MOBIDATALAB

Labs for prototyping future mobility data sharing solutions in the cloud

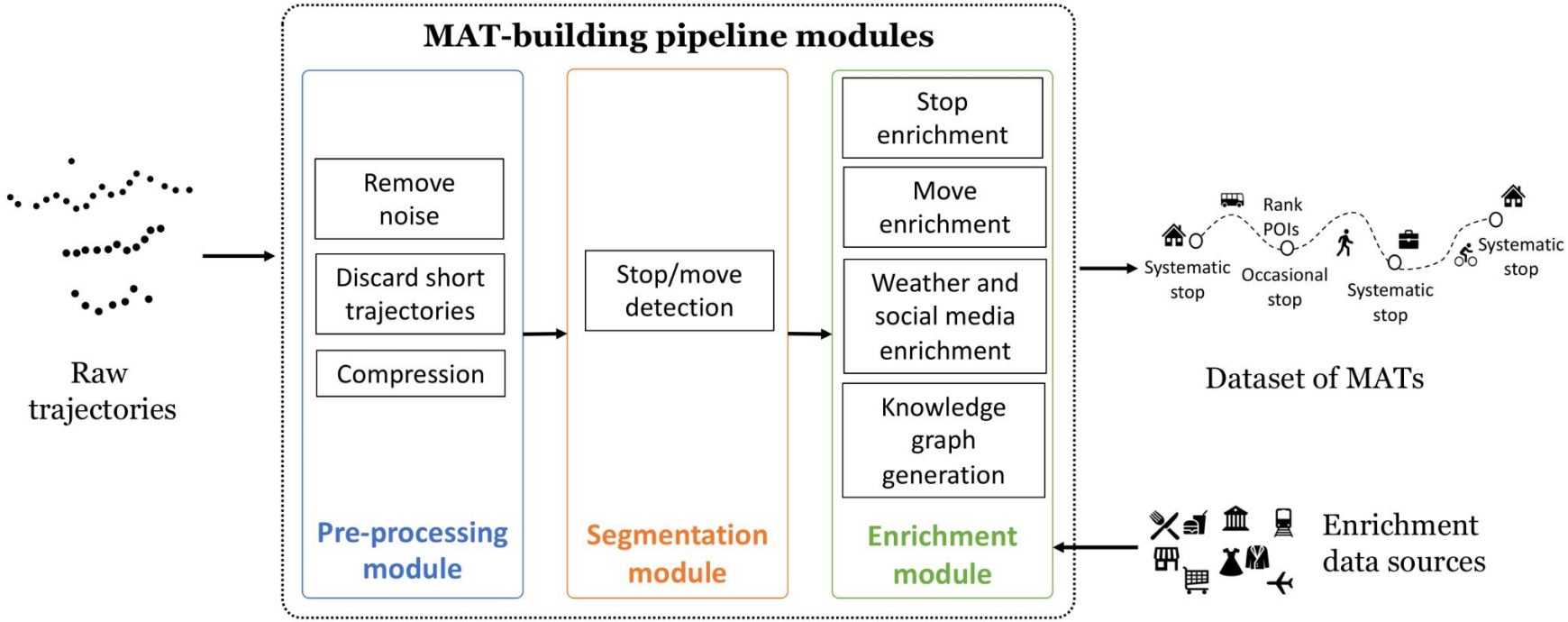
Semantic enrichment processor WebAPI

mobidatalab.eu

MobiDataLab is co-funded by the EU under the H2020 Research and Innovation Programme (grant agreement No 101006879).

Introduction

- The **semantic enrichment processor** enables to **enrich** trajectories with selected **aspects** (i.e., **semantic dimensions**). More on these later.
- The processor's objective is to help **uncover** interesting **movement behaviors** and **patterns** related to individuals or groups which would be otherwise unavailable.



GitHub repository: https://github.com/MobiDataLab/mdl-semantic-enrichment

The semantic enrichment processor webAPI

- The semantic enrichment processor WebAPI exposes **three different endpoints**, each related to one of the steps of the pipeline:
  - https://services.mobidatalab.eu:8443/semantic/Preprocessing
  - https://services.mobidatalab.eu:8443/semantic/Segmentation
  - https://services.mobidatalab.eu:8443/semantic/Enrichment

- The user can interact with each endpoint via **HTTP POST** and **GET requests**:
  - A **POST** request can be used to **initiate** a **task**.
  - **GET** requests can be used to **monitor** the status of a task. When the task terminates, the server **returns** the task's **results**.

WebAPI live demo – Preprocessing / 1

- **Python script** implementing the **live demo** available at: https://github.com/MobiDataLab/mdl-semantic-enrichment/blob/main/examples_api_request.py
- Using a dataset of trajectories moving within the city of Rome, Italy.
- Let's start! The **preprocessing step** takes as input a dataset of trajectories; **filters** out **noisy samples** and trajectories with **few samples**.
- It can also **compress** the preprocessed trajectories – this can speed up subsequent tasks.
- The final result is a **dataset** of **preprocessed trajectories**.

WebAPI live demo – Preprocessing / 2

- To **initiate** a **preprocessing task**: send a **HTTP POST** request sending the appropriate input parameters – see the documentation at: http://semantic.westeurope.cloudapp.azure.com:8000/docs#/default/preprocess_semantic_Preprocessing__post

- The user can then **monitor** the **task's status**, as well as get the **results** once they're ready, by sending a sequence of **HTTP GET** requests – see the documentation at: http://semantic.westeurope.cloudapp.azure.com:8000/docs#/default/preprocess_semantic_Preprocessing__get

WebAPI live demo – Segmentation

- Takes as input a dataset of (preprocessed) trajectories. For each trajectory, it finds out when:
  - the object is **staying** at some **location** for some **time** (**stop segment**)
  - or **moving** from some location to another one (**move segment**).

- To **initiate** a **segmentation task**: send a **HTTP POST** request sending the appropriate input parameters – see the documentation at: http://semantic.westeurope.cloudapp.azure.com:8000/docs#/default/segment_semantic_Segmentation__post

- The user can **monitor** the **task's status**, as well as **get** the **results** once they're ready, by sending repeated **HTTP GET** requests – see the documentation at: http://semantic.westeurope.cloudapp.azure.com:8000/docs#/default/segment_semantic_Segmentation__get

WebAPI live demo – Enrichment / 1

- Takes as input (1) a **dataset** of (preprocessed) **trajectories**, (2) the **stop** and **moves segments** detected in the segmentation step, plus (3) a few other input **parameters** and **data sources** used to enrich the trajectories with the following semantic aspects:
  - **Stops and their "regularity"** (i.e., if they are occasional or belong to a cluster of stops in a location). **Stops** are also **augmented** with the **POIs** located nearby their centroids.
  - **Moves**. Additionally, the processor attempts to **augment** each move with the **estimated transportation means** (optional).
  - **Weather conditions** (optional).
  - **Social media posts** (optional).

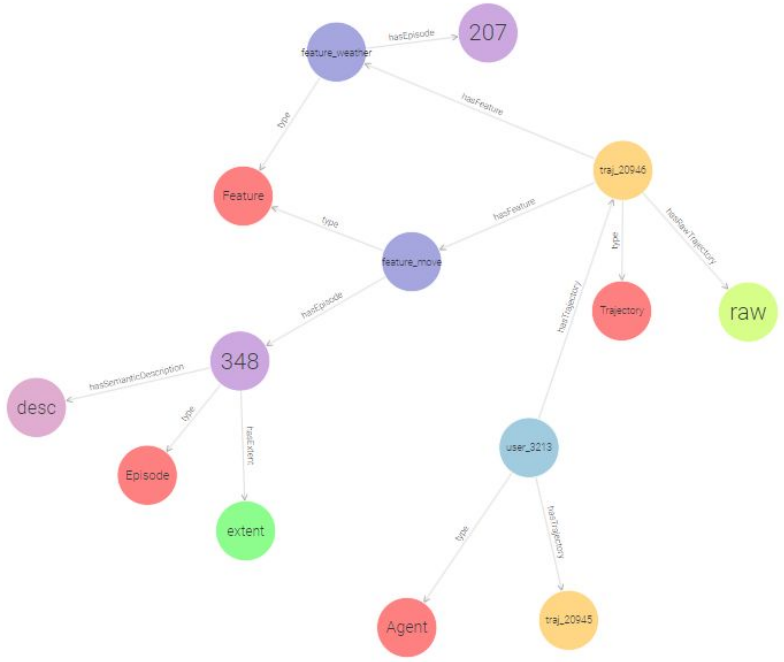
WebAPI live demo – Enrichment / 2

- To **initiate** an **enrichment task**: do a **HTTP POST** request sending the appropriate input parameters – see the documentation at: http://semantic.westeurope.cloudapp.azure.com:8000/docs#/default/enrich_semantic_Enrichment__post

- The user can then **monitor** the **task's status**, as well as **get** the **results** once they're ready, by performing repeated **HTTP GET** requests – see the documentation at: http://semantic.westeurope.cloudapp.azure.com:8000/docs#/default/enrich_semantic_Enrichment__get

Analysing the enriched trajectories...

- The enrichment step returns an **RDF knowledge graph** containing the dataset of **enriched trajectories**.
- The graph can be **imported** in a **triplestore** of choice (e.g., **GraphDB**), and then analysed by means of **SPARQL queries**. A few examples: https://github.com/MobiDataLab/mdl-semantic-enrichment/tree/main/misc/SPARQL



	type_move	t_start	t_end	duration_mins
1	step_specialized:Train	"2014-04-10T08:30:21+00:00"^^xsd:dateTime	"2014-04-10T09:15:52+00:00"^^xsd:dateTime	"45"^^xsd:long
2	step_specialized:Walk	"2014-04-10T09:31:13+00:00"^^xsd:dateTime	"2014-04-10T09:31:13+00:00"^^xsd:dateTime	"0"^^xsd:long
3	step_specialized:Walk	"2014-04-10T09:55:34+00:00"^^xsd:dateTime	"2014-04-10T10:05:37+00:00"^^xsd:dateTime	"10"^^xsd:long
4	step_specialized:Bus	"2014-04-10T10:36:33+00:00"^^xsd:dateTime	"2014-04-10T10:46:53+00:00"^^xsd:dateTime	"10"^^xsd:long
5	step_specialized:Walk	"2014-04-10T11:47:00+00:00"^^xsd:dateTime	"2014-04-10T12:32:48+00:00"^^xsd:dateTime	"45"^^xsd:long
6	step_specialized:Bus	"2014-04-10T12:53:40+00:00"^^xsd:dateTime	"2014-04-10T13:22:17+00:00"^^xsd:dateTime	"28"^^xsd:long
7	step_specialized:Bus	"2014-04-10T13:28:41+00:00"^^xsd:dateTime	"2014-04-10T13:28:41+00:00"^^xsd:dateTime	"0"^^xsd:long
8	step_specialized:Bus	"2014-04-10T13:44:43+00:00"^^xsd:dateTime	"2014-04-10T13:44:43+00:00"^^xsd:dateTime	"0"^^xsd:long
9	step_specialized:Bus	"2014-04-10T14:22:47+00:00"^^xsd:dateTime	"2014-04-10T14:45:36+00:00"^^xsd:dateTime	"22"^^xsd:long

MOBIDATALAB
Labs for prototyping future mobility data sharing solutions in the cloud

Further documentation and material...

- **More information** on the **RDF knowledge graphs** generated by the processor, and how to analyse them, available in the **MAT-Builder journal article**: https://ieeexplore.ieee.org/iel7/6287639/6514899/10227262.pdf

- **More information** on the **internal structure** of the **binary files** expected as input by the various endpoints: https://github.com/MobiDataLab/mdl-semantic-enrichment#overview-on-mat-building-pipeline-modules-and-input-datasets

- **Jupyter notebooks** with **examples** on how to **generate POI** and **weather datasets** to be given as input to the enrichment endpoint: https://github.com/MobiDataLab/mdl-semantic-enrichment/tree/main/misc/notebooks

- To **intuitively understand** the outputs generated by the processor: install and try out the **interactive user interface**: https://github.com/MobiDataLab/mdl-semantic-enrichment#use-of-the-semantic-enrichment-processor

Thank you for listening!

Francesco Lettich, ISTI-CNR
francesco.lettich@isti.cnr.it

The slides are available at: https://github.com/MobiDataLab/mdl-semantic-enrichment/tree/main/misc/slides

MOBIDATALAB
Labs for prototyping future mobility data sharing solutions in the cloud