IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 2 August 2023, accepted 10 August 2023, date of publication 23 August 2023, date of current version 30 August 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3307824

RESEARCH ARTICLE

Semantic Enrichment of Mobility Data: A Comprehensive Methodology and the MAT-BUILDER System

FRANCESCO LETTICH[1], **CHIARA PUGLIESE**[1,2], **CHIARA RENSO**[1], **AND FABIO PINELLI**[3]

[1]ISTI, Consiglio Nazionale delle Ricerche, 56124 Pisa, Italy
[2]Department of Computer Science, University of Pisa, 56126 Pisa, Italy
[3]SySMA Unit, IMT School for Advanced Studies, 55100 Lucca, Italy

Corresponding author: Francesco Lettich (francesco.lettich@isti.cnr.it)

This work was supported in part by the EC H2020 Projects MOBIDATALAB under Grant 101006879, in part by MASTER under Grant 777695, in part by the Scheme "INFRAIA-01-2018-2019 through the Integrating Activities for Advanced Communities" under Grant 871042, and in part by the "SoBigData++: European Integrated Infrastructure for Social Mining and Big Data Analytics" (http://www.sobigdata.eu).

ABSTRACT The widespread adoption of personal location devices, the Internet of Mobile Things, and Location Based Social Networks, enables the collection of vast amounts of movement data. This data often needs to be enriched with a variety of semantic dimensions, or *aspects*, that provide contextual and heterogeneous information about the surrounding environment, resulting in the creation of multiple aspect trajectories (MATs). Common examples of aspects can be points of interest, user photos, transportation means, weather conditions, social media posts, and many more. However, the literature does not currently provide a consensus on how to semantically enrich mobility data with aspects, particularly in dynamic scenarios where semantic information is extracted from numerous and heterogeneous external data sources. In this work, we aim to address this issue by presenting a comprehensive methodology to facilitate end users in instantiating their semantic enrichment processes of movement data. The methodology is agnostic to semantic aspects and external semantic data sources. The vision behind our methodology rests on three pillars: (1) three design principles which we argue are necessary for designing systems capable of instantiating arbitrary semantic enrichment processes; (2) the MAT-Builder system, which embodies these principles; (3) the use of an RDF knowledge graph-based representation to store MATs datasets, thereby enabling uniform querying and analysis of enriched movement data. We qualitatively evaluate the methodology in two complementary example scenarios, where we show both the potential in generating interesting and useful semantically enriched mobility datasets, and the expressive power in querying the resulting RDF trajectories with SPARQL.

INDEX TERMS Multiple aspect trajectory, semantic enrichment, trajectory enrichment, semantic enrichment processing, knowledge graph, resource description framework, python.

I. INTRODUCTION AND MOTIVATIONS

Tracking sensors have experienced consistent development in recent years. These devices generate high-frequency and high-volume data streams daily, capturing the movement of various objects, such as humans, animals, and different types of tracked vehicles like vessels, airplanes, and cars/trucks. Similarly, Internet of Things devices, including drones, smartwatches, airtags, smartbands, and cameras, produce large quantities of tracking data. Despite the massive volume of big spatio-temporal data these devices generate, their tracks often lack essential semantic contextual information. This information is currently sourced and stored through various means, including satellite images, weather

The associate editor coordinating the review of this manuscript and approving it for publication was Vlad Diaconita.

stations, points of interest, social media, web pages, web APIs, and data spaces, among others. The process of combining spatio-temporal tracks with semantic information is relatively unexplored. Consequently, semantically enriched traces, which can offer valuable insights across numerous domains, are underutilized. When movement data and contextual semantic information are synergistically combined, they lead to the generation of the so-called multiple aspect trajectories (MATs) [1]. MATs are location tracks semantically enriched with multiple heterogeneous semantic dimensions, or *aspects*.

The ability to construct MATs from different aspects and data sources enables the development of innovative applications aimed at extracting and analyzing mobility behaviours. Telco companies, for instance, utilize semantically enriched movement data. They collect vast amounts of data from their customers, aiming to combine this data with aspects related to their behaviours. For example, a company's marketing business unit may wish to design marketing campaigns based on the apps customers are using in specific locations. They might then enrich customers' movements with aspects dependent on the applications used. In this context, a semantic enrichment process might enrich customers' traces with semantic information such as the applications used at a specific location, customer profiles, or characteristics of the areas involved (e.g., by examining the categories of their points of interest (POIs)). The enriched data can provide valuable insights to the marketing business unit about where, when, and how to run specific marketing campaigns.

Other interesting application scenarios may focus on transportation. For example, telco companies may wish to better understand the usage patterns of public transportation to propose new routes or schedules. Similarly, in the tourism domain, enriching customer traces with aspects related to the POIs they visited, the transportation means used, or the weather conditions encountered could result in improved recommendations and more effective management of overtourism. The enriched data could then be leveraged to infer which kinds of tourists are visiting a city, how they visited its various areas, and their spending profiles (e.g., by analysing the expense level of the POIs they visited).

Overall, the capacity to generate different MAT dataset variants for each unique application scenario, ideally in a consistent format, would facilitate the pursuit of diverse objectives and improve the quality of subsequent analyses. While there is an increasing interest in the modelling and analysis of multiple aspect trajectories [1], [2], we observe that in the literature, there is no comprehensive established methodology that can guide users in building MAT datasets. Indeed, existing approaches are tied to specific datasets, aspects, sources, or application scenarios, rendering them unsuitable for different or more dynamic scenarios. As such, we argue that users should have a methodology that facilitates them in instantiating their own semantic enrichment processes, incorporating dynamic and heterogeneous aspects, with information sourced from multiple *external semantic data sources*.

In this paper, we address this problem and propose a methodology that focuses specifically on the notion of *semantic enrichment process*, enabling the creation of MAT datasets. Such a process should be highly configurable, allowing for easy setup of different enrichment processes leading to several variations of MAT datasets, depending on the specific aspects and sources considered for the application questions at hand. In more concrete terms, users should be able to dynamically define (1) the trajectory parts to be enriched, (2) the semantic aspects to use for enrichment, (3) the external data sources to build information for these aspects, and (4) the best approaches to enrich movement data. Additionally, users should be able to (5) rely on a uniform representation for MAT datasets, consistently enabling effective querying and analysis. To the best of our knowledge, such a methodology does not exist in the literature. Our proposed methodology rests on three pillars:

- The design principles of modularity, configurability, and extensibility, which are necessary for designing systems capable of instantiating *arbitrary* semantic enrichment processes.
- The MAT-Builder system, which embodies the above-mentioned principles.
- The use of a knowledge graph-based representation [3] based on the Resource Description Framework (RDF) formalism to store MAT datasets, allowing for uniform querying and analysis of enriched movement data.

In line with the previously mentioned related approaches, we assess the effectiveness of our methodology through an extensive qualitative empirical evaluation, highlighting its usefulness in constructing and analyzing enriched datasets. To this end, we provide two illustrative example scenarios to showcase the utility, versatility, and expressive power of our proposal, one in the tourism domain and the other in the urban mobility domain. These examples demonstrate how MAT-Builder can be used to effectively address various typical queries.

The rest of the paper is organized as follows. Section II presents some fundamental notions that are used throughout the paper, and introduces the problem statement. Section III gives an overview of the related literature and highlights the novelty of our contributions. Section IV presents the three pillars underpinning our methodology. Section V presents a concrete instance of a semantic enrichment process that can be implemented with our methodology. Such a process is then used in the qualitative experimental evaluation (Section VI) to generate knowledge graphs containing datasets of MATs, which are then employed to conduct analyses on the movement behaviours of selected individuals. Lastly, Section VII draws the final conclusions.

II. PRELIMINARIES AND PROBLEM STATEMENT

In this section, we first review some fundamental notions that will be used throughout the rest of the paper. These notions then set the stage for the final part of the section, which presents the problem statement that this study aims to address.

Definition 1 (Trajectory): We define a trajectory generated by a moving object mo to be $T = (mo, P)$, where $P = \langle p_1, p_2, \ldots, p_n \rangle$ represents the time-stamped sequence of mo's geographical locations (or samples). The sample $p_i = (x_i, y_i, t_i) \in P$ represents the i-th position of mo in space and time, with x_i and y_i providing the object's geographical location, and t_i the timestamp at which the sample was recorded.

Before we can semantically enrich a trajectory, we often need to identify relevant parts within the trajectory. This is achieved through a process known as trajectory segmentation.

Definition 2 (Segmented trajectory): Given a trajectory T, the process of identifying its *relevant segments*, i.e., subtrajectories in T uniquely characterized by one or more properties, is called *trajectory segmentation*. Such a process yields a segmented trajectory $T_{seg} = (T, S)$, where S denotes the set of such segments.

Numerous segmentation criteria exist in the literature. It is also worth noting that a segmented trajectory could simply be the original trajectory, in which case the segments correspond to the individual trajectory samples. For the purposes of this paper, we will assume that trajectories are segmented into sequences of *stop* (where the object is stationary) and *move* (where the object is moving from one stop to another) segments, following the criterion introduced by Spaccapietra et al. [4]. Having defined the notions of raw trajectory and segmented trajectory, we can now move to the concept of multiple aspect trajectory, where the segments of segmented trajectories are semantically enriched with *aspects* (semantic dimensions). We start by defining the notion of aspect.

Definition 3 (Aspect): An *aspect*, or semantic dimension, represents the *categorization* of a real-world fact, and it can be defined in terms of the *space of values* that instances belonging to the aspect can assume. We thus define an aspect as a pair $A = (desc, SAT)$, where $desc$ provides the description of the aspect and $SAT = \{a_1, \ldots, a_k\}$ specifies the set of attributes that represent the various properties characterizing the aspect. Accordingly, we define the *instance of an aspect* A to be a specific instantiation of its attributes in SAT.

The above definition implies that aspects can capture the complexity of real-world entities and phenomena due to their potential structural complexity and heterogeneity. For instance, consider the aspect *Points Of Interest*. The attributes of this aspect might include the POI name, description, photos, user reviews, geographical location, and opening and closing times. Another example is the *weather conditions* aspect, whose instances represent the meteorological conditions at a specific location and time and can have attributes such as temperature, weather conditions, humidity, dew point, and more.

We are now ready to introduce the concept of a *multiple aspect trajectory*, a segmented trajectory enriched with (possibly many) aspects.

Definition 4 (Multiple aspect trajectory): We define a *multiple aspect trajectory* to be a pair $MAT = (mo, E)$, where mo is the moving object that generated a segmented trajectory T_{seg}, while $E = \langle e_1, \ldots, e_n \rangle$ represents the sequence of T_{seg} *segments enriched* with zero or more aspects. More precisely, we define a part $e_i \in E \in MAT$ as a pair $e_i = (s_i, AI_i)$, where $s_i \in T_{seg}$ represents a segment from T_{seg} while AI_i represents the set of instances of the various aspects that enrich s_i. Note that AI_i can be the empty set – in other words, a part might not be semantically enriched.

Moving forward, we will use the terms semantic trajectory, used in older literature, and multiple aspect trajectory interchangeably. Observe that Definition 4 is general and thus applicable to a broad range of entities, e.g., vehicles, sea vessels, and animals. In urban contexts, particularly focusing on individual movement patterns within cities, a person's MAT might have a move segment enriched with an instance of the *move* aspect indicating the means of transportation (e.g., car, train, bike, or subway, each possibly having its own attributes). Stop and move segments might also be enriched with instances of the *weather conditions* aspect. A stop segment could be enriched with an instance of the *regularity* aspect to indicate whether the person regularly stays at the associated location, which can be useful to determine if the location is their home or workplace.

Information about the aspects often needs to be gathered from sources outside the trajectory or segmented trajectory datasets. To address this, we introduce the notion of external semantic data source.

Definition 5 (External semantic data source): Given an aspect A, we define the set of external semantic data sources associated with A as the set of sources that are external to the trajectory and segmented trajectory datasets and that are used to gather information on A. We denote such set as $ESS_A = \{so_1^A, \ldots, so_j^A\}$.

Examples of external semantic data sources can be social media, the semantic web, web APIs, local files, websites, data spaces, and more – for instance, collecting information about POIs typically involves accessing data sources like OpenStreetMap,[1] WikiData,[2] and possibly others. Let us now introduce the key concept of the *semantic enrichment process*, which pieces together the definitions introduced so far.

Definition 6 (Semantic enrichment process): A semantic enrichment process SEP takes three key inputs: a dataset of raw trajectories D, a set of aspects $ASP = \{A_1, \ldots, A_q\}$ for enriching the trajectories, and a set of external semantic data sources $ESS = \{ESS_{A_1}, \ldots, ESS_{A_q}\}$ to be accessed to gather information on the aspects. By means of the application of a sequence of operations $OP = \langle op_1, \ldots, op_r \rangle$, the process

[1] https://www.openstreetmap.org/
[2] https://www.wikidata.org/

first converts the raw trajectory dataset into a dataset of segmented trajectories, D_{seg}, and subsequently enriches each segment of a segmented trajectory $T_{seg} \in D_{seg}$ with the appropriate instances of the aspects in ASP. Aspect instances are created by taking into account the trajectories' characteristics as well as the information gathered from external semantic data sources. This ultimately leads to a dataset of multiple aspect trajectories D_{MAT}.

The definition of semantic enrichment process sets the stage for this work's problem: how can a user be supported in concretely implementing their semantic enrichment processes, given the different sets of aspects, external semantic data sources, sequences of operations, and parameter combinations they might choose to use? More precisely, suppose a user has a raw trajectory dataset D and intends to semantically enrich its trajectories in various ways. With this assumption, we can frame the problem as follows.

Definition 7 (Problem Statement): Let us represent the enrichment processes that the user intends to apply as $SEP_1, \ldots, SEP_z$. Each SEP_i incorporates a unique sequence of operations OP_i that enriches the trajectories in D with instances of the aspect set ASP_i, using information obtained from external semantic data sources in ESS_i. Here, we want to define a methodology and a system, MAT-Builder, that should be designed to assist users in implementing and executing any of their semantic enrichment processes SEP_i, which means:

$$\text{MAT-Builder}(SEP_i(D, ASP_i, ESS_i)) = D_{MAT_i},$$

where D_{MAT_i} represents the final dataset of multiple aspect trajectories.

It is evident that the system should support the user in implementing, reusing existing components, combining, and executing functionalities needed to perform the sequences of operations comprising their semantic enrichment processes. Accordingly, this paper addresses the problem by introducing a methodology that prominently features the MAT-Builder system as one of its core components. Details of this methodology are provided in Section IV.

III. RELATED WORK

Semantic enrichment can be described as a process in which some type of data is augmented with contextual, relevant, and meaningful information. The purpose of this process is to improve the utility of the data, making it more easily understandable and more effectively processed and analyzed. Semantic enrichment finds application across numerous types of data and contexts, for example social media [5], [6], images [7], [8], databases [9], simulations [10], data preparation in data science processes [11], mobility (which we will delve into more deeply later), and more. Despite the significant differences in the types of data and objectives across this vast body of literature, there are a few notable recurring themes pertinent to our work. These are: (1) the use of a system to enrich some type of data, where the components and algorithms used in the system depend on the type of data and the specificities of the problem considered. In some cases, the system is also designed to be open and extensible/customizable by end users; (2) the use of ontologies for semantic modelling of the entities and relationships of interest; (3) accessing external sources to enrich data with pertinent semantic information; (4) the use of some formalism to uniformly represent and query semantically enriched data.

When it comes to the semantic enrichment of mobility data, which is the focus of this work, one has to start from the concept of *semantic trajectory* introduced in the seminal work by Spaccapietra et al. [4]. In this work, trajectories are conceptually segmented into *stop segments*, i.e., sub-trajectories where moving objects remain stationary, and *move segments*, i.e., sub-trajectories where the objects change position. The fundamental intuition is that semantic trajectories can be characterized by key places visited by a moving object during stops, separated by segments where the object's position changes. Starting from this earlier definition, the notion of semantic trajectory has evolved into more complex definitions, with the semantic part gaining progressively more complexity.

The first attempt to propose a methodology and a system for constructing semantic trajectories is SeMiTri [12], which builds upon the foundation established in [4]. In this work, the identification of stop and move segments is leveraged as a means to enrich trajectories.

The first conceptual model of a generic semantic trajectory that goes beyond the basic stop and move segmentation was CONSTANT [13]. Here, a trajectory can be enriched with a limited set of predefined semantic aspects such as activities performed by an object, means of transportation, points of interest, trip purposes, and specific behavior patterns. These aspects are statically associated with trajectory segments and presented as textual labels. Although this approach first introduces the idea of a semantic trajectory characterized by more than its stop and move segments, it does not address how a semantic enrichment process should be conducted as it assumes that semantic trajectories already exist. Finally, both the set of aspects and the semantic sources are fixed.

A later approach called BAQUARA [14], instantiates the CONSTANT model into an ontological framework based on a predefined ontological model of a semantic trajectory. BAQUARA's principal innovation lies in annotating semantic trajectories with information retrieved via SPARQL queries from Linked Open Data sources, hence presenting a first example of a semantic enrichment process leveraging external data sources. However, BAQUARA does not consider the concept of a semantic enrichment process or how to conduct it. Additionally, the aspects considered by BAQUARA are pulled in by linking its ontology with a specific set of compatible concepts from the ontologies used by the three Linked Open Data sources it accesses: DBPedia, LinkedGeoData, and GeoCodes. This feature makes BAQUARA a monolithic framework, offering no support for the addition of new aspects or customization of existing ones.

TABLE 1. Comparison table of the main related works on semantic enrichment of mobility data.

Paper	Type of contribution	Semantic aspects	External semantic data sources	Uniform formalism	Semantic enrichment process
Spaccapietra et al., 2008 [4]	Conceptual model	Predefined (stop and move)	No	No	No
Yan et al, 2011 [12]	Methodology	Predefined (stop and move)	No	No	No
Bogorny et al., 2014 [13]	Conceptual model	Predefined	No	No	No
Fileto et al., 2015 [14]	Ontology	Predefined	Yes (limited to fixed LOD sources)	Yes (RDF)	No
Ruback et al., 2016 [15]	Ontology Mashup	Can be imported (via the ontology mashup)	Yes (limited to LOD sources)	Yes (RDF)	No
Noguiera et al., 2018 [16]	Ontology + Framework	Can be defined (limitedly to the ontology)	Yes (limited to fixed sources)	Yes (RDF)	No
Mello et al., 2019 [1]	Conceptual Model	Can be defined	No	Yes (RDF)	No
This paper	**Methodology**	**Can be defined**	**Yes**	**Yes (RDF)**	**Yes**

STEP [17] is an ontology-based model that can represent an arbitrary number of aspects to enrich movement data. In a later paper, the same authors propose the STEPv2 ontology [16], where they introduce the ability to choose the granularity of single occurrences of aspects and associate occurrences to a single point in space and time. In the same paper, they introduce FrameSTEP, a framework that enriches trajectories with a fixed set of aspects. FrameSTEP shares limitations seen with previous solutions in terms of being monolithic, offering no flexibility in the aspects and external semantic data sources used for enrichment.

In [15], the authors propose an approach to enrich movement data with information collected from various Linked Open Data sources, combined via *dynamic ontology mashups*. Unlike BAQUARA, where the ontology is predefined, this approach allows dynamic selection of the Linked Open Data sources and the aspects available from their ontologies to enrich mobility data. However, this approach is limited to external semantic data sources that are Linked Open Data and to the aspects available in their ontologies, therefore not allowing the instantiation of arbitrary semantic enrichment processes as per Definition 6.

In [1], the authors propose MASTER, a conceptual model for multiple aspect trajectories that is general enough to support an arbitrary number of aspects. In this model, various parts of a trajectory, such as points, segments, stops, and the moving objects themselves, can be enriched with aspects that are not predefined. Each aspect can also be represented via a complex object, thus going beyond label-based representations. Finally, MASTER can explicitly represent relationships between moving objects. The authors translated their conceptual model into an ontology and a logical schema based on the Resource Description Framework (RDF) standard. They also used a triplestore based on NoSQL databases to store multiple aspect trajectories. Overall, while the authors propose an ontology to model multiple aspect trajectories and demonstrate the advantage of storing them in RDF graphs, they do not address the notion of a semantic enrichment process nor propose a methodology and a system that support the instantiation of arbitrary semantic enrichment processes.

Table 1 provides a summary and comparison of the related works presented so far according to five key criteria:

1) **Type of contribution**: this criterion indicates the main contribution characterizing a work. This can be a conceptual model, an ontology, a system, a framework, or a comprehensive methodology.
2) **Semantic aspects**: this criterion assesses how a work addresses semantic aspects, i.e., if they are predefined or can be dynamically defined.
3) **Semantic enrichment process**: this criterion assesses whether a work supports the users in defining their own semantic enrichment process.
4) **External semantic data sources**: this criterion assesses whether a work uses external semantic data sources to enrich mobility data.
5) **Uniform formalism**: this criterion assesses whether a work uses a uniform formalism to store and query datasets of semantically enriched trajectories.

Overall, the reviewed approaches provide varying levels of semantic enrichment representation, ranging from the simplest, such as the stop and move segmentation proposed in [4], to the most complex ones such as MASTER [1]. The most recent approaches inspired the present work, which aims to build upon their strengths while addressing their limitations. These limitations predominantly stem from the absence of a common methodology to instantiate arbitrary semantic enrichment processes or the inability to go beyond a fixed set of aspects or external semantic data sources.

In this work, we propose a methodology for the semantic enrichment processing of movement data that is agnostic to the aspects and external semantic data sources that might be considered. Preliminary and limited parts of our system have already been introduced in a demo paper [18] and a poster paper [19]. In the demo paper, we first introduced an interactive tool designed to showcase the potential of a system, MAT-Builder, that facilitates the generation of datasets of MATs. However, compared to the present work, the version of MAT-Builder presented in the demo paper was neither extensible nor configurable, and it offered a specific instance of a semantic enrichment process. Moreover, it did not use a uniform formalism to store and analyze multiple aspect trajectories in knowledge graphs. The subsequent poster paper, instead, demonstrates an example of MAT-Builder's use in the tourism domain and uses RDF knowledge graphs for storing and querying multiple aspect trajectories in a uniform manner.

Compared to these two previous works, the present paper introduces the overarching methodology that encompasses MAT-Builder, and that allow users to instantiate arbitrary semantic enrichment processes leading to the generation of datasets of MATs. Said methodology is presented in Section IV. To the best of our knowledge, this is the first work that empowers practitioners to fully customize the semantic enrichment processing of movement data in a comprehensive, flexible, modular, and interactive manner.

IV. THE METHODOLOGY

In this section, we introduce the methodology to build datasets of MATs geared towards addressing the problem stated in Section II, Definition 7. To address the need to support and facilitate the user in instantiating different semantic enrichment processes, our methodology is founded on three pillars. The first pillar (Section IV-A) targets the need to implement different semantic enrichment processes, each corresponding (as per Definition 6) to a distinct sequence of operations. This is enabled by three design principles: modularity, extensibility, and configurability. We consider them essential for creating systems capable of effectively addressing the problem statement. The second pillar (Section IV-B) is MAT-Builder, a system that embodies these principles and provides a tangible answer to the problem statement. Lastly, the third pillar (Section IV-C) targets the need for a uniform representation of the final enriched trajectories. This pillar promotes the use of an RDF knowledge graph-based representation to store datasets of MATs. This not only offers uniformity but also facilitates querying and analysis of the enriched movement data.

A. THE DESIGN PRINCIPLES

Designing a system capable of supporting arbitrary semantic enrichment processes of movement data poses several challenges. First, a system might have to deal with a potentially vast amount of aspects; therefore, it must be flexible enough to include different sets of aspects for different semantic enrichment processes. Secondly, there is the need to possibly access multiple external data sources to dynamically gather the most appropriate information to associate with aspects during an enrichment process. Furthermore, users may also want to enrich the very same movement data in different ways, i.e., by choosing different aspects or external data sources. Thus, a system needs to offer users the flexibility to choose the specific operations they wish to employ for enriching movement data, while also allowing for easy incorporation of new operations. To face these challenges, we argue that a system must be designed according to three interconnected principles: *modularity*, *extensibility*, and *configurability*.

A system that generates datasets of MATs must arrange its operations in distinct modules, whereby each module is seen as a component dealing with a specific task. This represents the principle of **modularity**. For instance, one module might deal with trajectory pre-processing, another with trajectory segmentation, and another with semantic enrichment with a selected set of aspects whose information is gathered from appropriate external semantic data sources.

Considering the potential heterogeneity and dynamicity of aspects and external data sources, we also highlight how a system must allow developers to add new modules easily, as well as build on existing ones. For instance, a user might create a new module that includes operations present in existing modules. These requirements express the principle of **extensibility**.

Moreover, not all the modules might be suitable for a particular semantic enrichment process, and a module may depend on the output of other modules to compute its task. Thus, the system must ensure that any dependency between modules is satisfied. Consequently, starting from a set of modules available to the system, a user must be able to (1) pick the modules they need and (2) specify the order in which the modules should be executed (compatibly with any dependency between them). These requirements express the principle of **configurability**.

Finally, the intrinsic synergy between modularity and configurability is at the basis of the important concept of **MAT-building pipeline**. A MAT-building pipeline represents a specific semantic enrichment process, and can be practically seen as a sequence of chosen modules that implement the sequence of operations characterizing said process. Consequently, when integrated within a system the concept of a MAT-building pipeline empowers the user to implement their semantic enrichment processes, as required by the problem statement (Section II, Definition 7). For example, a MAT-building pipeline might first identify the trajectory parts to enrich (e.g., via segmentation), then integrate the external semantic data sources and enrich the segments, and finally output the resulting MAT dataset in an RDF knowledge graph.

The MAT-Builder system, introduced in Section IV-B, combines the three design principles with the concept of MAT-building pipeline to support users in the task of implementing highly customizable (and thus generic) semantic enrichment processing of movement data.

B. THE MAT-BUILDER SYSTEM

In this section we present MAT-Builder,[3] a system to generate datasets of MATs implemented following the design principles introduced in Section IV-A. The system is implemented in Python and consists of two components, the *user interface* (UI) and the *backend*. Figure 1 provides a sketch of the architecture.

The user interface is the system component in charge of graphically exposing to the user the operations of the modules making up some MAT-building pipeline. More specifically, it allows the modules to receive user inputs and provide corresponding feedback. An additional important aspect of the user interface is its capability to dynamically adjust the content

[3] Source code is available at: https://github.com/chiarap2/MAT_Builder

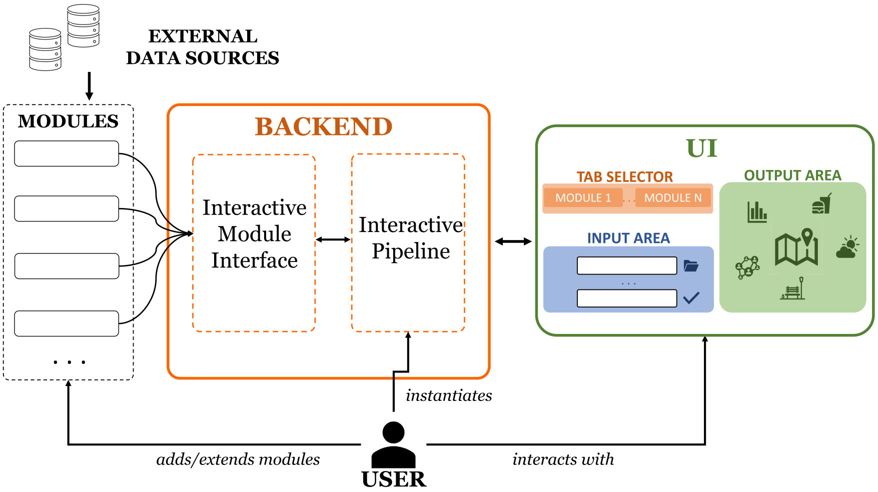


FIGURE 1. Overview of the MAT-Builder system architecture.

displayed on the screen according to the requirements of any MAT-building pipeline. For instance, different pipelines correspond to different sequences of modules – as such, the UI must appropriately represent such differences. Also, different modules typically require different inputs from users. Consequently, the UI must show a set of input fields that depend on the parameters required by the modules. Likewise, different modules usually require to output various kinds of feedback, e.g., plots and summaries.

The backend constitutes the core component of our system. It implements the MAT-Builder's processing engine according to the design principles introduced in Section IV-A. The backend is modular: the operations are distributed across separate modules, each addressing a specific task. Furthermore, the backend is extensible: combined with the open-source nature of MAT-Builder and the extensive library ecosystem of Python, this facilitates the seamless contribution of new operations by researchers and practitioners. Finally, the backend is configurable: users can instantiate their own MAT-building pipelines by picking up the modules they want among those available to the system.

In the rest of this section, we first focus on the UI (Section IV-B1), illustrating its layout and some implementation details. We then provide a detailed overview of the backend (Section IV-B2), showing how it embodies the design principles and how it enables the content displayed by the UI to be dynamically adaptable to any MAT-building pipeline.

1) THE USER INTERFACE

The user interface is built using the Dash library,[4] which provides graphical components with built-in callback mechanisms. These mechanisms enable developers to write functions that are executed whenever these components undergo a state change. This allows users to input information into the system and receive feedback interactively. Additionally, Dash's plotting capabilities are well-suited for representing graphical and geographical data related to MATs.

The layout of the user interface is structured in three areas (see the UI green block in Figure 1): *tab selector area*, *input area* and *output area*. The *tab selector area* presents a sequence of selectable tabs, each representing a single step (i.e., module) of the configured MAT-building pipeline. The *input area* provides the graphical components a module needs to get input from the user, e.g., input fields and execution button. Finally, the *output area* provides the feedback of a module once it terminates executing its task, e.g., summary data or plots.

2) THE BACKEND

As mentioned earlier, the MAT-Builder backend embodies the design principles outlined in Section IV-A. In the following, we discuss the technical implementation of these principles and how the backend enables the user interface to adapt to any MAT-building pipeline dynamically. Hereinafter, we will use object-oriented programming terminology. The main components of the backend are sketched in the orange block in Figure 1.

a: MODULARITY

The principle of modularity is applied by organizing different operations into separate modules (or classes), with each module targeting a specific task. The backend requires each module to provide a common set of management functionalities. These functionalities are used to connect modules in a MAT-building pipeline. This allows for effective control of module states at a higher level, and enables the user interface to dynamically adapt its input and output areas to accommodate any task and, thus, any enrichment process. In the backend's codebase, there is an interface called `InteractiveModuleInterface` which serves as a blueprint for modules: any module implemented in

[4] https://dash.plotly.com/

the backend must be a class that extends this interface and includes six specific methods.

The first method, `populate_input_area`, uses Dash's callback mechanisms to populate the user interface's input area with the graphical components required by the module to solicit user input. For instance, a module may necessitate the UI to show text labels representing input parameter names, accompanied by fields where users can supply the corresponding values.

The second one, `get_input_and_execute_task`, utilizes Dash's callback mechanisms to trigger the task's execution logic (i.e., the operations implemented within the module) once the user has provided all the required inputs. Once the task has been executed, the same method populates the UI output area with the necessary feedback for the user. For instance, the module may display a drop-down menu in the output area, allowing the user to select an enriched trajectory and subsequently plot it.

The third method, `get_dependencies`, provides the list of modules (in the form of type references) on whose output the module depends. The fourth method, `register_modules`, enables the module to register the instances of the modules on whose output it depends. The fifth method, i.e., `get_results`, makes the module's output accessible to other modules. Finally, the method `reset_state` resets the module's internal state.

b: CONFIGURABILITY

The principle of configurability ensures that users can easily define their semantic enrichment process by creating and customizing their MAT-building pipeline. The MAT-Builder backend implements this principle by exposing the `InteractivePipeline` component, which is a class part of the backend codebase. We explain the general characteristics of such component next.

The constructor of `InteractivePipeline` takes as input a list of type references, each representing a module that the user wants to employ within their MAT-building pipeline. The list implicitly specifies the execution order of the modules, and the logic within `InteractivePipeline` guarantees that any dependency between modules is satisfied. This is achieved at instantiation time by connecting each module's input to the output of the module(s) it depends on.

Moreover, an instance of `InteractivePipeline` directly manages and accesses the content of the UI areas by leveraging some of the six methods every module is required to implement. More specifically, the `InteractivePipeline` component must set up the initial state of the UI layout and then visualize the tabs in the tab selector area corresponding to the modules it must execute. Moreover, `InteractivePipeline` must ensure that the method within each module in charge of populating the input area of the UI, i.e., `populate_input_area`, is invoked when appropriate. Furthermore, the `InteractivePipeline` component must ensure that the task logic (i.e., `get_input_and_execute_task`) within a module is invoked when the user wants to execute it. Finally, this component supervises the execution of the MAT-building pipeline it represents – for instance, if some error occurs, `InteractivePipeline` must react with appropriate actions.

c: EXTENSIBILITY

The MAT-Builder backend incorporates the principle of extensibility by leveraging the object-oriented nature of Python and the modularity principle. Indeed, a user can add a new module (and thus operations) to the system by creating a class that extends `InteractiveModuleInterface` and contains the logic required to compute the task it addresses. Furthermore, users can alternatively provide new operations and functionalities by subclassing existing modules or by extending the capabilities provided by the backend codebase.

3) HOW TO CONFIGURE AND EXECUTE A MAT-BUILDING PIPELINE

Now that the components and inner workings of the MAT-Builder system have been introduced, we show how users can easily configure and execute their own interactive MAT-building pipeline with a few lines of code. Let us consider the slightly simplified Python code fragment shown below, which is the template of the *main* that a user can specialize to configure and execute their semantic enrichment process.

```
1 from dash import Dash
2 from backend import InteractivePipeline
3 from modules import m_1, ···, m_n
4
5 def main() :
6     UI_server = Dash()
7     pipeline = InteractivePipeline(UI_server,
        [m_1, ···, m_n])
8     UI_server.run()
```

The three initial lines (lines 1-3) import the classes needed to instantiate (1) a Dash UI server, (2) the `InteractivePipeline` class, and (3) the set of modules the user needs to instantiate their semantic enrichment process. Then, line 6 instantiates the user interface server, and line 7 instantiates an `InteractivePipeline` object. Here note that we pass to the `InteractivePipeline`'s constructor the reference to the UI server (required to manage the UI state) and a list of type references, i.e., the modules that will be used in the pipeline. The last line of code executes the UI server. As a result, the user can subsequently interact with the pipeline via a web browser. For all practical purposes, we emphasize that the only code the user needs to customize within the given fragment involves importing the necessary modules (line 3) and passing their type references to the `InteractivePipeline` instance (line 7).

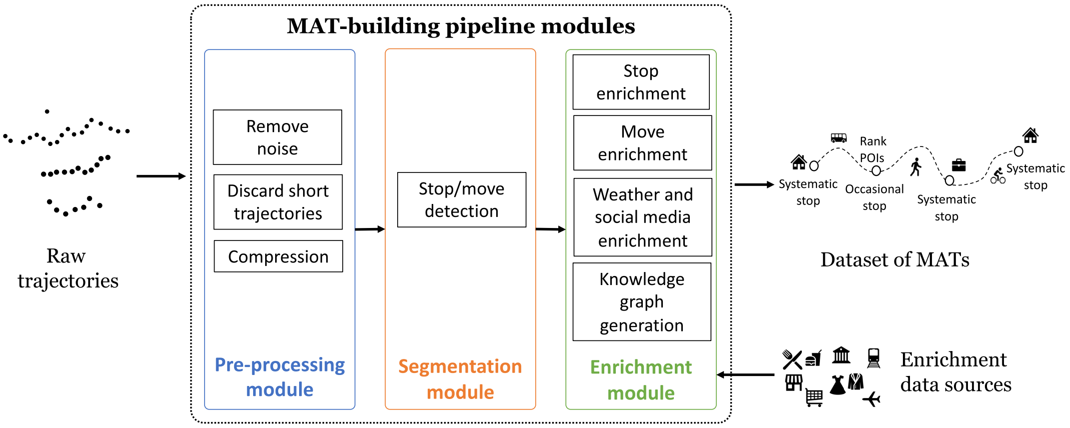


FIGURE 2. Our MAT-building pipeline instance.

We direct the reader to Section V for a concrete example of a MAT-building pipeline – the section includes screenshots showcasing the user interface, and provides a glimpse into the user's interaction with the pipeline.

a: CONSIDERATIONS ON COMPUTATIONAL COMPLEXITY

The computational complexity resulting from using MAT-Builder is dominated by the complexity of the algorithms used by the modules within the executed pipeline, as the system adds only a negligible overhead. As such, different pipelines are likely to exhibit varying performance profiles, reflecting the characteristics of their respective modules.

C. REPRESENTING DATASETS OF MULTIPLE ASPECT TRAJECTORIES

Knowledge graphs [3] represent a natural choice for MATs since they can support the representation of multitudes of aspects regardless of their heterogeneity and complexity. Moreover, knowledge graphs can be leveraged to conduct powerful analyses once they are imported in some triplestore of choice. Adopting a schema that gives proper structure to the information is one of the main problems when considering knowledge graphs for storage and querying. In the context of our work, we chose the STEPv2 ontology [16], which we lightly customized to suit this work's needs.[5] We refer to the original paper for the full details of the ontology. In the following, we provide a brief overview of the main customizations we made.

The first customization allows each instance of the *Agent* class (which represents a moving object in the ontology) to be related to instances of *Feature Of Interest*. This customization enables us to enrich with aspects not only the trajectories, but also the moving objects generating them. The second customization concerns the STEPv2 *Qualitative Description* class, which is a key class that allows trajectory segments to be enriched with aspect instances. In this work, we sub-classed *Qualitative Description* with several sub-classes to support the four aspects considered by the MAT-building pipeline described in Section V. Such aspects are: *regularity*, *move*, *social media posts*, and *weather conditions*. Extensive details on the second customization are provided in Section V.

V. A PRACTICAL MAT-BUILDING PIPELINE

In this section, we present a concrete MAT-building pipeline that implements the semantic enrichment process proposed in [20]. This serves as an example of how MAT-Builder effectively addresses the problem outlined in Section II, Definition 7.

The process consists of three steps, i.e., *trajectory pre-processing*, *trajectory segmentation* and *trajectory enrichment*. The steps are executed sequentially, therefore, the underlying operations are encapsulated into three distinct modules. In the following, we provide the main details of the modules' implementation and describe how they are connected to each other. We also report that the modules make extensive use of several functionalities available from Python's vast ecosystem of libraries. Among those used (via import statements), we report Pandas[6] and Geopandas,[7] plus a few others which will be mentioned later. Figure 2 shows an overview of the MAT-building pipeline, with the underlying information flowing across its modules.

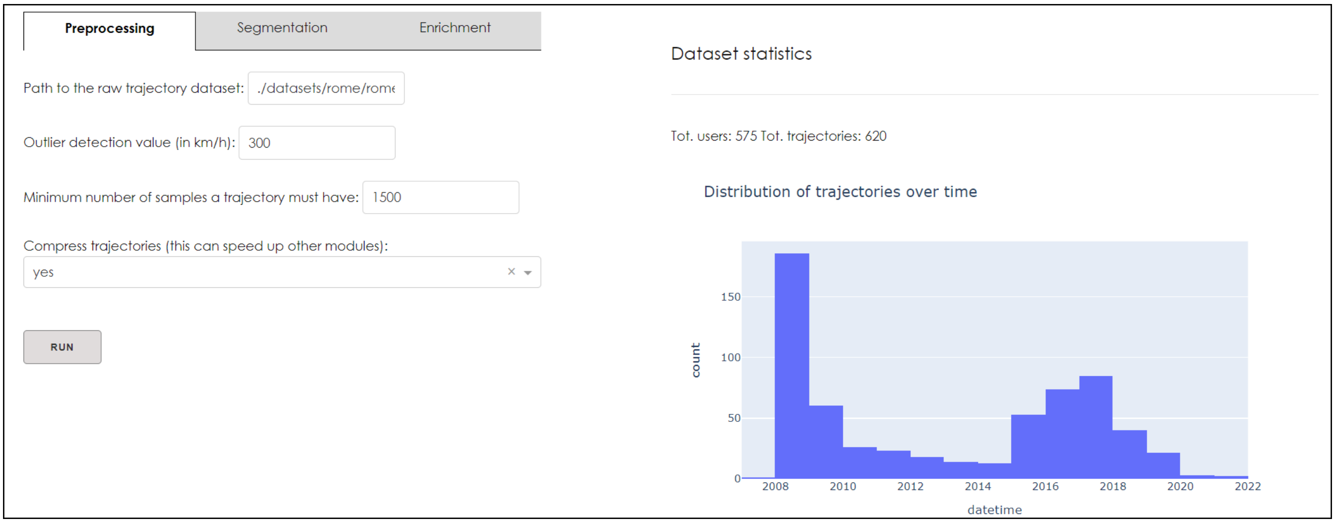
The **trajectory pre-processing** module (blue block in Figure 2) takes as input a set of raw trajectories and filters out noisy or unusable data to facilitate the activities of the other modules. Specifically, it can discard trajectories with insufficient sampling rate, filter out anomalous samples (i.e., those with unreasonable speeds), and compress trajectories. All the above operations correspond to functionalities imported from the scikit-mobility library [21]. A screenshot of the module being used via the system's UI is shown in Figure 3.

We highlight that the use of the pre-processing module is optional, as a dataset of pre-processed trajectories might

[5] The customized version of the STEPv2 ontology is provided in the MAT-Builder's GitHub repository.

[6] https://pandas.pydata.org/

[7] https://geopandas.org/



Path to the raw trajectory dataset: ./datasets/rome/rome

Outlier detection value (in km/h): 300

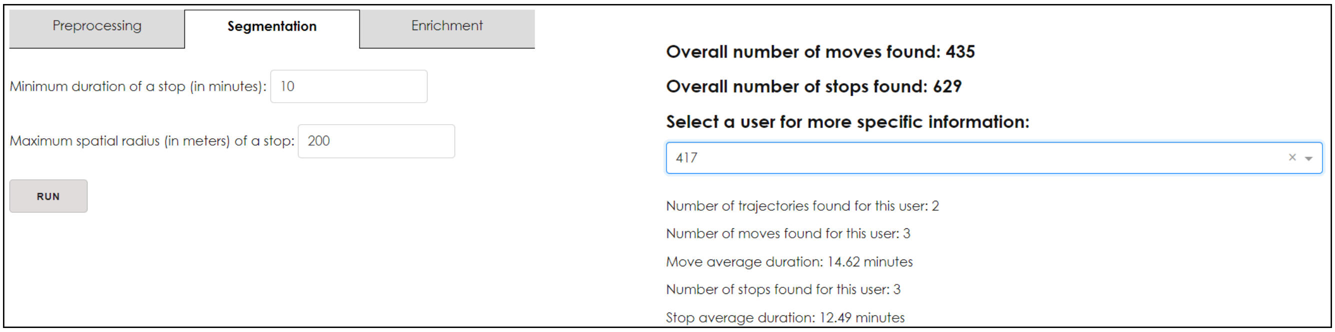
Minimum number of samples a trajectory must have: 1500

Compress trajectories (this can speed up other modules):
yes

Dataset statistics

Tot. users: 575 Tot. trajectories: 620

FIGURE 3. The preprocessing module, as shown in the MAT-Builder UI.



Minimum duration of a stop (in minutes): 10

Maximum spatial radius (in meters) of a stop: 200

Overall number of moves found: 435

Overall number of stops found: 629

Select a user for more specific information:
417

Number of trajectories found for this user: 2

Number of moves found for this user: 3

Move average duration: 14.62 minutes

Number of stops found for this user: 3

Stop average duration: 12.49 minutes

FIGURE 4. The segmentation module, as shown in the MAT-Builder UI.

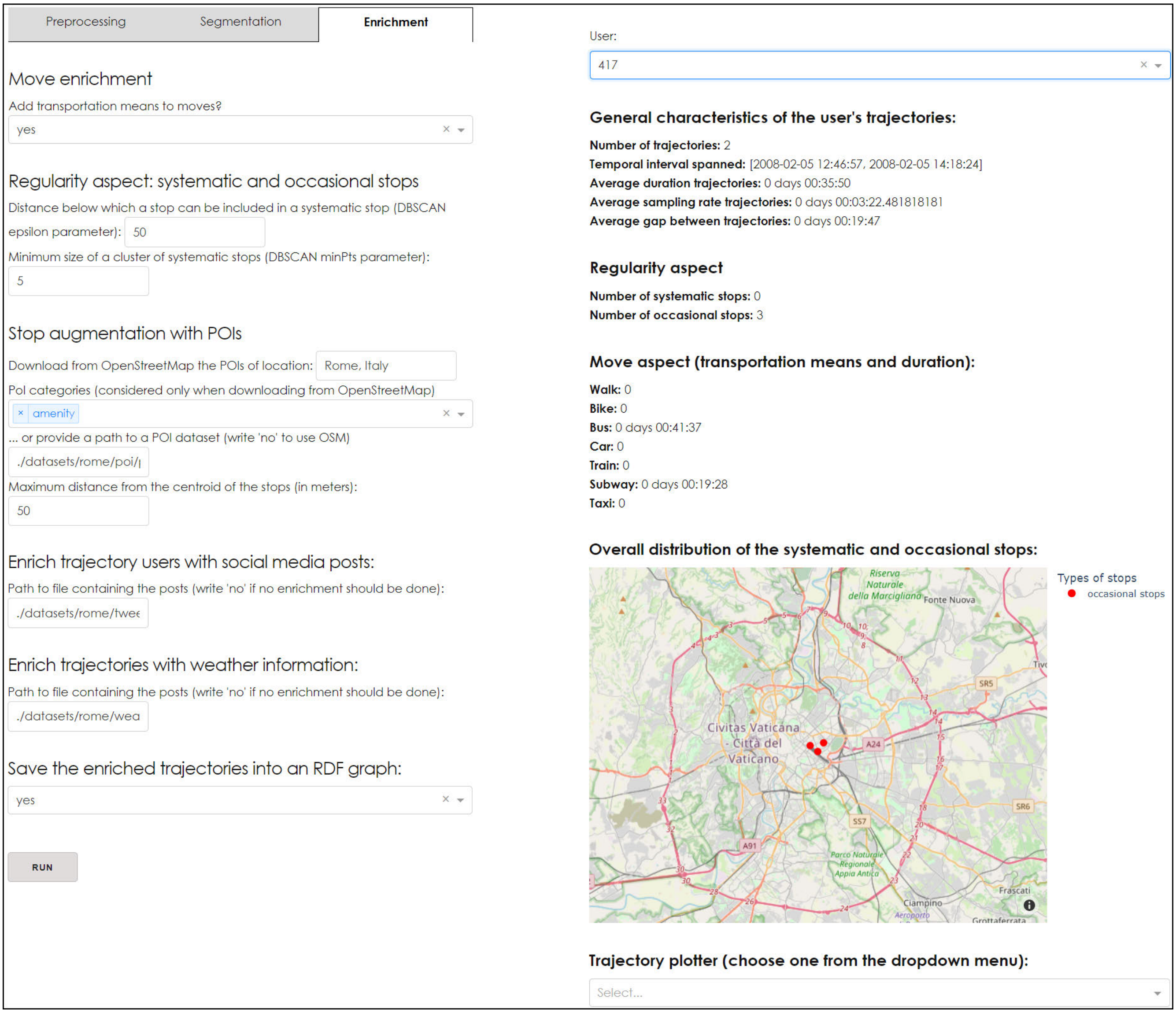
already available. In such a case, the user can take advantage of the modularity and configurability of MAT-Builder, and simply omit the pre-processing module when instantiating the `InteractivePipeline` object (see also Section IV-B3).

The **trajectory segmentation** module (orange block in Figure 2) takes as input a dataset of trajectories, and partitions every trajectory into sub-trajectories (or *segments*). The segmentation algorithm used by the module employs the *stop* and *move* criterion [4], and it is imported from the scikit-mobility library. A screenshot of this module being used via the system's UI is shown in Figure 4.

The **enrichment** module (green block in Figure 2) takes the output of the segmentation module and identifies the different segments to enrich, the aspects to consider, the datasets to be used to enrich the segments, and the enrichment criteria. The module is composed of five steps, each corresponding to a set of internal methods that implement the necessary operations: one dealing with the enrichment of stop segments, another with the enrichment of move segments, then two that enrich segments and trajectory users respectively with weather and social media information. The final step generates RDF knowledge graphs containing the final dataset of MATs. A screenshot of the enrichment module being used in the system's UI is shown in Figure 5.

The **stop enrichment** step first enriches the stop segments with the *regularity* aspect. More precisely, each stop segment is either considered a *systematic stop*, i.e., part of a cluster of stop segments that are located within a limited geographical area (and that very likely present some kind of temporal regularity), or an *occasional stop*, which is a stop segment that does not belong to any cluster. The regularity aspect is important, because it can provide valuable information on the long term behaviors of an individual. Systematic stops typically occur when an individual consistently stays at their home, workplace, or any other place that is part of their routines, while occasional stops describe a more irregular behavior. (e.g., the individual is on a leisure trip and is visiting some attractions).

The stop enrichment step differentiates between systematic and occasional stops via the scikit-learn's implementation of DBSCAN [22], a well-known density-based clustering algorithm: stop segments that belong to a cluster are considered systematic, while those that do not are considered occasional.



Preprocessing | Segmentation | Enrichment

Move enrichment

Add transportation means to moves?

yes

Regularity aspect: systematic and occasional stops

Distance below which a stop can be included in a systematic stop (DBSCAN epsilon parameter): 50

Minimum size of a cluster of systematic stops (DBSCAN minPts parameter):

5

Stop augmentation with POIs

Download from OpenStreetMap the POIs of location: Rome, Italy

PoI categories (considered only when downloading from OpenStreetMap)

amenity

... or provide a path to a POI dataset (write 'no' to use OSM)

./datasets/rome/poi/

Maximum distance from the centroid of the stops (in meters):

50

Enrich trajectory users with social media posts:

Path to file containing the posts (write 'no' if no enrichment should be done):

./datasets/rome/twee

Enrich trajectories with weather information:

Path to file containing the posts (write 'no' if no enrichment should be done):

./datasets/rome/wea

Save the enriched trajectories into an RDF graph:

yes

RUN

User:

417

General characteristics of the user's trajectories:

Number of trajectories: 2

Temporal interval spanned: [2008-02-05 12:46:57, 2008-02-05 14:18:24]

Average duration trajectories: 0 days 00:35:50

Average sampling rate trajectories: 0 days 00:03:22.481818181

Average gap between trajectories: 0 days 00:19:47

Regularity aspect

Number of systematic stops: 0

Number of occasional stops: 3

Move aspect (transportation means and duration):

Walk: 0

Bike: 0

Bus: 0 days 00:41:37

Car: 0

Train: 0

Subway: 0 days 00:19:28

Taxi: 0

Overall distribution of the systematic and occasional stops:

Trajectory plotter (choose one from the dropdown menu):

Select...

FIGURE 5. The enrichment module, as shown in the MAT-Builder UI.

Each systematic stop is then augmented with the *activity* that has been likely performed. The stop enrichment step currently contemplates three activities: *home*, *work*, and *other*. Estimating this information requires looking at the cluster containing the systematic stop – more precisely, it requires analysing how the cluster's systematic stops are temporally distributed. First, the stop enrichment step determines how many hours an individual has spent in each of the clusters of systematic stops that have been found: stop segments belonging to the two clusters in which the individual has spent the majority of their time are associated with the activities *home* or *work*, while the remaining ones are associated with the *other* activity. The underlying reasoning is that an individual spends most of their time either at home or work. Next, the step proceeds by associating the systematic stops belonging to the two temporally largest clusters either to the *home* or *work* activity. The *home* activity label is associated with the systematic stops belonging to clusters whose stops tend to occur outside working hours, i.e., late evenings, nights, early mornings, or weekends. On the contrary, the *work* activity label enriches the systematic stops that belong to clusters whose stops tend to occur during working hours, i.e., mornings and afternoons occurring during the weekdays.

Finally, both systematic and occasional stops are augmented with POIs: this is done by ranking the POIs by distance from the stops' centroids and then associating to each stop the top-k ones. We report that the stop enrichment step can retrieve POIs having any kind of geometric shape, either from OpenStreetMap (by importing and using the appropriate functionalities from the OSMnx library [23]) or from a dataset provided via a local file.

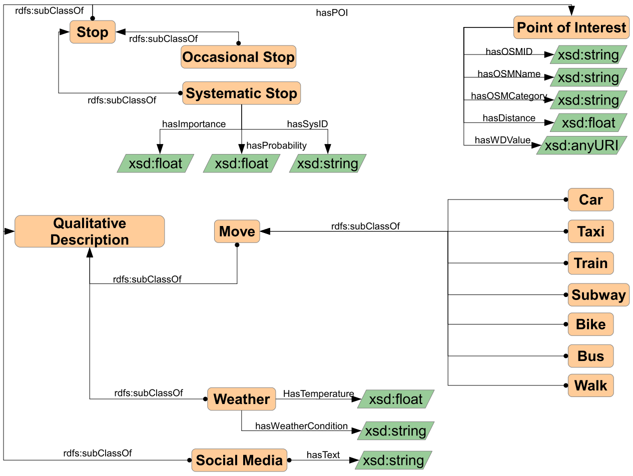


FIGURE 6. Classes, predicates, and properties that have been added to the STEPv2 ontology to model the four considered aspects.

The **move enrichment** step focuses on the move segments and the move aspect. In its present version, this step augments the move segments with two information, i.e., quantitative numerical measures and transportation means estimation. Numerical measures include maximum and average speed, acceleration, bearing rate, and total length, which are computed by importing some of the functionalities provided by the PTrail library [24]. Transportation means are estimated as the ones that have been likely used during each move segment. The module does the estimation via a random-forest classifier (created with the scikit-learn library [25]) that has been trained on the GeoLife dataset [26] with the classes *walk*, *car*, *bike*, *bus*, *subway*, and *train*.

The **weather** and **social media enrichment** steps enrich, respectively, segments and moving objects with the *weather* and *social media* aspects. The weather step takes as input a dataset containing historical weather data and enriches each trajectory segment with the weather information related to the time period and geographical area spanned by the segment. The social media step takes as input a dataset containing social media posts, whereby each post is associated with a user identifier (here we assume the identifiers of the trajectory users), a date of publication, and the text of the post, and enriches the moving objects directly.

The **knowledge graph generation** step finally stores datasets of MATs in RDF graphs according to the schema defined by the customized STEPv2 ontology (see also Section IV-C). Section IV-C summarily described how the STEPv2 ontology has been customized to suit the pipeline's needs. In the following, we focus on the subclasses of *Qualitative Description* that have been introduced to model the four considered aspects. Figure 6 provides an overview.

The first subclass is *Stop*, which provides a generic model for stop segments. An instance of *Stop* can be in relationship with one or more instances of *Point of Interest* via the *hasPOI* predicate – this allows to augment stop segments with POIs. Each instance of *Point of Interest* has, in turn, several data properties, i.e., the POI OpenStreetMap identifier *hasOSMID*, the POI category *hasOSMCategory*, the POI name *hasOSMName*, the POI WikiData identifier (if any) *hasWDValue*, and the distance *hasDistance* between the POI and the *Stop* instance it is associated with.

The *Stop* class is then further subclassed by the classes *Occasional Stop* and *Systematic Stop*: these model the *regularity* aspect. *Occasional Stop* does not have additional properties than the *Stop* class, while the *Systematic Stop* class has a few data properties and is further subclassed by three other classes, i.e., *Home*, *Work*, and *Other*. The three subclasses enable the augmentation of instances of systematic stops with the activity. The data properties of *Systematic Stop* are: *hasSysID*, which represents a cluster's identifier, *hasImportance*, which indicates the percentage of time the individual has spent in the cluster, and *hasProbability*, which measures the certainty that the estimated activity is correct.

The class *Move* models move segments and, more in general, the move aspect and has the subclasses: *Bike*, *Car*, *Train*, *Subway*, *Bus*, *Taxi*, and *Walk*.

The class *Weather* models the weather aspect. Each instance of this class has two data properties, i.e., *hasTemperature*, which reports the measured temperature, and *hasWeatherConditions*, which indicates the weather conditions.

Finally, the class *Social Media* models the social media aspect. It has a single data property, i.e., *hasText*, which provides the text associated with an instance of the class.

VI. EXAMPLE SCENARIOS

In this section, we present a qualitative empirical evaluation of our methodology, providing two example scenarios in which we apply the MAT-building pipeline described in Section V. Initially, the pipeline is employed to generate an RDF knowledge graph containing a MAT dataset. This dataset is subsequently imported into a triplestore, enabling in-depth analyses of selected individuals' mobility behaviors. Overall, the evaluation aims to highlight the utility of the MAT datasets generated by MAT-Builder and demonstrate our methodology's potential and versatility.

A. FIRST EXAMPLE SCENARIO: THE TOURIST

In the first scenario, we consider a dataset of publicly available trajectories retrieved from OpenStreetMap (OSM) and a few external semantic data sources used to gather information concerning the aspects.

1) BACKGROUND AND PREPARATORY STEPS

The trajectories cover the area of the province of Rome, Italy. The dataset has been downloaded from OSM and contains 26395 trajectories from 3181 distinct individuals, spanning a time interval between March 2007 and July 2021. Table 2 reports the main characteristics of the dataset. Out of the 3181 individuals, only 6 of them had data spanning more than 4 weeks. Additionally, data covering more than 1 week was available for only 13 individuals. These results indicate that the majority of individuals did not provide enough

TABLE 2. Metadata for the OpenStreetMap Rome dataset. The symbol *D* stands for day, *h* for hour, *m* for minutes, and *s* for seconds.

Individuals		3181
Trajectories		26395
Individuals with more than 4 weeks of data		6
Individuals with more than 1 week of data		13
Individuals with more than 1 day of data		356
	mean	**std.dev.**
Temporal interval spanned by an individual	1D 19h 18m	60D 7h 49m
Trajectories per individual	8.3	9.32
Samples per trajectory	513.33	722.82
Sampling rate trajectories	10 m 45s	6h 31m 10s
Duration trajectories	1D 8h 44m	60D 7h 40m
Gap between trajectories of an individual	1h 13m 6s	8h 56m 11s

information to enable the analysis of their movement behaviors over extended periods of time. This also explains the very large standard deviation observed with the trajectory duration. We also observe large variations for what concerns the trajectory sampling rate and the gaps between trajectories at the user level.

We employed two external data sources for semantic enrichment: OpenStreetMap and Meteostat.[8] We accessed OSM to construct a dataset consisting of 28787 POIs, which we used to augment the stop segments. We accessed Meteostat to generate a dataset of historical weather information. Finally, the social media post dataset is synthetic and contains simulated Twitter posts.

To produce a dataset of MATs, we use the pipeline introduced in Section V via the MAT-Builder user interface. We execute the pipeline's modules sequentially, in their intended order of use. In the pre-processing module, we set the minimum number of samples to *1500*, the maximum speed threshold to *300 km/h*, and enable trajectory compression. This yields a set of 620 pre-processed trajectories from 575 users. In the segmentation module, we set the minimum duration of a stop to *10 minutes*, while the maximum spatial radius a stop can have is set to *0.2 km*. This yields a set of 629 stops and 435 moves. Finally, the enrichment module enriches the segmented trajectories with all the aspects supported by the module, i.e., regularity, move (with transportation means estimation), weather, and social media.

The systematic and occasional stop detection is done via the DBSCAN clustering algorithm and is guided by two parameters. The first parameter, DBSCAN *epsilon*, sets the maximum distance within which two stops are deemed neighbors. We empirically fix this value at *50 meters*. The second parameter, DBSCAN *minPts*, sets the minimum count of neighboring stops needed for a stop to being tagged as a *core* point, thereby forming an initial cluster. When identifying systematic stops, this parameter specifies the minimum quantity of stops required to form a cluster, and we empirically set this value to 5. Both occasional and systematic stops are augmented with POIs located within *50 meters* of their centroids.

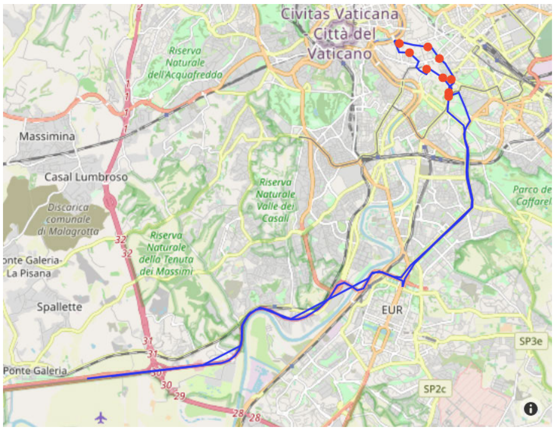


FIGURE 7. Plot of the MAT from MAT-Builder's user interface. The *red dots* represent *occasional stops*, while the *blue curve* represents the *moves*.

The MAT-building pipeline ultimately generates a dataset of enriched MATs, which is then stored in a RDF knowledge graph and finally imported in the GraphDB[9] triplestore. We utilize the triplestore to conduct various analyses using the SPARQL 1.1 query language.[10] The details and results of these analyses are presented next.

2) MOVEMENT BEHAVIOR ANALYSIS

From the dataset of MATs, we selected an individual (ID 2115) who produced a trajectory (ID 2652) that originates close to the Fiumicino Rome Airport in the early morning, then spends half of the day within the centre of Rome, and then goes back to the same airport in the early afternoon (Figure 7). The overall duration of the trajectory is around 6 hours. All such evidence could hint that the individual is some kind of tourist passing by the city. Further analyses of the individual's mobility behaviours are, however, required to reach any conclusion. Accordingly, we want to find out (1) which transportation means the individual has likely used during their trip, (2) the POIs the individual may have visited while staying in Rome, and (3) the weather conditions and social media posts related to their trip.

Before delving into the analyses, we briefly introduce the RDF namespaces repeatedly used in the queries. The namespaces refer to vocabularies provided by well-known ontologies, i.e., the RDF concepts vocabulary (*rdf*), the RDF schema vocabulary (*rdfs*), the Friend of a Friend (*foaf*) vocabulary, the Time (*time*) vocabulary, and the XML Schema representation vocabulary (*xsd*). The remaining two namespaces refer to the vocabulary provided by the customized STEPv2 ontology (*step*) and the vocabulary provided by the GraphDB's standard math functions extension (*ofn*), which enables the use of additional mathematical functions (some usefully dealing with data in the temporal domain).

We begin our analysis by first identifying the possible transportation means used by the individual through the

[8] https://meteostat.net/

[9] https://graphdb.ontotext.com/

[10] https://www.w3.org/TR/rdf-sparql-query/

```
1  SELECT ?type_move ?t_start ?t_end
   (ofn:asMinutes(?t_end - ?t_start) AS
   ?duration_mins)
2  WHILE
3  {
4    ?traj ˆstep:hasTrajectory / foaf:name “2115” ;
5          step:hasID “2652” ;
6          step:hasFeature ?feat.
7
8    ?feat step:hasEpisode ?ep.
9    ?ep step:hasSemanticDescription ?move ;
10       step:hasExtent ?ex.
11
12   ?move rdfs:subClassOf step:Move ;
13         rdf:type ?type_move.
14
15   ?ex step:hasStartingPoint / step:atTime /
      time:inXSDDateTime ?t_start ;
16       step:hasEndingPoint / step:atTime /
      time:inXSDDateTime ?t_end.
17 }
18 ORDER BY ASC(?t_start)
```

```
1  SELECT ?t_start ?t_end (ofn:asMinutes(?t_end -
   ?t_start) AS ?duration) ?poi_name ?poi_category
2  WHERE
3  {
4    ?traj ˆstep:hasTrajectory / foaf:name “2115” ;
5          step:hasID “2652” ;
6          step:hasFeature ?feat.
7    ?feat step:hasEpisode ?ep.
8
9    ?ep step:hasSemanticDescription ?stop.
10   ?stop rdf:type step:OccasionalStop.
11
12   ?ep step:hasExtent / step:hasStartingPoint /
      step:atTime / time:inXSDDateTime ?t_start ;
13        step:hasExtent / step:hasEndingPoint /
      step:atTime / time:inXSDDateTime ?t_end .
14
15   ?stop step:hasPOI ?poi.
16   ?poi step:hasOSMCategory ?poi_category ;
17        step:hasOSMName ?poi_name ;
18        step:hasOSMName ?poi_distance.
19 } ORDER BY ?t_start ?poi_distance
```

execution of the SPARQL query shown above. The query first finds out the trajectory of interest (lines 4-5), then retrieves all its aspect instances (lines 6-10), and finally filters out those that are not of the *move* aspect (line 12). The query then retrieves for each instance of the move aspect the estimated transportation means (line 13), and determines its starting and ending instants (lines 15-16). The SELECT finally returns a list of tuples, each representing a move instance with the estimated transportation means, its starting and ending instants, and its duration. From the results, we report that the query finds 9 move instances. By looking at them, we report that the individual appears to go from the airport to Rome's city centre by train, then mostly walked and used buses while moving in the city, and finally went back to the airport by bus.

Next, we want to find out the POIs the individual has possibly visited during their trip. Accordingly, the query presented next focuses on the regularity aspect – more specifically, on the individual's occasional stops. The query first finds out the trajectory of interest and keeps only the instances of *occasional stops* (lines 4-10). Successively, the query gathers information concerning the instances that have at least one POI and finally retrieves the names and categories of the POIs involved (lines 15-18). From the results, we report that the query finds 11 occasional stop instances.

By looking at the associated POIs, we report that the individual appears to have spent a good part of the morning visiting various monuments: the individual briefly stayed in the area surrounding the *Palatino* and then went to the *Tempio della Pace*. The individual then spent more than an hour in the area surrounding the *Altare della Patria* and then stayed nearby the *Pantheon* for around half an hour until lunchtime. After that, the individual appears to have dined at a restaurant for almost one hour. Finally, the individual appears to have stayed again in the vicinity of the *Palatino* and then went back to the airport. Overall, the individual has been repeatedly observed nearby famous monuments and appears to have walked and used public transportation means, thus reinforcing the initial impression that they were indeed a tourist.

Once the POIs the individual has most likely visited have been found, we might want to gather further information on their characteristics, especially if we consider that such POIs are known to be attractions of historical relevance. For instance, we might be interested to find out a photo of them, when they were built, what is their architectural style, if there is an entrance fee, and if they have a website or telephone number. Some of this information might not be available from the OSM POI dataset, thus requiring to access some other external semantic data source. We highlight that the POIs gathered from OSM come with a WikiData[11] identifier if a corresponding entity is present in its knowledge graph. WikiData is a Linked Open Data source whose knowledge graph provides large amounts of open information on real-world entities, each associated with an identifier and a set of properties. WikiData also conveniently exposes a SPARQL endpoint which can be used within *federated* SPARQL queries to access its knowledge graph content.

Let us, therefore, consider the previous query and turn it into a *federated* query to augment with further information the

[11] https://www.wikidata.org/

```
 1 OPTIONAL
 2 {
 3   ?poi step:hasWDValue ?WD.
 4   SERVICE <https://query.wikidata.org/sparql>
 5   {
 6     OPTIONAL ?WD wdt:P18 ?img_WD.
 7     OPTIONAL ?WD wdt:P2555 ?fee_WD.
 8     OPTIONAL ?WD wdt:P571 ?year_built_WD.
 9     OPTIONAL ?WD wdt:P856 ?url_WD.
10     OPTIONAL ?WD wdt:P1329 ?phone_WD.
11     OPTIONAL
12     {
13       ?WD wdt:P149 / rdfs:label ?style_WD.
14       FILTER(lang(?style_WD) = "en")
15     }
16   }
17 }
```

POIs that have been found. To this end, we insert the SPARQL fragment above, right before the end of the query's WHERE clause. The fragment is enclosed within the OPTIONAL graph pattern, which accounts for the possibility that certain POIs might not have a corresponding WikiData identifier. Subsequently, after establishing a connection with the WikiData SPARQL service (line 4), the fragment attempts to retrieve several pieces of information by using the appropriate WikiData property identifiers (note again the repeated use of OPTIONAL, lines 6–15). These include the URL of an image, entrance fee, year of construction, website URL, phone number, and architectural style (in English). Such information are finally stored in variables that can be used in the final SELECT statement.

In the final part of the running analysis, we aim to find out the weather conditions and the social media posts that the individual has respectively experienced and published during the trip. Let us focus on the weather conditions, as the strategy for the other aspect is similar. To this end, we can insert in either of the two queries shown before, right before the end of the WHILE loop, the SPARQL fragment presented next. The OPTIONAL keyword serves the purpose of not filtering out from the final results the segments for which no weather information is available. The FILTER keyword ensures that each segment gets associated with an instance of the weather aspect only if they have a non-empty temporal overlap. Finally, the *weather_conditions* variable can be integrated into the SELECT to report the weather conditions. All in all, we report that the individual experienced a sunny day during their trip.

```
OPTIONAL
{
  ?traj step:hasFeature / step:hasEpisode ?ep_w.
  ?ep_w step:hasSemanticDescription / rdf:type step:Weather ;
        step:hasWeatherCondition ?weather_conditions ;

        step:hasExtent / step:hasStartingPoint / step:atTime / time:inXSDDateTime ?tw_start ;
        step:hasExtent / step:hasEndingPoint / step:atTime / time:inXSDDateTime ?tw_end.

  FILTER((?t_start <= ?tw_end) && (?tw_start <= ?t_end))
}
```

B. SECOND EXAMPLE SCENARIO: THE UNIVERSITY STUDENT

In the second example scenario, our shift focuses on examining an individual's long-term mobility behaviors within an urban setting. Specifically, we aim to analyse their daily routines over extended time periods, seeking to understand where and when they consistently spent their free time and working hours. To this end, we consider a different dataset, GeoLife [26].

1) BACKGROUND AND PREPARATORY STEPS

The Geolife dataset contains 17621 trajectories from 178 distinct individuals and spans a time interval between April 2007 and August 2012, thus providing a far larger volume of movement data per individual than the OSM dataset. Table 3 shows the main characteristics of the dataset. The Geolife dataset complements the OpenStreetMap dataset used in the first scenario by providing substantially longer trajectories from a smaller group of individuals, making it ideal for studying individuals' long-term movement habits. Moreover, the Geolife dataset mainly consists of students' trajectories living in Beijing, while the OSM dataset includes trajectories from a very diverse group of individuals. In this scenario, we continue to use the same external semantic data sources considered in the first example scenario: OSM for POIs, Meteostat for weather conditions, and a synthetic dataset for social media posts.

TABLE 3. Metadata for the GeoLife dataset. The symbol *D* stands for day, *h* for hour, *m* for minutes, and *s* for seconds.

Individuals	178
Trajectories	17621
Individuals with more than 4 weeks of data	118
Individuals with more than 1 week of data	155
Individuals with more than 1 day of data	174

	mean	std.dev.
Temporal interval spanned by an individual	188D	438D 2h 52m
Trajectories per individual	102.58	250.06
Samples per trajectory	1817.97	2359.53
Sampling rate trajectories	48s	1m 22s
Duration trajectories	3h 44m 17s	2h 52m 40s
Gap between trajectories of an individual	3D 17h 45m	8D 22h 17m

To produce a dataset of MATs, we use the MAT-building pipeline via the MAT-Builder user interface and execute its modules in the same order used in the first example scenario. In the pre-processing module, we set the minimum number of samples to *1500*, the maximum speed threshold to *300 km/h*, and enable trajectory compression. This yields a set of 4331 pre-processed trajectories from 132 individuals. In the segmentation module, we set the minimum duration of a stop to *120 minutes*, while the maximum spatial radius a stop can have is empirically set to *0.2 km*. Observe that the minimum stop duration used in this scenario is larger than that used in the first one, since the goal is to focus on the systematic stops. This yields a set of 3355 stops and 3314 moves.

Finally, the enrichment module enriches the trajectories with all four aspects supported by the module, and the move occurrences have been augmented with the estimated transportation means. The distinction between systematic and occasional stop segments, which we recall concerns the regularity aspect, has been conducted by setting the DBSCAN *epsilon* parameter to *50* meters and the DBSCAN *minPts* parameter to *5*. Systematic stops have been further augmented with the *activity* information, according to the criteria outlined in Section V. Both occasional and systematic stops are augmented with POIs found to be less than *50* meters far from their centroids. Finally, trajectories have been enriched with the weather aspect, while moving objects have been enriched with the social media aspect.

The MAT-building pipeline ultimately yields a dataset of enriched MATs, which is then stored in a RDF knowledge graph and finally imported and analysed in the GraphDB triplestore.

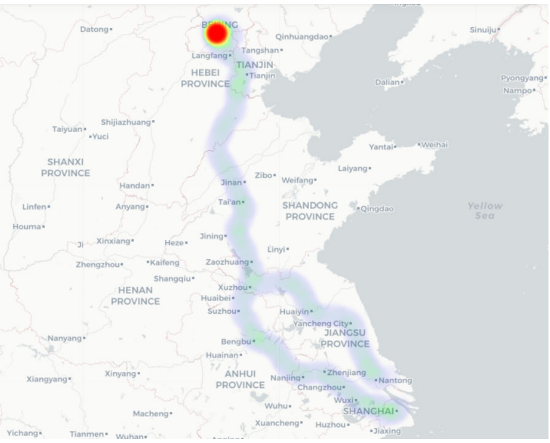


FIGURE 8. Heatmap of the individual's MATs locations.

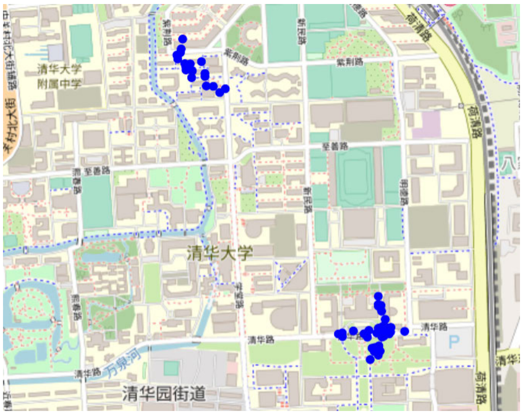


FIGURE 9. Distribution of the individual's systematic stops (blue circles). The systematic stops are located at the Tsinghua University's premises.

2) MOVEMENT BEHAVIOR ANALYSIS

The analysis aims to extract the long-term mobility behaviours of a selected individual. Accordingly, the analysis will focus on the individual's systematic stops. From the dataset, we selected an individual (ID 3) with 110 MATs. Such MATs span a time period between October 2008 and July 2009 for a total of 253 days and a half, therefore providing several months' worth of enriched mobility data. The average duration of the MATs is 12 hours and 11 minutes, their average sampling rate is below 372 seconds (although we report considerable variations among the MATs), and the average gap between the MATs is 43 hours: we argue that all these characteristics enable to properly reason on the individual's systematic stops (and thus on their long-term habits).

From the heatmap shown in Figure 8, observe that the vast majority of the individual's positions appear to be within the city of Beijing (red spot in the Figure), with a few others observed within the cities of Shanghai and Nantong (green shades in the Figure), both located in China. This suggests that the individual consistently lives in Beijing. For what concerns the individual's 182 stop segments, the enrichment module determines that 105 of them are occasional and 77 systematic. The systematic ones are all located in Beijing, and appear to be concentrated in two clusters, both located within Tsinghua University's premises. The two clusters of systematic stops are shown in Figure 9. In particular, the cluster on the upper part of the Figure consists of systematic stops located within a complex of dormitories, while those belonging to the cluster on the bottom-right part are located very close to the university's Department of Computer Science and Technology.

The locations of the two clusters suggest that the individual might be a student who spends their free time in the dormitory complex and spends their working hours attending classes or engaging in academic activities. To corroborate such conjecture, in the following we aim to (1) look at the *activity* information augmenting the systematic stops, therefore verifying when the individual happens to be located in the two clusters, and (2) analyze the POIs that have been associated with the systematic stops.

The SPARQL query presented next focuses on the first problem. The query is designed to retrieve information regarding the activities that enrich the systematic stops detected within all trajectories of the individual (lines 4–10).

```
1  SELECT ?sys_cluster_ID
   (SAMPLE(?sys_cluster_activity) AS
   ?cluster_activity) (COUNT(?sys_cluster_ID) AS
   ?num_stops_cluster) (SAMPLE(?sys_importance)
   * 100 AS ?cluster_importance)
   (SAMPLE(?sys_probability) * 100 AS
   ?correctness_activity) (AVG(ofn:asHours(?t_end -
   ?t_start)) AS ?average_duration_hrs)
   (SUM(ofn:asHours(?t_end - ?t_start)) AS
   ?sum_duration_hrs)
2  WHERE
3  {
4     ?traj ^step:hasTrajectory / foaf:name “3” ;
5            step:hasID ?traj_id.
6     ?traj step:hasFeature / step:hasEpisode ?ep.
7     ?ep step:hasSemanticDescription ?sys_stop.
8
9     ?sys_stop rdf:type ?sys_cluster_activity.
10    ?sys_cluster_activity rdfs:subClassOf
       step:SystematicStop.
11
12    ?sys_stop step:hasImportance ?sys_importance ;
13                step:hasSysID ?sys_cluster_ID ;
14                step:hasProbability ?sys_probability.
15
16    ?ep step:hasExtent / step:hasStartingPoint /
       step:atTime / time:inXSDDateTime ?t_start ;
17          step:hasExtent / step:hasEndingPoint /
       step:atTime / time:inXSDDateTime ?t_end.
18 }
19 GROUP BY sys_cluster_ID
```

It provides access to several key pieces of information, including the percentage of time spent by the individual within a particular cluster of systematic stops relative to all clusters (referred to as *sys_importance*) and the probability that the estimated activity is accurate (referred to as *sys_probability*, lines 12–14). Finally, the query computes the average (*average_duration_hrs*) and total (*sum_duration_hrs*) duration in hours that the individual has spent within the stops of each cluster.

We report that the query reveals the presence of two clusters, as expected from what has been shown in Figure 9. One cluster is associated with the activity *Home*, primarily located in the upper part of the figure, while the other cluster is associated with the activity *Work*. Within the *Home* cluster, the individual has spent approximately 31% of their total time, which is equivalent to 79 hours. This cluster contains a total of 21 systematic stops, with an average duration of 3.76 hours per stop. Conversely, the individual has spent around 69% of their time in the *Work* cluster, amounting to 164 hours. This cluster includes 56 systematic stops, with an average duration of 2.92 hours per stop. Overall, our analysis indicates that the individual has primarily spent their time, especially during working hours, in close proximity to the Department of Computer Science and Technology while spending their free time close to the dormitory complex.

```
1  SELECT ?sys_type ?poi_id (COUNT(?poi_id) AS
   ?poi_count) (SAMPLE(?poi_name) AS
   ?poi_name) (AVG(ofn:asHours(?t_end - ?t_start))
   AS ?average_duration_hrs) (AVG(?poi_distance)
   AS ?average_distance)
2  WHERE
3  {
4     ?traj ^step:hasTrajectory / foaf:name “3” ;
5            step:hasID ?traj_id.
6     ?traj step:hasFeature / step:hasEpisode ?ep.
7     ?ep step:hasSemanticDescription ?sys_stop.
8
9     ?sys_stop rdf:type ?sys_type.
10    ?sys_type rdfs:subClassOf step:SystematicStop.
11
12    ?sys_stop step:hasPOI ?poi.
13    ?poi step:hasOSMValue ?poi_id ;
14          step:hasOSMName ?poi_name ;
15          step:hasDistance ?poi_distance.
16
17    ?ep step:hasExtent / step:hasStartingPoint /
       step:atTime / time:inXSDDateTime ?t_start ;
18          step:hasExtent / step:hasEndingPoint /
       step:atTime / time:inXSDDateTime ?t_end.
19 }
20 GROUP BY ?sys_type ?poi_id
21 ORDER BY DESC(?poi_count)
```

The second problem requires identifying the POIs that augment the individual's systematic stops. To this end, consider the query presented next. The query first finds out the systematic stops and the associated activity (lines 7–10). Then, for each POI augmenting a systematic stop, the query retrieves its OSM identifier, name, and distance from the stop centroid (lines 12 – 15). The query subsequently GROUP BY the results according to the systematic stop activities and POI identifiers. This yields a set of *(activity, POI)* pairs, for each of which the query provides, via the final SELECT, the associated activity, POI identifier, frequency, name, average duration, and the average distance from the stop centroid. The pairs are finally ORDER(ed) BY their frequency.

From the results, we report that systematic stops augmented with the *Home* activity have been associated 12 times with the dormitory n.25 and 12 times with the dormitory n.26, while they have been associated 6 times with the dormitory n.24. Such stops have also been found having an average duration between 3.7 and 4.2 hours, and their centroids were found to be at an average distance from the aforementioned POIs comprised between 16 and 25 meters. Overall, the findings suggest that the individual has consistently spent a relevant part of their free time within the dormitory complex.

For what concerns the systematic stops augmented with the *Work* activity, 40 out of 56 have been associated with the university's Department of Computer Science and Technology, with an average duration of 3 hours and an average distance between their centroids and the building of 15 meters. This suggests that the individual consistently spent their working hours in that building, likely attending classes or engaging in other related academic activities.

In conclusion, all the findings strongly suggest that the individual has the profile of a student who studies computer science at Tsinghua University and lives in one of the university's dormitories.

C. FINAL CONSIDERATIONS

In the example scenarios, we qualitatively demonstrate how our methodology enables users to easily instantiate their semantic enrichment processes (i.e., MAT-building pipelines), involving different trajectory datasets, external semantic data sources, and specific application questions. The pipeline used in the scenarios enriches movement data by incorporating various aspects from different data sources, and stores the resulting MAT datasets in RDF knowledge graphs. Such uniform representation facilitates interesting, useful, and flexible analyses.

Finally, let us discuss the simplicity with which a pipeline, and thus the underlying process, can be customized or completely changed to address alternative analysis tasks. For example, if we shift our focus to studying how individuals use public transportation for intra-city travel, we might employ a set of modules that combine movement data with additional factors such as traffic conditions, weather, and public transport timetables. By executing this pipeline, we can obtain suitable MAT datasets that capture the relevant information for subsequent analysis.

VII. CONCLUSION

This paper introduces a comprehensive methodology for creating heavily semantically enriched trajectory datasets, also known as multiple aspect trajectories (MATs). The goal is to create datasets in which movement data is augmented with dynamic and heterogeneous aspects (semantic dimensions), the information for which can be derived from various external semantic data sources. Our proposed methodology is agnostic towards the types of moving objects, aspects, and external semantic data sources being used, making it universally applicable across different scenarios. This is achieved thanks to the three pillars on which the methodology is built: the design principles, the MAT-Builder system which embodies these principles, and the use of an RDF knowledge graph-based representation. This last pillar allows the storage and querying of MAT datasets in a unified manner.

To the best of our knowledge, this is the first methodology that empowers practitioners to fully customize the semantic enrichment processing of movement data, taking into account different data, scenarios, aspects, and external semantic data sources. In the final qualitative evaluation, we demonstrate how our methodology allows practitioners to implement various semantic enrichment processes and construct different versions of enriched movement data, based on their specific analytical needs. Using the MAT-Builder system, we instantiate particular semantic enrichment processes to generate MAT datasets stored in RDF knowledge graphs. We then conduct several analyses on the movement behaviors of specific individuals, revealing how the MATs generated by MAT-Builder are useful and insightful.

REFERENCES

[1] R. D. S. Mello, V. Bogorny, L. O. Alvares, L. H. Z. Santana, C. A. Ferrero, A. A. Frozza, G. A. Schreiner, and C. Renso, "MASTER: A multiple aspect view on trajectories," *Trans. GIS*, vol. 23, no. 4, pp. 805–822, May 2019.

[2] C. Renso, V. Bogorny, K. Tserpes, S. Matwin, and J. A. F. de Macedo, "Multiple-aspect analysis of semantic trajectories (MASTER)," *Int. J. Geograph. Inf. Sci.*, vol. 35, no. 4, pp. 763–766, Jan. 2021.

[3] L. Ehrlinger and W. Wöß, "Towards a definition of knowledge graphs," in *Proc. Joint Posters Demos Track 12th Int. Conf. Semantic Syst. (SEMANTiCS), 1st Int. Workshop Semantic Change Evolving Semantics*, 2016, pp. 1–4.

[4] S. Spaccapietra, C. Parent, M. L. Damiani, J. A. de Macedo, F. Porto, and C. Vangenot, "A conceptual view on trajectories," *Data Knowl. Eng.*, vol. 65, no. 1, pp. 126–146, Apr. 2008.

[5] M. A. Abebe, J. Tekli, F. Getahun, R. Chbeir, and G. Tekli, "Generic metadata representation framework for social-based event detection, description, and linkage," *Knowl.-Based Syst.*, vol. 188, Jan. 2020, Art. no. 104817.

[6] A. Preece, I. Spasic, K. Evans, D. Rogers, W. Webberley, C. Roberts, and M. Innes, "Sentinel: A codesigned platform for semantic enrichment of social media streams," *IEEE Trans. Computat. Social Syst.*, vol. 5, no. 1, pp. 118–131, Mar. 2018.

[7] J. Tekli, "An overview of cluster-based image search result organization: Background, techniques, and ongoing challenges," *Knowl. Inf. Syst.*, vol. 64, no. 3, pp. 589–642, Mar. 2022.

[8] Y. Abgaz, R. R. Souza, J. Methuku, G. Koch, and A. Dorn, "A methodology for semantic enrichment of cultural heritage images using artificial intelligence technologies," *J. Imag.*, vol. 7, no. 8, p. 121, Jul. 2021.

[9] F. Özcan, C. Lei, A. Quamar, and V. Efthymiou, "Semantic enrichment of data for AI applications," in *Proc. 5th Workshop Data Manage. End-to-End Mach. Learn.*, Jun. 2021, pp. 1–7.

[10] H. Noueihed, H. Harb, and J. Tekli, "Knowledge-based virtual outdoor weather event simulator using unity 3D," *J. Supercomput.*, vol. 78, no. 8, pp. 10620–10655, May 2022.

[11] M. Ciavotta, V. Cutrona, F. De Paoli, N. Nikolov, M. Palmonari, and D. Roman, "Supporting semantic data enrichment at scale," in *Technologies and Applications for Big Data Value*. Berlin, Germany: Springer, 2022, pp. 19–39.

[12] Z. Yan, D. Chakraborty, C. Parent, S. Spaccapietra, and K. Aberer, "SeMiTri: A framework for semantic annotation of heterogeneous trajectories," in *Proc. 14th Int. Conf. Extending Database Technol.*, A. Ailamaki, S. Amer-Yahia, J. M. Patel, T. Risch, P. Senellart, and J. Stoyanovich, Eds., Mar. 2011, pp. 259–270.

[13] V. Bogorny, C. Renso, A. R. de Aquino, F. de Lucca Siqueira, and L. Alvares, "Constant—A conceptual data model for semantic trajectories of moving objects," *Trans. GIS*, vol. 18, no. 1, pp. 66–88, Feb. 2014.

[14] R. Fileto, C. May, C. Renso, N. Pelekis, D. Klein, and Y. Theodoridis, "The Baquara^2 knowledge-based framework for semantic enrichment and analysis of movement data," *Data Knowl. Eng.*, vol. 98, pp. 104–122, Jul. 2015.

[15] L. Ruback, M. A. Casanova, A. Raffaetà, C. Renso, and V. Vidal, "Enriching mobility data with linked open data," in *Proc. 20th Int. Database Eng. Appl. Symp. (IDEAS)*, E. Desai, B. C. Desai, M. Toyama, and J. Bernardino, Eds., Jul. 2016, pp. 173–182.

[16] T. P. Nogueira, R. B. Braga, C. T. de Oliveira, and H. Martin, "FrameSTEP: A framework for annotating semantic trajectories based on episodes," *Exp. Syst. Appl.*, vol. 92, pp. 533–545, Feb. 2018.

[17] T. P. Nogueira and H. Martin, "Querying semantic trajectory episodes," in *Proc. 4th ACM SIGSPATIAL Int. Workshop Mobile Geographic Inf. Syst.*, Nov. 2015, pp. 23–30.

[18] C. Pugliese, F. Lettich, C. Renso, and F. Pinelli, "MAT-BUILDER: A system to build semantically enriched trajectories," in *Proc. 23rd IEEE Int. Conf. Mobile Data Manage. (MDM)*, Jun. 2022, pp. 274–277.

[19] F. Lettich, C. Pugliese, C. Renso, and F. Pinelli, "A general methodology for building multiple aspect trajectories," in *Proc. 38th ACM/SIGAPP Symp. Appl. Comput.*, Tallinn, Estonia, Mar. 2023, pp. 515–517.

[20] A. Ibrahim, H. Zhang, S. Clinch, and S. Harper, "From GPS to semantic data: How and why—A framework for enriching smartphone trajectories," *Computing*, vol. 103, no. 12, pp. 2763–2787, Dec. 2021.

[21] L. Pappalardo, F. Simini, G. Barlacchi, and R. Pellungrini, "scikit-mobility: A Python library for the analysis, generation, and risk assessment of mobility data," *J. Stat. Softw.*, vol. 103, no. 4, pp. 1–38, Jul. 2022.

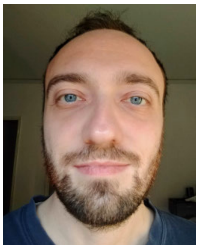
[22] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. KDD*, vol. 96. 1996, pp. 226–231.

[23] G. Boeing, "OSMnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks," *Comput., Environ. Urban Syst.*, vol. 65, pp. 126–139, Sep. 2017.

[24] S. Haidri, Y. J. Haranwala, V. Bogorny, C. Renso, V. P. da Fonseca, and A. Soares, "PTRAIL—A Python package for parallel trajectory data preprocessing," *SoftwareX*, vol. 19, Jul. 2022, Art. no. 101176.

[25] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, and D. Cournapeau, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2011.

[26] Y. Zheng, X. Xie, and W.-Y. Ma, "GeoLife: A collaborative social networking service among user, location and trajectory," *IEEE Data Eng. Bull.*, vol. 33, no. 2, pp. 32–39, Jun. 2010.



FRANCESCO LETTICH received the bachelor's, master's, and Ph.D. degrees from Universita Ca' Foscari, Venice, Italy. He has been a Postdoctoral Researcher with Universita Ca' Foscari, the Federal University of Ceara', Brazil, and the University of Alberta, Canada. He is currently a Researcher with ISTI-CNR, Italy. His research interests include spatial, spatio-temporal, and mobility data.

CHIARA PUGLIESE received the bachelor's and master's degrees in digital humanities from the University of Pisa, in 2018 and 2020, respectively, where she is currently pursuing the Ph.D. degree in computer science. She is also a member of the High Performance Computing Laboratory, ISTI Institute of CNR, Pisa, Italy. Her research interest includes data mining applied to semantically enriched mobility data.

CHIARA RENSO received the Ph.D. degree in computer science. She is currently a Senior Researcher with the ISTI Institute of CNR, Italy. She has more than 100 peer-reviewed publications in the areas of mobility analysis, machine learning and artificial intelligence methods for mobility data, analysis of geolocated social media, semantic enrichment of trajectories, and privacy. She is an Associate Editor of *Viewpoints* of the *Communications of the ACM* and an Editorial Board Member of the *International Journal of GIS*.

FABIO PINELLI received the M.Sc. degree in computer science and the Ph.D. degree in information engineering from the University of Pisa, in 2005 and 2010, respectively. He is currently an Assistant Professor with the SySMA Group, IMT School for Advanced Studies. He was a Research Scientist with IBM-Research Ireland and a Senior Data Scientist with Vodafone Italia. His research interests include data mining and machine learning methods applied to different domains, from economics to urban environments.

Open Access funding provided by 'Consiglio Nazionale delle Ricerche-CARI-CARE-ITALY'
within the CRUI CARE Agreement