Practica de obtencion de datos

Practica de obtencion de datos
- 0. Autores
- 1. Descripcion
- 2. Instalacion
  - a. Instalacion python 3
  - b. Instalacion de librerias
  - b. Preparacion de archivos
- 3. Uso
  - Ejecucion del *scraper.py*:
  - Ejecucion del *merger.py*:
  - Ejecución del *rdf_generator.py*
- 5. Figuras generadas
- 6. Posibles mejoras
- 7. Otros enlaces de interes

0. Autores

Veronica Gomez Gomez, Pablo Olmos Martinez y Carlos Grande Nuñez

1. Descripcion

La practica de la asignatura tiene como objetivo proporcionar datos en formato RDF de los transportes
METRO, METRO Ligero y Cercanias RENFE, para su uso posterior en la obtencion de rutas accesibles.

La entrega se compone de 9 ficheros:

1. scraper.py (codigo del *scraper*)
2. merger.py (codigo integrador para la unión de los csv)
3. rdf_generator.py (codigo generador del RDF)
4. stops_metro (datos GTFS del metro)
5. stops_ligero (datos GTFS del metro ligero)
6. stops_cercanias (datos GTFS de cercanias renfe)
7. scraper_output.csv (fichero salida del scraper)
8. transportes.csv (fichero con la base de datos final tras la unión)
9. obd.rdf (fichero RDF)

2. Instalacion

a. Instalacion python 3

Asegurse de tener Python 3.x.x instalado en el PC. Puedes descargar las últimas versiones desde los siguientes enlaces:

https://www.python.org/downloads/

- Windows: https://www.python.org/downloads/windows/
- Linux/UNIX: https://www.python.org/downloads/source/
- Mac OS X: https://www.python.org/downloads/mac-osx/

b. Instalacion de librerias

Para la correcta ejecución de los codigos python serán necesarias las siguientes librerias:

- BeautifulSoup4
- request
- re
- pandas
- rdflib
- csv

Para instalar una librería debe ejecutarse en la consola de python el siguiente comando cambiando la palabra 'libreria' por el nombre de la libreria que desea instalarse:

```
$ pip install libreria
```

b. Preparacion de archivos

Descargar y guardar todos los ficheros mencionados en la descripcion en la misma carpeta.

Los ficheros fundamentales para la correcta ejecucion son:

- scraper.py
- merger.py
- rdf_generator.py
- stops_metro
- stops_ligero
- stops_cercanias

3. Uso

Una vez guardados todos los ficheros en el mismo directorio se deberán ejecutar los archivos Python en el orden descrito a continuación.

Ejecucion del *scraper.py*:

1. Ejecuta el *scraper.py* con python 3 mediante la consola o algún IDE como *Spyder* o *Pycharm*
2. El scraper generará un fichero *scraper_output.csv* con los datos de las webs del Consorcio Regional de Transporte de Madrid

Ejecucion del *merger.py*:

3. Una vez se ha generado el fichero *scraper_output.csv* ejecutar el *merger.py* para combinar los ficheros stops con el fichero del scraper.
4. El *merger* generará el fichero final *transportes.csv*, que contendrá toda la información de las estaciones y de cada uno de los transportes tal y como se pedía en el ejercicio.

Ejecución del *rdf_generator.py*

5. Finalmente se ejecutará el *rdf_generator.py*
6. El *rdf_generator* generará el fichero rdf.xml con los datos estructurados de la práctica.

5. Figuras generadas

A partir del RDF se ha generado un grafo de las líneas de metro ligero con sus paradas correspondientes. Para una correcta visualización, se ha seleccionado solo este transporte, ya que los demás transportes complicaban su lectura debido a sus grandes dimensiones.

Este grafo ha sido generado a través de la web: https://www.w3.org/RDF/Validator/

6. Posibles mejoras

El *scraper* se podria realizar con la libreria scrapy en lugar de usar BS4 y expresiones regulares y teniendo ambos codigos comparar cual de los dos es mas eficiente.

El *merger* actualmente funciona con una limpieza exhaustiva de cada una de las columnas con el nombre de las estaciones. Sería interesante probar expresiones regulares para la combinación de estas dos columnas reduciendo así el código.

El *rdf_generator* podría exportarse en formato turtle para conseguir una visualización basada en clusters y no tan lineal.

No siendo necesariamente una mejora, en lugar de fraccionar el codigo en 3 archivos diferentes y ejecutarlos en un orden determinado, podra haberse realizado todo en un solo codigo. Aunque hace más compleja la lectura del código y su detección de errores, si que permite generar todos los archivos necesarios en una sola ejecución.

7. Otros enlaces de interes

- Datos abiertos del Consorcio Regional de Transporte de Madrid (CRTM), Datos.
- Google Transit Feed Specification (GTFS), GoogleTransit.
- Información de los medios de transporte en el CRTM, CRTM.
- Visualizador de RDF: RDF Grapher
- Repositorio con el conenido de esta práctica, Repositiorio.