Hindawi
Mobile Information Systems
Volume 2018, Article ID 6294872, 17 pages
https://doi.org/10.1155/2018/6294872

Research Article

The Urban Nexus Approach for Analyzing Mobility in the Smart City: Towards the Identification of City Users Networking

Federica Burini, Nicola Cortesi, Kevin Gotti, and Giuseppe Psaila

University of Bergamo, Bergamo, Italy

Correspondence should be addressed to Giuseppe Psaila; giuseppe.psaila@unibg.it

Received 1 December 2017; Accepted 25 February 2018; Published 8 May 2018

Academic Editor: Joaquin Huerta

We present an interdisciplinary approach that makes possible to learn how citizens live in the city by the means of mobile social media data, that is, volunteered geographical information provided by the inhabitants through social media and mobile apps, by adopting a new reticular approach to spatial analysis. In particular, we present the general notions as background of our work, an investigation methodology to apply whenever such an analysis task must be performed, and a digital environment of tools and frameworks to support the methodology.

1. Introduction

One of the main long-term goals of city administrators is to make their city attractive for residents and city users (tourists, commuters, migrants, etc.). Nowadays, the buzzword of *smart city* is used, to characterize ideal cities of the future. However, since real cities are far away from the utopian smart cities of the future, in order to understand how to improve and make the city *smarter* than it currently is, administrators need information about how the city is lived now, what are the critical issues, and what are the positive aspects to develop.

Nowadays, city users can significantly yet unconsciously contribute to the production of data useful for administrators, by the use of social media. In fact, city users often post geolocated messages that describe their activities throughout the city, by means of their smartphones. Such a big quantity of information could provide unexpected insights concerning how people live their city or the city they are visiting, and could be exploited to improve services, public transportations, viability, and so on.

However, to achieve this ambitious goal, many contributions are needed: the first technical contribution is given by the ability of gathering mobile data; the second is the ability of transforming and querying gathered mobile data; and the third is a flexible tool that provides analysts with the ability of studying *Big Geo-Data* by means of a simply-to-read visualization system. From a methodological point of view, there is a need to find an approach helping researchers in the investigation and analysis of the user-generated contents, in order to provide unexpected knowledge to administrators for understanding urban-mobility issues. In other words, an interdisciplinary approach is needed in order to make data scientists and computer scientists be aware of the main elements that are useful to geographers and urban space analysts to understand mobility and spatiality of individuals. Then, an interdisciplinary work can produce a methodology as well as software tools and frameworks to conduct on such an investigation.

In this paper, we present the *Urban Nexus Approach*, developed within the *Urban Nexus* project, an *Excellence Initiative* of the *University of Bergamo* (Italy) that involves *Anglia Ruskin University* (Cambridge, UK) and *EPFL* (Lausanne, Switzerland). In short, the goal of this project is to develop a methodology and tools to study how city users move within the city, use the urban space, and share their experience in places. The *Urban Nexus Approach* is a modular methodology that relies on a set of software tools and frameworks for gathering data sets coming from various sources of information (in particular, mobile social media), transforming and querying possibly geotagged data, visualizing them on maps, and analyzing them in order to

reconstruct networking of individuals within the city (i.e., spaces and connections they produce during their movement).

In this paper, first of all we introduce the general conceptual framework that provides basic concepts as background of our work. On these concepts, we define our approach and define the investigation methodology that constitutes the first main contribution of the paper. Then, the *Urban Nexus digital environment* is extensively presented: it is a pool of software tools and frameworks specifically devised to support the investigation methodology; the digital environment is the second main contribution of this paper. In fact, we remark that we do not want to present specific results concerning specific cities; we present a general approach that is both methodological and technical.

The paper is organized as follows. Section 2 introduces the multidisciplinary background of our research project. Section 3 introduces the investigation methodology that is driving our work. Section 4 presents the *Urban Nexus digital environment*, that is, the pool of tools and frameworks that together make possible to perform the analysis. Finally, Section 5 addresses concluding remarks.

2. Background

In this section, we present the background of our research. It is a multidisciplinary background, since the research project is multidisciplinary.

2.1. A New Perspective about the Contemporary City. A great question geographers are trying to give an answer is "what is the contemporary city?" [1]. An approach could be to interpret the "city" as a node of a globalized network, where a local dimension and a global dimension no longer exist separately; rather, these dimensions interact, by reconfiguring urban contexts with their centralities, their axis, their full and empty spaces, and their internal and external connection [2].

In other words, if the base of contemporary life is movement, the elements from which to explore the city are its people (stakeholders of mobility) and the places they visit, viewed as nodes of a network that creates unity and cohesions [3].

2.1.1. The Concept of Rhizome. In the context of globalization, the "contemporary city" assumes a polycentric and reticular configuration: it is no longer subdivided into center and periphery; rather, it is viewed as an "osmotic-centered system" of mobility. In fact, it is inserted into a globalized network, where local scale and global scale interact by reconfiguring centrality, axes, and internal and external connections of the city [1].

The creation of networks among the multiple places of the contemporary city is one of the processes that characterize the mobility of the inhabitants in the era of globalization: a new reticular dimension emerges, based on connections activated among places, exploited by individuals in their life experience; connections could be real (transportation infrastructures) and virtual (information about places published on the web or on social media, possibly produced by citizens).

Such a reticularity, produced by experience of individuals in urban space, is termed *rhizomatic*, resuming a concept born in the field of botany and then reelaborated in the philosophical field: "Compared to centric (even polycentric) systems, hierarchical communication, and predetermined connections, a *rhizome* is an acentric, nonhierarchical, and not meaningful system" ([4], page 33). The concept of *rhizome* is refined in spatial terms by Jacques Lévy: "a rhizome is the space of individual action in mobility, but also in the multiform relationship with other individuals" ([5], page 19). A further definition could be the following: "a rhizome is a family of networks, characterized by the absence of identifiable boundaries and a meeting between topological metric inside and topographic metrics outside" ([5], pages 18-19). In other words, a rhizome belongs to the topology metric, that is, to a discontinuous space, based on nodes and connections that produce a network without beginning, without end, and without well-defined boundaries because it is the result of the experience in space of individuals. From a very pragmatical point of view, we can guess that "a rhizome is a set of places frequently lived by a single user and by many users, on the basis of material and virtual connections among them."

Based on this perspective, understanding the contemporary city means understanding city users, that in turn means understanding their rhizomes.

2.2. The Role of Social Media and Big Data. It is now clear that the identification of rhizomes is the crucial point of our research, but this is not an easy task. The reason is that each person has his/her way to experience the city, conditioned by the places where he/she lives, the places where he/she works, his/her interests, and so on.

An unexpected help comes from the popularity of social networks that stimulated new behaviors by people. In fact, many social network users continuously post messages about their day-by-day life. Through mobile apps for smartphones, they can post georeferenced messages that, if gathered, could reveal their movements and their habits. Many social networks provide an API (application programming interface) for getting posted messages; the result is that a huge amount of data can be easily gathered about habits of single users (people become sensors of themselves [6, 7]).

Consequently, in order to study rhizomes, we could rely on the potential of Big Data [8–11], by exploiting techniques and Big Data mapping systems. Big Data could foster the analysis of function and use of urban spaces based on the needs of inhabitants and city users. Data sets coming from an incredible variety of sources, such as social networks, mobile phone companies, public authorities, and national statistical institutes, could be cross-analyzed, aiming at understanding habits, flows, and relationships of inhabitants and city users (residents, commuters, tourists, migrants, etc.); the goal is to reveal different ways of experiencing the world and of managing the distances within it [12].

Such a kind of research has been already introduced in Europe by the *European Statistical System* and accepted by the Italian National Statistics (named ISTAT), particularly from the "Commission to study and guide the choices of ISTAT on Big Data." Public administrations could get great advantages in taking decisions about mobility infrastructures and many other issues, if they were provided with tools to exploit to improve their knowledge about how city users live in the city (a smart city should learn about itself).

2.3. The Urban Nexus Project: An Interdisciplinary Approach to Study the Smart City. The *Urban Nexus* project (see the seminal paper [13] for more information) aims at developing a scientific and educational cooperation among *University of Bergamo* (Italy), *Ecole Polytechnique Fédérale de Lausanne* (Switzerland), and *Anglia Ruskin University* in Cambridge (UK) (the research project is an initiative of the University of Bergamo in Italy, under the scientific coordination of Emanuela Casti and Federica Burini at the CST-DiatheisLab; see: http://urbannexus.unibg.it). The overall goal is to *develop a methodology* to analyze contemporary cities through an integrated and interdisciplinary approach, in order to foster renovation and improvement of accessibility to the city. The three cities hosting the involved universities, that is, Bergamo (Italy), Lausanne (Switzerland), and Cambridge (UK) will be three real case studies.

The Urban Nexus project aims at involving researchers with different competences, mainly, geographers, spatial analysts, and computer scientists. In fact, in order to collect and cross-analyze large amounts of data, skilled computer scientists are necessary; they have to provide flexible tools, able to manage many and possibly huge data sets, each of them possibly having different formats and heterogeneous structures. Nevertheless, the choice and analysis of data must be carried on, and the results are interpreted by geographers and spatial analysts.

The activities of the project are devoted to provide two main contributions:

(i) The definition of an *investigation methodology* that, relying on the concept of rhizome, is able to drive researchers to analyze data coming from various Big Data sources in order to understand the contemporary city.

(ii) The development of a novel *digital environment* that provides tools and frameworks able to effectively support the investigation methodology; in fact, nowadays, powerful software tools are a necessary condition to effectively perform social and geographical analysis based on Big Data.

2.4. Technological Context. The term *Big Data* is a buzzword very poplar nowadays. Behind it, many aspects are hidden, so that many researchers have been working on the topic.

In particular, a famous paper is the study by Kitchin [14], where the 3 Vs model is proposed, in order to characterize the topic: Big Data means *Volume*, *Velocity*, and *Variety*. This observation is important, because people usually think about Big Data more or less only in terms of volume; however, data change very fast and can come very fast; furthermore, they can be really various, that is, a large number of different data sets coming from a large number of sources should be integrated together, in order to get the desired results. Our context is characterized by volume and variety.

Currently, technological approaches to data exchange and diffusion has converged on JSON (JavaScript Object Notation) as de facto standard for information interchange. In fact, social network APIs use JSON to represent data sets they can export. Many web services adopt the same approach, and on open data portals, JSON data sets are becoming more and more popular, in place of CSV and XML. In particular, as far as XML is concerned, JSON is substantially playing the role for which XML was designed: the reason is that an XML document is hard to manage within programs; on the contrary, object-oriented data structures of programming languages can be easily serialized into JSON, and JSON files can be easily deserialized into object-oriented main memory data structures.

On the same track, the GIS community as well has adopted JSON: the recent standard named *GeoJSON* [15] is a JSON-based format to describe geographical information layers.

The consequence of the diffusion of JSON is that DBMSs able to store and manage large collections of JSON data sets have become necessary. On this technology track, MongoDB [16] is certainly the most famous NoSQL database system (see [17] for an overview about NoSQL databases): in fact, it is able to store and query heterogeneous collections of JSON objects (i.e., objects with different structures) in the same collection. In spite of its popularity, the query language is specifically taught for programmers, being based on JavaScript; consequently, it is not suitable for geographers and analysis that are not familiar with hard procedural programming.

The reader can guess that large data sets, as well as a large number of data sets, are difficult to analyze by hands. Useful techniques that could greatly help in analyzing such a mess of data could come from the Information Retrieval area: in particular, the well known *Page-Rank* algorithm [18] could be a good starting point. The first version of Google search engine was based on it; its strength is the reticular view of linked web pages; consequently, we expect that the approach could be adopted for and adapted to the analysis of reticularity of contemporary cities.

More than twenty years ago, the area of data mining has provided very interesting techniques to analyze frequent cooccurrences of items in market-basket analysis, known as *itemset mining* and *association rule mining* [19–21]. Originally developed for the retail industry and applied to relational databases [22], they are very useful for a variety of problems. In our context, they could be applied to address the problem of identifying rhizomes of city users, moving from traces of people: in fact, a rhizome could be seen as the frequent cooccurrence of places in tours/trips performed by city users. Furthermore, other techniques developed on the side of itemset mining and association rules mining could provide inspiration for developing novel techniques and

tools. In particular, the need for integrating several data sets may recall the techniques that make effective the integration of association rules extracted from within multiple transaction databases (see [23–26]).

3. Methodological Approach

The first contribution provided by the project is the definition of an investigation methodology, toward the identification of rhizomes from within data describing mobile users of social media. Independently of the specific data set and independently of the specific goal of the analysis, the analysis process should proceed performing the following steps:

(1) *Identification of sources*: it is necessary to identify sources of potentially useful data sets that should be as meaningful as possible.

(2) *Selection, transformation, and analysis* of data produced by mobile social media users and their *cartographic representation* to understand mobility in urban spaces.

(3) *(Towards the) reconstruction of rhizomes* from mobile data produced by social media users.

All these steps will be separately described in the next subsections.

3.1. Identification of Sources. The first step to perform is the collection of data sets, from relevant data sources. However, it is not easy to understand what data sources are relevant: different data sources provide different data sets that could give different points of view of the problem.

In particular, in order to analyze the mobility of inhabitants, the following sources could be considered.

(i) *Statistical sources*. National and European Statistics Institutions continuously provide data sets about citizenship in general and about mobility in particular. These data sets are *certified* in the sense that their quality is controlled, and they can be considered *reliable*. Of course, since they are the result of a rigorous selection and analysis process, they are not up to date. Furthermore, they do not describe habits of single people.

(ii) *Collaborative sources* rely on the wish of people to participate to an investigation, typically through web applications, social media, and mobile apps. The participatory process can generate very useful crowd-sourced information and Volunteered Geographic Information (VGI) concerning single-city users [27–30].

(iii) *Private sources* could effectively integrate previous data sets. In particular, as far as detection of mobility is concerned, mobile phone companies own very detailed data sets about movements of phone users. The data sets they could provide are very large and very rich, in terms of information about the sequence of visited places and how long a person stays in a place. However, these data sets are usually difficult to be obtained because phone companies are jealous of their data.

Thus, depending on the selected data sets, more or less complex and more or less long acquisition processes must be carried on to get the desired data sets.

3.2. Selection, Transformation, Analysis, and Cartographic Representation. Once gathered, data sets must be put in a form suitable for getting useful information. Several activities could be done.

(i) Selection. Among all data sets, it is necessary to select only the relevant data. Many considerations can affect this task, for example, the typology of places, the typology of city users, the time period of interest, and so on.

(ii) Transformation. To analyze selected data (or, better, to cross-analyze selected data), a process of transformation and integration is necessary. For example, if places visited by users are characterized by the latitude and longitude, perhaps it could be necessary to perform a geo-coding activity, that is, substitute punctual coordinates with the name of places.

(iii) Analysis. Transformed data sets should be analyzed by analysis, in order to get some useful hints about what interesting information could it be possible to extract.

(iv) Cartographic Representation. The analysis could be effectively supported by a cartographic representation of selected and transformed data, in order to conduct a visual-analysis process.

3.3. (Towards the) Reconstruction of Rhizomes. Following a reflective approach [31], the goal of this step is to analyze the rhizomatic spatiality that emerges from data sets describing city users. We identified a multiparadigm approach, that is, multiple paradigms can be jointly applied to analyze data from different perspectives.

3.3.1. Site Analysis. Places that emerge from data sets (e.g., visited by single users) constitute a reticular view of the space, where they are the nodes of a network. Several dimensions characterize the analysis.

(a) Localization: by analyzing images and text, locations could be identified or better characterized with respect to simple coordinates.

(b) Time analysis: time elapsed on a node and distance in time or in space between two nodes could reveal interesting information (e.g., about accessibility of nodes).

(c) Categories of nodes: nodes (places) can be categorized, in order to perform a more specific analysis; for example, a place can be categorized as *public spaces* (e.g., open spaces such as squares and roads

and mobility places such as stations, airports, and public urban transport stops), *semipublic spaces* (such as monuments, religious, and historical places), or *semiprivate places* (such as hotels, restaurants, and shops), as identified by Jaques Lévy ([1], page 57).

(d) Opinion of users: the opinion of users in relation to places could give a sentiment polarity about it.

3.3.2. Connection Analysis (Networks). The reticular view of cities demands for analyzing connections between nodes. Aspects to discover are hereafter reported.

(a) *Accessibility* (cycle-pedestrian, automobile, public transport on wheel or on iron, air).

(b) *Quantitative relationships between nodes*: a ranking method could reveal the strength of relationships between nodes, on the basis of the reticular perspective. In this respect, we defined an algorithm named *Node Rank*, described in Section 4.5.1.

On the same line, data mining tools, such as frequent itemset mining [19] are necessary, in order to reconstruct rhizomes based on a quantitative approach.

3.3.3. Cartographic Representation. This is a necessary support to analysts. However, traditional cartography could not be satisfactory, in order to visualize relationships on the reticular space; for this reasons, we are going to perform research work by experimenting representations that possibly meet the reflexive approach (inspired by [31]).

The investigation methodology so far presented cannot be deployed without software tools that assist analysts during the overall process.

4. The Urban Nexus Digital Environment

The investigation methodology described in Section 3 must be supported by several software tools. These tools constitute the *Urban Nexus Digital Environment*. From the technical point of view, this environment constitutes the main result of the project, with tools and frameworks specifically developed moving from needs merged during the definition of the investigation methodology.

Figure 1 depicts the digital environment, showing the components that we describe hereafter.

The *Urban Nexus digital environment* is divided in four sections, depending on the task performed by tools and analysts:

(i) On the left-hand side, we find the *Data Acquisition* section that covers tasks devoted to acquiring data sets. In particular, we consider, as data source, *Open Data Portals*, that have become valuable and precious data sources, and *Twitter*. We also generically consider any kind of data source that provides useful data.

(ii) On the top of the figure, we find the Integration and Transformation section, devoted to integrate and transform the collected data. Usually, integration and transformation tasks are complex and, often, tedious activities, but they are crucial for conducting investigation activities.

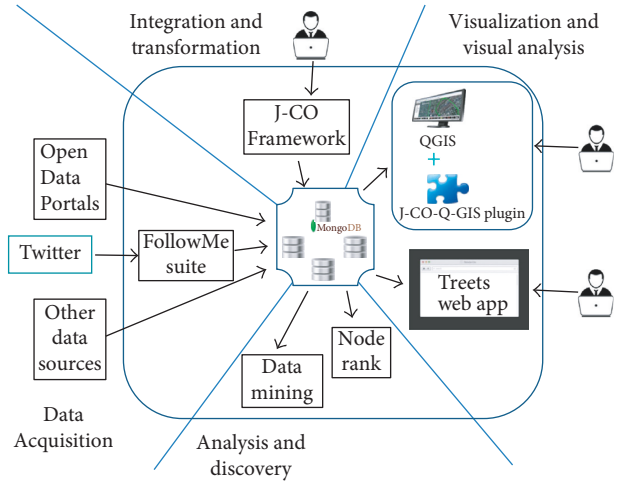

Figure 1: The *Urban Nexus digital environment*.

(iii) On the right-hand side, we find the Visualization and Visual Analysis section that copes with those tasks performed by analysts based on a visual analysis of the data, as well as with mapping of data.

(iv) On the bottom of the figure, we find the Analysis and Discovery section that copes with tasks requiring massive analysis of data, based on possibly novel data mining and knowledge discovery algorithm.

In the center of the digital environment, we find data sets: in fact, source data, intermediate results, and final results are the value of investigation activities. Nevertheless, data must be gathered, integrated, transformed, analyzed, and visualized. Such tasks are performed by the tools that constitute the *Urban Nexus digital environment*.

The investigation methodology asks to collect data coming from many sources, such as social media like *Twitter*, open data portals, and many other public sources of information. Such data sets are usually provided as JSON collections, due to its capability of describing complex and heterogeneous data. Nowadays, JSON has become the de facto standard representation adopted to share social media data and open data sets. Consequently, the storage service in the *Urban Nexus digital environment* is provided by *MongoDB*, the very popular NoSQL DBMS that natively stores collections of JSON objects.

Let us briefly describe the components of the *Urban Nexus digital environment*.

(i) The FollowMe Suite gathers data from Twitter, in order to discover traveling users and track them. It monitors a pool of airports to detect users that posted geolocated posts in the airport area and tracks these users for 8 days, in order to find out traveling users.

(ii) The J-CO Framework is devoted to transform and query collections of possibly geotagged JSON data sets. It provides a query language, named J-CO-QL,

that allows analysts to specify complex high-level queries by means of declarative operators that do not require programming skills; operators of the language deal with spatial representation in a native way.

(iii) QGIS is the very famous free GIS software that can be used to visualize geolocated data so far gathered and stored within MongoDB databases. A plug-in, named J-CO-QGIS, was developed to allow easy visualization of JSON geotagged data sets obtained by means of the J-CO Framework.

(iv) The Treets web app is a visual exploratory tool that allows analysts to explore traces of single users, by analyzing their posts (texts and pictures), by means of a cartographic representation of posts and traces.

(v) Node Rank is an algorithm developed to study the centrality of places in travelers' trajectories. Furthermore, other data mining tools, such as frequent itemset mining algorithms, are going to be used, in order to discover frequent patterns of places visited by tourists and, more in general, city users.

In the rest of this section, we introduce the components of the *Urban Nexus digital environment*.

4.1. The FollowMe Suite. The *FollowMe* suite [32] is an open and interoperable pool of tools [33], developed to discover and track social network users through their geotagged posts. The suite originates from the idea of trying to track *Twitter* users, but the suite is easily extensible and its architecture is open, so that new external components can be easily added (in fact, for a period, we also tracked *Instagram* users [34]). The goal is to gather nonauthoritative information about tourists that visit a given city. In particular, we are interested in tracking flying tourists, that post when they are waiting for their plane in the origin airport, or when they are waiting for their luggages in the destination airport.

We now illustrate the general overview of the approach and of the suite, illustrated in Figure 2.

The general vision is that *Twitter* users post a geotagged message when they are either in the origin or destination airport, at the beginning of their trips; then, during their trips, they capture pictures or write comments about a place, possibly by means of a geotagged post as well. All these posts are sent by the *Twitter* mobile app to the *Twitter Farm* that hosts servers and stores data. At this point, it is the turn of the *FollowMe* suite.

(i) The Hang Post Finder is responsible to query the Twitter API to look for messages posted in the area of a monitored airport (Twitter API provides a service to search for geolocated tweets, given the coordinates of the center and the radius of an area of interest). These messages are called hang posts, because they are the hang to discover users to follow.

(ii) The Timeline Tracker follows each user identified by means of hang posts, by inspecting his/her timeline, that is, the history of messages posted by the user to find out geolocalized messages posted in the next 8 days after the date of the hang post.

(iii) Posts are stored within a MongoDB database as JSON objects.

(iv) The Trip Builder is the component invoked to query the database of gathered messages and reconstruct trips, in order to later study them. The *Trip Builder* can be invoked through the API that provides an external access to this query service. Again, JSON is adopted to generate output data sets, as well as classical CSV files (loosing, in this case, some information and flexibility of representation).

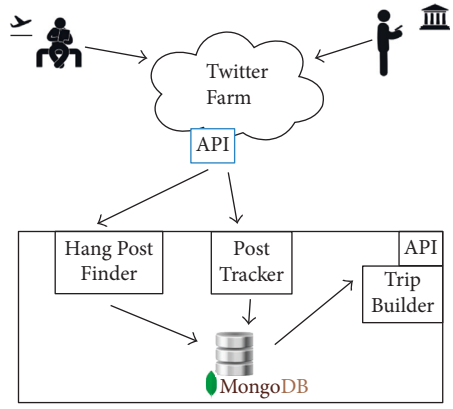

Figure 2: The *FollowMe* suite.

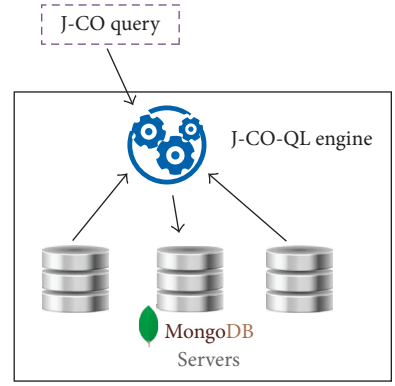

Figure 3: The *J-CO* framework.

4.2. The J-CO Framework. The framework we conceived for cross-analyzing multisource georeferenced data is named *J-CO* (*J*SON *CO*llections) and is constituted by the following main components depicted in Figure 3:

(i) One or more *NoSQL* databases, namely *MongoDB*, to support JSON objects storage from multiple sources

(ii) A *Data Model* that makes explicit the role of geotagging (when present) in collected JSON objects, as well as a clear *Execution Model* on top of which to build a novel query language

(iii) A novel query language, named *J-CO-QL*, that provides high-level operators for transforming heterogeneous collections of JSON objects, able to deal in a native and simple way with geotagging and spatial relationships

(iv) The *J-CO-QL* Engine that evaluates queries by reading/writing data in a *MongoDB* database (it receives queries and executes them on top of the NoSQL servers)

4.2.1. Data Model. The basic concept on which we rely to define the *J-CO-QL* language is the JSON object. Fields (object properties) can be simple (numbers or strings), complex (i.e., nested objects), and vectors (of numbers, strings, and objects). As far as the georeference field is concerned, we rely on the *GeoJSON* standard [15, 35]. In particular, we assume an object's geometry field named `~geometry`, defined as a *GeometryCollection* type for the *GeoJSON* standard. The absence of this top-level field means that the object does not have an explicit geotag.

As an example, consider the JSON objects reported in Figure 4, describing *Points of Interest* (POIs), in the city of Bergamo. In this example, two shops are represented with a point-like georeference: notice the `~geometry` field, that describes the longitude and latitude (resp.) of the centroids of the shops' locations on Earth, as defined by *WGS* (*World Geodetic System*) *84* (this standard is our default *CRS*–*Coordinate Reference System*).

In a *NoSQL* environment such as *MongoDB*, a *Database* is defined as a set of collections c, while a *Collection* is represented by a name c.name and its instance, that is, a vector of JSON objects. To manipulate JSON collections and to store their results in new collections, we need operators (Section 4.2.2) that satisfy the *closure property*, that is, they get collections and generate collections. This is a first characteristic of the *J-CO-QL* language.

J-CO-QL queries transform collections stored in databases and generate new collections which will be stored in turn into a possibly different database, to achieve data persistence. For simplicity, we call such databases as *Persistent Databases*.

4.2.2. Query Process. A *state s* of a query process is a pair $s = (\text{tc}, \text{IR})$, where tc is a collection named *Temporary Collection*, while IR is a database named *Intermediate Results database*.

Each operator application starts from a given query process state and generates a new query process state. When a query is executed by the *J-CO-QL* Engine, the resulting tc (*Temporary Collection*) will be optionally stored to an IR (*Intermediate Results*) database, that could be taken as input by a subsequent operator application.

The query process starts from the empty temporary collection $s_0 \cdot \text{tc} = \varnothing$ and the empty intermediate results database $s_0 \cdot \text{IR} = \varnothing$. Thus, *J-CO-QL* provides operators (named *start operators*) able to start the computation, taking collections from the persistent databases, while other operators (named *carry-on operators*) carry on the process continuously transforming the temporary collection and possibly saving it into the persistent databases or, for temporary results, into the intermediate results database IR. This way, new subtasks can be started, by taking collections both from persistent databases and from IR.

```
[
   {
      "id": 321,
      "category": "shop",
      "name": "Shop 321",
      "~geometry":
      {
         "coordinates": [
            9.186973,
            45.467843
         ],
         "type": "Point"
      }
   },
   {
      "id": 456,
      "category": "shop",
      "name": "Shop 456",
      "~geometry":
      {
         "coordinates": [
            9.205654,
            45.477872
         ],
        "type": "Point"
      }
   }, ...
]
```

Figure 4: Excerpt of collection `POIs`.

Innovative Features. The need to support complex transformation processes that typically pass through the generation of several intermediate results motivates the intermediate results database IR. Intermediate collections are stored explicitly into IR, to be later used for creating the target collection. In fact, it would be inappropriate to store them into the persistent databases that should store source and target collections. Collections stored in IR are clearly intermediate, and will disappear from the system at the end of the process.

4.2.3. J-CO-QL: Language and Execution Engine. The key components of the *J-CO* Framework are the query language, named *J-CO-QL*, and its execution engine. With respect to other query languages for *JSON* objects, the main innovations provided by *J-CO-QL* are the following:

(i) Typically, other query languages for JSON collections are unable to deal with heterogeneous objects in the same collection at the same time (in general, several queries must be written and then their results united together). J-CO-QL provides operators

[key]	date	postId	time	userId	~geometry
0	2015-04-01	58313545517835690	06:10:20	John	[-] Object, 2 properties: coordinates: [-] Array, 2 items: 0: 45.66561062; 1: 9.69930355; type: Point
1	2015-04-06	58313905695025460	06:54:01	Alketa	[-] Object, 2 properties: coordinates: [-] Array, 2 items: 0: 45.66561062; 1: 9.69930355; type: Point

(a)

[key]	category	id	~geometry
0	shop	7547	[-] Object, 2 properties: coordinates: [-] Array, 2 items: 0: 9.44838445411; 1: 45.7409545188; type: Point
1	shop	7562	[-] Object, 2 properties: coordinates: [-] Array, 2 items: 0: 9.696728220119999; 1: 45.6921785928; type: Point

(b)

Figure 5: (a) Objects from collection `TrackedPosts`, describing tracked posts. (b) Objects from collection `POIs`.

specifically designed to deal with objects with different structure within the same operator application.

(ii) Second, other languages for querying JSON data are conceived for programmers, or for people having programming skills. In contrast, operators provided by J-CO-QL are high-level operators that allow analysts to think directly to objects structure; they do not have to write low-level procedures.

(iii) Finally, but not less important, J-CO-QL directly deals with georeference possibly contained in JSON objects because the data model explicitly deals with them through the `~geometry` field.

In the rest of the paper, we will present relevant operators of *J-CO-QL* by explaining their use in transformation tasks. The reader can refer to [36], where the language is introduced in more details.

4.2.4. J-CO-QL by Example: Cross-Analyzing Tweets and POIs. In this section, we will show the power of *J-CO-QL*, by showing its application to a real case concerning smart cities. Meanwhile, a sketch of fundamental operators is given; for a detailed introduction, refer to our internal report [37].

The first collection to consider is named `TrackedPosts`; it contains georeferenced posts gathered from *Twitter* by the *FollowMe* suite (Section 4.1) since May 1, 2015. The second collection is named `POIs` and contains georeferenced Points of Interest. We have two specific goals: (a) we want to discover the most attractive POIs in the area of the city of Bergamo; (b) we want to discover tourist traces that mostly visited POIs.

Each post in collection `TrackedPosts` is described by a JSON object having several fields, that is, `postId`, `userId`, `date`, and `time`. An excerpt of this collection is reported in Figure 5(a), where objects are pretty printed in a graphical way. Notice the `~geometry` field: it was not present in the original JSON representation of posts; it has been added by performing some preprocessing activities by means of *J-CO-QL* (they are not reported here because they could be tedious for the reader).

As far as Points of Interest (POIs) are concerned, we started from the 4 collections we gathered from Open Data Portals, one for each POI category: *hotel*, *shop*, *museum*, and *architectural*. Similarly to the preprocessing task performed for posts, by means of the *J-CO-QL* we preprocessed them in order to obtain a unique and homogeneous collection of JSON objects, containing all the POIs, named `POIs`. Fields of objects are `id`, `category` (which denotes the category of the POI, e.g., `hotel`, `shop`, `museum`, and `architectural`), and `~geometry`. An excerpt of collection `POIs` is reported in Figure 5(b).

Moving from collections `TrackedPosts` and `POIs`, we can perform transformations suitable to reach our goals, that is, (a) to rank POIs on the basis of the number of posts nearby and (b) rank user traces with respect to the number of POIs of a given type their posts are nearby. Both goals rely on the ability of discovering which posts are nearby which POI. To this end, the *J-CO-QL* language provides the `SPATIAL JOIN` operator, able to couple objects coming from two collections on the basis of a (metric or topological) spatial condition on geometries. This operator is the key of the process that is reported in Figures 6 and 7. Hereafter, we will describe the process in details, by introducing the *J-CO-QL* operators.

4.2.5. Ranking POIs and Posts. The general goal of the analysis process is to compute a score for POIs and a score for traces. The score for POIs is defined by

$$\mathrm{Sc}(p) = \left| \{\mathrm{tp} \in \texttt{TrackedPosts} \mid \mathrm{dist}(p, \mathrm{tp}) \leq 100\} \right|. \quad (1)$$

The score $\mathrm{Sc}(p)$ of a POI p is the number of posts with distance less than 100 meters from the POI's coordinates (function dist gives the distance in meters).

As far as users traces are concerned, we rank them with respect to the four categories of POIs we considered, separately. The score is defined in

$$\mathrm{Sc}(t,\ c) = \sum_{\mathrm{tp}_j \in t, p_i \in \mathrm{POIs100}\left(c, \mathrm{tp}_j\right)} \frac{1}{1 + \mathrm{dist}\left(p_i,\ \mathrm{tp}_j\right)}. \quad (2)$$

For a trace t and a POI category c, the score $\mathrm{Sc}(t,\ c)$ is the sum of inverse of the distance (in meters) between

```
SPATIAL JOIN OF COLLECTIONS
  TrackedPosts@SmartData as post,
  POIs@SmartData as poi
 ON DISTANCE(M) <= 100 SET GEOMETRY RIGHT
 CASE WHERE
      WITH  .poi.category, .poi.id,
            .post.postId, .post.userId
      GENERATE
        {category: .poi.category,
         poiId: .poi.id
         postId: .post.postId,
         userId: .post.userId,
         dist: DISTANCE(M),
         inv_dist: 1 / (1 + DISTANCE(M))}
      KEEPING GEOMETRY
 DROP OTHERS;
SET INTERMEDIATE AS PostsAndPOIs;

GROUP
 PARTITION  WITH STRING .poiId
  BY  .poiId, .category
  INTO  posts
  GENERATE {.category, .poiId, .posts,
            count: SIZE(.posts)}
    SETTING GEOMETRY .posts[1].(~geometry)
DROP OTHERS;
SAVE AS WeightedPOIs@SmartData;
```

Figure 6: Query that computes the scores for POIs.

a point tp_j in trace t and a POI p_i of category c at distance no greater than 100 meters from tp_j. (POIs $100(c, \text{tp}_j)$ is the set of POIs of category c at distance no greater than 100 meters from tp_j), plus 1 meter. This way, each term cannot be infinite, and it is always less than or equal 1 (it is 1 when tp_j and p_i have the same coordinates). Note that we have four scores for each trace, one for each POI category.

So, the concrete problem of the analyst is to compute the scores for POIs and traces. In the following, we will describe how a *J-CO* user could perform the analysis described above, by executing two simple *J-CO-QL* queries.

(1) Computing Scores for POIs. The first query (Figure 6) has the goal of creating a new collection named `WeightedPOIs`, where each object describes a POI and has a score field which is the number of posts nearby to the POI. We can split the query in two parts: the first one executes a `SPATIAL JOIN` between collections `TrackedPosts` and `POIs`; the second part performs a `GROUP` operation, in order to group together all posts nearby to the same POI and derive the `score` field (i.e., the number of nearby posts).

The key operator for this query is the `SPATIAL JOIN` operator (introduced in our language in [38]). It takes two collections (either stored in a *MongoDB* persistent database or in the intermediate results database IR) and builds pairs of objects coming from the two input collections, on the basis of the relationship existing between their geometries. In this case, we join objects l_i from collection `TrackedPosts`, with objects r_j from collection `POIs`, creating an object $t_{i,j}$ for every i, j. Both source collections are stored in the persistent database `SmartData`. The two collections are aliased, within the operator, as post and poi, respectively. As far as the

```
GET COLLECTION PostsAndPOIs;
GROUP
 PARTITION  WITH STRING .poiId
  BY  .userId, .category
  INTO  posts
  GENERATE {.category, .userId, .posts,
            score: SUM(.posts, .inv_dist)}
    DROPPING GEOMETRY
DROP OTHERS;

FILTER
 CASE WHERE score>= 0.03
  GENERATE {.userId, .category, .score}
DROP OTHERS;
SET INTERMEDIATE AS RankedTraces;

JOIN OF COLLECTIONS
    TrackedPosts@SmartCities AS post,
    RankedTraces AS RT
 CASE WHERE .post.userId = .RT.userId
 GENERATE {score: .post.score,
           postId: .post.postId,
           userId: .post.userId,
           category: .RT.category}
   SETTING GEOMETRY AS .Post.~geometry
DROP OTHERS;

GROUP
 PARTITION WITH .score, .postid,
                .userId, .~category
 BY .score, .userId, .category
 INTO posts
 GENERATE {.score, .userId, .category}
    SETTING GEOMETRY AS POLYLINE(.posts)
 DROP OTHERS;
SAVE AS RankedTracks@SmartCities;
```

Figure 7: Query that computes the scores for tourists' traces.

geometry of $t_{i,j}$ is concerned, in our case, we specified `SET GEOMETRY RIGHT`, meaning that we keep the geometry of the right object r_j (in this case, the POI); alternatively, we can specify `LEFT, INTERSECTION, ALL` (which merges the two geometries). Notice that the output object $t_{i,j}$ has three fields: one has the name of the left collection alias (i.e., `post`) and its value is the left object l_i; one has the name of the right collection alias (i.e., `poi`) and its value is the right object r_j; the third one is the `~geometry` field. Finally, the object $t_{i,j}$ is generated (i.e., the left and right objects are joined) only if the spatial join condition specified after the `ON` keyword is met: in this case, the distance between the geometries of the two objects must be no greater than 100 meters.

The `WHERE` condition after the `CASE` keyword selects output objects having the desired fields (`WITH` predicate): these objects are restructured by the `GENERATE` action, that also adds two fields, that is, the distance value and its inverse, respectively, named `dist` and `inv_dist`.

We then store those results in a new collection named `PostsAndPOIs` into the intermediate results database *IR*.

After that, the GROUP operation takes the before-generated collection and, by grouping the objects in the

collection by values of fields, poiId, and category, creates a collection of distinct POIs. This collection is similar to the original POIs collection, but each object has a count field, that is, the number of posts with distance equal or less than 100 meters from the POI. Notice that field posts is the array of all posts grouped together; since in the previous spatial join, the geometry of POIs was kept, all grouped objects in array posts have the same geometry; thus, we take the ~geometry field of the first object in the array as ~geometry field of the overall object (SETTING GEOMETRY option in the GENERATE action of the GROUP operator).

The collection so far obtained is saved into the persistent database SmartData with name WeightedPOIs.

(2) Computing Scores for Traces. The second query (Figure 7), computes the scores, for each POI category, of tourists' traces. The query is composed by four parts:

(1) First of all, moving from the intermediate collection PostsAndPOIs, the GROUP operator computes the score of each trace for each category.

(2) The FILTER operator selects traces having score of at least 0.03 for at least one category, meaning that we want only traces with at least 3 posts in the surrounding of at least one of the given category.

(3) Tourists' traces are cloned, in order to associate them with all the different scores (one for each category), by the JOIN operator.

(4) Finally, the GROUP operator generates one object for each trace for each score category, in order to generate a geometry representing the trace by means of a polyline, to show it on the map.

In details, the first GROUP operator takes posts in collection PostsAndPOIs and groups objects having the same value for fields userId and category, generating the array field named posts, containing all posts grouped together. The GENERATE action adds the score field, by summing the values of field inv_dist of objects in the array posts.

The FILTER operator takes the temporary collection and keeps only objects that satisfy the WHERE condition; then, it generates a restructured version of them. In this case, we want traces (identified by userId and labeled by field category) with a score of at least 0.3. By using the GENERATE action, we obtain objects having only fields userId, category, and score. The output collection is stored into the intermediate results database with name RankedTraces.

In the third part of the query, the JOIN operator joins the intermediate collection RankedTraces (aliased as RT) with collection TrackedPosts (aliased as post) in the SmartCities persistent database. The WHERE condition specifies that we join objects of the two input collections if they have the same value for fields userId in both objects. The behavior is similar to the SPATIAL JOIN, apart from the ~geometry field, that is not automatically dealt with. In practice, we extend posts with the category and the score of the trace for that category. Generated objects have fields score, userId, and category; the ~geometry field is taken form the ~geometry of the post, as specified by the SETTING GEOMETRY clause.

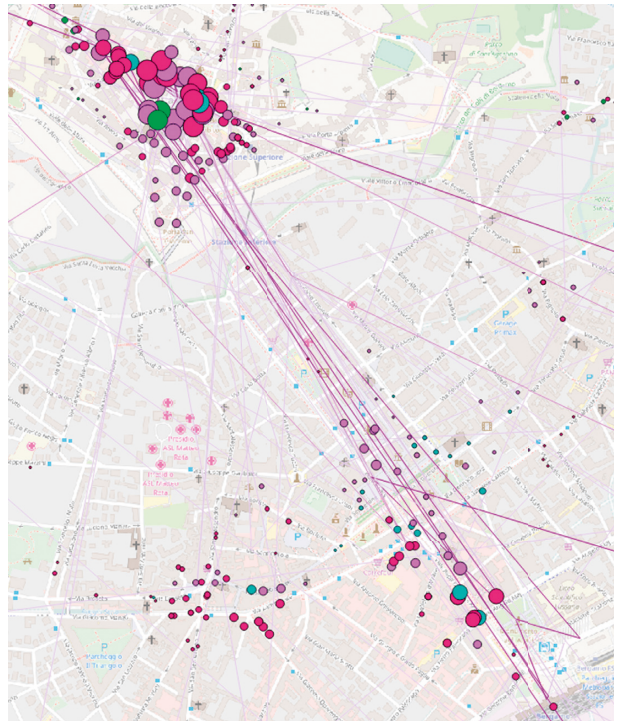

Figure 8: Visualization of POIs (circles), having more than one posts in its surroundings, with size proportional to their score (colored points) and tourists' traces (lines) having the best score for architectural POIs (the greater the score is, the higher the score is).

Finally, we can create a collection of objects that describe a trace labeled with the category and the score, where ~geometry is a *polyline* representing the track of all posts in the trace. This is done by the last GROUP operator that groups objects (posts) on the basis of fields score, userId, and category, deriving the final ~geometry by specifying SETTING GEOMETRY AS POLYLINE, which derives a polyline from each point grouped into array posts. We finally store the output collection as RankedTraces in the SmartCities persistent database.

Actually, we clustered tourists' traces by the category of their favored POIs. In particular, if a user mostly visits architectural POIs (and posted georeferenced content nearby them), his/her track will be classified as an architectural trace. Otherwise, if a user only posts social georeferenced content when near to a shop, the *shop* category will be assigned to his/her trace.

In Figure 8, we can see a visualization for the POIs in distinct colors depending on their category (red, green, blue, and purple circles represent distinct categories of POIs), where each circle size represents the score (proportional to the number of posts nearby). Tourists' traces (lines) are depicted with the color of the POIs' category: it can be seen that the violet user traces come closer to many architectural POIs, than to other categories of POIs, since the violet color is associated with architectural POIs and the stronger is the color of the trace, the higher is the score of the trace.

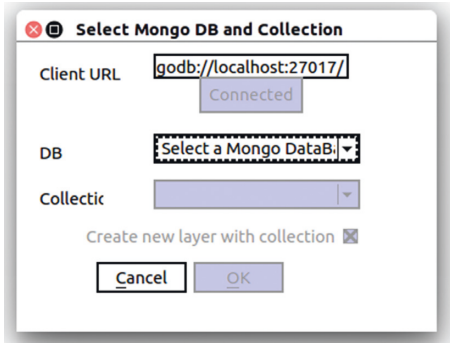

Figure 9: *J-CO-QGIS*: Db browser.

4.3. The J-CO-QGIS Plug-In for QGIS. In order to provide analysts with a powerful tool for querying and visualizing georeferenced JSON data within classical *GIS* software, we developed a plug-in for *QGIS* (currently the most popular open-source free *GIS* software) named *J-CO-QGIS*.

J-CO-QGIS provides several features, in particular

(i) a DB Browser, which allows users to connect to MongoDB persistent databases and to select the collections to show;

(ii) a Collection Viewer, which takes collections from MongoDB persistent databases and loads those objects with geometrical representation (having the `~geometry` field) into QGIS, so as to map them;

(iii) a Query Issuer, which is a text editor that allows users to write J-CO-QL queries and to send them to the *J-CO-QL Engine*.

When the plug-in is launched, it shows the *DB Browser* window (Figure 9): first of all, the user must specify the connection string to the *MongoDB* server (the connection string has the form mongodb://user:password@example.com/the_database? authMechanism = SCRAM-SHA-1). By clicking on the *Connect* button, the plug-in actually connects to the specified *MongoDB* server and shows the list of available databases. By browsing this list, shown in the *DB* list-box, the user chooses the database from which to get the desired collection. The content of the *Collection* list-box is updated, with the list of collections available in the chosen database.

At this point, the *Collection Viewer* window is open (Figure 10). In this window, it is possible to select objects of interest to show as a *QGIS* information layer; the right-hand side area shows the full structure of selected *JSON* objects. The *Insert selection to MongoDB* button reexports the selected objects into a new collection within the persistent database, whose name will be asked in a prompt. The *OK* button shows the layer containing the selected objects on the map.

As an example, suppose we have two collections, named `POIs` and `TrackedPosts` (deeply described in Section 4.2.4). The former describes Points of Interest and the latter the location from which tourists wrote geotagged posts to a social network. Figure 11 depicts the layers created by selecting objects in the two collections: POIs are depicted as colored circles, while posts' locations are depicted as red stars (the user can choose how to visualize geotags).

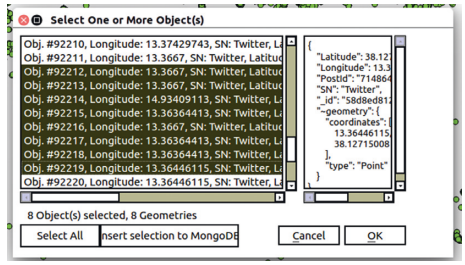

Figure 10: *J-CO-QGIS*: collection viewer.

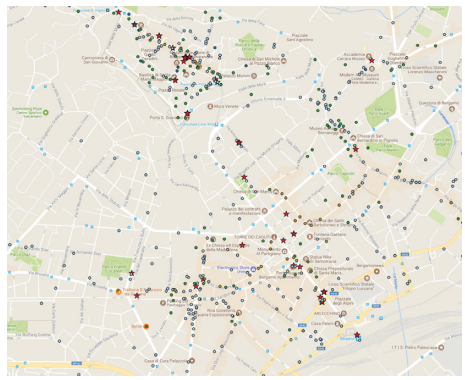

Figure 11: Visualization of selected georeferenced posts (red stars) over points of interests in Bergamo (colored points).

4.4. The Treets Web Application. *Treets* is a web application developed to provide analysts with a visual-analysis tool for geolocated tweets gathered by the *FollowMe* suite (Section 4.1). It has been designed to enable people, that do not have coding or GIS skills to explore data, find out information from trajectories, texts, and pictures in tweets, to help them in the analysis process. To achieve this goal, *Treets* provides features such as filtering, visualization, and data exporting.

In this section, we present functionalities of *Treets*, to show how this visual-analysis tool can help analysts in their exploratory activity.

In the left-hand side of the home page (Figure 12), a map dynamically shows traces and tweets. The panel in the right-hand side of the home page shows some information and provides filters (later discussed and shown in Figure 13).

On the map, tweets are represented as points. Clicking on a point, a pop-up window appears, containing details of the chosen tweet, including text, images, and external links (Figure 14). By clicking on the pop-up window, a new browser tab is open, and the user is redirected to *Twitter* web site, namely to the page showing the tweet itself.

On the map in the home page of *Treets*, tweets posted by the same user are linked in the chronological order by colored lines forming a *trace*; each trace has a different color. By clicking on a trace, a pop-up window appears with the user name, the number of tweets in the trace and a button to export tweets in the trace as a CSV (Comma Separated Value) file.

As far as filtering of tweets shown on the map is concerned, *Treets* provides five different alternatives:

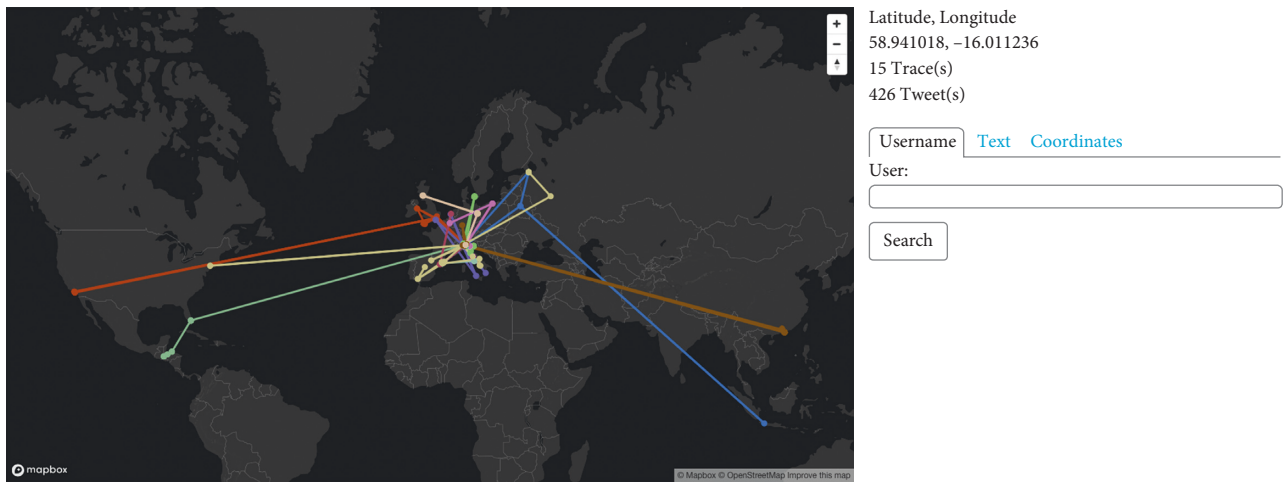

Figure 12: *Treets* visual-analysis tool: home page.

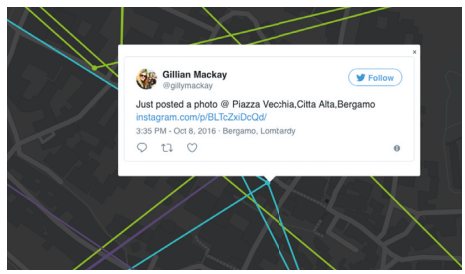

Figure 13: *Treets*: search form.

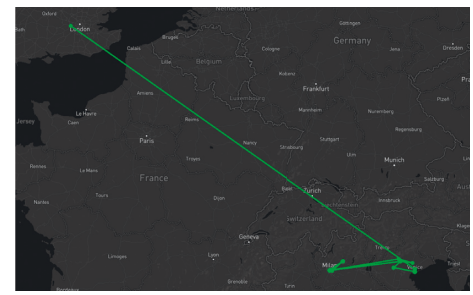

Figure 15: *Treets*: trace of a Californian tourist.

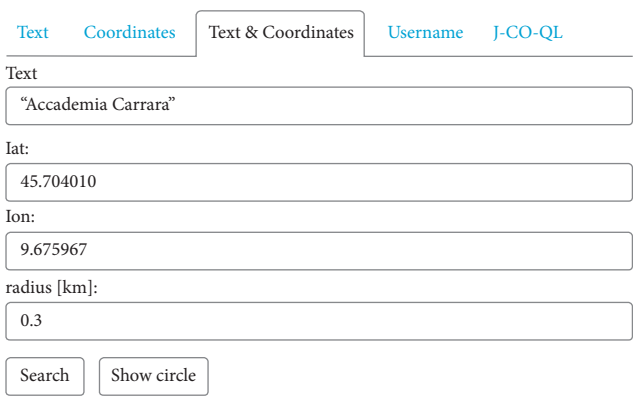

Figure 14: *Treets*: visual inspection of a tweet on the map.

(i) *By Username*: to retrieve the trace of a desired user (see, e.g., the trace in Figure 15, that shows the trip of a Californian tourist).

(ii) *By Text*: to find tweets containing the desired text and display the full traces they are part of on the map. Double quoted texts permit to find an entire sentence within texts e.g., to look for Accademia Carrara, the ancient art museum in Bergamo, the search string is "`Accademia Carrara`," which is different from "`Carrara Academy`" (anyway, the search is performed in a case-insensitive way).

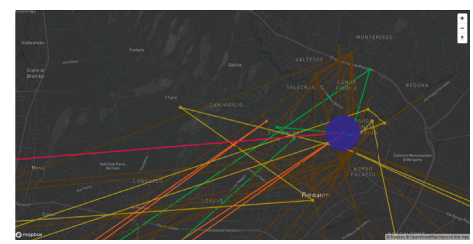

Figure 16: *Treets*: traces with at least a tweet in the search (blue) area.

(iii) *By Location*: by double clicking on the map, the Latitude and Longitude fields in the search form will be automatically filled (they can also be manually edited); thus, the desired radius of the circumference can be specified; the *Show Circle* button shows the search circle on the map; the *Search* button will filter tweets falling in the search circle and all traces with at least one selected tweet will be shown on the map (Figure 16).

(iv) *By Location and Text*: this search option allows analysts to combine location-based and text-based filtering.

(v) *By J-CO-QL*: *J-CO-QL* queries can be specified; furthermore, it is also possible to select a *MongoDB* server and a database, from which to select a collection of JSON objects to show (only geotagged

objects) on the map; this way, more skilled analysts can make use of one single web tool for both performing integration and transformation of data sets and to visualize them on the map.

As an example, we investigated tweets gathered by the *FollowMe* suite in order to discover unusual tourists that visited the city of Bergamo, with respect to our expectation of *usual tourist*. By performing a text search, looking for tweets talking about *Accademia Carrara*, that is, the ancient art museum in Bergamo, we found out the tourist whose trace is shown in Figure 15: this is a Californian tourist, that landed in Milan Linate airport, visited some museums in Milan, then went to Bergamo to spend some days, and then moved to Venice making some intermediate stages. Looking at texts, it is clear that this tourist is an art passionate. Notice that even though this tourist is not statistically relevant alone, an analysis could reveal interesting outcomes, suggesting improvement of synergy among museums and cities, building a network in northern Italy to attract such kind of tourists. Furthermore, from a technical point of view, it suggests to develop tools to perform analysis and discovery tasks on an automated basis that would never be possible to conduct by hand.

4.5. Analysis and Discovery Tools. In the *Urban Nexus digital environment*, this section is the newest and less developed. The idea is to develop tools that perform knowledge discovery tasks. The reader can think about data mining tools, but also other kind of analysis are acceptable; it depends on what the methodology asks for.

Currently, we identified the following needs:

(i) Analysis of paths followed by users, in terms of visited places and visited POIs.

(ii) Discovering the most frequent patterns of personal spatial dimensions, by means of frequent itemset mining techniques.

(iii) Trajectory analysis techniques could be developed, in order to better understand how city users move in the city.

Currently, we have worked on data collected by the *FollowMe* suites, that is, geolocated tweets of possibly traveling users. The analysis of such a data set inspired a novel technique to study the *centrality* of places, that is, rank places to let the central places in tours emerge; this technique is described in Section 4.5.1.

As far as discovering the most frequent patterns of personal spatial dimensions is concerned, we are going to address this problem. In our opinion, having the set of places visited by one single person, the well-known *frequent itemset mining* technique [19–21] could be very helpful in revealing common patterns of space usage. In particular, we could find out common patterns of one single person, as well as common patterns of groups of people; we also foresee that unexpected communities could emerge. Certainly, we are going to apply this approach to collected tweets, but we plan to adopt a volunteered approach to gather more detailed data.

Finally, better insights could be provided by trajectory analysis techniques. In previous works [39, 40], some algorithms for clustering trajectories detected from tweets gathered by the *FollowMe* suite were developed and tested. However, we think that the results are not satisfactory for the investigation methodology developed within the *Urban Nexus* project; consequently, they are not part of the *Urban Nexus digital environment*.

4.5.1. Node Rank. We now present the *Node Rank* technique, developed to discover centrality of places visited by tourists.

Consider a corpus of *geolocated* tweets, gathered by the *FollowMe* suite (Section 4.1), containing the traces of tourists that visited a given city. A tweet t has a *date_time* property, that is, the date and time the tweet was posted, as well as a *user_id*, denoting that the user that posted the tweet. For each user, it is possible to sort his/her tweets on the basis of property *date_time*, thus obtaining the *trip* of user *user_id*:

$$\text{Trip}(\text{user\_id}) = \langle t_1, \ldots, t_n \rangle, \tag{3}$$

where $n \geq 1$ and such that, for each pair t_i and $t_{(i+1)}$, it is $t_i \cdot \text{date\_time} < t_{(i+1)} \cdot \text{date\_time}$.

Being geolocated, that is, with associated latitude and longitude of a point on the earth, a tweet t corresponds either to the place from which the user posted the tweet, or to the place the user selected while geotagging the tweet. The consequence is that a trip describes in which order a user moved and which places he/she visited, or places he/she was referring to.

Thus, we can represent a trip on a graph, where nodes correspond to places (on the basis of their coordinates), while edges represent a move of a user from a place denoted by a tweet t_i to a place denoted by a tweet $t_{(i+1)}$. Obviously, in our analysis, it is not relevant to study a single trip, but all trips together, so in the same graph, we represent all collected trips.

As an example, Figure 17 shows a sample graph that represents 13 places (nodes) and 12 moves (edges) that are obtained from several trips of different users described by their tweets. As far as nodes are concerned, the bold number in the top left corner is the node identifier, while the number reported in the center of nodes is the number of tweets posted in that place. As far as edges are concerned, the number reported beside an edge denotes the number of moves emerged from tweets.

At this point, we can formulate the problem: *we want to define a ranking measure for places, depending on the movement of citizens, able to measure if a place is central, on the basis of incoming and outgoing moves (edges).*

Hereafter, we present the formal definition of the *Node Rank* approach. A complete description can be found in [41].

The Node Rank Method. The *Node Rank* method was inspired by the *Page-Rank* method [18, 42], due to the similarity between hypertextual links and moves of tourists in a network modeling a city. The idea is that when a tourist is visiting a city, his/her decision to visit a place could be influenced not only by the popularity of places from which he/she come from, but also by the popularity of places he/she

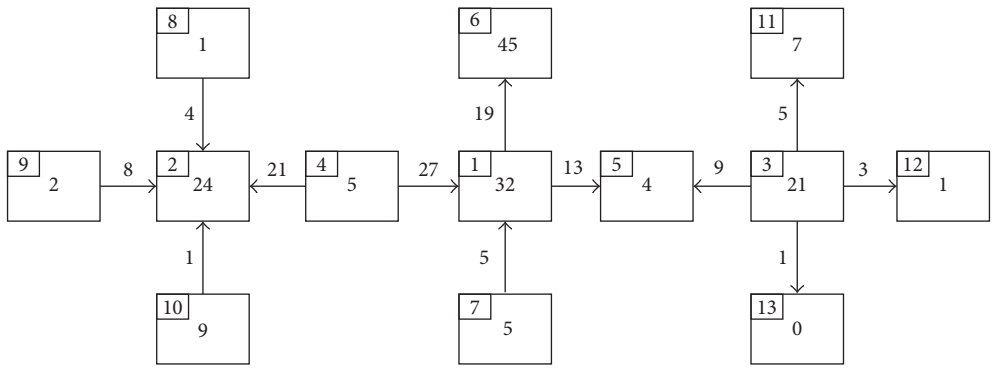

Figure 17: Sample graph of visited places.

is going to visit. Then, it is necessary to consider different kinds of relevance:

(i) *Inbound Relevance.* A place (a node in the graph) receives an inbound relevance, that is, a relevance influenced by the popularity of nodes from which inbound edges start.

(ii) *Outbound Relevance*. A place receives an outbound relevance, that is, a relevance influenced by the popularity of nodes to which outbound edges point to.

(iii) *Compound Relevance*. The relevance of each place is obtained by composing inbound relevance and outbound relevance.

To illustrate the idea, consider the sample graph depicted in Figure 17. If we consider the inbound relevance, node 2 should be intuitively the most relevant, since it has four incoming edges; this means that the associated place is of interest for many tourists and is the target where to end the trip. If we consider node 3, we can see the opposite situation, that is, it has four outgoing edges, so it represents a place that is important as starting point for tourist trips. Therefore, we argue that the rank of node 2 and node 3 should be the same.

But what about node 1? It has two incoming nodes and two outgoing nodes, so its role is different with respect to the role played by node 2 (a favorite target) and node 3 (a favorite starting point): it is crucial as intermediate place through many trips, that is, it is the place where many tourist passed, during their trips, in the middle of their paths. Since the goal of our research is to discover such crucial places, we need to combine both inbound relevance and outbound relevance.

The basic set of equations computes either the inbound relevance or the outbound relevance of each node u, denoted as $\mathrm{NR}_{\mathrm{in}}(u)$ and $\mathrm{NR}_{\mathrm{out}}(u)$, respectively.

$$\mathrm{NR}_{\mathrm{in}}(u) = (1-d) + d \sum_{v \in \mathrm{IB}(u)} \frac{\mathrm{NR}_{\mathrm{in}}(v)}{N_{\mathrm{in}}(v)},$$
$$\mathrm{NR}_{\mathrm{out}}(u) = (1-d) + d \sum_{v \in \mathrm{OB}(u)} \frac{\mathrm{NR}_{\mathrm{out}}(v)}{N_{\mathrm{out}}(v)}, \tag{4}$$

where $N_{\mathrm{in}}(u)$ is the number of incoming edges, while $N_{\mathrm{out}}(u)$ is the number of outgoing edges. $\mathrm{IB}(u)$ and $\mathrm{OB}(u)$ denotes, respectively, nodes connected to u by an inbound edge and an outbound edge for node u.

Note that we obtain two systems of equations: one for inbound ranks and one for outbound ranks.

For each node, it is necessary to combine inbound and outbound ranks, by means of the following formula.

$$\mathrm{NR}(u) = \left(\mathrm{NR}_{\mathrm{in}}(u) + \mathrm{NR}_{\mathrm{out}}(u)\right) \times \mathrm{adj}(u). \tag{5}$$

The sum of the inbound and outbound ranks is corrected by means of an adjustment factor:

$$\mathrm{adj}(u) = 1 - \frac{1}{1 + \log_k\left(1 + N_s(u) + N_{\mathrm{in}}(u) + N_{\mathrm{out}}(u)\right)}, \tag{6}$$

where $N_s(u)$ is the sum of number of single tweets, and self-loops for node u. The adj parameter has been introduced to deal with single tweet trips, that is, with users that posted only one tweet in their trips, as well as self-loops, that is, users that posted two or more consecutive tweets from the same place. The base of the logarithm k is based on the maximum value of

$$N_{\mathrm{tot}} = \max_{u \in B}\left(N_s(u) + N_{\mathrm{in}}(u) + N_{\mathrm{out}}(u)\right), \tag{7}$$

that is, the place with maximum number of tweets (B is the total set of nodes in the graph). So, the base of the logarithm k is defined as

$$k = \sqrt[2]{N_{\mathrm{tot}}}. \tag{8}$$

The final rank of node is the sum of the inbound rank and the outbound rank. However, we could obtain ranks that still precisely do not represent the actual importance of place: from our perspective, a place must be valuable for both tourism and mobility. For these reasons, given the same number of links, a node with variety of incoming and outgoing links is more important than uniform ones. Thus, the final ranks $\overline{\mathrm{NR}}(u)$ are given by the following equations:

$$\overline{\mathrm{NR}}(u) = \left(\mathrm{NR}_{\mathrm{in}}(u) + \mathrm{NR}_{\mathrm{out}}(u)\right) \times \left(1 + \frac{\left(N_{\mathrm{in}}(u)\right)/\left(N_{\mathrm{out}}(u)\right)}{\mathrm{perc}}\right), \tag{9}$$

$$\overline{\mathrm{NR}}(u) = \left(\mathrm{NR}_{\mathrm{in}}(u) + \mathrm{NR}_{\mathrm{out}}(u)\right) \times \left(1 + \frac{N_{\mathrm{out}}(u)/N_{\mathrm{in}}(u)}{\mathrm{perc}}\right). \tag{10}$$

If $N_{in}(u) < N_{out}(u)$, (9) is used to compute the final score for node u, otherwise (10) is used. The correction factor (by which $(NR_{in}(u) + NR_{out}(u))$ is multiplied) is introduced, in order to benefit nodes with more variety, for instance nodes with the same number of incoming and outgoing edges receive the maximum boost, meanwhile nodes with only one of the two types of edges do not receive any boost. The factor *perc* must not be chosen equal to zero: if its value is positive and near to zero, it will produce a great boost, instead if its value is a large number it will nullify the possible increase; moreover, we can also choose a negative value if we want to give a negative boost.

For our purpose, perc = 5 is suitable, since it gives a maximum boost of 20%, with respect to the noncorrected ranks $(NR_{in}(u) + NR_{out}(u))$.

The algorithm we developed to solve the equations so far defined is outside the scope of this paper (see [41]). However, we report a sample result obtained by analyzing a pool of tweets concerning the city of Bergamo. Table 1 shows how places are ranked by the *Node Rank* approach, on the basis of analyzed tweets, while Table 2 shows the same places ordered on the basis of the simple number of tweets posted in that places. The reader can see how the order changes: the places that gain the top-most positions emerge as the key places through which most of tourists pass before visiting other places during their trip.

The potential applications of this approach are manifold, at various orders of magnitude: changing the scale, we can analyze local areas (such as cities), regions, countries, and so on.

Table 1: Rank places of Bergamo.

Place	Score
Aeroporto, Orio al Serio	121,60,882
Oriocenter, Orio al Serio	1,915046
Piazza vecchia, Città Alta	1,203989
Stazione Inferiore Funicolare	1,187952
Sistema Piacentiniano	0.852875
Porta Nuova	0.698300
Piazza Mercato delle Scarpe, Città Alta	0.521350
Fiera di Bergamo	0.490758
Stazione FS	0.475586
Piazza Giacomo Matteotti	0.459371

Table 2: Number of tweets for place in Bergamo.

Place	Number of tweets
Aeroporto, Orio al Serio	1448
Stazione Inferiore Funicolare	103
Oriocenter, Orio al Serio	100
Piazza Vecchia, Città Alta	70
Piazza Giacomo Matteotti	46
Piazza Pontida	28
Da Mimmo	23
Sistema Piacentiniano	23
Castello di San Vigilio	20
Stazione FS	16

5. Conclusions

The analysis of mobility by exploiting user-generated data through mobile devices is an ambitious task. In fact, it demands for an integrated multidisciplinary approach: geographers and analysts must define a suitable methodology to gather mobile-users data sets and investigate them; computer scientists must develop novel and powerful tools and frameworks to perform very complex tasks on data, such as data gathering, transformation, integration, analysis, and visualization. This is the ambition of the *Urban Nexus* project.

Based on this premise, we can summarize the contributions of this paper. First of all, we introduced a novel perspective toward smart cities, where a city is smart since it learns about how city users live in the city, in order to improve infrastructures and services on the basis of needs in terms of mobility; the concept of rhizome, born in other scientific areas, is providing a novel and crucial approach, supporting the reticular view of city-life experience.

Second, the paper provides an investigation methodology that could be used to drive any kind of analysis process of mobile-user data, towards the reconstruction of rhizomes.

Third of all, we show the large variety of tools necessary to perform such an investigation, moving from a real-life data set gathered from *Twitter*; this data set contains traces of traveling people through Europe, reconstructed by following georeferenced messages they posted during their trips. Then, the *J-CO* framework supports the integration and transformation of data sets, while the *Treets* web application allows analysts to deeply investigate traces and habits of mobile users. Finally, towards the reconstruction of collective rhizomes, we presented the *Node Rank* method, that is able to give a quantitative evaluation of visited places on the basis of the reticular approach. In practice, we could say that a large variety of Big Data requires a large variety of tools to analyze them.

Obviously, we cannot consider our research to be at the end. Paradoxically, we are at the beginning of a long research activity. This is true in relation to several aspects. First of all, traces of traveling users detected through *Twitter* are not the only possible ones. Many other sources could be used, such as the one produced by *Google Timeline*, that users could voluntarily provide; they could give a different perspective about habits of users, but the investigation methodology does not change.

Second, novel strategies to transform and integrate data could be developed, on the basis of automatic algorithms that could cluster data and characterize them based on texts and images; furthermore, sentiment analysis techniques could be applied to detect sentiment polarity of messages.

Finally, as sketched throughout the paper, data mining techniques based on the concept of frequent itemset mining could be developed and applied, so as to actually reconstruct rhizomes from global perspectives.

We are confident in the development of the three areas mentioned above. Thus, many research activities could be and will be performed, definitely enlarging the *Urban Nexus digital environment* and, at the end, the possibility for smart

cities to learn from city users' experience. Nevertheless, the overall integrated approach, which is the main contribution of this paper, is still suitable and perfectly adequate to support our future work.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] J. Lévy, *L'Invention du Monde: Une Géographie de la Mondialisation*, Presses de Sciences Po Paris, Paris, France, 2008.

[2] E. W. Soja, *Postmetropolis Critical Studies of Cities and Regions*, Nlackwell, London, UK, 2000.

[3] E. Casti and F. Burini, *Centrality of Territories: Verso la Rigenerazione di Bergamo in un Network Europeo*, Bergamo University Press, Bergamo, Italy, 2015.

[4] G. Deleuze, F. Guattari, J. V. Pérez, and U. Larraceleta, *Rizoma: (Introducción)*, Pre-Textos, València, Spain, 2003.

[5] J. Lévy, T. P. L. Romany, and O. P. Maitre, "Rebattre les cartes. topographie et topologie dans la cartographie contemporaine," *Réseaux*, vol. 34, pp. 17–52, 2016.

[6] F. Girardin, F. Calabrese, F. Dal Fiore, C. Ratti, and J. Blat, "Digital footprinting: uncovering tourists with user-generated content," *IEEE Pervasive Computing*, vol. 7, no. 4, 2008.

[7] M. F. Goodchild, "Citizens as sensors: the world of volunteered geography," *GeoJournal*, vol. 69, no. 4, pp. 211–221, 2007.

[8] M. Batty, "Big data, smart cities and city planning," *Dialogues in Human Geography*, vol. 3, no. 3, pp. 274–279, 2013.

[9] M. Batty, "Big data and the city," *Built Environment*, vol. 42, no. 3, pp. 321–337, 2016.

[10] M. Graham and T. Shelton, "Geography and the future of big data, big data and the future of geography," *Dialogues in Human Geography*, vol. 3, no. 3, pp. 255–261, 2013.

[11] R. Kitchin, "Big data and human geography: opportunities, challenges and risks," *Dialogues in Human Geography*, vol. 3, no. 3, pp. 262–267, 2013.

[12] M. Lussault, *L'Homme Spatial: la Construction Sociale de l'Espace Humain*, Vol. 363, Seuil Paris, Paris, France, 2007.

[13] F. Burini, D. E. Ciriello, A. Ghisalberti, and G. Psaila, "The urban nexus project: when urban mobility analysis, vgi and data science meet together," in *Mobile Information Systems Leveraging Volunteered Geographic Information for Earth Observation*, pp. 111–130, Springer, Cham, Switzerland, 2018.

[14] R. Kitchin, *What Does Big Data Mean for Official Statistics?*, Discover Society, Bristol, England, 2015.

[15] H. Butler, M. Daly, A. Doyle, S. Gillies, S. Hagen, and T. Schaub, "The geojson format," Tech. Rep., Internet Engineering Task Force, Fremont, CA, USA, 2016.

[16] K. Banker, *MongoDB in Action*, Manning Publications Co., Shelter Island, NY, USA, 2011.

[17] C. Strauch, *Nosql Databases*, 2011, http://www.christof-strauch.de/nosqldbs.pdf.

[18] S. Brin and L. Page, "Reprint of: the anatomy of a large-scale hypertextual web search engine," *Computer Networks*, vol. 56, no. 18, pp. 3825–3833, 2012.

[19] R. Agrawal, T. Imieliński, and A. Swami, "Mining association rules between sets of items in large databases," *ACM SIGMOD Record*, vol. 22, no. 2, pp. 207–216, 1993.

[20] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," in *Proceedings of 20th International Conference on Very Large Data Bases, VLDB*, vol. 1215, pp. 487–499, San Francisco, CA, USA, September 1994.

[21] G. Grahne and J. Zhu, "Fast algorithms for frequent itemset mining using fp-trees," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 10, pp. 1347–1362, 2005.

[22] R. Meo, G. Psaila, and S. Ceri, "A new SQL-like operator for mining association rules," in *Proceedings of the 22st VLDB Conference*, Bombay, India, September 1996.

[23] X. Wu, C. Zhang, and S. Zhang, "Database classification for multi-database mining," *Information Systems*, vol. 30, no. 1, pp. 71–88, 2005.

[24] X. Wu and S. Zhang, "Synthesizing high-frequency rules from different data sources," *IEEE Transactions on Knowledge and Data Engineering*, vol. 15, no. 2, pp. 353–367, 2003.

[25] S. Zhang, Q. Chen, and Q. Yang, "Acquiring knowledge from inconsistent data sources through weighting," *Data & Knowledge Engineering*, vol. 69, no. 8, pp. 779–799, 2010.

[26] S. Zhang, X. Wu, and C. Zhang, "Multi-database mining," *IEEE Computational Intelligence Bulletin*, vol. 2, no. 1, pp. 5–13, 2003.

[27] M. Haklay, "Citizen science and volunteered geographic information: overview and typology of participation," in *Crowdsourcing Geographic Knowledge*, pp. 105–122, Springer, Berlin, Germany, 2013.

[28] B. Hecht and M. Stephens, "A tale of cities: urban biases in volunteered geographic information," *ICWSM*, vol. 14, pp. 197–205, 2014.

[29] D. Z. Sui, S. Elwood, and M. Goodchild, *Crowdsourcing Geographic Knowledge: Volunteered Geographic Information (VGI) in Theory and Practice*, Springer Science & Business Media, Berlin, Germany, 2012.

[30] M. Zook, M. Graham, T. Shelton, and S. Gorman, "Volunteered geographic information and crowdsourcing disaster relief: a case study of the Haitian earthquake," *World Medical & Health Policy*, vol. 2, no. 2, pp. 7–33, 2010.

[31] E. Casti, *Reflexive Cartography: A New Perspective in Mapping*, Vol. 6, Elsevier, New York, NY, USA, 2015.

[32] A. Cuzzocrea, G. Psaila, and M. Toccu, "Knowledge discovery from geo-located tweets for supporting advanced big data analytics: a real-life experience," in *Model and Data Engineering*, pp. 285–294, Springer International Publishing, Berlin, Germany, 2015.

[33] G. Bordogna, A. Cuzzocrea, L. Frigerio, G. Psaila, and M. Toccu, "An interoperable open data framework for discovering popular tours based on geo-tagged tweets," *International Journal of Intelligent Information and Database Systems*, vol. 10, no. 3-4, pp. 246–268, 2017.

[34] A. Cuzzocrea, G. Psaila, and M. Toccu, "An innovative framework for effectively and efficiently supporting big data analytics over geo-located mobile social media," in *Proceedings of the 20th International Database Engineering & Applications Symposium*, pp. 62–69, ACM, Montreal, QC, Canada, ACM, July 2016.

[35] T. E. Chow, "Geography 2.0: a mashup perspective," in *Advances in Web-Based GIS, Mapping Services and Applications*, S. Li, S. Dragićević, and B. Veenendaal, Eds., pp. 15–36, Taylor & Francis Group, London, ISBN 978-0-415-80483-7, 2011.

[36] G. Bordogna, S. Capelli, D. E. Ciriello, and G. Psaila, "A cross-analysis framework for multi-source volunteered, crowdsourced, and authoritative geographic information: the case study of volunteered personal traces analysis against transport network data," *Geo-spatial Information Science*, pp. 1–15, 2017.

[37] S. Capelli, P. Fosci, F. Marini, and G. Psaila, "J-co-ql: A flexible query language for complex geographical analysis of heterogeneous geo-tagged json data sets," Tech. Rep., University of Bergamo, Bergamo, Dalmine, Italy, 2017.

[38] G. Bordogna, S. Capelli, and G. Psaila, "A big geo data query framework to correlate open data with social network geo-tagged posts," in *Proceedings in AGILE 2017 International Conference*, Paris, France, July 2017.

[39] G. Bordogna, L. Frigerio, A. Cuzzocrea, and G. Psaila, "Clustering geo-tagged tweets for advanced big data analytics," in *Proceedings of Big Data (BigData Congress), 2016 IEEE International Congress*, pp. 42–51, IEEE, San Francisco, CA, USA, June 2016.

[40] G. Bordogna, L. Frigerio, A. Cuzzocrea, and G. Psaila, "An effective and efficient similarity-matrix-based algorithm for clustering big mobile social data," in *Proceedings of Machine Learning and Applications (ICMLA), 2016 15th IEEE International Conference*, pp. 514–521, IEEE, Anaheim, CA, USA, December 2016.

[41] N. Cortesi, K. Gotti, G. Psaila, F. Burini, K. T. Lwin, and M. Hossain, "A network-based approach to discover rank places visited by tourists from geo-located tweets," in *Proceedings of Software, Knowledge, Information Management and Applications (SKIMA 2017), 11th International Congress, IEEE, Chengdu, China*, 2017.

[42] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: bringing order to the web," Tech. Rep., Stanford InfoLab, Stanford, CA, USA, 1999.