Novena práctica de laboratorio de Sistemas de Información para la Web

Motivación

En la práctica anterior se trabajó con microdatos y JSON-LD, un formato de serialización RDF. En la sesión de hoy ahondaremos en el ecosistema RDF mediante la creación de tripletas de forma manual, así como explorando distintas tecnologías que aspiran a generar datos estructurados procesando lenguaje natural.

Textos de prueba

Para las distintas fases del ejercicio se usarán los siguientes textos[1]:

"Miles Davis was an american jazz musician."

"President Barack Obama and European Union leaders huddled in Washington amid growing fears over the future of the euro, which closed greater than 1.3 dollars."

"The New York Times reported that John McCarthy died. He invented the programming language LISP."

Descripción del ejercicio

Primera fase: obtención manual de la información estructurada

1. Obtener toda la información estructurada posible de cada texto usando al menos tipos de `schema.org`, describirla en lenguaje natural, **¡Atención!** No usar conocimiento ajeno al texto (p.ej. No incluir la fecha de defunción de Miles Davis o John McCarthy).

2. Modelar en RDF la información obtenida usando el formato Turtle. (Se recomienda el uso del editor Atom para el cual se ha activado sintaxis coloreada y autocompletado para ficheros `.ttl`).

[1] Obtenidos de `http://wit.istc.cnr.it/stlab-tools/fred/demo/?`

3. Validar primero la sintaxis de los datos RDF y luego verificar que una máquina puede extraer la semántica deseada (usar la *Google Structured Data Testing Tool* sobre la conversión automática de Turtle a JSON-LD).

Segunda fase: obtención automática información estructurada

Existen diversos servicios web que permiten obtener información estructurada (p.ej. en formato RDF) a partir de texto en lenguaje natural. En esta sesión probaremos los siguientes:

- Open Calais.
- DBpedia Spotlight.
- FRED.

¡Atención! Aunque todos los servicios pueden consumirse programáticamente para esta sesión podrán consumirse mediante sus respectivos formularios.

- Para cada servicio y cada texto es necesario obtener un documento RDF (resulta indiferente su formato) con la información extraída.
- Dicha información será traducida (mediante RDF Translator) a N3 (superconjunto de Turtle) para su examen manual y a JSON-LD para su validación por medio de la Structured Data Testing Tool de Google.

A la vista de las pruebas realizadas responde a las siguientes preguntas de forma razonada y con algunos ejemplos:

- ¿Qué ontologías usa cada servicio para "tipar" las instancias detectadas en el texto?
- ¿Existe algún tipo de dichas ontologías que pudiera considerarse equivalente a otro tipo en schema.org? Señala todos los casos que te hayas encontrado con cada servicio, indicando también (si fuera posible) qué propiedades de un tipo mapearían sobre propiedades del otro.
- Reflexiona acerca de los motivos que pueden llevar a cada equipo de desarrolladores a producir una ontología propia. Investiga (someramente) sobre el problema de alineación de ontologías.
- Utiliza el servicio sameAs.org para localizar equivalencias[2] para los tipos fundamentales de Open Calais, DBpedia Spotlight y FRED, así como los homólogos de Schema.org que detectaste. ¿Existe algún servicio para el que no se disponga de información en sameAs?

[2] La URI debe ser absoluta, no es lo mismo schema.org/City que http://schema.org/City…