nombre). Los resultados de las consultas SPARQL pueden ser conjuntos de resultados o grafos RDF.

Términos de conjuntos de colecciones de datos semánticos

Model: Colección de sentencias.

DataSource: Colección de Modelos. Siendo uno de ellos el modelo por defecto y el resto modelos con nombre. El DataSource es de lectura y escritura, por lo que se pueden agregar tripletas.

Dataset: Lo mismo que el DataSource pero estático, por lo que es de solo lectura.

Graph: Colección de tripletas. Cualquier modelo puede ser transformado a un grafo, y así tener una representación más cercana de RDF.

DatasetGraph: Un contenedor de Grafos, similar al DataSource (También de lectura y escritura) que provee una estructura para un grafo por defecto y grafos con nombre.

3.2.4. semantichtml

HTML semántico

El uso y participación en la Web Semántica a travez de documentos RDF no es atractivo para la mayoría, porque no parece proveer un beneficio directo, ya que esto implica, aprender un lenguaje nuevo para generar documentos que sólo le será de utilidad a usuarios con un grado de conocimiento muy alto.

Para resolver este problema, existen etiquetas de marcado en HTML que le proporcionan información semántica al documento, que copmo se verá más adelante puede ser extraída para generar documentos RDF. Existen distintos tipos:

Microdatos: Son un grupo de pares nombre-valor anidados dentro del documento, en paralelo con el contenido existente. Dichos grupo son también llama-

dos ítems y cada par nombre-valor es la propiedad. Los atributos más utilizados son:

Para crear un ítem, se utiliza el atributo de HTML “itemscope”.

Para crear una propiedad se utiliza el atributo “itemprop”.

Luego para definir los valores, se utilizan según sea una URL o un String los siguientes atributos: “a” con “href”, “src” o “img” para definir un valor URL “value” para un String.

También se le puede establecer el tipo a un ítem mediante el atributo “item-type”

Microformatos: Nacen con la intención acercar más personas al uso de la web semántica a través de una forma sencilla de utilizar una serie de atributos de las etiquetas de XHTML:

class: especifica el nombre de la clase

rel: Se utiliza en los enlaces para indicar relaciones en los documentos.

rev: Igual al rev pero en sentido inverso.

Con el uso del atributo class se puede entonces especificar un microformato a utilizar. Existen muchos tipos como por ejemplo: hCard (para describir personas), hreview (para describir reviews), geo (para describir posiciones), etc. Cada uno de ellos con distintas propiedades.

RDF-a: Permite directamente embeber tripletas (sujeto-predicado-objeto) utilizando namespaces de XML. Modela la información en forma de grafo, a diferencia de microdatos y microformatos que lo hacen en forma de árbol, lo que hace que el mapeo a RDF sea claro, ya que en los otros casos puede ser problemático. También tiene la opción de darle tipo a los literales. Como contrapartida tiene la desventaja de ser mucho más complejo de utilizar.

3.2.5. extractores

Extratores

Son herramientas que permiten extraer tripletas de los lenguajes

3.3. Tecnologías

Triplestores, frameworks, extractores, crawlers, motores de búsqueda semánticos...

3.4. Estrategia propuesta

Con el fin de crear una aplicación que satisfaga los requerimientos mencionados anteriormente, se debe encontrar un procedimiento que incluya desde obtener los datos relevantes de la web hasta llevarlos a un estado que permita una explotación satisfactoria.

El procedimiento debería incluir los siguientes pasos

Selección de vocabularios Tanto la recolección como la publicación de datos en la web semántica involucra la elección del o de los vocabulario/s que mejor modelen el dominio de problema, con la excepción que para su publicación existe la posibilidad de desarrollar uno propio que se ajuste correctamente en caso de que no se encuentre uno existente.

Recolección y extracción de los datos ?

Evaluación de Calidad de los Datos ???

Curado de los Datos ???

Integración de los datos ??

Publicación del Dataset Curado ??

Explotación del Dataset ??

acá sería bueno incluir un diagrama que muestre el pipeline. En la lista de arriba con un párrafo se explica cada paso. Luego, en las secciones que sigue se analiza mas cada paso, pero se lo plantea como un problema.. se analizan los retos; entonces los capítulos 4 a 9, explican como los resolviste. Alternativamente, se puede hablar menos acá, solo lo suficiente para que se entienda la estrategia general, y luego en los capítulos 4 a 9 se analiza el problema en detalle y se propone la solución. Con esta ultima forma, todo lo que se refiere a los vocabularios quedaría en el capitulo 4 y no acá. Puede ser mejor.

3.5. Selección de vocabularios

Seleccionar un vocabulario implica analizar varios aspectos del mismo, no sólo su definición e implementación, sino también el uso práctico dado por sus usuarios. En primer lugar se debe comprobar que los nombres de las propiedades que posee sean correctamente auto-explicativas. Supongamos por ejemplo que existe un ítem con un rating agregado modelado por una ontología que posee las siguientes propiedades:

1. minrating
2. ratingValue
3. countRating

Las dos últimas propiedades resultan fácilmente identificables, ratingValue se trata del promedio de puntaje, y ratingCount la cantidad de puntajes que le fueron otorgados, pero la propiedad minRating podría generar distintas formas de interpretación, alguien podría suponer que se trata del valor mínimo que fue

adquirido por un usuario, o el valor mínimo que un usuario puede otorgar. Y muchas veces la documentación de la ontología (si es que existe) no es suficiente.

Luego se deberá analizar si existen las propiedades para cubrir las necesidades mínimas de los casos de uso.

Y por último se debería intentar buscar ejemplos reales que muestren el uso que le dieron los usuarios a la ontología, para determinar qué propiedades están incorrectamente interpretadas o también para los casos donde las propiedades que se encuentren en desuso.

Con estas precauciones en mente se puede emprender la búsqueda, que podría tener como comienzo búsquedas en search engines. Existen dos buscadores específicos para esta tarea:

3.5.1. lov

Linked Open Vocabulary (LOV)

Proporciona una plataforma técnica de búsqueda y evaluación de calidad sobre un dataset extraído de linked data cloud que contiene descripciones de vocabularios RDFS y ontologías OWL. Esas descripciones están en forma de metadatos y pueden ser generados tanto por los autores de los vocabularios como por curadores de LOV. Posee además de la búsqueda las funciones de estadística o sugerencia.

Actualmente el dataset está integrado por 475 namespaces distintos que contienen una media de 10 clases y 20 propiedades, siendo schema.org el más grande de ellos.

3.5.2. vocabcc

Vocab.cc

Vocabcc es un proyecto opensource que permite a los desarrolladores de RDF realizar búsquedas de vocabularios de Linked Data.

Para facilitar la decisión de seleccionar un vocabulario deseado, proporciona además información estadística de cada uno sobre el dataset Billion Triple Challenge (BTCD). Esta información incluye el número de apariciones globales de la URI dada en el BTCD, así como el número de documentos dentro de la BTCD, que contiene el URI dado. Estos números permiten una clasificación de propiedades y clases, respectivamente, con respecto a su uso. También se proporciona información acerca de la posición de una URI dada en estos rankings.

Los desarrolladores pueden buscar las URIs con queries arbitrarias o búsquedas de URIs específicos (prefijos comunes se resuelven automáticamente con datos de prefix.cc).

Para permitir una fácil integración de la funcionalidad vocab.cc, toda la información está disponible como RDF y se puede acceder como Servicio Vinculado.

3.6. Recolección y extracción de los datos

Como se mencionó antes, la web contiene grandes cantidades de documentos publicados con información semántica. Pero la tarea de encontrarlos, con el agregado de que sólo una pequeña porción de ellos será relevante para los requerimientos no es trivial en lo absoluto debido a la inmensidad del universo en el que se encuentran. La forma de llevar a cabo este objetivo está atada al hardware disponible, tanto para almacenar los datos, como para el tiempo que va a emplear la ejecución de esta tarea.

Dado que las bases de datos semánticas sólo almacenan información en forma de tripletas o cuádrupletas, los documentos encontrados deberán someterse a un proceso de extracción que seleccione las sentencias HTML y las convierta a alguno de los lenguajes que soportan tripletas o cuádrupletas. Para esto existen múltiples herramientas.

Una vez encontrados, descargados y transformados los documentos HTML a documentos semánticos puede construirse la base de datos semántica con la información recolectada.

Estos son los cuatro pasos necesarios para lograr tener el dataset semántico con el cual se puede empezar a trabajar. Cada uno de ellos posee distintas alternativas para su realización, algunas se describirán a continuación

3.6.1. Búsqueda

OJO: ESTO ESTABA EN EL ARCHIVO `busqueda.tex` no en el opciones-`Busqueda`

La forma de ejecución de esta tarea dependerá de algunos aspectos:

Recursos de hardware disponibles

Cantidad y calidad de la información requerida

El grado de atemporalidad mínima tolerable en los datos

El primer paso para realizar la recolección es intentar responder la siguiente pregunta: ¿Dónde encuentro la información?

Una vez seleccionados los vocabularios, se necesitará obtener fuentes de datos que contengan sus datos publicados en esos vocabularios.

Para lograrlo se podrá utilizar como punto de partida:

Sitios indexadores: Son algunos sitios que disponen de un dataset muy grande procesado con documentos indexados, que ofrecen consultar dicho dataset mediante servicios web. Generalmente proveen una API donde se pueden consultar los datos mediante distintos grados de flexibilidad.

Síndice, LOD cloud cache y UriBurner son algunos ejemplos de estos sitios. Se puede utilizar entoces, los servicios web a fin de obtener una lista de URL que se cree que tendrán la información necesaria.

Síndice: Es una herramienta creada en conjunto por la Universidad de Deri, Fondazione Bruno Kessler y Openlink Software que proporciona múltiples tipos de API ofreciendo acceso a su dataset de la web de datos. Este dataset contiene información recolectada de la web en múltiples formatos de la web semántica y

puede ser accedido a travez de un search engine, una API restful o un SPARQL endpoint. En la actualidad posee indexados 708.26 millones de documentos.

Link Open Data Cloud Cache (LOD): Al igual que sindece proporciona acceso a un dataset recolectado de la web de datos pero de una manera mucho más acotada. Se dispone de un text search (funciona como un search engine) o de un SPARQL Endpoint bastante restringido. Sólo se puede acceder al dataset entero mediante federated queries, en caso de querer buscar sobre todo el dataset, el endpoint limita la consulta sólo a una parte del mismo. Tambien posee muchas restricciones respecto a los time outs, por lo que las consultas no pueden ser muy flexibles. Para poder disponer de la funcionalidad completa del endpoint y así poder aprovechar tanto el dataset como las consultas SPARQL deberá comprarse una licencia.

Sitios autoritativos: Son sitios conocidos que generan información relevante y la publican en las ontologías y vocabularios seleccionados. Ejemplos aplicados al caso de estudio podrían ser IMDB (que publica reviews de películas bajo el vocabulario schema), o Rottentomatoes (que hace lo mismo pero no solo con películas). Se podrán utilizar entonces estos sitios como punto base para un posible crawling.

Catálogos de enpoints: Son catálogos provistos por algunos sitios que mantienen una lista actualizada de SPARQL endpoints junto con el estado de disponibilidad en el que se encuentran. Por ejemplo los sitios <http://labs.mondeca.com/sparqlendpointsstatus/> y <http://www.w3.org/wiki/SparqlEndpoints> que este último además provee detalles sobre la información de los mismos.

Consultar estos sitios entonces generará una lista de endpoints SPARQL que se podrá utilizar para consultar la información necesaria.

Volcados de datos: Son datasets muy grandes que fueron el resultado de un web crawling, los cuales están disponibles para su descarga y se pueden utilizar para procesarlos y obtener los datos requeridos. El más importante es <http://challenge.semanticweb.org/2014/>.

3.7. Evaluación de Calidad de los Datos

Como sabemos los reviews son creados por usuarios que en su mayoría no están familiarizados con el desarrollo de una aplicación, esto causa que una parte muy significativa del contenido publicado no esté de la forma adecuada para ser procesado. La falta de calidad en el contenido publicado puede deberse tanto a errores por parte del usuario como por parte del publicador.

El publicador deberá asegurarse que los datos respeten rigurosamente las ontologías en las cuales se publican.

El sitio web en el cual el usuario se encuentre realizando el review deberá guiarlo en todo lo posible para lograr que la evaluación quede en un formato adecuado.

Aunque también existen muchos problemas que no dependen del sitio web, generalmente errores semánticos de calidad, donde lo redactado esta hecho de forma inconsistente o insuficiente.

Este paso entonces tiene por objetivo encontrar todos los problemas de calidad que pueda haber en el dataset que acaba de ser descargado y extraído y que generen inconvenientes para una posterior integración/explotación.

Por lo general se utilizará la base de datos mediante consultas SPARQL estratégicamente generadas para lograr sacar estadísticas y detectar los problemas.

Existen frameworks para realizar estas operaciones.

3.8. Curado de los Datos

En el paso anterior se describieron los problemas en la calidad del dataset que tendrán un impacto en la aplicación resultante, o que, podrían limitar o imposibilitar la realización de la misma. Dichos problemas pueden ser muchos, y algunos muy difíciles de solucionar, será parte entonces del proceso, reconocer aquellos que su resolución sea viable y además encontrar la forma de implementarla.

Cabe destacar que el proceso de evaluar el dataset en búsqueda de problemas y implementar soluciones es iterativo, debido a que una mejora puede conllevar a la detección de nuevos problemas.

3.9. Integración de los datos

Esto va acá o es parte de Explotación?

Los datos recolectados fueron generados por distintos usuarios, en múltiples sitios y bajo distintas ontologías y estándares. Estos aún curados, necesitan un último paso para poder realizar una explotación, y es la integración.

Este gran y dificultoso proceso abarca cualquier operación que intente dar una visión más unificada de la información. Este proceso puede tener distintos niveles y aspectos a integrar, como podría ser por ejemplo, unificar las ontologías de los reviews, o bien en una nueva ontología de review, o bien en una ya existente, para lograr tener información semánticamente más parecida. Ya que este proceso puede ser muy dificultoso, habrá que ver en base a los requerimientos, qué aspectos de los datos se integrarán.

Una vez realizada la integración mínima necesaria para los requerimientos, el dataset ya estará listo para su explotación.

3.10. Publicación del Dataset Curado

???

3.11. Explotación del Dataset

Acá es donde efectivamente se discute que implica cubrir los requerimientos de la aplicación . Puede ser que acá también te refieras a como la forma en la que querés explotar los datos impacta en las fases anteriores. Incluso, habría que pensarlo, en este capítulo tal vez conviene primero hablar de explotación y luego de las otras fases - porque explotación determina mucho que es lo que vas a mirar en las fases anteriores, ¿no?

Capítulo 4

Selección de Vocabularios

En la actualidad existen cuatro vocabularios que cumplen los requerimientos mínimos para modelar el dominio de problema planteado.

4.0.1. Review Ontology

RDF Review Vocabulary

También conocido como Review Ontology, es una de las ontologías más antiguas de review, que fue pensada para uso del lenguaje

RDF y está definido bajo el namespace <http://purl.org/stuff/rev#>.

Fue utilizado para la construcción Revyu, y sirvió como guía para otras ontologías.

Consta de tres clases y trece propiedades.

Clases

Comment: Un comentario sobre el review.

Feedback: Expresa la utilidad del review.

Review: El review mismo.

Propiedades

commenter: Especifica el usuario que realizó el comentario del review. Tiene dominio Feedback o Comment y rango foaf:Agent.

Actualmente se encuentra en desuso.

hasReview: Enlaza el ítem evaluado con el review. Su dominio es rdfs:Resource (siendo ésta la clase del ítem evaluado) y su rango es Review. Es una de las propiedades principales.

hasComment: Idem anterior pero con el comentario en lugar del review.

Se encuentra actualmente en desuso.

hasFeedback: Idem anterior pero con feedback en lugar de comentario.

Se encuentra actualmente en desuso.

maxRating: Establece el puntaje máximo que es posible otorgar por un usuario a través de la propiedad rating. Tiene como dominio Review y como rango Literal siendo este último un número positivo. Su ausencia en un review

asume su valor por defecto (5). Por lo que si bien no es indispensable que esté, se deberá respetar la convención a la hora de generar el rating.

minRating: De la misma forma que el anterior, sólo que establece el puntaje mínimo y su valor por defecto es (1).

positiveVotes: Se refiere a la cantidad de votos positivos que tuvo el review, otorgados por usuarios que lo leyeron y lo encontraron útil. Su dominio es Review y su rango es Literal siendo este último un número positivo.

Se encuentra actualmente en desuso.

rating: Una de las propiedades principales, indica el valor numérico otorgado por el creador del review sobre el ítem evaluado. Su dominio es Review y su rango es Literal siendo este último un número entre los valores de minRating y maxRating.

reviewer: Especifica el usuario que realizó el review. Tiene dominio Review y rango foaf:Person.

text: Otra de las propiedades principales, define el texto que describe el sentimiento del usuario hacia el ítem. Tiene como dominio Review y como rango Literal.

totalVotes: Exactamente igual a positiveVotes.

title: El título del review . Tiene dominio Review y rango Literal. Subclase de dc:title

No tiene demasiada utilidad para el caso de estudio y además se encuentra en desuso.

type: Enuncia el tipo de ítem que clasifica taxonómicamente al ítem evaluado. Su dominio es rdfs:Resource (siendo ésta la clase del ítem evaluado) y su rango no se encuentra especificado.

Es una propiedad muy útil pero actualmente se encuentra en desuso.

date: Si bien no está definida dentro del vocabulario, es correcto utilizar <http://purl.org/dc/terms/date>, implica la fecha en la que se realizó el review.

4.0.2. hReview

Como se mencionó anteriormente, está establecido por convención, que los microformatos son una forma de publicar información en la web semántica, pero al no disponer de namespaces, no pueden ser representados por ninguna ontología o ningún otro lenguaje de la misma.

Al no poder utilizar ontologías que modelen reviews surgió la necesidad de crear un estándar específico para embeberlos dentro de HTML utilizando microformatos.

Dicho estándar se encuentra actualmente en la versión 0.4 y propone el uso de 10 propiedades, que se supone, deberían ser suficientes para cubrir todas las necesidades a la hora de generar un review.

summary (opcional): Puede ser el título o nombre del review, o es posible también hacer una pequeña sinopsis del mismo. Se encuentra en desuso.

type (opcional): Representa el tipo de ítem evaluado, pero se encuentra acotado a alguno de estos `product` — `business` — `event` — `person` — `place` — `website` — `url` .

También está en desuso.

item: Esta propiedad enuncia toda la información que se crea necesaria para identificar al ítem, mínimamente los atributos nombre, url y foto.

Para lograr bajo una propiedad cubrir todos los atributos, el rango de la misma debería ser un hCard, que luego contendrá las propiedades mínimas necesarias: fn, url, photo y cualquier otra que se quiera adicional.

reviewer (opcional): Al igual que ítem, indica todo lo necesario para identificar a la persona autora del review, para lo cual también deberá representarse con un hCard.

dtreviewed (opcional): Se refiere a la fecha en la que fue creado el review.

rating: El valor numérico con el cual el usuario expresa su satisfacción con el ítem, y está formado por un entero con un solo decimal de precisión, que se encuentra dentro del rango 1.0 a 5.0. Dicho rango puede ser alterado con la presencia de las propiedades worst y best que restringen el valor mínimo y máximo respectivamente.

descripción (opcional): Establece el valor textual con el cual el usuario expresa su satisfacción con el ítem, creando una sinopsis detallada del mismo.

tags (opcional): Una etiqueta intenta establecer una idea acerca de qué se trata el contenido en una sola palabra para una rápida identificación o para mejorar las búsquedas.

permalink (opcional): Genera una URI que identificará al review creándole una especie de ID, que será útil para los casos donde se podría repetir la publicación del mismo.

license (opcional): Expresa la licencia del review.

El indicador “(opcional)” se refiere a si es indispensable para conformar un hReview o no, de manera tal que si no se encuentra una propiedad que no está marcada como opcional no podrá ser considerado un hReview.

Cabe destacar que muchas de las propiedades opcionales, podrían ser necesarias para el caso de estudio.

El problema con este vocabulario surge a la hora de trabajar con la información obtenida, que no puede ser representada por ningún otro lenguaje de la web semántica, por lo que se vio la necesidad de mapear las propiedades de hReview, a otro vocabulario que sí pueda.

Esto llevó a que en la web donde definen hReview lo consideren compatible con RDF Review Vocabulary, por lo que en Noviembre de 2007 se creó una herramienta que transforma de uno al otro hreview2rdfxml.xsl, pero existe un problema con el rango de algunas propiedades, por ejemplo reviewer (propiedad homónima en ambos vocabularios, pero una con rango hCard y otra con rango foaf:Person).

En general si bien este vocabulario bien utilizado puede ser efectivo, resulta poco flexible y compliado de manejar por parte de quien quiera explotarlos, el motivo por el cual se ha vuelto muy popular es su facilidad para generarlos, dado que microformatos es un lenguaje muy sencillo y cualquier persona con un relativamente mínimo conocimiento de HTML puede generar sin problemas un hReview, teniendo además herramientas online como opción, que generan el código a travez de un formulario. Como es el caso de

<http://microformats.org/code/hreview/creator>.

4.0.3. dataVocabulary

RDF Data Vocabulary

En Mayo de 2009 Google anuncia la introducción de los llamados “Google Rich Snippets”, estos fragmentos enriquecidos son una convención de etiquetas (con soporte para RDFa-lite y microdata) que permitían agregar información útil a los SERP del buscador de Google. De manera tal que los datos que contenían estos fragmentos, recibían un tratamiento especial.

Sin embargo en el anuncio, Google revela que el soporte se limitó al uso de las clases y propiedades del vocabulario definido en una página notoriamente improvisada llamada <http://rdf.datavocabulary.org/>.

En ella se establecían modelos de clases para varios tipos de ítems, tales como Persona, Organización o Producto y también para los Reviews.

La clase Review, quedó definida bajo el namespace <http://data-vocabulary.org/Review> e incluía las siguientes propiedades:

itemreviewed: Enlace al ítem que está siendo evaluado.

rating: El valor numérico con el cual el usuario expresa su satisfacción con el ítem, tiene como rango un valor numérico bajo la clase xsd:string o Rating, y los valores posibles se encuentran en escala de 1 a 5, pudiendo la misma ser alterada con la presencia de las propiedades worst y best que restringen el valor mínimo y máximo respectivamente.

reviewer: El autor del review, su rango es dvocab:Person o xsd:string.

dtreviewed: La fecha en la que se realizó el review, no contiene un rango específico pero aclara que debe respetar el formato ISO para las fechas.

description: El cuerpo del review que representa el valor textual de satisfacción del usuario con el ítem.

summary: Un resumen corto del review.

Se puede notar la excesiva similitud de este vocabulario con hReview, queda claro que no hubo una intención de innovar algo, sino de representar el hReview en microdatos, probablemente por el apuro en la que data vocabulary fue creado.

Vale aclarar que limitar las clases y propiedades posibles en RDFa es básicamente hacerle perder el sentido al lenguaje (la descentralización de los vocabularios sobre los términos) haciendo que el lenguaje se utilice como si fuese microformatos, pero perdiendo su valor más improtante (la simplicidad) de manera tal que tomó la inflexibilidad de microformatos y la complejidad de RDFa.

Más adelante los Google Rich Snippets incluyeron también soporte para microformatos (lo que incluía hReview).

4.0.4. Schema.org

Schema.org

Luego de los fragmentos enriquecidos y el improvisado vocabulario data-vocabulary creado para soportar los fragmentos, Google en conjunto con Bing y Yahoo crearon en el 2011 Schema.org, con el objetivo de obtener un vocabulario

más completo y organizado para la implementación de los snippets con RDFa y microdata.

Su lanzamiento provocó la inmediata obsoletización de data-vocabulary cuyas clases fueron todas reemplazadas por equivalentes dentro de la nueva ontología:

- <http://www.data-vocabulary.org/Address> -¿<http://schema.org/PostalAddress>
- <http://www.data-vocabulary.org/Geo> -¿<http://schema.org/GeoCoordinates>
- <http://www.data-vocabulary.org/Organization> -¿<http://schema.org/Organization>
- <http://www.data-vocabulary.org/Person> -¿<http://schema.org/Person>
- <http://www.data-vocabulary.org/Event> -¿<http://schema.org/Event>
- <http://www.data-vocabulary.org/Product> -¿<http://schema.org/Product>
- <http://www.data-vocabulary.org/Review> -¿<http://schema.org/Review>
- <http://www.data-vocabulary.org/Offer> -¿<http://schema.org/Offer>

Este nuevo vocabulario, recibe constantes actualizaciones, al punto que, al día de la fecha, la ontología cuenta con 946 clases.

En particular, la clase review, definida bajo el namespace <http://schema.org/Review> y es subclase de `schema:CreativeWork` que a su vez es subclase de `schema:Thing`

.

Propiedades de Review

- `itemReviewed` El ítem que está siendo evaluado, tiene rango `schema:Thing`
- `reviewBody` El valor textual del review de la evaluación, que tiene rango `schema:Text`
- `reviewRating` El valor numérico del review de la evaluación, que tiene rango `schema:Rating`

Propiedades de CreativeWork

- `about` El tema del contenido. Rango `schema:Thing`.
- `aggregateRating` El promedio de la acumulación de uno o más ratings dentro de un review. Tiene rango `schema:AggregateRating`
- `author` El autor del contenido. Tiene rango `schema:Person` o `schema:Organization`
- `comment` Comentarios sobre el contenido. De rango `schema:Comment` o `schema:UserComments`.
- `commentCount` Cantidad de comentarios. Rango `schema:Integer` .
- `creator` El autor del contenido. Tiene rango `schema:Person` o `schema:Organization`
- `dateCreated` Fecha en la que fue creado. Rango `schema:Date` .
- `dateModified` Fecha en la que fue modificado por última vez. Rango `schema:Date`

.

- `datePublished` Fecha en de la primer publicación. Rango `schema:Date` .
- `publisher` El publicador. Tiene rango `schema:Organization`
- `review` El review sobre el contenido. Tiene rango `schema:Review` .
- `text` Contenido textual. Tiene rango `schema:Text` .

Properties de Thing

- `additionalType` Tipos adicionales para el ítem que en general son utilizados para especificar tipos externos, como podría ser por ejemplo una película de clase <http://schema.org/Movie> con el tipo adicional <http://dbpedia.org/ontology/> . Tiene rango `schema:URL` .
- `alternateName` Especifica un alias. Rango `schema:Text` .
- `description` Una descripción corta del ítem. Rango `schema:Text`

name Establece el nombre. Rango schema:Text .

sameAs Una referencia a una url que desambiguadamente represente al ítem, como podría ser una URL de wikipedia, dbpedia, freebase, etc . Rango schema:URL.

url URL del ítem. Rango schema:URL

Unas 60 propiedades de CreativeWork fueron omitidas por no iban al caso ya que sólo tienen razón de ser para otras clases que también heredan de CreativeWork. Lo importante aquí es remarcar un par de situaciones interesantes:

=Las propiedades Review:reviewBody, CreativeWork:text y Thing:description podrían contener el valor textual de la evaluación del review semánticamente correcta por la definición de cada una.

=Las propiedades CreativeWork:author y CreativeWork:creator son exactamente iguales.

=Los valores CreativeWork:dateCreated CreativeWork:dateModified y CreativeWork:datePublished podrían generar confusión.

=Sería sintácticamente correcto que un Review utilice la propiedad Review

Más adelante se mostrarán y enumerarán los problemas que estas cuestiones (causadas por el afán de hacer uso de la herencia lo más posible) generan.

Capítulo 5

Recolección de Datos

Objetivo

Como se indicó anteriormente la información semántica que va a ser necesaria para construir se encuentra en la web en forma de documentos HTML que tienen la particularidad de ser muy efímeros, de manera tal que un documento que poseía datos relevantes a la fecha, puede al día siguiente, o dejar de estar disponible on-line, o haber cambiado de forma tal que la información de éste ya no es relevante, o ya no la posee.

En [] sección 4.2 Challenges for the selection of data sources se generó una estadística de este caso, donde se estableció que en promedio 62 % de los documentos encontrados, continuaban on-line luego de un año, y de estos, sólo un 56 % aún poseían datos relevantes.

Si bien armar un dataset con sólo información extraída de los documentos sin descargar estos últimos es posible, la situación anterior genera la necesidad de mantenerlos en una copia local para evitar una posible pérdida de los mismos.

El objetivo entonces será armar un repositorio local con los documentos on-line descargados que se cree que tienen la información necesaria.

Estrategia

Recolección y extracción de los datos Como se mencionó antes, la web contiene grandes cantidades de documentos publicados con información semántica. Pero la tarea de encontrarlos, con el agregado de que sólo una pequeña porción de ellos será relevante para los requerimientos no es trivial en lo absoluto debido a la inmensidad del universo en el que se encuentran. La forma de llevar a cabo este objetivo está atada al hardware disponible, tanto para almacenar los datos, como para el tiempo que va a emplear la ejecución de esta tarea. Dado que las bases de datos semánticas sólo almacenan información en forma de tripletas o cuádrupletas, los documentos encontrados deberán someterse a un proceso de extracción que seleccione las sentencias HTML y las convierta a alguno de los lenguajes que soportan tripletas o cuádrupletas. Para esto existen múltiples herramientas. Una vez transformados los documentos HTML a documentos semánticos puede construirse la base de datos semántica con la información recolectada.

Capítulo 6

Evaluación de Calidad de los Datos

Capítulo 7

Curado de los Datos

Capítulo 8

Integración de los Datos

Capítulo 9

Publicación del Dataset Curado

Capítulo 10

Explotación del Dataset

Capítulo 11

Conclusiones y trabajo futuro

Capítulo 12

Bibliografía

Bibliografía

Apéndice A

Publicaciones