Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Computer Science 00 (2017) 000–000

Procedia
Computer Science

www.elsevier.com/locate/procedia

The 8th International Conference on Ambient Systems, Networks and Technologies (ANT 2017)

Reputation evaluation of georeferenced data for crowd-sensed applications

Marco Gusmini[a], Nafaâ Jabeur[b], Roula Karam[a], Michele Melchiori[a,*], Chiara Renso[c]

[a]Dept. of Inf. Engineering, Università degli Studi di Brescia, Brescia, Italy
[b]German University of Technology in Oman (GUtech), Athaibah, Sultanate of Oman
[c]ISTI, CNR, Pisa, Italy

Abstract

Volunteered Geographic Information (VGI) is a process where individuals, supported by enabling technologies, behave like physical sensors to harvest georeferenced content in their surroundings. The value of this, typically heterogeneous, content has been recognized by both researchers and organizations. However, in order to be fruitfully used in various VGI-based types of application reliability and quality of particular VGI content (i.e., Points of Interest) have to be assessed. This evaluation can be based on reputation scores that summarize users' experiences with the specific content. Following this direction, our contribution provides, primarily, a new comprehensive model and a multi-layer architecture for reputation evaluation aimed to assess quality of VGI content. Secondly, we demonstrate the relevance of adopting such a framework through an applicative scenario for recommending touristic itineraries.

Peer-review under responsibility of the Conference Program Chairs.

Keywords: Social Sensors; Reputation Evaluation; Volunteered Geographic Information; Mobile Crowdsourcing; Tourism Planning;

1. Introduction

Citizens as sensors[1] is a terminology introduced in a seminal paper by Goodchild[9] with the idea that each individual, supported by enabling technologies like smartphones and wearable sensors, behaves like physical sensors reporting through the Internet detailed information about the environment he/she is living in. The user-reported data can be distinguished into passively or actively collected. Passive data collection is when we share our whereabouts, even without knowing we are doing so, like the Google Maps real-time traffic information. On the other hand, data is actively collected when there is an explicit action of the individual in reporting some information. This last case, when

* Corresponding author. Tel.: +39-030-3715845.
E-mail address: michele.melchiori@unibs.it

[1] https://www.greenbiz.com/news/2013/03/20/how-citizens-have-become-sensors

1877-0509
Peer-review under responsibility of the Conference Program Chairs.

related to the active sensing of geographic-related information, is also known as Volunteer Geographical Information (VGI). VGI, or geospatial crowdsourcing, is "where citizens (volunteers) contribute data and information about the earth and environment that is explicitly or implicitly georeferenced and then disseminated via collaborative projects, such as OpenStreetMap or social media such as Flickr, Twitter and Facebook"[12]. Active user-generated data collection is a recent phenomenon growing rapidly in terms of citizen adoption and volume of data/information produced in the direction of the so called Web 3.0 or Collective Intelligence[2]. As a consequence, it is more and more attracting the attention of researchers and organizations[11].

A relevant aspect of VGI for users and applications is that in many cases it provides more timely, updated and detailed content than authoritative sources[9]. On the negative side, it is known that the process of VGI generation is intrinsically subjective and loosely controlled; it relies often on devices having variable levels of precision and on untrained volunteers. As a consequence, VGI data are highly heterogeneous in coverage, density and quality. Techniques to estimate and improve the quality of these data are therefore needed[1,13] in order to decide whether they can be suitable for a given domain or application.

A classification of approaches for assessing overall quality of VGI covering a geographical area is discussed in the work of Goodchild and Li[10]. However, for various types of user-oriented applications that focus on particular geospatial objects, it is important to get an estimation of the quality of descriptions about specific points of interest (PoI). For instance, it can be required to know whether to trust or not the VGI description of an object. Examples can be found in applications providing information on the location and type of architectural barriers[6] or the potability of water wells[2]. In the following, we hint at another example about Spatial Decision Support Systems. Quality indicators for VGI data focus on aspects influencing the quality without measuring it directly. Just to cite few: *lineage*, which relies on the history and evolution of the dataset describing for example a PoI, *quality of textual descriptions*[3], *experience*[3], *trustworthiness* and *reputation*[7].

The remainder of this paper is structured as follows. Section 2 provides additional research context on evaluating reputation for crowd-sensed applications. In Section 3, we propose a multi-layer architecture to enhance VGI reputation. In Section 4, the reputation model requirements and metrics are defined and assessed by some experimentation presented in Section 5. In Section 6, an applicative scenario about recommending touristic itineraries is described to illustrate the potentiality of including our reputation mechanism on citizen-generated data. The paper concludes in Section 7 with some suggestions to extend our model and future research directions.

Motivating scenario. Spatial Decision Support Systems are commonly created to allow several decision-makers to collaboratively plan their actions.[4] Because of the increasing complexity in several real-life scenarios, including hazard management and road traffic monitoring, stakeholders are relying on people with advanced technologies (e.g., smartphones, wearables) to collect VGI data anytime, anywhere, about events and objects of interest. Although some progress is being achieved, several obstacles are still challenging the efforts to integrate VGI capabilities with Spatial Data Warehouses (SDW) for more advanced analytics. For instance, the VGI datasets are commonly unstructured, with varying qualities, formats, granularities, and trustworthiness levels. As this investigation is expected to happen frequently, assigning reputation values to VGI participants and their contributions is a relevant add-on feature. These values are particularly important to filter, prioritize and lighten data loading into the SDW.

2. Related Work

Several research and development works have attempted to estimate reputation of user-generated georeferenced content in different applications. For example, Bishr and Khun[2] described a reputation model based on coherence of volunteers' reports on the potability of water wells in developing countries. The potability status has a simple boolean value and time is explicitly included in the model. For instance, trustworthiness in the reports about potability is reduced proportionally to the elapsed time since the creation of such reports. Our model considers more general VGI scenarios allowing for complex descriptions of objects. Another approach is given in Zhao et al.[18] They estimate trustworthiness of VGI data based on contributor's reputation as well as on analyzing several versions of VGI data. This is similar to the approaches proposed by D'Antonio et al.[7] Each version of VGI description is created by a

[2] https://glennremoreras.com/tag/collective-intelligence/

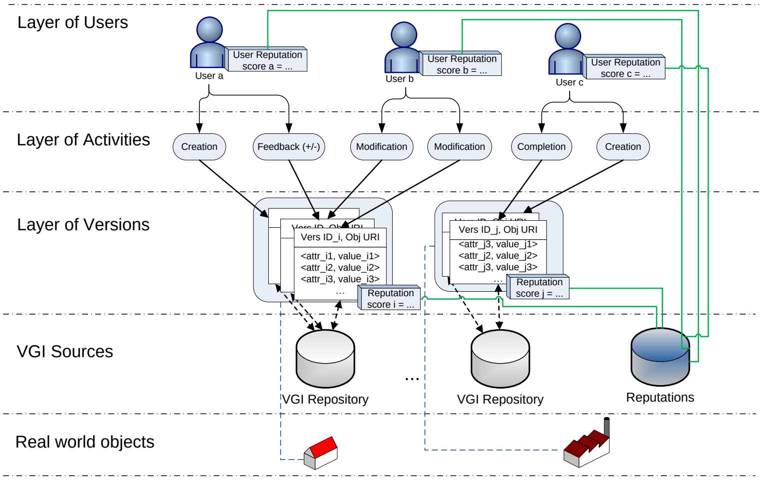


Fig. 1. Reference multi-layer architecture for VGI reputation.

contributor and reports the current status of a geospatial object. The level of trust on a specific version depends on: (i) contributor's reputation; (ii) similarity distance between this version and the previous one for the same object; and (iii) level of trust in the previous version. As per our approach, these authors distinguish between implicit and explicit assessment of contributors.

Trustworthiness and reputation have been studied in other contexts. We can mention crowdsourcing [16], multi-agent systems [17] and social networks [14] where a trust value is typically defined with reference to an ordered pair of users or to pair user-object. Differently, reputation is a community-based value associated with a user. In particular, CrowdRec [16] is a recent task-oriented recommendation approach that recommends reliable workers for a specific task assigned by a requester. A worker is recommended for a task when he has received positive evaluations from trusted requesters on tasks of the same type. The approach defines a graph where edges are representing the trust relationships between requesters. Basically, edges are weighted proportionally to the number of times the requesters evaluated in a coherent way a worker on similar tasks. Similar to our approach, CrowdRec deals with dishonest behaviors. However, the problem setting is different from ours because there is no an overall reputation value associated with a worker, but rather a *prediction of a worker's performance* is doing a specific type of task for a given requester is computed.

3. An architecture for reputation-enhanced VGI

In this section, we present a multi-layer architecture (see Fig. 1) for VGI enhanced with reputation scores. This architecture briefly introduces the main concepts and their relationships for evaluating reputation. For the scope of this paper, we will highlight the layers of activities and versions. According to [18], a version of a VGI description, hereafter simply a *version*, is a description of state of a geospatial real world object at a certain time as inserted by a given user (the correspondent author). For the same state, more versions can be produced, as far as the state of an object can change during its lifespan (e.g., its address can change), thus making out of date all previous versions describing it. In our approach, we compute reputation of both authors/users and the correspondent versions. Basically, user's reputation is based on other peers' feedbacks who had judged his/her versions. Similarly, the reputation of a version describing an object state depends: (i) on the peers' feedbacks on it and (ii) on the activities of authors who tried to update the version by producing other versions referring the same object state.

In layer of *versions*, we model a version (e.g., the description of a georeferenced PoI or any element on a map, as a street) as a set of pairs $< attribute, value >$ with a URI identifying the described object. Versions are stored in VGI repositories, like OpenStreet Map (OSM) and WikiMapia.

In layer of *activities*, we register the user's activities. We consider four user's activities: (i) *creation* of a new version, (ii) *modification* of an existing version, (iii) *completion* of an existing version, or (iv) *direct feedback* on a version. With the creation activity, a user produces a new set of pairs $< attribute, value >$, that is a new version. A modification refers to an existing version and consists of new values for pairs $< attribute, value >$ already existing in the version (e.g., the pair <opening, Monday-Friday> modifies <opening, Monday-Saturday>). Similarly, a completion refers to an existing version and defines a set of new pairs $< attribute, value >$ meant to be added to an existing version (e.g., <phone2, +39303333020> is added to an existing version). Every time a user makes a modification (respectively, a completion) to a version *A*, a new version *A'* is generated from *A* by making a copy of *A* and then modifying the original pairs (respectively, adding new pairs). Finally, a user's direct feedback on a version *A* assesses either that the version is correct (positive feedback) or not (negative feedback) according to this user. For each version (e.g., *A*, *A'*), its creation time is stored. A reputation value in $[0, 1]$ is associated with both user and version. Modifications, completions and direct feedbacks on a version *A* modify both the reputations of *A* and the reputation of its author. This is a remarkable feature for our model. In the architecture depicted in Fig. 1, the content of the versions layer is stored in the VGI sources and the reputation information is stored in a separated repository. In fact, reputation scores of authors and versions are stored and managed by a platform separated from the VGI source repositories.

Concerning the layer of *users* we can distinguish between active users, who have performed any type of activity on versions, and non active ones, who have given only direct feedbacks to versions produced by other users. In the Fig. 1, *User b* and *User c* are active while *User a* is not active. We also distinguish between cooperative users, behaving honestly, and non-cooperative users, behaving according to a malicious attitude (e.g., giving a positive feedback to a clearly wrong version in order to increase the reputation of another dishonest user).

4. Reputation evaluation model

In this section, we assess some requirements and metrics for the reputation evaluation model based on the multi-layer architecture. As mentioned before, the reputation of versions depends not only on feedbacks being considered as direct evaluations, but also on modifications and completions considered as indirect feedbacks. A modification reduces the reputation of any version since we can infer that the user making the modification has judged the original version presenting errors or requiring some update. On the contrary, a completion activity tends to recognize the original content as correct but incomplete. Therefore, a completion should increase the reputation of the original version whereas a modification should decrease it. Modifications and completions are then considered in our model as implicit feedbacks. The impact of a modification/completion action should depend on how much the new resulting version diverges from the original one. For example, a small modification should only reduce slightly the reputation of the original version.

To make the user's contribution consistent and reliable, the activities on versions are also subject to constraints. For example, a user is not allowed to give directly feedback to another user nor evaluate a version more than once. This is set to avoid a malicious user to deliberately increasing or decreasing the reputation of other specific peers or their contents too. In the following, we formalize the metrics.
User reputation. Every time a feedback, completion or modification are made on a version, the *User reputation* score of the author is updated as follows:

$$uR_u = \begin{cases} uR_0 \\ (1 - e^{-n}) \cdot \frac{POS_u}{POS_u + NEG_u} + e^{-n} \cdot uR_0 \end{cases} \tag{1}$$

where $uR_u \in [0, 1]$ and $uR_u = 1$ is the maximum reputation. We are not classifying explicitly the honest/malicious nature of users based on the reputation score, but all the user's activities are weighted proportionally to his/her reputation. The first case of Eq. (1) applies when no activities have been made yet to any versions authored by the user u, i.e., the positive and negative feedbacks are zero (i.e., $POS_u = 0$ and $NEG_u = 0$). The term uR_0 is the initial user reputation, usually set to a quite low reputation value (e.g., 0.3) for every new user. Alternatively, the initial reputation could be set based on automated analysis of the user information (e.g., their completeness) provided in filling in the registration profile[3]. In the second case of Eq. (1), the parameter n is the total number of versions produced by the user u. This includes new versions and those produced due to the modifications or completions of other versions. The objective of the exponential coefficient e^{-n} is to reduce the amplitude of the initial oscillations of the uR_u value when n is low (e.g., $n < 3$). This is to prevent that a single positive or negative feedback can increase or decrease a lot the

[3] The features in user profile can also analyzed with machine learning techniques in order to detect fake profiles or multiple registrations of the same user, as proposed in Xiao al Al.[15]. Detected fake profiles then can be disabled or submitted to human checking.

reputation uR_u when u has produced just few versions. Moreover, as n increases, the initial reputation uR_0 has less impact so the user's reputation score depends more and more on the feedbacks provided by other users.
The definition of the positive and negative feedbacks terms POS_u and NEG_u are reported below:

$$POS_u = \sum_{v \in V(u)} POS_v \cdot h(t_v) \qquad NEG_u = \sum_{v \in V(u)} NEG_v \cdot h(t_v) \tag{2}$$

where $V(u)$ is the set of contributions produced by the user u and v is the version. POS_v and NEG_v summarize, respectively, positive and negative feeedbacks being expressed on the version v. $h(t_v)$ is a coefficient weighting the contribution of POS_v and NEG_v that depends on the creation time t_v of v (the aging of version), as formalized below.
Aging of versions. The coefficient $h()$ assigns a higher weight to feedbacks expressed on recent versions. This value decreases linearly with the difference between the evaluation time t and the creation time t_v of a version v. In particular, $h(t_v)$ is equal to 0 when this difference is larger than a predefined value α. The reason is that feedbacks on old versions should impact less the author's reputation compared to feedbacks on new versions because the content will obviously become obsolete/old with time. In our experiments, we have chosen α equal to 180 days; feedbacks on a version older than 180 days are no more changing the author's reputation. This coefficient is defined as:

$$h(t_v) = \begin{cases} \frac{\alpha-(t-t_v)}{\alpha} & \text{if } t - t_v < \alpha \\ 0 & \text{if } t - t_v \geq \alpha \end{cases} \tag{3}$$

where $h(t_v) \in [0, 1]$. We notice that since POS_v and NEG_v are always positive, POS_u and NEG_u are positive as well.
Reputation of versions. The reputation score vR_v of a version v is based on (implicit and explicit) feedbacks given to v by other peers.

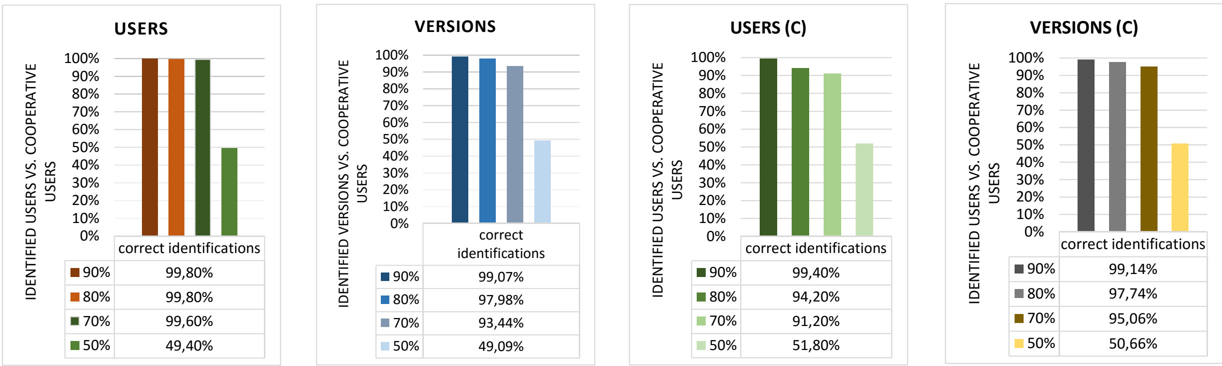
$$vR_v = \begin{cases} vR_0 \\ (1 - e^{-k}) \cdot \frac{POS_v}{POS_v + NEG_v} + e^{-k} \cdot vR_0 \end{cases} \tag{4}$$

where $vR_v \in [0, 1]$ and k is the number of feedbacks on v. The first case in Eq. (4) applies when $k = 0$. In the second case, when $k \neq 0$, reputation score vR_v is defined according to an expression similar to the one for user's reputation score, as in Eq. 1.

The initial reputation $vR_0 \in [0, 1]$ is a combination of the activities on the version (new version, modification, completion) and the author's u reputation as follows:

$$vR_0 = \begin{cases} uR_u & \text{if } vers(v) \\ (1 - Sim(v, vPrev)) \cdot uR_u + \\ Sim(v, vPrev) \cdot min(vR_{vPrev}, uR_u) & \text{otherwise} \end{cases} \tag{5}$$

where $vers(v)$ is true when v is a new version and, in this case, the initial reputation vR_0 is set to be equal to the reputation uR_u of the author. Otherwise, when v is obtained by modification or completion of a previous version, vR_0 is computed as weighted mean of: (i) the author reputation uR_u and (ii) the minimum between the reputation of the previous contribution $vPrev$ and uR_u. The rationale for selecting the minimum is the following. If reputation of $vPrev$ is high and uR_u is low, a user u could maliciously modify a version with high reputation in order to produce a version v with some wrong content (e.g., a different website address) but having initially a high reputation. By taking the minimum, this cannot happen since the user's reputation uR_u poses an upper bound to the initial reputation of v. $Sim(v, vPrev) \in [0, 1]$ is a similarity measure of the current and the previous version. The more these versions are dissimilar (i.e., $Sim(v, vPrev) \sim 0$), the closer the reputation score to the author's reputation score uR_u and thus independent of the reputation of $vPrev$. This is because if the version v modifies significantly $vPrev$ then the reputation of v should be more depending on the author than on the previous version $vPrev$. We will not detail here how to calculate $Sim(v, vPrev)$ due to page limitation. It is proportional to the number of pairs $< attribute, value >$ and identical for both versions v and $vPrev$.
Feedback evaluation. The values of POS_v and NEG_v, summarizing the positive and negative amount of direct and indirect feedbacks on version v, are updated every time a new feedback on v is produced. The impact of a feedback by a user u_f is proportional to reputation of u_f. In particular, the equation for updating the current POS_v when either a positive feedback or a completion f is submitted by a user u_f with reputation uR_f, is the following:



(a) (b) (c) (d)

Fig. 2. Identification of: (a) users, (b) versions, in the model without aging, and of: (c) users, (d) versions, in the model with aging.

$$POS'_v = POS_v + \begin{cases} uR_f & \text{if } f \text{ is pos. feedback} \\ Sim(v', v) \cdot uR_f & \text{if } f \text{ is completion} \end{cases} \tag{6}$$

In case of completion, v' denotes the upgraded version of v after this operation. More the versions v and v' are similar, higher the increasing of POS_v due to the presence of the similarity coefficient $Sim(v', v)$. The rationale is that if v' is very different from v, it is not considered a completely positive evaluation of v, i.e., v was evaluated as incomplete. In this case, POS'_v has not to differ a lot from POS_v.

We define as well the equation for updating NEG_v when either a negative feedback or a modification f is submitted by u_f with reputation uR_f:

$$NEG'_v = NEG_v + \begin{cases} uR_f & \text{if } f \text{ is neg. feedback} \\ (1 - Sim(v', v)) \cdot uR_f & \text{if } f \text{ is modification} \end{cases} \tag{7}$$

5. Experimental evaluation

This section provides preliminary experimental results to evaluate our approach. We implemented in Matlab[4] the reputation model described in the previous section and tested its accuracy in two different experimental settings. In the simulation, a version is a-priori labeled either as correct (i.e., supposed to represent truthfully the reality) or incorrect. Similarly, a user is labeled either as cooperative or non-cooperative (i.e., malicious) and behaves consequently. For example, non-cooperative users can: (i) create incorrect versions, (ii) give negative feedbacks, or modify, correct versions in order to reduce the reputations of versions and, as a consequence, of their authors. On the contrary, cooperative users can: (i) create correct versions and (ii) perform activities on existing versions coherent with their status. Moreover, a user can be either active, allowed performing any type of action on versions, or passive, allowed giving direct feedbacks only. By running the experiments we measure the accuracy of the reputation model in identifying correctly, i.e., according to the labels, cooperative/non-cooperative users and correct/incorrect versions. We recall that the objective of the proposed reputation model is to assign high reputation scores to cooperative users and to correct versions and low reputation scores otherwise. In the experiments, we assumed a cooperative user is identified correctly if the reputation score is in $(0.5, 1]$ at the end of the simulation; a non-cooperative user is identified correctly if the reputation score is in $[0, 0.5]$. We use the same criteria for identifying versions.

We present two experiments where we aim to observe the level of accuracy under different conditions. All the numerical results presented in the following are obtained as averages by running ten simulations[5].

In the first experiment, we want to compute the accuracy in identifying cooperative and non-cooperative users by varying the number of cooperative users. In this experiment, we disregard the aging of versions feature (i.e., we set $h(t_v) = 1$). The objective is to test the levels of accuracy even in a simplified version of the model. We set the

[4] https://www.mathworks.com/products/matlab.html

[5] The detailed parameters and output datasets of the simulations are available at: http://tinyurl.com/ant2017exp

percentage of active users to 10%, so 90% of users are not active. Approximately, this proportion between active users, contributing as reviewers, and users giving only feedbacks, reflects the Amazon community[6]. The type of an activity performed during the simulation is chosen randomly based on the probability distribution: *creation of a new version*, 1%; *modification*, 1%; *completion*, 1%; *direct feedback*, 97%. We set the total number of activities to 10000 and the number of users to 500. The results reporting the amount of users identified correctly is shown in Fig. 2(a). The result reporting the amount of correctly identified versions is in Fig. 2(b). We can notice the high performances of the model when cooperative users are respectively the 90%, 80% and 70% of the total users. The results are even better when we consider the amount of correct identification of versions receiving at least 5 feedbacks (not reported in the figure). On the contrary, when cooperative users decrease to 50% the performance falls down. In this case, the amounts of cooperative and non-cooperative users are equally balanced and every user is assigned with the same initial reputation score uR_0, so the model is not able to identify who is cooperative and who is not.

In the second experiment, we simulate the reputation model including also the mechanism of aging of versions. The parameter α influencing the aging rate is set to 180 days. In particular, aging of versions becomes relevant when we think about a version that in the past was describing correctly a real world object, but now is obsolete because the object changed meanwhile its status (e.g., the street name changed). So, it is desirable that negative feedbacks and modifications on a version older than the α period does not reduce the reputation of the author. Our current simulation software does not include this scenario where an object can change its status. Therefore, aging of versions does not bring a clear average in these simulations. We obtain slightly lower accuracies than in the previous experiment when we have 80% and 70% of cooperative users, even though the results look still good. The percents of correct identifications are shown in Fig. 2(c) for users and in Fig. 2(d) for contributions.

6. Applicative scenario: user-generated data for tourism planning

The widespread diffusion of personal location-aware devices combined with the advent of smart cities, where several kinds of sensors collect environment measures in real time, and the user generated geographical data shared through the Internet are creating a new kind of mobile, semantically rich sensed data. This opens the path to new application scenarios where the combination of accurate device-sensed data has to be combined with the less accurate user generated data. Improving the quality of volunteer collected data is a necessary step towards the the real usability of such data. An example of an application scenario that is gaining more and more attraction is the field of tourism itinerary recommendations.[7]

Tourists approaching their destination for the first time deal with the problem of planning a sightseeing itinerary that covers the most subjectively interesting attractions while fitting the time available for their visit. TripBuilder (*http://tripbuilder.isti.cnr.it*) is an unsupervised system helping tourists to build their own personalized sightseeing tour described in[5]. Given the target destination, the time available for the visit, and the tourist's profile, TripBuilder recommends a time-budgeted tour that maximizes tourist's interests and takes into account both the time needed to enjoy the attractions and to move from one PoI to the next one. A distinctive feature of TripBuilder is that the knowledge base feeding the recommendation model is entirely and automatically extracted from publicy available crowdsourced data like Flickr and Wikipedia.

However, Wikipedia as a source of PoIs for tourism recommendations has some limitations as some world parts like latin-america or asian countries are not covered by a sufficient number of Wikipedia pages describing tourist attractions and, because of this sparsity, need to rely on other local sources of crowd data. The integration of these local-based, up-to-date, user-generated and reliable Volunteer Geographical data source capable of describing the Points of Interests into the recommendation engine is crucial for supporting Tripbuilder in these areas. Another important aspect to be considerend in tourist travel planning is the collection of environmental quality measures that can affect the suggested paths to the most pleasant areas for the tourist. These environmental information can be sensed not only by physical sensors thanks to smart cities, but also from citizens like the discovering of the polluted or noisy areas[8].

[6] https://www.quora.com/What-percentage-of-buyers-write-reviews-on-Amazon
[7] https://get.google.com/trips/

In all these cases, it is crucial to assess dynamically the validity of these citizen-generated data[8]. This problem can be tackled in a tightly coupled architecture where TripBuilder dynamically interact with the reputation layer to asses the reputation level of the Points of Interests to be recommended to the user. When the version reputation value of a PoI goes below a fixed threshold, the recommendation engine immediately excludes the PoI from the suggested itinerary. In turn, users exploiting Tripbuilder to visit the PoI may contribute to the user-generated data with specific actions to modify, update or complete the attraction information.

7. Conclusions and perspectives

In this paper, a novel comprehensive model and its architecture for reputation evaluation of VGI content has been discussed and preliminarily evaluated by simulation. Our approach defines metrics to score a composite reputation of VGI data coming from unknown contributors. We promote the usefulness of our model by discussing the integration of VGI data with its reputation scores in applicative scenarios. Future work includes dealing with inconsistency that may arise due to repeated updates and completions for the same PoI. We are also planning to perform a deeper evaluation of the model, studying its possible joint use with other techniques for quality assessment in VGI data, and improving our architecture in order to integrate it with other crowd-sensed applications. From a broader perspective, we plan to integrate our findings in Mobile Crowdsourcing Systems in order to support processes of creation of Collective Intelligence.

References

1. Jamal Jokar Arsanjani, Peter Mooney, Alexander Zipf, and Anne Schauss. Quality assessment of the contributed land use information from openstreetmap versus authoritative datasets. In *OpenStreetMap in GIScience*, pages 37–58. Springer, 2015.
2. Mohamed Bishr and Werner Kuhn. Trust and reputation models for quality assessment of human sensor observations. In Thora Tenbrink, John Stell, Antony Galton, and Zena Wood, editors, *Spatial Information Theory: 11th International Conference, COSIT 2013, Scarborough, UK, September 2-6, 2013. Proceedings*, pages 53–73. Springer International Publishing, Cham, 2013.
3. Gloria Bordogna, Paola Carrara, Laura Criscuolo, Monica Pepe, and Anna Rampini. A linguistic decision making approach to assess the quality of volunteer geographic information for citizen science. *Information Sciences*, 258:312 – 327, 2014.
4. Boubaker Boulekrouche, Nafaâ Jabeur, and Zaia Alimazighi. Toward integrating grid and cloud-based concepts for an enhanced deployment of spatial data warehouses in cyber-physical system applications. *Journal of Ambient Intelligence and Humanized Computing*, 7(4):475–487, 2016.
5. Igo Ramalho Brilhante, Jose Antonio Macedo, Franco Maria Nardini, Raffaele Perego, and Chiara Renso. On planning sightseeing tours with tripbuilder. *Information Processing & Management*, 51(2):1 – 15, 2015.
6. Maria Antonia Brovelli, Marco Minghini, and Giorgio Zamboni. Public participation in GIS via mobile applications. *ISPRS Journal of Photogrammetry and Remote Sensing*, 114:306 – 315, 2016.
7. Fausto D'Antonio, Paolo Fogliaroni, and Tomi Kauppinen. VGI edit history reveals data trustworthiness and user reputation. In *17th AGILE International Conference on Geographic Information Science (Short Paper)*, 2014.
8. Jennifer Gabrys, Helen Pritchard, and Benjamin Barratt. Just good enough data: Figuring data citizenships through air pollution sensing and data stories. *Big Data & Society*, 3(2):2053951716679677, 2016.
9. Michael F Goodchild. Citizens as sensors: the world of volunteered geography. *GeoJournal*, 69(4):211–221, 2007.
10. Michael F. Goodchild and Linna Li. Assuring the quality of volunteered geographic information. *Spatial Statistics*, 1:110 – 120, 2012.
11. Denis Havlik, Javier Soriano, Carolos Granell, Stuart E. Middleton, Hylke van der Schaaf, Arne Berre, and Jasmin Pielorz. Future internet enablers for VGI applications. In *EnviroInfo 2013*. September 2013.
12. Peter Mooney, Padraig Corcoran, and Blazej Ciepluch. The potential for using volunteered geographic information in pervasive health computing applications. *Journal of Ambient Intelligence and Humanized Computing*, 4(6):731–745, 2013.
13. Hansi Senaratne, Amin Mobasheri, Ahmed Loai Ali, Cristina Capineri, and Muki Haklay. A review of volunteered geographic information quality assessment methods. *International Journal of Geographical Information Science*, May 2016.
14. Wanita Sherchan, Surya Nepal, and Cecile Paris. A survey of trust in social networks. *ACM Comput. Surv.*, 45(4):47:1–47:33, August 2013.
15. Cao Xiao, David Mandell Freeman, and Theodore Hwa. Detecting clusters of fake accounts in online social networks. In *Proceedings of the 8th ACM Workshop on Artificial Intelligence and Security*, AISec '15, pages 91–101, New York, NY, USA, 2015. ACM.
16. B. Ye and Y. Wang. Crowdrec: Trust-aware worker recommendation in crowdsourcing environments. In *2016 IEEE International Conference on Web Services (ICWS)*, pages 1–8, June 2016.
17. H. Yu, Z. Shen, C. Leung, C. Miao, and V. R. Lesser. A survey of multi-agent trust management systems. *IEEE Access*, 1:35–50, 2013.
18. Yijiang Zhao, Xiaoguang Zhou, Guangqiang Li, and Hanfa Xing. A Spatio-Temporal VGI Model Considering Trust-Related Information. *ISPRS International Journal of Geo-Information*, 5(2):10, 2016.

[8] Rise of the citizen scientist, http://www.nature.com/news/rise-of-the-citizen-scientist-1.18192