PUBLIC TRANSPORTATION AND ANALYSIS

Phase 2: Innovation

Intoduction:

Transportation analysis provides the basis for transportation planning. Transportation planning is a process of finding feasible alternatives and components of a transportation system to support human activities in a community.

Core features:

- *Import one or multiple GTFS feeds into one SQLite database for efficient querying of the data.*
- *Augment the sqlite with real walking distances between PT stops using Open Street Map (OSM) data.*
- *Compute simple statistics for the public transport networks such as number of stops, routes, network length.*
- *Filter databases spatially and temporally to match your area and time region of interest.*
- *Perform accessibility analyses using a routing/profiling engine*
    - *Adapted from the Connection Scan Algorithm (CSA).*
    - *Compute all Pareto-optimal journey alternatives between an origin-destination pair, and summarize connectivity with measures on travel time and number of transfers.*
- *Produce data extracts in various formats (network edge lists, geojson).*

Prerequisites

- *Python 3.8*
- *Supported platforms: Linux, OSX & Windows*
- *Optional: git is used for development.*

Install

Linux and MacOS

pip install gtfspy

Windows

Windows should work, but has not been tested or and may not be supported as much. Please report problems.

Windows users may need to install Shapely library first. Download Shapely wheel and then run:

```
pip install wheel
pip install {path to the Shapely wheel file on your PC}
```

If you come across the Microsoft Visual C++ 14.0 is required error, you may need to download the latest Microsoft Visual C++ Build Tools. You can download it from here.
After that, continue with:

```
pip install gtfspy
```

Development quickstart

Use this if you want to be able to edit gtfspy's source code.

```
git clone git@github.com:CxAalto/gtfspy.git
cd gtfspy/
pip install -r requirements.txt # install any requirements
nosetests . # run tests
```

Remember to also add the gtfspy directory to your PYTHONPATH environment variable.

Examples

- *Importing a GTFS feed into a sqlite database*
- *TODO: Validate an imported feed*
- *Compute and plot temporal distance profiles between an origin--destination pair*
- *Visualizing the public transport network on map*
- *Filter GTFS feed spatially and temporally*
- *Extract a network / a temporal network from the GTFS database*
- *TODO! Run a simple accessibility analysis pipeline!*

Contributing

We welcome contributions as GitHub pull requests. In your pull request, please also add yourself as a contributor in the list below.

Versioning

This library is not yet stabilised, and new features are being developed. Thus code organization and interfaces may change at a fast pace. More precise versioning scheme will be decided upon later.

Changelog

View the changelog.

Authors

Package maintainers

- *Rainer Kujala*
- *Richard Darst*
- *Christoffer Weckström*

Other contributors

- *Manuel Rios (marz7002)*
- *Nils Haglund*
- *Michaela Ockova (evelyn9191)*

Acknowledgments

- *The development of this Python package has benefited from the support by Academy of Finland through the DeCoNet project.*
- *For running the Java routing, we use the Graphhopper routing library.*
- *You?*

NOTE:

File Naming Convention: TechnologyName_Phase2 After completion upload your file to your private GitHub account. Please give access to your faculty evaluators[facultyevaluator@gmail.com] and industry evaluator [IndustryEvaluator@skillup.online] to your private GitHub repository for evaluation process