Location-based Mobile Recommendations by Hybrid Reasoning on Social Media Streams

Tony Lee, Seon-Ho Kim, Marco Balduini, Daniele Dell'Aglio,
Irene Celino, , Yi Huang, Volker Tresp, Emanuele Della Valle

Duckil Bldg., 976 Deachi-dong, Kangnam-gu Seoul, Korea
{tony, shkim}@saltlux.com, {marco.balduini,
daniele.dellaglio, emanuele.dellavalle}
@polimi.it, irene.celino@cefriel.com,
{yihuang, volker.tresp}@siemens.com
http://www.saltlux.com

Abstract. In this paper, we introduce BOTTARI: an augmented reality application that offers personalized and location-based recommendations of Point Of Interests based on sentiment analysis with geo-semantic query and reasoning. We present a mobile recommendation platform and application working on semantic technologies (knowledge representation and query for geo-social data, and inductive and deductive stream reasoning), and the lesson learned in deploying BOTTARI in Insadong. We have been collecting and analyzing tweets for three years to rate the few hundreds of restaurants in the district. The results of our study show the commercial feasibility of BOTTARI.

Keywords: Social media analytics, Mobile recommendation, Stream reasoning, Hybrid reasoning, Machine learning, Semantic Web, Ontology

1 Introduction

When a tourist visits new place, they would face the challenges to discover proper restaurants, shops, or other tourist attractions. Usually, if you want to have a dinner in Seoul, it's quite hard to select a preferred restaurant among a hundreds restaurants in the district they are visiting (see Figure 1). BOTTARI is an augmented reality application for personalized and localized restaurant recommendations, experimentally deployed in the Insadong district of Seoul. At a first look, it may appear like other mobile apps that recommend restaurants, but BOTTARI is different: BOTTARI uses inductive and deductive stream reasoning [1] to continuously analyze social media streams (specifically Twitter) to understand how the social media users collectively perceive the points of interest (POIs) in a given area, e.g., Insadong's restaurants. In this paper, we describe the choices we made in designing BOTTARI and the lessons we learned by experimentally deploying it in Insadong.

Fig. 1. A picture of Indadong: the density of restaurants is very high

2 Background Work

When reasoning on massive data streams, well known artificial intelligence techniques have the right level of expressivity, but their throughput is not high enough to keep pace with the stream. The only technological solutions with the right throughput are Data Stream Management Systems (DSMS) [3] and Complex Event Processing [4], but, on the other hand, they are not expressive enough. A new type of inference engines is thus needed to reason on streams. Della Valle et al. named them *stream reasoners* [2]. A number of stream reasoning approaches have been developed. They share three main concepts: (a) they logically model the information flow as an RDF stream, i.e. a sequence of RDF triples annotated with one or more non-decreasing timestamps, (b) they process the RDF streams “on the fly”, often by re-writing queries to the raw data streams, and (c) they exploit the temporal order of the streaming data to optimize the computation. BOTTARI uses both a deductive and an inductive stream reasoner. The deductive stream reasoner is based on Continuous SPARQL (C-SPARQL) [5] – an extension of SPARQL that continuously processes RDF streams observed through windows (as done in DSMS) - and exploits the Streaming Linked Data (SLD) framework [6]. The inductive stream reasoner is based on SUNS (Statistical Unit Node Set) approach [7, 8] – a scalable machine learning framework for predicting unknown but potentially true statements by exploiting the regularities in structured data. The SUNS employs a modular regularized multivariate learning approach able to deal with very high-dimensional data [9] and to integrate temporal information using Markov decomposition [10].

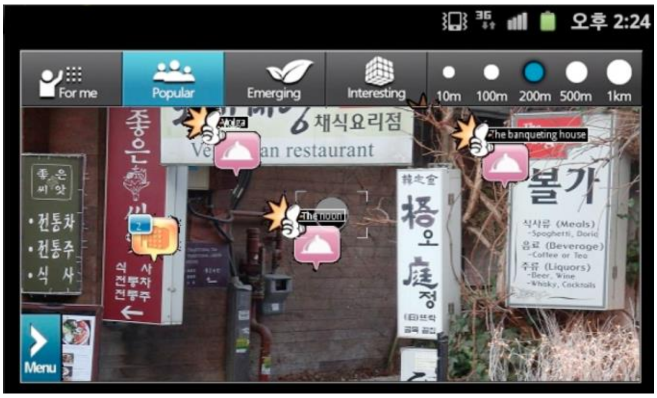
The LarKC platform [12] is used for orchestrating SLD, SUNS and the geo-spatial query engine and exposing their aggretated capabilities as a SPARQL endpoint. It is a pluggable Semantic Web framework that can be deployed on a high-performance computing cluster. The LarKC platform is the main result of the EU FP7 integrated project, Large Knowledge Collider [11].

3 The BOTTARI Mobile App

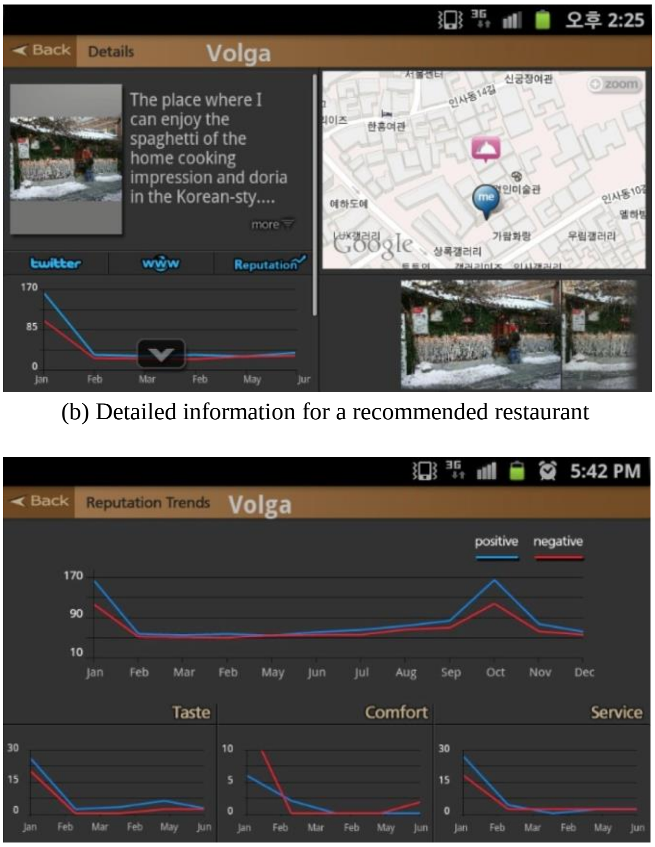
As shown in Figure 2, BOTTARI is an Android application (for smart phones and tablets) in augmented reality (AR) that directs the users' attention to restaurants. In Korean language, "bottari" is a cloth bundle that carries a person's belongings while travelling. BOTTARI carries the collective perceptions of social media users about POIs in an area and uses them to recommend POIs. As shown in Figure 2(a), BOTTARI users can search POIs in their proximity using four buttons:

1. *'For Me'* that emphasizes the personalization of POI suggestions as in local search;
2. *'Popular'* that emphasizes the presence of positive ratings of social media users;
3. *'Emerging'* that focuses on the most recent ratings posted on social media capturing seasonal effects (e.g., Insadong people seems to prefer meat restaurants in winter rather than in summer) or POIs "on fashion" for a limited period; and
4. *'Interesting'* that returns the POIs described with a category of interest for the user.

Users can see recommended POIs based on their preferences in AR, as shown in Figure 2(a). In this view, the POIs are indicated with different icons. Thumb-up and thumb-down icons indicate social reputation as positively or negatively perceived on social media. Moreover, given the importance of the distance between the user and the recommended POIs, BOTTARI offers functionality for distance-based filtering of the recommended POIs; see the circles in the right-upper side of Figure 2(a). The user can learn more about a POI as shown in Figure 2(b). Figure 2(c) shows a peculiar feature of BOTTARI: the trend over time of the POI reputation as collectively perceived on social media.



(a) Recommendations with AR function



The place where I can enjoy the spaghetti of the home cooking impression and doria in the Korean-sty....

(b) Detailed information for a recommended restaurant

(c) Reputation trend analysis

Fig. 2. Screenshots of the BOTTARI Android application

A video displaying BOTTARI at work, on a mobile phone and on a tablet, is available on YouTube at http://www.youtube.com/watch?v=c1FmZUz5BOo.

4 Data Set Used in BOTTARI

BOTTARI is built on two types of data: the geo ontology for the POIs and the social media streams.

4.1 Geo Ontology for the POIs

Insadong is a 2 km^2 district with a high density of restaurants. For BOTTARI, the information about 319 restaurants and 1850 tourist attractions of Insadong were collected with a considerable manual effort. The result is a manually curated high quality geo-referenced knowledge base where each restaurant is described by 44 properties (e.g., name, images, position, address, ambiance, specialties, categories, etc.). We designed geo-context model based on RDF/OWL and SKOS. All the POIs were classified into 600 taxonomic classes described by SKOS.

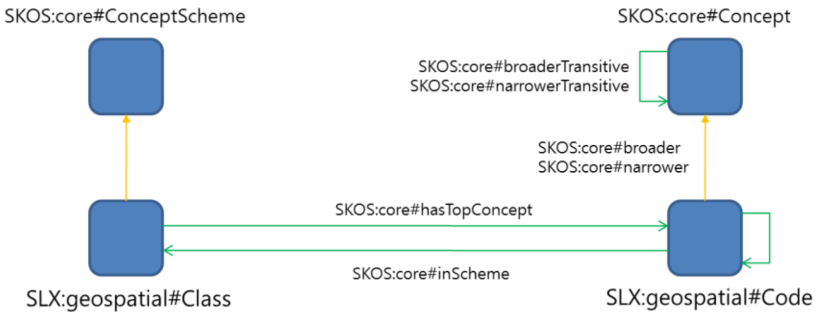
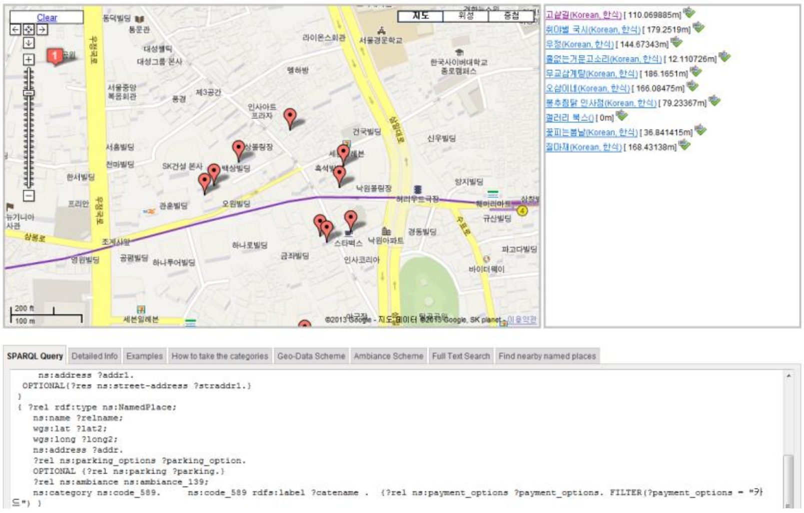


Fig. 3. Geo Context Model for BOTTARI

We also introduced Geo-SPARQL to query proper POI sets with sophisticate conditions including GPS coordinate, distance, category, ambiance, menu, price, parking, smoking and etc. For example, we can query to find “a Korean restaurant within 5 minutes’ walk from Gallery Books which is possible parking, credit card, price is between 10,000 and 30,000, is also good for teenager or baby” like below. The query is presented hereafter, while a screenshot of the BOTTARI workbench used for testing Geo-SPARQL queries in show in Figure 4.

```
PREFIX ns: <http://lod.saltlux.kr/geospatial/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX wgs: <http://www.w3.org/2003/01/geo/wgs84_pos#>
PREFIX f: <http://www.saltlux.com/geo/functions#>
SELECT *
WHERE {
 { ?res ns:name ?name. FILTER (?name = "갤러리 북스")
   ?res rdf:type ns:NamedPlace.
   ?res wgs:lat ?lat1;
        wgs:long ?long1;
        ns:address ?addr1.
   OPTIONAL{?res ns:street-address ?straddr1.}
 }
 { ?rel rdf:type ns:NamedPlace;
    ns:name ?relname;
    wgs:lat ?lat2;
    wgs:long ?long2;
    ns:address ?addr.
    ?rel ns:parking_options ?parking_option.
    OPTIONAL {?rel ns:parking ?parking.}
    ?rel ns:ambiance ns:ambiance_139;
    ns:category ns:code_589.     ns:code_589
    rdfs:label ?catename .  {?rel ns:payment_options
    ?payment_options. FILTER(?payment_options = "카드") }
 {?rel ns:price ?price. FILTER(f:between(?price, 10000,
    30000) )}
 {?rel ns:good_for ?goodfor. FILTER(?goodfor = "미성년"
    || ?goodfor = "유아") }
 OPTIONAL{?rel ns:street-address ?straddr.}
}
FILTER (f:within_distance(?lat1, ?long1, ?lat2, ?long2,
        200) )
} ORDER BY ?relname
```



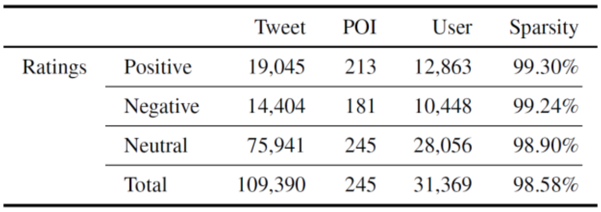
고삼길(Korean, 한식) [110.069885m]
취야벌 국시(Korean, 한식) [179.2519m]
우정(Korean, 한식) [144.67343m]
줄없는거문고소리(Korean, 한식) [12.110726m]
무교삼계탕(Korean, 한식) [186.1651m]
오심이네(Korean, 한식) [166.08475m]
봉추찜닭 인사점(Korean, 한식) [79.23367m]
갤러리 북스() [0m]
꽃피는봄날(Korean, 한식) [36.841415m]
질마재(Korean, 한식) [168.43138m]

```
    ns:address ?addr1.
  OPTIONAL{?res ns:street-address ?straddr1.}
 }
 { ?rel rdf:type ns:NamedPlace;
   ns:name ?relname;
   wgs:lat ?lat2;
   wgs:long ?long2;
   ns:address ?addr.
   ?rel ns:parking_options ?parking_option.
   OPTIONAL {?rel ns:parking ?parking.}
   ?rel ns:ambiance ns:ambiance_139;
   ns:category ns:code_589.     ns:code_589 rdfs:label ?catename .  {?rel ns:payment_options ?payment_options. FILTER(?payment_options = "카
드") }
```

Fig. 4. Geo-SPARQL query results on BOTTARI workbench

4.2 Social Media Streams

The social media stream is gathered from the Web (in particular from Twitter) and converted into an RDF stream using the proprietary crawling and sentiment analysis infrastructure of Saltlux, Inc. The data used for the experiments were collected in 3 years, from February 4th, 2008 to November 23rd, 2010 (1,023 days). 200 million tweets were analyzed and, as a result, 109,390 tweets posted by more than 31,369 users were discovered to positively, neutrally or negatively talk about 245 restaurants in Insadong. More information about the technique employed for sentiment analysis is provide in section 5.



		Tweet	POI	User	Sparsity
Ratings	Positive	19,045	213	12,863	99.30%
	Negative	14,404	181	10,448	99.24%
	Neutral	75,941	245	28,056	98.90%
	Total	109,390	245	31,369	98.58%

Table 1. Statistics of sentiment analysis results from Twitter

This data stream is characterized by:

- *high sparsity* – defining sparsity as *1 - #Ratings / (#POIs* × *#Users)*. For instance, the sparsity of the positive ratings is 99.3%;
- *incompleteness* – only 41% of users positively rated at least one POI;
- *inconsistencies* – the same user can rate a particular POI several times expressing different opinions;
- *exponential growth over time* – data shows the exponential growth in the usage of Twitter in Korea starting from December 2009 to 2012; and
- *long-tail distribution* – ratings follows a long-tail distribution with few users that rated many POIs, and many users that rated one or two POIs.

Approaches for sentiment analysis could be divided in 2 types: machine learning based or rule (pattern matching) based. We applied hybrid model consists of syllable kernel based SVM and NLP rule engine to analyze sentiments from real-time tweet streams. We identify linguistic patterns of positive or negative sentiments in text string and encoded them into the rule set:

- adjectives (e.g. 맛있다/tasty, 재미있다/funny, 편하다/comfortable) are used to identify the polarity
- adverbs generally preceding amplify the strength of the polarity beard by the adjective.
- some noun phrase (e.g. 마음에 들어요/like it, 문제가 많아요/have many problems) are also used to identify polarity; and
- three different ways to make negative sentiment.

1) post derivation ‘지 않다’ (e.g. ‘마음에 들지 않아요’)

2) prefix ‘안’ (e.g. ‘안 불편해’)

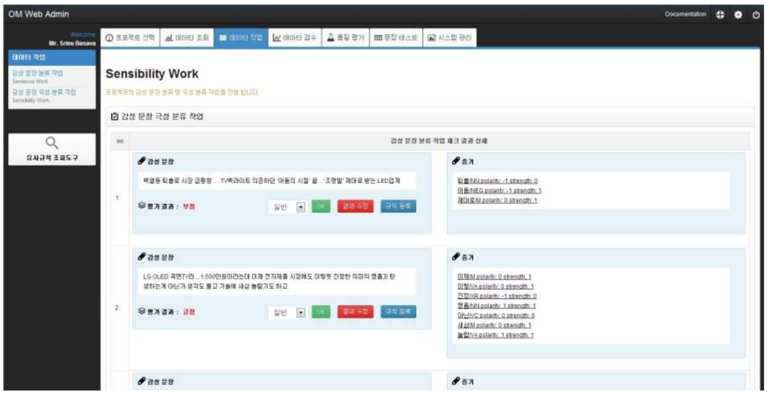
3) prefix ‘못’ (e.g. ‘못 해봤어’)

More specifically, each rule consists of one linguistic pattern and two attributes like:

- Polarity (positive, neutral, negative that we encoded respectively as: +1,0,-1)

- Strength (a positive number reflecting how strong is the sentiment. We encoded the range from 0 to 5)

The grammar for defining linguistic pattern should be considered Part-Of-Speech (POS) and other linguistic features because the adjectives or verbs in Korean can have almost infinite flectional derivations (called 'eomi') and the original form of the verb has to retrieved effectively. Same canonical form of a token can correspond to several different meanings and POSs (e.g. 먹/V, 먹/N). We designed tree based pattern matching algorithm where the match can check up to 3 consecutive tokens. For extracting linguistic patterns and sentiment words, we ran a quite big Korean corpus composed of several domains (food, restaurant, purchasing, movies, and etc.) and applied a mutual information measure to extract the top most candidates. Figure 5 whos the workbench for sentiment analysis used in BOTTARI.



Sensibility Work

프로젝트의 감성 문장 분류 및 극성 분류 작업을 진행 합니다.

감성 문장 극성 분류 작업

감성 문장 분류 작업 체크 결과 상세

1

감성 문장

백열등 퇴출로 시장 급팽창 … TV백라이트 의존하던 '어둠의 시절' 끝… '조명발' 제대로 받는 LED업계

평가 결과 : 부정

증거

퇴출/NN polarity: -1 strength: 0
어둠/NEG polarity: -1 strength: 1
제대로/M polarity: 0 strength: 1

2

감성 문장

LG OLED 곡면TV란… 1,500만원이라는데 이제 전자제품 시장에도 이렇듯 진정한 의미의 명품기 탄생하는게 아닌가 생각도 들고 기술에 새삼 놀랍기도 하고

평가 결과 : 긍정

증거

이제/M polarity: 0 strength: 1
이렇/VA polarity: 0 strength: 1
진정/XR polarity: -1 strength: 0
명품/NN polarity: 1 strength: 1
아닌/VC polarity: 0 strength: 0
새삼/M polarity: 0 strength: 1
놀랍/VA polarity: 1 strength: 1

감성 문장

증거

Fig. 5. Workbench for sentiment analysis

5 Ontology for BOTTARI

We designed BOTTARI following an ontology-based information access architecture[13]. BOTTARI ontology is represented in Figure 6. It extends the SIOC vocabulary defining *TwitterUser* as a special case of *UserAccont* and the concept of Tweet as being equivalent to Post. It models the notion of POI as *NamedPlace* extending *SpatialThing* from the W3C WGS-84 vocabulary. A *NamedPlace* is enriched with a categorization (e.g., the ambience describing the atmosphere of a restaurant) and the count of positive/negative/neutral ratings. The most distinctive feature of BOTTARI ontology is the object property *talksAbout* – and its sub-properties for positive, negative and neutral opinions – that allows to state that a Tweet (positively, negatively or neutrally) talks about a *NamedPlace*.

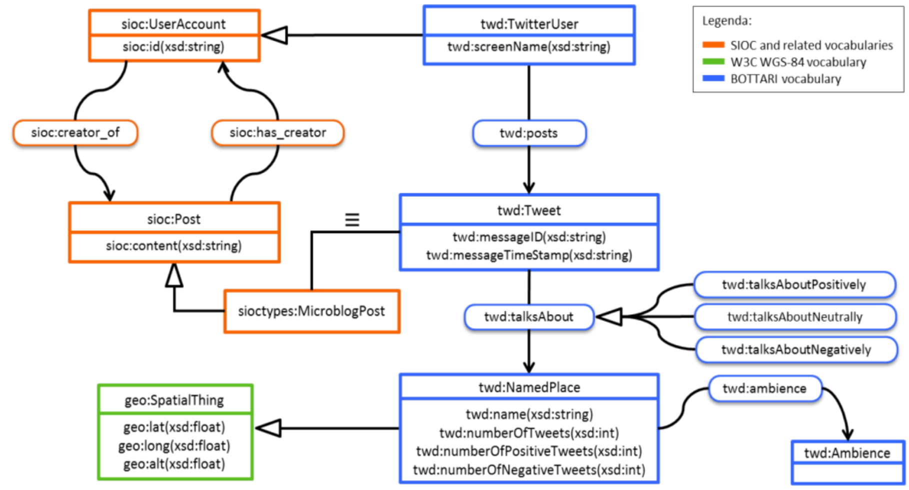


Fig. 6. Ontology model for BOTTARI

6 Architecture and Components

BOTTARI architecture consists of three parts: (a) client side that interacts with the user and communicates to the back-end sending SPARQL queries, (b) a data initiated segment (PUSH) that continuously analyses the social media streams, and (c) a query initiated segment (PULL) that uses the LarKC platform to answer the SPARQL queries of the client by combining several forms of reasoning.

6.1 The PUSH segment

The PUSH segment continuously analyses the social media streams crawled from the Web. The SEMANTIC MEDIA CRAWLER AND OPINION MINER crawl 3.4 million tweets/day related to Seoul, identifies the subset related to the Insadong restaurants (thousands per day) and extracts the users' opinions. The result is an RDF stream of positive, negative and neutral ratings of the restaurants of Insadong.

The RDF stream flows at an average rate of a hundred tweets/day and peaking at tens of tweets/minute. The RDF stream is processed in real-time by the SLD by means of the network of C-SPARQL queries where:

- a first continuous query counts the positive ratings for each POI in one day.
- a second query aggregates the result of the previous one over a week,
- a third query computes this aggregation over a month, and
- finally, a last query further aggregates the results of the queries upstream over one year.

The results of each of the four queries are published as linked data. The results of the first three queries are used in the PULL segment to answer the SPARQL queries for the Emerging recommendations and to display the trend lines illustrated in Figure 2(c). The results of the last query are used to compute the Popular recommendations in the PULL segment.

The last component of the PUSH segment is BOTTARI inductive stream reasoner SUNS. It daily takes the results of the first query and updates the inductive materialization used for the For me recommendations

6.2 The PULL segment

The PULL segment is based on the LarKC platform, which acts as an ontology-based information-integration platform: BOTTARI ontology logically integrates the data models of the different plug-ins involved in computing a given type of recommendations. Whenever a user presses one of the four recommendation buttons in BOTTARI interface, the client issues a query using the BOTTARI ontology. When the LarKC platform receives the query, it decomposes it into a set of queries, one for SLD, one for SUNS and one for the Geo-SPARQL engine. The queries are executed in parallel. Each plug-in receives its re-written query and sends its partial results to the a plug-in that joins the partial results and returns the complete answer to the client, as if the query had been evaluated on a single integrated knowledge base. Caching of entire queries and intermediate results is applied in order to minimize query latency.

More specifically, the Geo-SPARQL engine, given a location, a spatial orientation and a POI category, returns a list of POIs ordered by distance from the location. It delegates the query execution to SOR, the spatial-aware RDF store by Saltlux. SUNS, given the id of the user (e.g., Alice), returns a list of POIs ordered by the estimated probability that the user will like them. SLD, given a period (i.e., a day, a week, a month, or a year), returns a list of POIs ordered by the number of tweets that talk positively about the POI in that period. As explained above, the linked data published by the PUSH segment are used.

To better clarify how we configured the LarKC platform to evaluate the BOTTARI client requests, let us consider the query that represent a mix of the queries the client sends for *Interesting* (lines 3-6), *For me* (lines 7-8) and *Emerging* (line 10) recommendations.

```
1. SELECT ?poi ?name ?lat ?long
2. WHERE {
3. { ?poi a ns:NamedPlace ; ns:name ?name ;
4.        geo:lat ?lat ; geo:long ?long ;
5.        ns:category :InterestingForForeigners . }
6.   FILTER(:within_distance(37.5,126.9,?lat,?long,200))
7. { :Alice sioc:creator_of
      [twd:talksAboutPositively ?poi]}
8.    WITH PROBABILITY ?p ENSURE PROBABILITY [0.5..1)
9. {  ?poi twd:numberOfPositiveTweetsInTheMonth ?n }
10. }
11. ORDER BY
    DESC(?n*?p*:distance(37.5,126.9,?lat,?long,200))
12. LIMIT 10
```

LarKC platform extracts lines 7-8 from the above query and rewrites them in the SPARQL with probability query below and issues it to SUNS.

```
1. CONSTRUCT {:Alice twd:talksAboutPositively
2.             [ ns:about ?poi ; ns:withProbability ?p ] }
3. WHERE {
4.  { :Alice sioc:creator_of
      [ twd:talksAboutPositively ?poi ] }
5.  WITH PROBABILITY ?p } ENSURE PROBABILITY [0.5..1)
6. }
7. ORDER BY DESC(?p)
```

The peculiarity of the query are the WITH/ENSURE PROBABILITY clauses (Line 5) and the CONSTRUCT clause (Lines 1-2). The former are part of the query language exposed by SUNS: SPARQL with probability. They allow to bind the probability that the binding exist in the inductive materialized and to ensure that such a probability is in a given range. The latter allows representing the probability values returned by SUNS without using annotations or reification.

7 Evaluation

The quality and the efficacy of BOTTARI recommendations was comparatively evaluated using the data set described in Section 4.

The *For me*, *Popular* and *Interesting* recommendations were compared with two baselines: random guess (Random) and k-nearest neighbour (KNNItem). The combination of *For me* and *Popular* recommendations was also considered. For all recommendations, the distance filter was not applied, because our data set does not contain the user position at twitting time.

A key aspect of BOTTARI is the adoption of stream reasoning techniques that build on the hypothesis that a long enough window can capture all the information needed for a given task, while the rest can be forgotten. In a first set of experiments, we targeted the evaluation of the *Emerging* recommendations, which use a time window. A set of ground truths was created by withholding the newest rating for each user. Different time frames were considered: 1 day, 2 days, 7 days, 30 days, 90 days and 180 days. Table 2 shows the number of ratings in the different time frames.

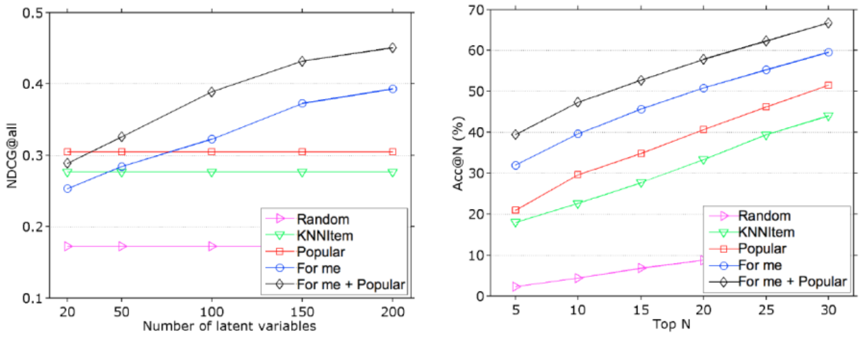
As a result, we discovered that the *Emerging* recommendations with a 90 days window are as effective as the *Popular* recommendations that keep the full history (i.e., two years of data).

In a round of experiments we evaluated also the inductive stream reasoning. We followed the standard method of splitting the data into a training set and a test set was used. In this case, a ground truth contains one positive rating for each user randomly withheld from the data set. We repeated this data split five times.

	Nr. of ratings	%
Last day	188	0.17
Last 2 days	703	0.64
Last 7 days	5,057	4.62
Last 30 days	27,049	24.73
Last 90 days	65,600	70.01
Last 180 days	93,696	85.65
Total	109,389	100.00

Table 2. Number of ratings with different time frames

We evaluated *For me* recommendations produced by SUNS with 20, 50, 100, 150 and 200 latent variables. As expected, Random was the worst. The *Emerging* recommendations on the last 90 days were slightly better than KNNItem. This might be due to the "bandwagon effect" that exists in many social communities. The *For me* recommendations significantly outperformed all the others after the number of the latent variables reached 100. The best ranking ever was produced by the combination of both *For me* and *Emerging* recommendations on the last 90 days. These results confirm the idea that a combined approach of deductive and inductive stream reasoning works best.



(a) (b)

Fig. 7. BOTTARI evaluation results

8 Conclusions

BOTTARI is a sophisticated application of semantic technologies that makes use of the rich and collective knowledge obtained by continuously analysing social media streams. We believe it was important to hide this complexity from the user using an intuitive and easy to use interface.

Inspired by the literature on ontology-based information access, we design BOTTARI ontology as driver of both data and service integration. It allows for combining real data sources at real scale, i.e. location-specific static information about hundreds of POIs with the results of continuous analysis of dynamic social media streams. However, we believe that the BOTTARI ontology was also crucial in handling the heterogeneous data models of the plug-ins. For instance, the inductive reasoner annotates triples in the inductive materialization with their probability to be true, but the other plug-ins cannot understand these annotations, unless they are transformed into commonly described data.

BOTTARI is engineered for scalability. Both SUNS and SLD show a scalability that goes largely beyond the actual needs of the BOTTARI deployment in Insadong. Training SUNS over two years of data takes 1.5 minutes. SLD can handle a flow of 15,000 tweets/second when the actual rate is tens of tweets/day. These results convinced Saltlux to start a large-scale deployment of BOTTARI in Korea.

Acknowledgments
This work was partially supported by the LarKC project (FP7-215535) and Mobile Cognition and Learning project in Korea.

References

1. D. F. Barbieri, D. Braga, S. Ceri, E. Della Valle, Y. Huang, V. Tresp, A. Rettinger, H.Wermser, Deductive and Inductive Stream Reasoning for Semantic Social Media Analytics, IEEE Intelligent Systems 25 (6) 32–41 (2010)
2. E. Della Valle, S. Ceri, F. van Harmelen, D. Fensel, It's a Streaming World! Reasoning upon Rapidly Changing Information, IEEE Intelligent Systems 24 (6) 83–89 (2009)
3. M. Garofalakis, J. Gehrke, R. Rastogi, Data Stream Management: Processing High-Speed Data Streams, Springer-Verlag New York, Inc., (2007)
4. D. Luckham, The power of events: An introduction to complex event processing in distributed enterprise systems, in: N. Bassiliades, G. Governatori, A. Paschke (Eds.), Rule Representation, Interchange and Reasoning on the Web, Vol. 5321 of Lecture Notes in Computer Science, Springer Berlin / Heidelberg, pp. 3–3. (2008)
5. D. F. Barbieri, D. Braga, S. Ceri, E. Della Valle, M. Grossniklaus, Querying rdf streams with c-sparql, SIGMOD Record 39 (1), 20–26 (2010)
6. M. Balduini, E. Della Valle, D. Dell'Aglio, M. Tsytsarau, T. Palpanas, C. Confalonieri: Social listening of City Scale Events using the Streaming Linked Data Framework, in: Proceedings of ISWC 2013 (2013)

7. V. Tresp, Y. Huang, M. Bundschus, A. Rettinger, Materializing and querying learned knowledge, in: Proc. of IRMLeS 2009, (2009)
8. Y. Huang, V. Tresp, M. Bundschus, A. Rettinger, H.-P. Kriegel, Multivariate prediction for learning on the semantic web, in: P. Frasconi, F. A. Lisi (Eds.), ILP, Vol. 6489 of Lecture Notes in Computer Science, Springer, pp. 92–104 (2010)
9. Y. Huang, M. Nickel, V. Tresp, H.-P. Kriegel, A scalable kernel approach to learning in semantic graphs with applications to linked data, in: Proc. of the 1st Workshop on Mining the Future Internet (2010)
10. V. Tresp, Y. Huang, X. Jiang, A. Rettinger, Graphical models for relations - modeling relational context, in: International Conference on Knowledge Discovery and Information Retrieval (2011)
11. D. Fensel, F. van Harmelen, B. Andersson, P. Brennan, H. Cunningham, E. Della Valle, F. Fischer, Z. Huang, A. Kiryakov, T. K. il Lee, L. School, V. Tresp, S.Wesner, M.Witbrock, N. Zhong, Towards LarKC: a Platform for Web-scale Reasoning, in: Proc. of ICSC 2008, (2008)
12. A. Cheptsov, et al., Large Knowledge Collider. A Service-oriented Platform for Large-scale Semantic Reasoning, in: Proceedings of WIMS 2011 (2011)
13. M. Lenzerini, Data integration: A theoretical perspective, in: L. Popa (Ed.), PODS, ACM, pp. 233–246 (2002)