Análisis y enriquecimiento de información – Taller 3

Nicolas Jimenez, Oscar Forero
MINE4102 – Análisis de información sobre Big Data
Universidad de los Andes, Bogotá, Colombia
of.forero41@uniandes.edu.co
en.jimenez@uniandes.edu.co
Fecha de presentación: diciembre 09 de 2020

Tabla de contenido

Enlace a aplicación: http://mine4102-9.virtual.uniandes.edu.co:9000/

Enlace commit:

Introducción

El desarrollo de este taller se enmarca en la integración, análisis y enriquecimiento de fuentes de datos semiestructuradas y no estructuradas. Bajo este marco se planteó el objetivo de construir una aplicación WEB que nos permita integrar varias fuentes de información financiera, analizar el contenido de esta información, enriquecerla semanticamente y finalmente visualizarla en un dashboard interactivo. A continuación, se describe el proceso y los resultados encontrados en la construcción de la aplicación Web.

Recolección de datos y uso de tecnología NoSQL

Las fuentes de datos utilizadas para alimentar esta aplicación web son: money.stackexchange.com y DBPedia. Para recolectar la información se trabajó a traves de las APIs disponibles de cada una de las fuentes y se utilizaron dos técnicas de consulta diferentes. En el primer caso, con money.stackexchange.con se realizó una conexión a traves de la API de este sitio web y se realizaron consultas que nos permitieron descargar las preguntas y respuestas relacionadas a temas financieros.

Para el caso de DBpedia igualmente se trabajó a través de la API disponible y en este caso se utilizó el lenguaje de consulta SparQL que nos permitió realizar consultas complejas de información. La información que se recolectó se guardó en una base de datos MongoDB. A continuación, se presenta la ficha técnica de las colecciones más relevantes para los datos recolectados:

Colección "answers"	
id	Id
tags	Tags de las respuestas
comment_count	Número de comentarios
down_vote_count	Cuenta de votos en contra
up_vote_count	Cuenta de votos a favor
is_accepted	Validación de la respuesta
score	Puntaje
last_activity	Fecha de última actividad
creation_date	Fecha de creación
answer_id	Id de la respuesta
question_id	Id de la pregunta
content_license	Licencia para el contenido
share_link	Link para compartir
link	Link de la respuesta
title	Título
body	Cuerpo de la respuesta
user_id	Id del usuario
last_edit_date	Fecha de la última edición

Tabla 1: Ficha técnica colección "answers"

Colección "questions"	
id	Id
tags	Tags de las preguntas
comment_count	Cuenta de comentarios
delete_vote_count	Cuenta de votos ppara eliminar
reopen_vote_count	Cuenta de votos para re abrir
close_vote_count	Cuenta de votos para cerrar
is_answered	Sí tiene respuestas
view_count	Cuenta de vistas
creation_date	Fecha de creación
last_edit_date	Fecha de última edición
question_id	Id de la pregunta
link	Link pregunta
title	Título
body	Cuerpo de la pregunta
user_id	Id del usuario
extracted	Extraida
migrated_to	Migrada a
closed_date	Fecha de cierre
locked_date	Fecha de bloqueo
closed_reason	Razón de cierre
bounty_amount	Monto del bono
bounty_closes_date	Fecha de cierre de bono
accepted_answer_id	Id respuesta aceptada
protected_date	Fecha protección
favorite_count	Cuenta favorita
down_vote_count	Cuenta de votos en contra
up_vote_count	Cuenta de votos a favor
answer_count	Cuenta de respuestas
score	Puntaje
last_activity_date	Fecha de última actividad
creation_date	Fecha de creación

Tabla 2: Ficha técnica colección "questions"

Colección "users"	
id	id
badge_counts	Cuenta de insignias
view_count	Cuenta de vistas
down_vote_count	Cuenta de votos en contra
up_vote_count	Cuenta de votos a favor
answer_count	Cuenta de respuestas
question_count	cuenta de preguntas
account_id	Id de la cuenta

Tabla 3: Ficha técnica colección "users"

Análisis sintáctico de contenido y relacionamiento de entidades

Para realizar el análisis sintáctico de contenido y poder identificar las entidades se utilizó la herramienta de Dandelion.eu. Esta herramienta realizó el análisis sobre los datos extraidos de las preguntas y respuestas de nuestra fuente de datos money.stackexchange.com y encontró una lista de entidades que mostramos parcialmente en la tabla 1. Además, para cada entidad se calculó un coeficiente de confianza. Una vez realizada la identificación de entidades se construyó una herramienta de visualización de datos en donde se pueden identificar la cantidad de respuestas por pregunta, los usuarios por ubicación y la cantidad de puntos por respuesta.

Entidades	Confidence
tax deferral	0,8372
passive income	0,8279
money supply	0,823
par value	0,7595
income tax	0,7469
personal finance	0,7238
employer	0,7046
savings	0,6904
stock	0,6813
tax-advantaged	0,6725
investments	0,6634
401k	0,6318
salary	0,6118
gift tax	0,6071

Tabla 4: Entidades

Con el objetivo de realizar un relacionamiento de entidades se construyó la pestaña “Entidad” en la aplicación en la cual podemos observar una gráfica que muestra la cantidad de entidades presentes en los tags de las preguntas. De esta manera se puede observar que la entidad que más se repite o que más se utiliza en los tags de las preguntas es la entidad “Stock” seguida de “taxes”.

Enriquecimiento semántico y utilización de SparQL

Para realizar el enriquecimiento semántico se trabajó con DBpedia, la cual nos permite hacer consultas complejas a través de varios EndPoint. Para esta aplicación web solo se trabajó a través del EndPoint con información en inglés. En primer lugar, se realizó una exploración de los posibles temas financieros que más podían aportar al enriquecimiento. Después de este paso, se realizarón tres consultas a DBpedia con la ayuda del lenguage de consulta SparQL. El objetivo de estas consultas era el de obtener la mayor cantidad de información posible a cerca de tres temas encontrados en las entidades que identificamos, estos tres temas son: “currency”, “salary” y “employee”. Una vez realizadas las consultas se recolectó la información y se guardó en nuestra base de datos de mongoDb.

Consultas sobre Mongo DB

Una vez recolectados los datos de nuestras fuentes de información y habiendo identificado las entidades se procedió a realizar las consultas sobre nuestra base de datos en mongoDb. La primera consulta que se realizó fue sobre la moneda que se maneja por paises, gracias a esta consulta se puede ver que la moneda que más se maneja por paises es el Euro seguido del dólar. Para esta consulta y gracias al enriquecimiento semantico se pudo encontrar la información de los códigos de las monedas de cada país. La segunda consulta que se realizó fue sobre los salarios que tienen las personas. Para el

caso de esta consulta, encontramos que teníamos información sobre salarios de diferentes personas y en diferentes monedas, por esta razón se clasificó la información y se construyerón tres gráficas diferentes, una para el dólar americano, otra para el Euro y la última para la libra esterlina. Se encontró que en los tres casos de las monedas la persona que ganaba más salario era un director ejecutivo de alguna empresa multinacional conocida. Finalmente, la tercera consulta que se realizó fue sobre el número de empleados que tienen las empresas. Esta consulta nos permitió encontrar información de diferentes empresas sobre el número de empleados. Todas las consultas que realizamos a MongoDB las hicimos a través de la operación mapReduce para el procesamiento escalable.

Visualización de resultados

Para la visualización de resultados se construyó una aplicación Web con la ayuda del framework Django y en el lenguaje de programación python. Esta aplicación Web esta dividida en 4 pestañas llamadas: RSS, Dbpedia, Tag Cloud y Entidad. Dentro de estas pestañas se puede visualizar información sobre las preguntas y respuestas hechas en el sitio Web money.stackexchange.com, sobre las monedas por países, salarios en diferentes monedas, empleados por empresa, tags populares en money.stackexchange.com y las entidades presentes en los tags de las preguntas. Para la construcción y visualizar la información se utilizo la librería chart.js.

Análisis de resultados

Al realizar el análisis de los resultados obtenidos podemos ver que el alcance de esta aplicación Web resulta ser insuficiente para los objetivos propuestos al inicio de este proyecto. En primer lugar y aunque se cuenta con varias fuentes de información, en la aplicación Web no se logró mostrar una correcta relación entre las diferentes fuentes. Tampoco fue posible realizar la integración de una tercera fuente de información como Twitter, el cual pudo haber aportado información sobre noticias o novedades financieras. Por otra parte, se logró identificar la entidades en las preguntas encontradas en money.stackexchange.com, lo cual nos permitió identificar temas clave para el enriquecimiento semantico. Sin embargo, también hay que mencionar que este enriquecimiento semantico no se logró visualizar de forma ideal en la plataforma. Puesto que, se encontraron dificultades al momento de relacionar la información para mostrarla en la plataforma. Como logros obtenidos se puede resaltar el uso de SparQl como lenguaje de consulta sobre la fuente de datos Dbpedia. Además, se logró implementar un Tag Cloud con los temas populares de money.stackexchange.com. Dentro del campo de mejoras posibles se pueden incluir temas como la integración de más fuentes de información, visualización del enriquecimiento semantico en la aplicación web y la creación de un menú para generar interacción con el usuario de la aplicación Web.

Conclusiones

- El uso de diferentes fuentes de información nos permite tener una base de datos con volumen y variedad de datos.

- El uso correcto de una fuente de información como money.stacjexchange nos permite tener un flujo de datos constante.
- El conocimiento de diferentes lenguajes de consultas de información es ideal para poder recolectar la información adecuada en la fuente de datos ideal.
- La correcta visualización de información requiere trabajo en el back-end de la aplicación, el cual en algunas ocasiones no se puede evidenciar de parte del usuario.
- Aunque las herramientas de analisis de sintactico son populares en internet, es dificil encontrar una herramienta que se adapte a las necesidades especificas de cada proyecto.
- El uso de SparQl con una fuente de datos como Dbpedia permite generar volumenes de información relevante para los usuarios.

Bibliografía

- Chart.js (2020). Chart.js documentation. URL: https://www.chartjs.org/docs/latest/
- Dandelion API (2020). Dandelion Documentation. URL: https://dandelion.eu/docs/
- Dbpedia. (2020). DBpedia. URL: https://wiki.dbpedia.org/
- Django (2020). Django documentation. URL: https://docs.djangoproject.com/en/3.1/
- MongoDB (2020). MongoDB Documentation. URL: https://docs.mongodb.com/
- SparQL Query Language for RDF. (2013). W3. URL: https://www.w3.org/TR/rdf-sparql-query/