Clustering Geo-Tagged Tweets for Advanced Big Data Analytics

Gloria Bordogna
CNR-IREA, Italy
bordogna.g@irea.cnr.it

Luca Frigerio
CNR-IREA, Italy
frigerio.l@irea.cnr.it

Alfredo Cuzzocrea
DIA Department
University of Trieste and ICAR-CNR, Italy
alfredo.cuzzocrea@dia.units.it

Giuseppe Psaila
DIGIP Department
University of Bergamo, Italy
psaila@unibg.it

***Abstract*—In this paper, we introduce an original approach that exploits timestamped geo-tagged messages posted by *Twitter* users through their smartphones when they travel to trace their trips. An original clustering technique is presented, that groups similar trips to define tours and analyze the popular tours in relation with local geo-located territorial resources. This objective is very relevant for emerging *big data analytics tools*.**

Tools developed to reconstruct and mine the most popular tours of tourists within a region are described which identify, track and group tourists' trips through a knowledge-based approach exploiting timestamped geo-tagged information associated with *Twitter* messages sent by tourists while traveling.

The collected tracks are managed and shared on the Web in compliance with OGC standards so as to be able to analyze the characteristic of localities visited by the tourists by spatial overlaying with other open data, such as maps of Points Of Interest (POIs) of distinct type. The result is an novel Interoperable framework, based on web-service technology.

***Keywords*-Big Data Analytics; Knowledge Discovery from Geo-Located Tweets; Intelligent Systems.**

I. Introduction

Tourism is an important driver for the development of a territory. Being able to answer the question "where do most tourists actually go and what are the most popular tours?" could be really useful to public and local administrators for understanding the dynamics of mass tourism and making better territorial planning.

Nowadays, we are entering the so called *big geo-data era* (e.g., [1], [2]), i.e. we are in the condition of having a huge amount of information that could be exploited to answer the above question, whenever appropriately filtered and analyzed. This aspect plays a critical role, especially when dealing with *Big Data Analytics* (e.g., [3], [4], [5], [6], [7], [8], [9], [10]). In this respect, the widespread diffusion of smart devices has favored a new way of sharing impressions about places visited by travelers on social networks; people can post geo-tagged (i.e., geo-localized) and time-stamped (i.e., with a date) messages and pictures; these meta-data give a ready to use key to know where and when the messages were sent.

Among all social networks, *Twitter* (as well as other social networks that adopt the same approach) is particularly attractive to the purpose of searching messages that have something to do with travels: in fact, the limitation on the length of tweeter messages makes them a suitable means to exchange impressions in a couple of words, when people are moving around and are not keen to write long posts. Furthermore, they represent a kind of voluntary contribution, because users voluntarily install the (*Twitter*) app and voluntarily post messages (tweets) so that every user can see messages by other users without limitations. Even if only 20 % of the tweets are geo-tagged, this posts' information, hardly acquirable with traditional survey methods.

The above considerations motivate the research that we describe in this paper. It is necessary to design an architecture of tools that: 1) are able to continuously gather geo-localized posts from social networks; 2) provide novel methods to analyze travelers' trips, on the basis of gathered posts; 3) enable to visualize results in an integrated fashion, i.e., exploiting open geo-data sources in an integrated way; 4) make results available to other geographic services. This objective is very relevant for emerging *big data analytics tools* (e.g., [11], [12], [13], [14], [15]).

In order to be compliant with modern standards and to enable interoperability with open data sources and other services, we defined a *web service architecture*, in which standard components and external sources are integrated with services and suites of tools specifically developed for this research. One suite (called the *FollowMe Suite* has been described in a previous work (see [16]), queries social networks to discover posts sent by travelers from specific geographic areas, identifies the possible tourists and follow their tweets to reconstructs their time lines, i.e., the history of posted messages. A new suite (called the *Trip Analysis Suite*) proposes and implements an original method for knowledge-based trip clustering. Finally, a geo-portal (named *Tourist Tracker Geo-portal*) enables to analyze most popular tours (i.e., clustered trips) contextualizing them with the characteristics of the territory ([17]).

978-1-5090-2622-7/16 $31.00 © 2016 IEEE
DOI 10.1109/BigDataCongress.2016.78

A main characteristic of the approach is that it is a knowledge-based approach. In facts, our goal is to semantically analyze the tracked tourist trips, by correlating them with information regarding POIs such as commercial, social and natural attractions of a specific territory of interest. The final and ambitious goal is to be able to guess which are the most plausible resources the tourists have visited.

Different knowledge of the territorial resources can be exploited, in order to perform distinct analysis of the tourists' trip. For example, in our experiments we represented tours as sequences of zip areas they crossed; however, it could be possible to represent trips as sequences of the closest POIs they came across. This is the knowledge-based geo-clustering method, developed as part of the *Trip Analysis Suite* and presented in Section IV.

Furthermore, by publishing the popular tours by an interoperable Web GIS enables their mapping and analysis contextually to other multi-source open geo-spatial data themes relative to the territory resources such as hysterical and world heritage, naturalistic places, shopping centers. This is provided by the *Tourist Tracked Geo-portal*, presented in Section III, A case study, discussed in Section V, shows the effectiveness of the approach. Extensive experiments are reported in Section VII.

II. Problem Definition

In the paper [16], we started our research addressing the problem of identifying *Twitter* users possibly traveling to regions of interests. The method we defined is summarized by the following three steps.

1) *Hang Tweet Search.* First of all, we identified a pool of airports, in order that we collect geo-localized tweets posted in their geograpic area. These tweets, called *hang tweets*, are collected daily.
2) *Tracking Users.* For each hang tweet, we follow the timeline (i.e., the history of posted tweets) of the posting user, for the next 8 days after the hang tweet date. This tweets are called *tracked tweets*.
3) *Trip Querying.* Once hang tweets and users timelines are collected, trips that touched a region of interest are built, by filtering tracked tweets.

Completed these preliminary, yet fundamental, tasks, we are now ready to address the main focus of this paper, i.e., discovering frequent trips performed by users on the basis of *Geographic Partitioning*.

A. Geographic Partitioning

Tweets can be semantically correlated with the territories if the punctual position given by latitude and longitude is generalized into a *geo-slot*, i.e., an area whose label (also called *geoslotID*) is meaningful for humans.

For example, if administrative entities are used, such as small municipalities, trips become sequences of visited municipalities, irrespective of the specific POI. But if the scale of analysis is continental, wider administrative areas, such as regions, are more suitable for the analysis (trips become sequences of visited regions).

Finally, a region slots representation could be a grid layer of regularly shaped cells that contain a specific POI (e.g., a commercial area) and its neighborhood; this choice could be useful to infer which POIs might have been visited.

B. Tour Mining

After geo-partitioning, trips are represented as sequences of *geoslotID*. At this point the problem becomes to discover which are the most popular *geoslotID* in trips. This is a clustering problem.

Thus, our aim is to group trips (in the form of *geoslotID* sequences) into clusters that contains, as much as possible, similar trips; the similarity measure (see Section IV) is based on the inclusion relationships between trips. We call each cluster a *popular tour* (or simply *tour*), because it encompasses trips that share many visited *geoslotID*s.

C. Formal Definitions

First of all, we define the concept of geo-tagged tweets.

Definition 1: **Geo-Tagged Tweet.** A geo-tagged tweet d is defined as a tuple:

$$d\text{=(}\ u,\ c,\ t,\ tslot,\ geo,\ geoslotID\ \ )$$

where:

- *u* is the unique identifier of the author of the tweet;
- *c* is the textual content of the tweet represented by a string of 160 char length;
- *t* is the timestamp of the tweet, i.e., its date of creation;
- *tslot* is the time period when the tweet was sent;
- *geo* is the pair of geographic coordinates (latitude nd longitude) detected by the GPS from where the tweet was created.
- *geoslotID* is the geographic region identifier from where the tweet was sent.

□

On the basis of the concept of geo-tagged tweet, it is possible to define the next concepts.

Definition 2: **Trip.** A trip tr is then defined by the pair:

$$tr\text{=(}o\_airport,\ \langle d,order\rangle\ )$$

where:

- *o_airport* is the originating airport of the hang tweet of the trip;
- *¡d, order¿* is a list of pairs of tweets d and associated natural number (*order*) identifying the time order in the tourist trip relative to the hang tweet (*order=1*).

□

Finally, we can formally define the concept of *tour*.

Definition 3: **Tour.** A tour to is finally defined as a set of similar trips pair:

$$to\ =tr_1,...,\ tr_n$$

where

- *n* is the cardinality of the tour, which we name as the degree of popularity of the tour to.

- for any pair of trips tr_i, tr_j belonging to a tour to a minimum percentage of common geoslots exists: $\exists\ k$ *and* h such that $tr_i.d_k.geoslotID$ = $tr_j.d_h.geoslotID$.

□

III. The *Interoperable Framework* for Trips Tracking and Analysis

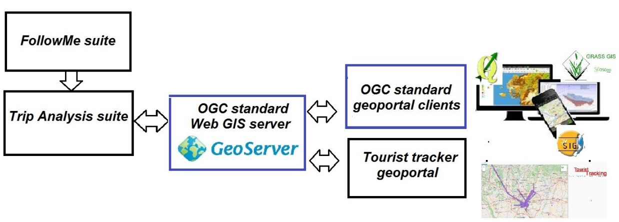

FollowMe suite
Trip Analysis suite
OGC standard Web GIS server
GeoServer
OGC standard geoportal clients
Tourist tracker geoportal
GRASS GIS
SIG
Tourist Tracking

Fig. 1. Architecture for the *trips identification and analysis*.

The Interoperable Framework we devised for tracking and analysing trips is depicted in Figure 1.

It is constituted by two main suites of tools: the *FollowMe Suite* [16] for trips identification and the *Trips Analysis Suite* for its geographic analysis.

A third component of the architecture is the *Geoserver* open source Web GIS, that is used to publish on the Web the analyzed trips in order to share then as open data, so that any OGC compliant geo-portal client, such as QGIS, can visualize them and spatially analyze trips with respect to other geo-spatial data.

A forth component of the framework is the *Tourists Tracker Portal*, that provides users with specific functionality to analyze popular tours, getting them from *Geoserver*.

The communication is based on the web service paradigm and related protocols and formats. The joint adoption of a web service architecture and of the OGC standard for open geographic data enables the easy extension with new dedicated services; this choice guarantees flexibility in the future development of the project. This way, the framework is actually interoperable, both internally (new services and components can be easily added) and externally (easy bidirectional exchange of data with external source and services).

In the rest of this section, we describe the tasks performed by each service.

A. The FollowMe Suite

This is the first service developed during the project. The interested reader can find details in [16]. Here, we briefly describe its three main functionalities.

1) The *FollowMe* suite queries *Twitter* API to find *hang tweet*, i.e., tweets posted in the area of the monitored airports.
2) For each user identified by means of hang tweets, the *FollowMe* suite queries (through *Twitter* API) his/her *Timeline*, i.e., the history of tweets posted by the user, to get *tracked tweets*: these are geo-localized tweets posted in the next 8 days after the date of the hang tweet. Both *hang tweets* and *tracked tweets* are stored in a local storage area.
3) Given an area of interest, trips that occur in that area are reconstructed and extracted, by querying *hang tweets* and *tracked tweets* previously stored in the local storage area. Reconstructed trips are exposed and exported through the web service interface.

B. The Trip Analysis Suite

On the functionalities provided by the *Trip Analysis Suite* this paper is mainly focused. It performs the geographical analysis and tour discovery on trips tracked by the *FollowMe* suite. Its internal architecture and functionality is described in Section IV. As a brief introduction, we can say that:

- first of all, tweets are geo-partitioned with a label that identifies the geo-slot that contains it;
- trips are clustered in order to discover popular tours, on the basis of the geo-slots.

Results of the tour discovery tasks are deployed to *Geoserver*.

C. Tourists Tracker Geo-Portal

Finally, the last service in our architecture is an OGC compliant geo-portal named *Tourists Tracker*. it provides the end user interface to visualize tours, by getting them from *Geoserver*.

Mainly, trips belonging to the same tour are depicted with the same color, while each tour is visualized with a different color; this way, for users it is easier to analyze different tours.

IV. The Trip Analysis Suite in Depth

In the previous section, we gave an overview of the Interoperable framework. In this section, we focus on the *Trip Analysis Suite*, the service that performs the tour discovery task.

A. Tour Discovery Methodology and Process

The *Trip Analysis Suite* actually performs the activities of knowledge discovery on trips collected by the *FollowMe Suite*.

As previously introduced (Section II), we propose a knowledge-based clustering method, where semantics is given by geo-slots in which punctual coordinates of tweets fall. Different geo-slots partitions gives different results. Analysts can import geographic description of geo-slots from external interoperable sources. For example, it is possible to analyze tours with respect to municipalities, regions, countries, neighborhoods, etc..

This means that row trips must be preprocessed before discovering tours. The tasks hereafter described constitute the knowledge discovery process performed by the *Trip Analysis Suite*.

1) Fetching of the *Geographic Slots* descriptions.
2) Fetching (from the *FollowMe Suite*) of the gathered trips w.r.t. an area of interest.

3) Geo-partitioning of tweets in trips, in order to semantically labeling tweets with geographic slots.
4) Clustering trips in order to identity *tours* (see Section II), i.e., groups of trips that mostly visited the same geographic slots.

B. Trip Geo-Partitioning

This task identifies which tweets visited which geographic slots. These are described by a vector representation. To this end, the *Trip Geo-partition* associates a region slot identifier, $geoslotID$, with each tweet by evaluating the inclusion/closeness of the tweets geotag geo inside the boundaries of the regions; this way, all tweets that are included in the same region slot get the same $geoslotID$. After this phase, a trip tr is represented by a string of $geoslotID$s sequentially ordered according to tweets timestamp *order*. This function may cause distinct trips visiting the same regions in the same order to be represented by equal strings. Nevertheless, if the order by which the regions are visited is different the strings only share sub-strings.

C. Trip Clustering/Tour Mining

The *Trip Clustering* task actually groups trips into tours based on their path similarity with respect to the geographic slots used for geo-partitioning.

This clustering algorithm is applied to discover the most popular tours followed by the tourists. It groups the trips whose strings of $geoslotID$s are similar. Basically it is an original hierarchical agglomerative single link clustering algorithm working on strings. The similarity between two trips tr_1 , tr_2, represented by the sequences of their *geoslotID*, is defined based on the fuzzy inclusion of $geoslotID$s in the two strings, i.e., the ratio of the number of common $geoslotID$s shared by a pair of strings with respect to the number of $geoslotID$s of the shortest string.

Formally, if $< geoslotID_{1,1}, ..., geoslotID_{1,n} >$ and $< geoslotID_{2,1}, ..., geoslotID_{2,m} >$ are the two strings (of respective length n and m) of geoslots of trips tr_1 and tr_2, then we define the similarity between the two trips as follows:

$$sim(tr_1, tr_2) = \frac{\sum_{i=1,...,n, j=1,...,m} (geoslotID_{1,i} = geoslotID_{2,j})}{2 \times min(n, m)}$$

Notice that $sim(tr_1, tr_2) = 0$ when no common geoslots exists between the two trips while $sim(tr_1, tr_2) = 1$, the maximum value, when all the geoslots of one of the two trips are included in the set of geoslots of the other trip.

We can now informally describe the clustering algorithm. The algorithm is based on an agglomerative approach.

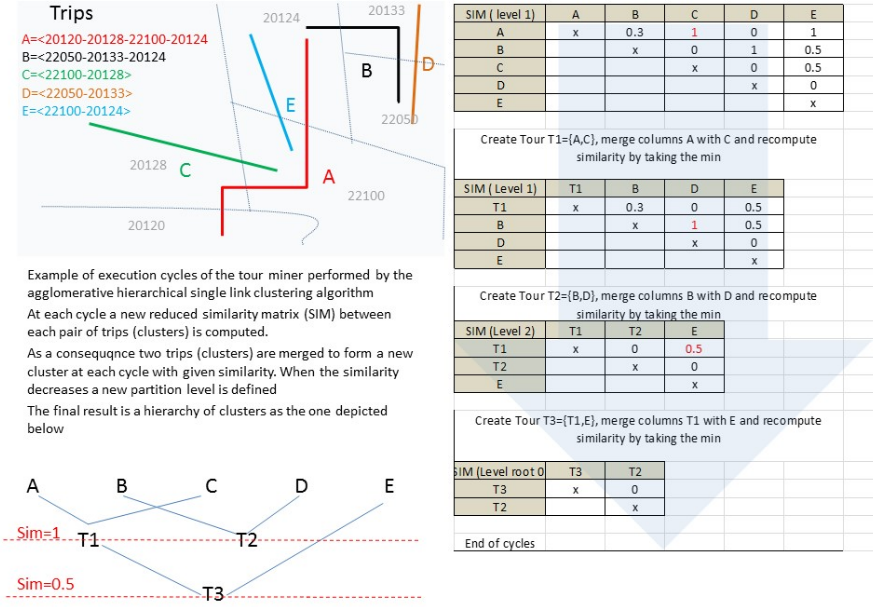
- The algorithm is starts by considering each trip as a singleton cluster, and then groups the pair of trips (clusters) sharing the greatest similarity $sim(tr_1, tr_2)$ at each cycle.
- The cycles are repeated until all trips belong to one single cluster (the root).
- Each time a cluster $c = \{tr_1, tr2\}$ is created, the similarity of any other trip (clusters) tr to the newly created cluster c is computed as follows:
$$sim(tr, c) = min(sim(tr_1, tr), sim(tr_2, tr))$$
By this computation of the similarity of a trip to a cluster we guarantee the generation of compact clusters: no trip is added to an already created cluster if it does not share at least a single geoslot with all the trips in the cluster, thus preventing to generate long and thin tours, consisting of trips which share geoslots only with one or a few other trips of the cluster.
- This way a hierarchy of clusters is built, a dendrogram, in which at each level we get a partition of the set of trips into a distinct set of clusters. Specifically, by descending the dendrogram from the leaves to the root, the clusters are characterized by equal or decreasing similarity degree at each level.
- Into the geodatabase, We store only the clusters of the partitions corresponding to a decrease of the similarity level with respect to the previous level.

At this point, the decision to select a partition of clusters to deploy on the Web must be made. Several criteria may be applied: for example, one can select the partition with a similarity degree above a desired threshold, so that the greater the threshold, the greater is the number of clusters and the smaller are the clusters in terms of contained trips. Another choice could be to compute a quality indicator of each partition, such as its entropy and partition coefficient, and then select the partition with greatest quality. Finally, a practical choice can be deploying on the Web more partitions of clusters by selecting from each of them only clusters with a minimum degree of popularity, i.e., a minimum number of contained trips. This allows the stakeholders to visually inspect the most popular tours with distinct levels of granularity (similarity).

To illustrate the rationale behind the clustering technique, in Figure 2 we report five sample trips, labeled from A to E (left upper corner). Matrices in the right side of Figure 2 reports the similarity measures at each step, and the dendogram in the bottom left corner shows the hierarchy of generated clusters. The algorithm proceeds as follows.

1) At first (top matrix) pairwise similarity measures among the five trips are computed. The pair with highest similarity is chosen (trips A amd C) and grouped together into the new tour $T1$.
2) A new matrix is computed, where trips A and C no longer appear; instead, tour $T1$ is reported and compared (in terms of similarity) with trips B, D and E.
Again, the pair with highest similarity is chosen (trips B and D), and the new cluster $T2$ generated.
3) In the third version of the matrix, the pair with highest similarity is constituted by cluster $T1$ and trip E. Consequently, the new cluster $T3$ is build, by aggregating cluster $T1$ and trip E. Cluster $T3$ is at a higher hierarchical level w.r.t. cluster $T1$, since it is included in cluster $T3$; $T1$ is child of $T3$.
4) Finally, the last matrix, compares clusters $T3$ and $T2$. They are completely disjoint (similarity is 0) and the clustering process ends.

The reader can notice how the similarity level decreases



Example of execution cycles of the tour miner performed by the agglomerative hierarchical single link clustering algorithm

At each cycle a new reduced similarity matrix (SIM) between each pair of trips (clusters) is computed.

As a consequqnce two trips (clusters) are merged to form a new cluster at each cycle with given similarity. When the similarity decreases a new partition level is defined

The final result is a hierarchy of clusters as the one depicted below

SIM (level 1)	A	B	C	D	E
A	x	0.3	1	0	1
B		x	0	1	0.5
C			x	0	0.5
D				x	0
E					x

Create Tour T1={A,C}, merge columns A with C and recompute similarity by taking the min

SIM (Level 1)	T1	B	D	E
T1	x	0.3	0	0.5
B		x	1	0.5
D			x	0
E				x

Create Tour T2={B,D}, merge columns B with D and recompute similarity by taking the min

SIM (Level 2)	T1	T2	E
T1	x	0	0.5
T2		x	0
E			x

Create Tour T3={T1,E}, merge columns T1 with E and recompute similarity by taking the min

SIM (Level root 0	T3	T2
T3	x	0
T2		x

End of cycles

Fig. 2. Example of Tour clustering with 5 sample trips.

moving from the leaves to the root of the hierarchy.

D. Internal Architecture

As a consequence of the tasks to be performed, the *Internal architecture* of the *Trip Analysis Suite* results as depicted in Figure 3.

- *Data Storage.* This component is necessary because the suite must perform several processing steps and need to store intermediate results, as well as preliminary data.
  In particular, it is necessary to store raw trips provided by the *FollowMe Suite*, the vector representation of geographic slots, the string representation of geo-tagged trips and, finally, the hierarchy of discovered tours.
  We decided to adopt the PostgreSQL free DBMS powered with the PostGIS extension. The advantage of this choice is that PostGIS provides the support for geographic objects representation in an obect relational database and extend SQL with spatial queries. Any any kind of geographic information (from 0 to 3 dimensional spatial objects) can be stored in the PostGres/PostGIS database and can be easily managed by SQL spatial queries. This way we decouple the management and analysis of the tweets contents, i.e., text of the post from the management and analysis of the geographic component of tweets.
- *Geo-Partitioner.* This component actually performs the Ge-Partitioning task of tweets, i.e., it associates with any tweet a *geoslotID*. To this end it exploits the vector representation of the geographic slots, and directly uses PostGIS capabilities to retrieve the *geoslotID* from latitude and longitude.
  Then, the string representing any trip as sequence of *geoslotID*s is generated and stored again into the PostgreSQL database.
- *Tour Miner.* This component is then responsible to perform *Trip Clustering* and generates tours. It takes string representations of trips (generated by the *Geo-Partitioner*) from the PostgreSQL database. The resulting hierarchy of clusters/tours is then stored into the database as well, in order to allow later deploy to map services.
- *Web Deploy to Geoserver.* Discovered tours are be deployed to the *Geoserver* publishing service, so that they can be visualized and analyzed both in geo-portals and in other GIS systems connected to the internet. In our architecture, this role is performed by the component named *Web Deploy*, that reads the most popular tours from the PostGres/PosGIS database and transfers them to *Geoserver* (recall that *Geoserver* us an OGC standard GIS Web server, whose goal is to share these tours as open data by means of the Wed Map Service).
  Tours are selected as popular tours if they contain a minimum number min_{tr} of trips, and belong to a partition with similarity degree min_s (the cluster similarity level).

V. Case Study

In order to illustrate the effectiveness of our approach, we built a simple case study on the basis of a small set of geo-located tweets gathered by the *FollowMe Suite*.

The goal of the case study is to discover the most popular tours of travelers coming to Lombardy, the region in the center of Northern Italy where the main city is Milan, that during

Partition Similarity	Discovered Tours
0.076	128
0.1	139
0.375	307

TABLE I
RELATIONSHIPS BETWEEN PARTITION SIMILARITY AND DISCOVERED TOURS.

2015 was the location of the EXPO 2015 event. Therefore, we identified a pool of 30 European airports with flights connected to the Lombardy airports of *Milano Linate* (airport code LIN), *Milano Mapensa* (airport code MXP), and *Milano Orio al Serio* (airport code BGY, that in effect is located very close to Bergamo, a city located 40 km east of Milan).

We collected tweets in the period between April 20, 2015, and October 22, 2015. By performing a query to discover trips in the bounding box of Lombardy, the *Trip Builder* generated a result set of 9127 tweets, formed by a total of 2568 trips.

The *Trip Analysis Suite* processed the trips, by geo-partitioning tweets with zip codes as geographic slots. The goal was to discover administrative entities such as cities or quarters in big cities frequently visited by tourists. Them, the *Tour Miner* generated the hierarchy of tours that we discuss in the following.

Table I reports three levels of the hierarchy of clusters that give the most popular tours. It can be observed that by increasing the minimum required intra-cluster similarity among the trips belonging to the same tour, i.e., cluster, the number of tours of the correspondent partition increases. It must be pointed out that 5% of the zip areas were visited by a single trip represented by just one single Tweet, 1% by a single trip with two Tweets in the zip area. There is the outlier zip area 24010, small municipalities in Brembana Valley (Valle Brembana, in Italian), north of Bergamo, visited by just one trip with 26 tweets.

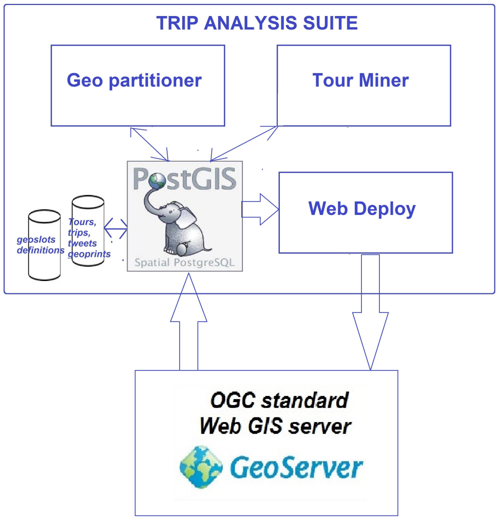


Fig. 3. Internal Architecture of the *Trip Analysis Suite*.

Zip Code	Number of Trips	Number of Tweets	City
21010	347	375	Malpensa
20121	183	270	Milano Center -North District
20090	125	144	Linate
24050	115	126	Orio al Serio Airport
20122	80	104	Milano Center-SE District
21019	66	70	Malpensa North
20124	64	80	Milano NE District
20123	56	72	Milano Center -SW District
20157	48	106	EXPO area

TABLE II
MOST VISITED ZIP AREAS.

Landed At	Number of Trips	Number of Tweets
Milan Malpensa MXP	1506	5203
MIlan Linate LIN	448	1277
Bergamo Orio BGY	310	1481

TABLE III
TRIPS DETECTED FROM LANDING AIRPORTS IN THE MONITORED AREA.

Figure 4 reports a screen-shot of the *Tourists Tracker Portal*, that displays the footprints of the tweets (as red stars) and the zip code areas boundaries . It can be observed that the areas most dense of tweets are Milano city, EXPO site, close to north-west of Milano, Malpensa airport, Como and Bergamo city surroundings. The visualized table of attributes (containing tweets identifier, username, i.e., nick name of the traveler, date, time, start date of the trip, and airport, refers to the tweets (pointed by the red arrow) sent by the same user during his trip to the Brembana valley, North of Bergamo city.

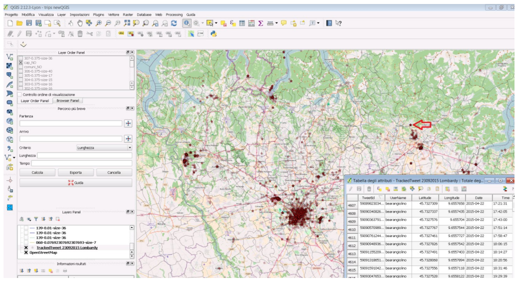


Fig. 4. The *Tourists Tracker* Geo-Portal.

Table II reports the zip code areas most visited by trips (by more than 50 Trips) and with most Tweets. It can be seen that travelers sent more tweets from EXPO area and Milano Centre -North District, probably due to free WI-FI connections.

Table III reports the first three most popular airports of Lombardy region, where most of the trips land/depart from. It can be observed that more than 88% of all trips pass from

Departed From	Number of Trips	Number of Tweets
Barcelona BAR	75	270
Munich (MUC)	51	253
Athens ATH	30	110

TABLE IV
TRIPS DETECTED FROM DEPARTURE AIRPORTS IN THE MONITORED AREA.

Tour UD	Airport	Number of Trips	Average Tweets per Trip
128	Malpensa	418	4
124	Linate	152	7
120	Malpensa	139	5
121	Malpensa	113	3
123	Malpensa	110	5

TABLE V
LANDING AIRPORTS OF THE MOST POPULAR TOURS (TOURS WITH MORE THAN 100 TRIPS.

these airports. Table IV reports the first three most popular foreigner airports, from where the hang tweet was sent before taking off for Lombardy destinations.

Table V reports the landing airport of the biggest tours (clusters) consisting of more than 100 trip, and the average length of trips (in number of tweets) within the tour of partition level with similarity 0.375. It can be observed that the most popular tours (defined by considering with minimum similarity among trips above 100) constitutes 36% of all tours and all landed from Malpensa airport (3 tours), but one, that landed to Linate airport. No popular tour so defined departed from Orio al Serio airport. The first most popular tour from Orio al Serio airport is only half popular according to the definition of popularity, since it has only 59 trips of an average length of 6 tweets.

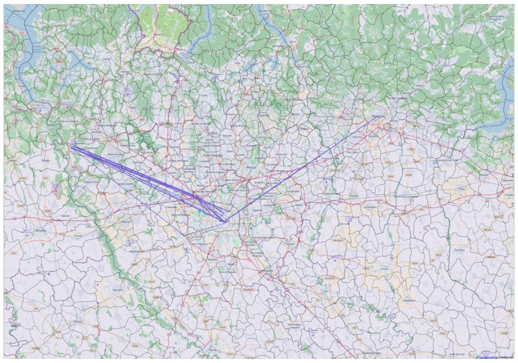


Fig. 5. Tour-cluster 256 (in partition with similarity 0.375) containing 42 trips landed in Milan Malpensa airport.

In order to show the exploratory facilities offered by sharing on the Web the tours, i.e., the clusters of trips grouped by their common visited regions, we discuss some examples.

Figure 5 reports the map of the tour-cluster 256 - of partition 0.375 containing 42 trips. It is interesting to note that its trips started in Malpensa airport, visited EXPO site and some of them reached Bergamo city. Notice that each visualized tour is composed by several polylines, each one depicted with the same color of the tour.

However, the *Tourists Tracker Geo-portal* permits to personalize the background map and default bounding box. Furthermore, being based on the OGC standard, it can integrate several information layers, coming from possibly external OGC services. This interoperability permits to cross analyze the trips and open geo-data from other sources.

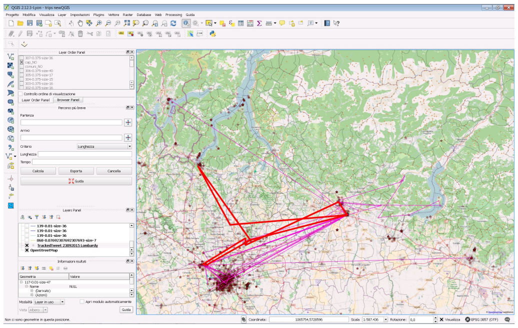


Fig. 6. Visualizing a single trip in details.

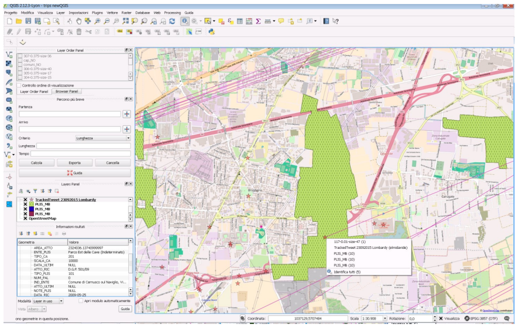


Fig. 7. Visualizing a single trip detail with an external information layer describing naturalistic parks.

As an example, Figure 6 shows a single trip selected among whose in a tour. By selecting to show a new information layer describing naturalistic parks in Lombardy (provided as Open Data by the Open Data Service of Regione Lombardia) and zooming in, it is possible to observe that (see Figure 7) a tweet in the trip was posted while visiting a naturalistic park.

VI. RELATED WORK

In this section we have selected a brief collection of works that are correlated with our research.

[18] analyze geo-located *Twitter* messages in order to uncover global patterns of human mobility. Based on a dataset of almost a billion tweets recorded in 2012, they estimate the volume of international travelers by country of residence. Mobility profiles of different nations were examined based on such characteristics as mobility rate, radius of gyration, diversity of destinations, and inflow-outflow balance. Their analysis of the community structure of the *Twitter* mobility network reveals spatially cohesive regions that follow the regional division of the world.

[19] try to understand how racial segregation of the geographic spaces of three major US cities (New York, Los Angeles and Chicago) affect the mobility patterns of people living in them. Collecting over 75 million geo-tagged tweets from these cities during a period of one year beginning October 2012 they identified home locations for over 30,000 distinct users, and prepared models of travel patterns for each of them. Dividing the cities' geographic boundary into census tracts and

grouping them according to racial segregation information they try to understand how the mobility of users living within an area of a particular predominant race correlate to those living in areas of similar race, and to those of a different race.

[20] use an advanced data-mining framework with a novel use of social media data retrieval and sentiment analysis to understand how geo-located tweets can be used to explore the prevalence of healthy and unhealthy food across the contiguous United States. Additionally, tweets are associated with spatial data provided by the US Department of Agriculture (USDA) of low-income, low-access census tracts (e.g. food deserts), to examine whether tweets about unhealthy foods are more common in these disadvantaged areas.

[21] monitor all posts on *Twitter* issued in a given geographic region and identify places that show a high amount of activity. In a second processing step, they analyze the resulting spatio-temporal clusters of posts with a Machine Learning component in order to detect whether they constitute real-world events or not. They show that this can be done with high precision and recall. The detected events are finally displayed to a user on a map, at the location where they happen and while they happen.

[22] present a study that focuses on these questions: are users who are similar from the geospatial perspective (i.e., who send messages from nearby locations) similar from the textual perspective (i.e., send messages with similar textual content)? Do posts with similar content have a spatial distribution similar to that of any random set of posts? The authors provide statistical tests to examine the correlation between textual content and geospatial locations in tweets.

[23] compare the social properties of *Twitter* users' networks with the spatial proximity of the networks. Using a comprehensive analysis of network density and network transitivity they found that the density of networks and the spatial clustering depends on the size of the network; smaller networks are more socially clustered and extend a smaller physical distance and larger networks are physically more dispersed with less social clustering.

[24] aim at developing a geo-social event detection system by monitoring crowd behaviors indirectly via *Twitter*. In particular, they attempt to find out the occurrence of local events such as local festivals; a considerable number of *Twitter* users probably write many posts about these events. To detect such unusual geo-social events, we depend on geographical regularities deduced from the usual behavior patterns of crowds with geo-tagged microblogs.

In [25], an approach to detect events of both periodic and aperiodic nature has been defined and applied based on first capturing Tweeter posts and then applying spatio-temporal clustering at global and local scale to identify singularities.

As far as our work on trip clustering and identification of popular trips it shares the same objective of the work [26] that identifies the most frequent travel routes and the top interesting sites in a given geo-spatial region by mining trajectory patterns from photos on $Flickr$. They extract the semantics of the sites from the photo captions created by users, while the geo-tagged tweets we collect and analyze do not provide such information and thus we associate the semantics of locations by mapping the tweets in relevant geographic units, for example in the study case we used zip code areas.

Another approach is proposed in [27], whose aim is to analyze trajectories of users extracted from geo-tagged tweets to discover people and community behaviors and regularities in moving trajectories.

VII. Experiments

In order to evaluate the feasibility of our approach on real data, we performed a severe experimental evaluation. From within the dataset gathered by the FollowMe suite since May 1, 2015, we extracted a pool of 300000 tweets dispersed in western Europe. Then, in order to make the running conditions really severe, we performed experiments on a regular PC equipped with 4 GB RAM and a CUDA Indiva graphic board, which allow to make parallel some phases of the process. Let us discuss performance of each phase in details.

A. Preparing and Uploading Data into the PostgreSQL server

First of all, the Trip Analysis Service needs to upload data received from teh FollowMe Suite into the PostgreSQL database which it relies on. However, data cannot be uploaded as they are, but the need to be preprocessed to adapt to the specific data structure.

The time is not negligible: as the reader can se from Figure 8, it can take up to 12 million msecs, i.e., a little bit more than 3 hours in the worst case of 300000 tweets. Of course, we expect that on a more performing and multi-CPU server the execution time of this preliminary phase can significantly reduce: in fact, the execution time scales more or less linearly w.r.t. the number of tweets.

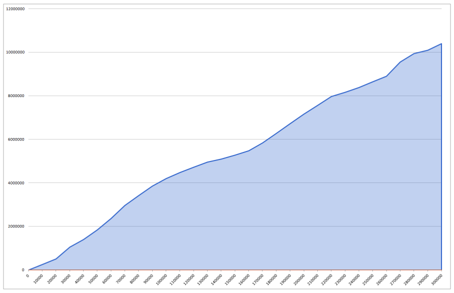


Fig. 8. Execution Time for Uploading Yweets (Trips) into PostgreSQL Server (x axis: number of tweets, y axis: execution times in msec).

B. Geo-Slot Assignment

In order to assign geo-slots to tweets distributed on Western Europe, we used the NUST-2013 classifications for European Regions [28]; in particular, we used the Level 3 classification (NUTS3), which encode small European regions. Spatial definition of regions are stored within the PostGreSQL server enriched with PostGIS plug-in.

Figure 9 reports the execution times w.r.t. the number of tweets. As we expected the chart confirm that the process is substantially linear, since coordinates of each tweet are matched against spatial definitions of each region, by means of PostGIS spatial functions. Notice that even in the worst case (300000 tweets) the execution time does not exceeds 2.5 million msec, i.e., 41 minutes, and the process scales linearly.

The execution time of the subsequent step are reported in Figure 10: geo-codes (i.e., NUTS codes) are used to build representation of trips as strings of geo-codes. In the worst case of 11000 trips the process takes a little bit more than 1.5 million of msecs, i.e., 26 minutes.

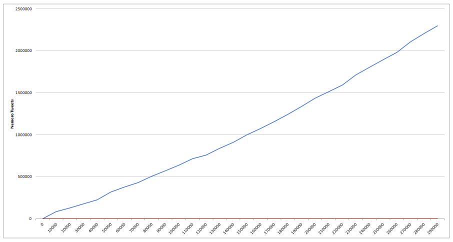


Fig. 9. Execution Time for Assigning geo-Slots to Tweets (x axis: number of tweets, y axis: execution times in msec).

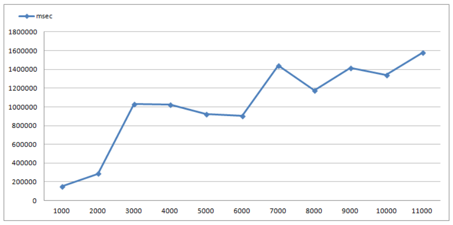


Fig. 10. Execution Times for Building String Representations of Trip (x axis: number of trips, y axis: execution times in msec).

C. Tour Mining (Geoclustering)

The last phase actually mines tours. It can be subdivided into two separate steps: 1) first of all, it is necessary to build the *similarity matrix*, which represents the similarity measure for every pair of trips (look at the description in Section IV); 2) second, the hierarchical representation is build, based on the similarity matrix previously computed.

Due to the large number of pairs, the similarity measure is huge and this forced us to limit the size of the matrix by limiting the maximum number of trips for our experiments. This is the reason why we scaled experiments in this phase up to 11000 trips. We will investigate in the future how to make parallel this phase.

Let us consider the outcome of the experiments. In Figure 11, we report the behavior of the first step, i.e., building the similarity matrix. Notice that with the dataset of 11000 trips it takes a little bit more than 259000 msec, i.e., 4.3 minutes. This is due to the fact that this task is performed in main memory, and it suffer of space limitation in main memory.

Then. the subsequent step of tour building, where the hierarchical representation of tours/clusters is actually built, is significantly more efficient, as shown in Figure 12. The reader can see that it is even more efficient than building the similarity matrix. In effect, with the dataset of 11000 trips, the execution time is 161000 msecs, i.e., 2.5 minutes. Figure 13 is obtained by adding the chart reported in Figure 11 (execution time for similarity matrix building) to the execution time for tour building (drawn in Figure 12 and reported in Figure 13 as lower line). With the dataset of 11000 trips, the total execution time of geo-clustering is 420000 msec, i.e., 7 minutes.

Summarizing the outcome of our experiments, we can say that at the current stage of development, pre-processing, uploading and geo-coding are very time consuming, but scales linearly and do not suffer for main memory space limitations. In our opinion, they could be easily distributed on several computers, in order to make them more efficient.

In contrast, geo-clustering suffer for main memory space limitations, while it is very efficient in terms of execution times. So, next work will be devoted to modify the algorithms in order to both define a compact representation of the similarity matrix and to distribute the computation among several machines.

Anyway, the decision to perform experiments in a sever environment such as a regular PC without strong computational resources allowed us to identify the bottlenecks of the process,but at the same time allows us to claim that the process is suitable to be performed, as it is, on datasets containing several thousands of trips.

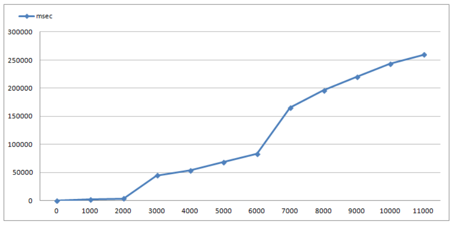


Fig. 11. Execution Time for Computing the Similarity Matrix (x axis: number of trips, y axis: execution times in msec).

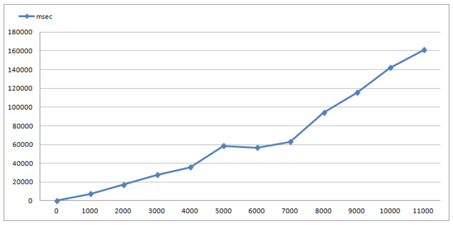


Fig. 12. Execution Time for Tour Building (x axis: number of trips, y axis: execution times in msec).

VIII. Conclusions and Future Work

In this work, we developed an original approach that permits to follow traveling *Twitter* users by tracking their geo-tagged messages posted on *Twitter* during their trips. Several tools have been developed to capture and then spatially analyze the

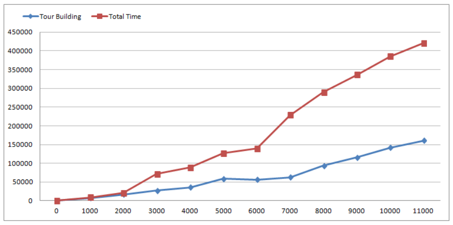


Fig. 13. Total Execution Time for Tour Building (x axis: number of trips, y axis: execution times in msec).

geo-tagged tweets in order to trace the trips, to mine the most popular tours, and to visually correlate them to open territorial data. The approach is knowledge-based and this allows its customization to distinct regions of interest, and its tuning to various desired angularities of the spatial analysis.

As far as future work is concerned, we have to consider that the project is only at the beginning steps. The main efforts will be devoted to connect with other social networks and gather posts from them. This way, we should obtain a wider spectrum of information, by integrating several sources of information. Nevertheless, We will also continue to investigate clustering techniques, in order to develop new analysis techniques able to reveal unexpected knowledge from gathered trips.

References

[1] L. Zhao, L. Chen, R. Ranjan, K. R. Choo, and J. He, "Geographical information system parallelization for spatial big data processing: a review," *Cluster Computing*, vol. 19, no. 1, pp. 139–152, 2016.

[2] P. Baumann, P. Mazzetti, J. Ungar, R. Barbera, D. Barboni *et al.*, "Big data analytics for earth sciences: the earthserver approach," *Int. J. Digital Earth*, vol. 9, no. 1, pp. 3–29, 2016.

[3] A. Cuzzocrea, D. Sacca, and J. D. Ullman, "Big data: a research agenda," in *17th International Database Engineering & Applications Symposium, IDEAS '13, Barcelona, Spain - October 09 - 11, 2013*, 2013, pp. 198–203.

[4] A. Cuzzocrea, L. Bellatreche, and I. Song, "Data warehousing and OLAP over big data: current challenges and future research directions," in *Proceedings of the sixteenth international workshop on Data warehousing and OLAP, DOLAP 2013, San Francisco, CA, USA, October 28, 2013*, 2013, pp. 67–70.

[5] A. Cuzzocrea, "Analytics over big data: Exploring the convergence of datawarehousing, OLAP and data-intensive cloud infrastructures," in *37th Annual IEEE Computer Software and Applications Conference, COMPSAC 2013, Kyoto, Japan, July 22-26, 2013*, 2013, pp. 481–483.

[6] ——, "Big data mining or turning data mining into predictive analytics from large-scale 3vs data: The future challenge for knowledge discovery," in *Model and Data Engineering - 4th International Conference, MEDI 2014, Larnaca, Cyprus, September 24-26, 2014. Proceedings*, 2014, pp. 4–8.

[7] A. Cuzzocrea and I. Song, "Big graph analytics: The state of the art and future research agenda," in *Proceedings of the 17th International Workshop on Data Warehousing and OLAP, DOLAP 2014, Shanghai, China, November 3-7, 2014*, 2014, pp. 99–101.

[8] G. L. Andrienko, N. V. Andrienko, H. Bosch, T. Ertl, G. Fuchs, P. Jankowski, and D. Thom, "Thematic patterns in georeferenced tweets through space-time visual analytics," *Computing in Science and Engineering*, vol. 15, no. 3, pp. 72–82, 2013.

[9] S. Kumar, X. Hu, and H. Liu, "A behavior analytics approach to identifying tweets from crisis regions," in *25th ACM Conference on Hypertext and Social Media, HT '14, Santiago, Chile, September 1-4, 2014*, 2014, pp. 255–260.

[10] Y. Lu, X. Hu, F. Wang, S. Kumar, H. Liu, and R. Maciejewski, "Visualizing social media sentiment in disaster scenarios," in *Proceedings of the 24th International Conference on World Wide Web Companion, WWW 2015, Florence, Italy, May 18-22, 2015 - Companion Volume*, 2015, pp. 1211–1215.

[11] S. Yang and A. L. Kavanaugh, "Collecting, analyzing and visualizing tweets using open source tools," in *Proceedings of the 12th Annual International Conference on Digital Government Research, DG.O 2011, College Park, MD, USA, June 12 - 15, 2011*, 2011, pp. 374–375.

[12] D. Cameron, A. Finlayson, and R. Wotzko, "Visualising social computing output: Mapping student blogs and tweets," in *Social Media Tools and Platforms in Learning Environments.*, 2011, pp. 337–350.

[13] A. Cuzzocrea, C. D. Maio, G. Fenza, V. Loia, and M. Parente, "Towards OLAP analysis of multidimensional tweet streams," in *Proceedings of the ACM Eighteenth International Workshop on Data Warehousing and OLAP, DOLAP 2015, Melbourne, VIC, Australia, October 19-23, 2015*, 2015, pp. 69–73.

[14] L. Wang, R. Ranjan, J. Kolodziej, A. Y. Zomaya, and L. Alem, "Software tools and techniques for big data computing in healthcare clouds," *Future Generation Comp. Syst.*, vol. 43-44, pp. 38–39, 2015.

[15] K. Slavakis, G. B. Giannakis, and G. Mateos, "Modeling and optimization for big data analytics: (statistical) learning tools for our era of data deluge," *IEEE Signal Process. Mag.*, vol. 31, no. 5, pp. 18–31, 2014.

[16] 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 - 5th International Conference, MEDI 2015, Rhodes, Greece, September 26-28, 2015, Proceedings*, 2015, pp. 285–294.

[17] G. Bordogna, T. Kliment, L. Frigerio, and P. Carrara, "Volunteered geographic information and spatial data infrastructures to promote s-low resources: the case study of the -orti di bergamo smart application-," in *Centralities of Territories, E. Casti and F. Burini eds.* Bergamo University Press, Sestante edizioni, 2015, pp. 165–182.

[18] N. Bora, Y. Chang, and R. Maheswaran, "Mobility patterns and user dynamics in racially segregated geographies of us cities," in *Social Computing, Behavioral-Cultural Modeling, and Prediction, Washington, DC (USA)*, April 2014.

[19] I. Grabovitch, Y. Kanza, E. Kravi, and B. Pat, "On the correlation between textual content and geospatial locations in microblogs," in *Social Computing, Behavioral-Cultural Modeling, and Prediction, Washington, DC (USA)*, June 2014.

[20] B. Hawelka, I. Sitko, E. Beinat, S. Sobolevsky, P. Kazakopoulos, and C. Ratti, "Geo-located twitter as proxy for global mobility patterns," *Cartography and Geographic Information Science*, vol. 41, no. 1, pp. 260–271, 2014.

[21] R. Lee and K. Sumiya, "Measuring geographical regularities of crowd behaviors for twitter-based geo-social event detection," in *Social Computing, Behavioral-Cultural Modeling, and Prediction, Washington, DC (USA)*, November 2010.

[22] M. Stephens and A. Poorthuis, "Follow thy neighbor: Connecting the social and the spatial networks," *Computers, Environment and Urban Systems*, vol. 41, no. 1, pp. 260–271, 2014.

[23] M. Walther and M. Kaisser, "Geo-spatial event detection in the twitter stream," in *Social Computing, Behavioral-Cultural Modeling, and Prediction, Washington, DC (USA)*, 2013.

[24] M. Widener and W. Li, "Using geolocated twitter data to monitor the prevalence of healthy and unhealthy food references across the us," *Applied Geography*, vol. 54, pp. 189–197, 2014.

[25] G. Bordogna, S. Sterlacchini, and P. Arcaini, "User driven query framework of social networks for geo-temporal analysis of events of interest," in *L. Yan (Ed.), Handbook of Research on Innovative Database Query Processing Techniques*. Information Science Reference, 2016, pp. 224–249.

[26] Z. Yin, L. Cao, J. Han, J. Luo, and T. S. Huang, "Diversified trajectory pattern ranking in geo-tagged social media." in *SDM*. SIAM, 2011, pp. 980–991.

[27] C. Comito, D. Falcone, and D. Talia, "Mining popular travel routes from social network geo-tagged data," in *Intelligent Interactive Multimedia Systems and Services*, 2015, vol. 40, pp. 81–95.

[28] "Nuts-2013 classification - eurostat," http://ec.europa.eu/eurostat/web/nuts/overview.