Towards BOTTARI: Using Stream Reasoning to Make Sense of Location-Based Micro-Posts

Irene Celino[1], Daniele Dell'Aglio[1], Emanuele Della Valle[2,1], Yi Huang[3], Tony Lee[4], Seon-Ho Kim[4], and Volker Tresp[3]

[1] CEFRIEL – ICT Institute, Politecnico of Milano, Milano, Italy
[2] Dip. di Elettronica e dell'Informazione – Politecnico di Milano, Milano, Italy
[3] SIEMENS AG, Corporate Technology, Muenchen, Germany
[4] Saltlux, Seoul, Korea

Abstract. Consider an urban environment and its semi-public realms (e.g., shops, bars, visitors attractions, means of transportation). Who is the maven of a district? How fast and how broad can such maven influence the opinions of others? These are just few of the questions BOTTARI (our Location-based Social Media Analysis mobile app) is getting ready to answer. In this position paper, we recap our investigation on deductive and inductive stream reasoning for social media analysis, and we show how the results of this research form the underpinning of BOTTARI.

1 Introduction

In the last few years, we have been witnessing the increasing popularity and success of Location-based Services (LBS), especially of those with a Social Networking flavour. Twitter, Facebook Places, foursquare, Gowalla are only a few application examples; those services bring a wide range on useful information about tourist attractions, local businesses and points of interests (POIs) in the physical world. Although these services are enormously popular, users still suffer from a number of shortcomings. The overwhelming information flow coming from those channels often confuses users; it is also very difficult to distinguish between a fair personal opinion and a malicious or opportunistic advice. This might be the reason why users primarily link to people they know personally since, in an on-line social network, there is no clear way to know who to trust.

In this paper, we present our collaborative effort to the design and development of the BOTTARI application, a Location-based Service for mobile users that exploit Social Media Analysis techniques to identify the "mavens" of a specific geographical area, i.e. those people who can be considered as experts of the POIs in this area. BOTTARI was conceived by Saltlux, a Korean Knowledge Communication Company. The application is still under development and it will be made available to Korean users in the Seoul area. BOTTARI exploits hybrid Stream Reasoning both on heterogeneous social network data [1] and geo-location data. The hybrid reasoning engine combines deductive and inductive

R. García-Castro et al. (Eds.): ESWC 2011 Workshops, LNCS 7117, pp. 80–87, 2011.

techniques. Since the input data are huge and change in real-time, the reasoning engine works by processing streaming data. The hybrid reasoning engine is developed on top of the LarKC platform [2], a pluggable architecture to build applications with Semantic Web technologies.

The remainder of the paper is organised as follows. Section 2 explains the concept of stream reasoning and delineates the system architecture. Section 3 describes the BOTTARI app. Section 4 details some user questions in terms of queries to our stream reasoner. Finally, Section 5 concludes the paper.

2 System Architecture

Continuous processing of information flows (i.e. **data streams**) has widely been investigated in the database community. [3]. In contrast, continuous processing of data streams *together with rich background knowledge* requires semantic reasoners, but, so far, semantic technologies are still focusing on rather static data. We strongly believe that there is a need to close this gap between existing solutions for belief update and the actual need of supporting decision making based on data streams and rich background knowledge. We named this little-explored, yet high-impact research area **Stream Reasoning** [4]. The foundation for Stream Reasoning has been investigated by introducing technologies for wrapping and querying streams in the RDF data format (e.g., using C-SPARQL [5]) and by supporting simple forms of reasoning [6] or query rewriting [7].

We are developing the Stream Reasoning vision on top of LarKC [8]. The LarKC platform is aimed to reason on massive heterogeneous information such as social media data. The platform consists of a framework to build workflows, i.e. sequences of connected components (plug-ins) able to consume and process data. Each plug-in exploits techniques and heuristics from diverse areas such as databases, machine learning and the Semantic Web.

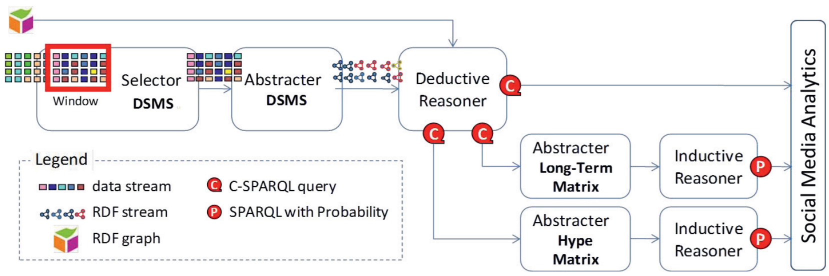

Fig. 1. Architecture of our Stream Reasoner

We built our Stream Reasoning system by embedding a deductive reasoner and an inductive reasoner within the LarKC architecture (see Figure 1). First, BOTTARI pre-processes the micro-posts by extracting information[1] whether a micro-post expresses a positive or a negative feeling of its author about a certain POI.

[1] Those technological details are Saltlux trade secrets.

After BOTTARI data arrives to the stream reasoner as a set of data streams, a selection plug-in extracts the relevant data in each input window of the stream. A second plug-in abstracts the window content from the fine grain data streams into aggregated events and produces RDF streams. Then, a deductive reasoner plug-in is able to register C-SPARQL queries, whose results can be of immediate use (cf. Section 4) or can be processed by other two sub-workflows. Each sub-workflow is constituted by an abstracter and an inductive reasoner, which uses an extended version of SPARQL that supports probabilities [9].

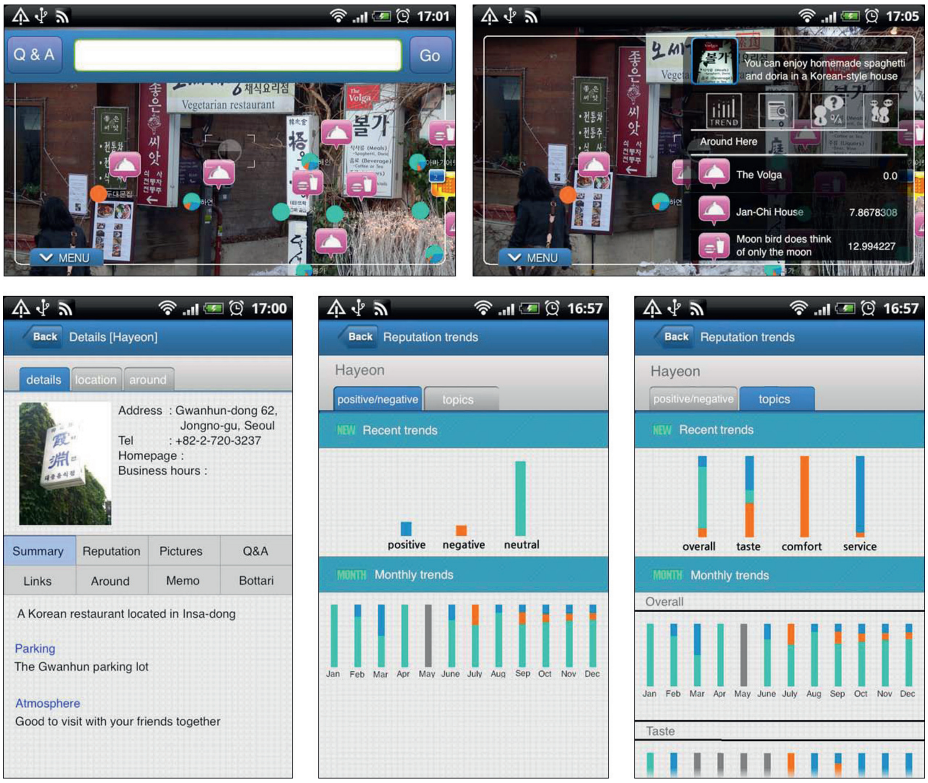

Fig. 2. Some screenshots of the BOTTARI Android application

3 The BOTTARI Mobile App

Bottari is a Korean word that refers to a bundle or container made from patterned cloth that is used to transport a one's belongings when travelling. The BOTTARI mobile app is a location-based service that exploits the social context to provide relevant contents to the user in a specific geographic location; as such, BOTTARI lets the user "transport" the location-specific knowledge, derived from the local mavens' expertise, when moving in the physical space.

The purpose of the BOTTARI service is to provide recommendations on local context information to users through an augmented reality interface. BOTTARI gives detailed information on local POIs, including trust or reputation information. In Figure 2, we provide some sample screenshots on how the BOTTARI mobile application will look like once completed. The screenshots in the upper part of Figure 2 show how a user searches for POIs of a given kind (e.g., restaurants or snack bars) around her position and explores them using augmented reality. A small pie graph shows the results of the sentiment analysis for each POI: blue for positive, red for negative, and green for neutral feeling. The screenshots in the bottom part of Figure 2 show how a user visualizes more detailed information about a POI. They are, from left to right, the POI identity card, the global sentiment analysis (again, blue, red and green represent positive, negative and neutral feeling respectively) and the detailed sentiment analysis on different topics (e.g., taste, comfort and service for a restaurant).

The input data for the BOTTARI service come from public social networks and location based services (Twitter, local blogs and Korean news). They are converted into RDF streams and then processed and analysed by the system described in Section 2. The RDF-ized data are modelled with respect to the ontology represented in Figure 3, which is an extension to the SIOC vocabulary [10]. Our model takes into account the specific relations of Twitter (follower/following, reply/retweet); it adds the geographical perspective by modelling the POIs; it includes the "reputation" information by means of positive/negative/neutral reviews.

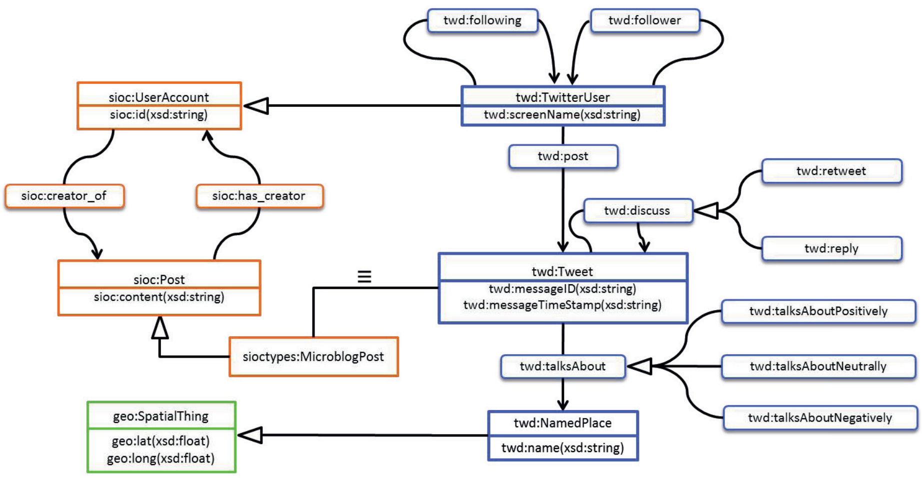

Fig. 3. Ontology modelling of BOTTARI data

4 Computing Answers to User Questions

The hybrid Stream Reasoning solutions we are developing is able to answer questions like: Who are the opinion makers (i.e., the users who are likely to influence the behaviour of their followers with regard to a certain POI)? How fast and how wide are opinions spreading? Who shall I follow to be informed about a given category of POIs in this neighbourhood? Which persons similar to me are nearby at an interesting POI?

In the rest of the section we show how to formulate the four queries above using C-SPARQL and SPARQL with probabilities.

Who are the Opinion Makers?
Lines 1 and 4 of the following listing tell the C-SPARQL engine to register the continuous query on the stream of micro-posts generated by BOTTARI considering a sliding window of 30 minutes that slides every 5 minutes. Lines 2 and 3 tells the engine that it should generate an RDF stream as output reporting the opinion makers for a certain POI.

```
1. REGISTER STREAM OpinionMakers COMPUTED EVERY 5m AS
2. CONSTRUCT { ?opinionMaker a twd:opinionMaker ;
3.                    twd:posts [ twd:talksPositivelyAbout ?poi ] . }
4. FROM STREAM <http://bottari.saltlux.com/posts>  [RANGE 30m STEP 5m]
5. WHERE {
6.       ?opinionMaker a twd:TwitterUser ;
7.                      twd:posts [ twd:talksPositivelyAbout ?poi ] .
8.       ?follower sioc:follows ?opinionMaker;
9.                  twd:posts [ twd:talksPositivelyAbout ?poi ] .
10.      FILTER (cs:timestamp(?follower) > cs:timestamp(?opinionMaker))
11. }
12. HAVING ( COUNT(DISTINCT ?follower) > 10 )
```

The basic graph pattern (BGP) at lines 6–7 matches positive micro-posts of the potential opinion makers about a set of POIs. The BGP at lines 8–9 looks up the followers of the opinion makers who also positively posted about the same set of POIs. The FILTER clause at line 10 checks whether the micro-posts of the followers occur after those from the opinion makers. Finally, at line 12 the clause HAVING promotes to *true* opinion makers those who have at least ten such followers.

How Fast and Wide Opinions are Getting Spread?
Using the RDF stream computed by the previous query, the query in the following listing informs about how wide the micro-posts of an opinion maker are getting spread in half an hour. To do so, it considers the reply and re-tweet relationships among tweets (i.e., tweets linked by the `discuss` property in BOTTARI data model). Being `discuss` a transitive property, the C-SPARQL engine uses the materialization technique presented in [6] to incrementally compute the transitive closure of `discuss`.

```
1.  REGISTER STREAM OpinionSpreading COMPUTED EVERY 30s AS
2.  SELECT ?opinionMaker ?opinionMakerTweet
3.         COUNT(?positiveTweet) COUNT(?negativeTweet)
4.  FROM STREAM <http://bottari.saltlux.com/posts> [RANGE 30m STEP 30s]
5.  FROM STREAM <http://bottari.../OpinionMakers [RANGE 30m STEP 30s]
6.  WHERE {
7.        ?opinionMaker a twd:opinionMaker ;
8.                      twd:post ?opinionMakerTweet .
9.        { ?positiveTweet a twd:Tweet ;
10.                        twd:discuss ?opinionMakerTweet ;
11.                        twd:talksAboutPositively ?poi .  }
12.       UNION
13.       { ?negativeTweet a twd:Tweet ;
14.                        twd:discuss ?opinionMakerTweet ;
15.                        twd:talksAboutNegatively ?poi .  }
```

Lines 1, 4 and 5 tell the C-SPARQL engine to register the continuous query on the stream of micro-posts generated by BOTTARI and on the streaming results of the opinion makers query. In both cases, a sliding window of 30 minutes, which slides every 30 seconds, is considered. The BGP at lines 7–8 matches the micro-posts of the opinion makers. The group pattern at lines 9–11 and the group pattern at lines 13–15 look up other micro-posts that, respectively, positively and negatively discussed those of the opinion makers. Lines 2–3 ask the engine to generate a variable binding reporting how many positive and negative micro-posts are discussing the micro-posts of the current opinion makers.

Who shall I Follow?

Let us consider now a specific BOTTARI user named Giulia. In the following listing we show a query that asks for the mavens Giulia should follow to be informed about attractions for kids, even among people she does not know. The system uses the social network of Giulia and the last window in the stream (generated by the query in the first listing) to determine such predicted probability.

```
1.  SELECT ?user ?prob
2.  FROM STREAM <http://bottari.../OpinionMakers [RANGE 30m STEP 30s]
3.  WHERE{
4.      ?opinionMaker a twd:opinionMaker ;
5.                    twd:posts [ twd:talksAboutPositively ?poi ] .
6.      ?poi skos:subject twd:attractionsForKids .
7.      { :Giulia twd:following ?opinionMaker.
8.        WITH PROBABILITY AS ?prob
9.        ENSURE PROBABILITY [0.8,1] }
10. } ORDER BY DESC(?prob)
```

The BGP at lines 4–6 matches those opinion makers that have recently been expressing positive opinions about attractions for kids. The group pattern at lines 7–9 makes use of SPARQL with probability [9]. The triple pattern at line

7 matches BOTTARI users that Giulia is following. Note that the `following` relationship may have not been asserted yet, the WITH PROBABILITY clause at line 8 extends SPARQL by letting it query an induced model. The variable `?prob` may assume values between 0 and 1, where the value 1 means that she already follows that user. The ENSURE PROBABILITY clause at line 9 accepts only those solutions whose estimated probabilities is larger or equal to 0.8 and less than 1, i.e. those mavens who should be recommended to Giulia. Finally, the ORDER BY clause is used to return users sorted by decreasing probability. The query answer includes pairs of users and predicted likelihoods (e.g. `:Alice` with probability 0.99, `:Bob` with probability 0.87).

What People Similar to Me Are Nearby in an Interesting POI?
A more complex example of query soliciting all BOTTARI features is as follows. Let's consider that Giulia is now in a specific location and she is looking for people who share her preferences and who are nearby in an interesting POI. Both physical proximity and recency of micro-posting are to be considered.

```
1.  PREFIX ogc: <http://www.opengis.net/geosparql#>
2.  PREFIX ogcf: <http://www.opengis.net/geosparql/functions#>
3.  SELECT ?poi1 ?user ?prob
4.  FROM STREAM <http://bottari.../streamOftweets> [RANGE 1h STEP 10m]
5.  WHERE {
6.    ?user twd:post [ twd:talksPositivelyAbout ?poi1 ] .
7.    ?poi1 geo:lat ?lat1; geo:long ?long1 ; skos:subject ?category .
8.    :Giulia twd:post [ twd:talksAbout ?poi2 ] .
9.    ?poi2 geo:lat ?lat2; geo:long ?long2 ; skos:subject ?category .
10.    FILTER( ogcf:distance(ogc:Point(?lat1,?long1),
11.                          ogc:Point(?lat2,?long2), ogc:km) < 0.1 )
12.   { :Giulia twd:following ?user.
13.     WITH PROBABILITY AS ?prob
14.     ENSURE PROBABILITY [0.8,1) }
15. }
16. ORDER BY DESC(?prob)
```

Line 4 indicates that activities should be related to the latest period; lines 8–9 determine where Giulia is, while lines 6–7 ask for users who recently tweeted positively about a POI of the same category of the one where Giulia is. Lines 10–11 make sure that the POI is nearby by using GeoSPARQL, a proposal by the Open Geospatial Consortium[2]. The group pattern at lines 12–14 leverages inductive reasoning to ensure at least a 80% similarity between Giulia and the nearby tweeters; this probabilistic value is used to rank results (cf. line 16).

5 Conclusions and Future Works

In this paper we presented BOTTARI, a location-based mobile application which is able to supply contents and personalized suggestions to its users. We explained

[2] Cf. `http://www.opengeospatial.org/projects/groups/geosparqlswg`

the processing of new recommendations, based on the elaboration of data streams generated by microblogging platforms like Twitter and foursquare. The computation is defined as a workflow combining Semantic Web and machine learning techniques and it is executed on top of the LarKC platform.

Our future work will focus on the development of the first stable version of the BOTTARI application and its release as Android app. The initial release will focus on Korea and will be evaluated by following a user-centered approach: a set of users will try out the application, supplying us feedbacks via a survey with questions about the system and its accuracy in providing suggestions.

Acknowledgments. This work was partially supported by the EU project LarKC (FP7-215535).

References

1. Barbieri, D.F., Braga, D., Ceri, S., Della Valle, E., Huang, Y., Tresp, V., Rettinger, A., Wermser, H.: Deductive and Inductive Stream Reasoning for Semantic Social Media Analytics. IEEE Intelligent Systems 25(6), 32–41 (2010)
2. Cheptsov, A., et al.: Large Knowledge Collider. A Service-oriented Platform for Large-scale Semantic Reasoning. In: Proceedings of WIMS 2011 (2011)
3. Garofalakis, M., Gehrke, J., Rastogi, R.: Data Stream Management: Processing High-Speed Data Streams. Springer-Verlag New York, Inc. (2007)
4. Della Valle, E., Ceri, S., van Harmelen, F., Fensel, D.: It's a Streaming World! Reasoning upon Rapidly Changing Information. IEEE Intelligent Systems 24(6), 83–89 (2009)
5. Barbieri, D.F., Braga, D., Ceri, S., Della Valle, E., Grossniklaus, M.: C-SPARQL: a Continuous Query Language for RDF Data Streams. Int. J. Semantic Computing 4(1), 3–25 (2010)
6. Barbieri, D.F., Braga, D., Ceri, S., Della Valle, E., Grossniklaus, M.: Incremental Reasoning on Streams and Rich Background Knowledge. In: Aroyo, L., Antoniou, G., Hyvönen, E., ten Teije, A., Stuckenschmidt, H., Cabral, L., Tudorache, T. (eds.) ESWC 2010. LNCS, vol. 6088, pp. 1–15. Springer, Heidelberg (2010)
7. Ren, Y., Pan, J.Z., Zhao, Y.: Towards Scalable Reasoning on Ontology Streams via Syntactic Approximation. In: Proc. of IWOD 2010 (2010)
8. Fensel, D., et al.: Towards LarKC: a Platform for Web-scale Reasoning. In: Proc. of ICSC 2008 (2008)
9. Tresp, V., Huang, Y., Bundschus, M., Rettinger, A.: Materializing and querying learned knowledge. In: Proc. of IRMLeS 2009 (2009)
10. Berrueta, D., et al.: SIOC Core Ontology Specification. W3C Member Submission, W3C (2007)