Personalized Filtering of the Twitter Stream

Pavan Kapanipathi[1,2], Fabrizio Orlandi[1], Amit Sheth[2], and Alexandre Passant[1]

[1] Digital Enterprise Research Institute, Galway, Ireland
{fabrizio.orlandi, alexandre.passant}@deri.org
[2] Kno.e.sis Center, Dayton, OH - USA
{pavan, amit}@knoesis.org

Abstract. With the rapid growth in users on social networks, there is a corresponding increase in user-generated content, in turn resulting in information overload. On Twitter, for example, users tend to receive uninterested information due to their non-overlapping interests from the people whom they follow. In this paper we present a Semantic Web approach to filter public tweets matching interests from personalized user profiles. Our approach includes automatic generation of multi-domain and personalized user profiles, filtering Twitter stream based on the generated profiles and delivering them in real-time. Given that users interests and personalization needs change with time, we also discuss how our application can adapt with these changes.

Keywords: Semantic Web, Social Network, Twitter, PubSubHubbub, User Profiling, Personalization

1 Introduction

Online Social Networks have become a popular way to communicate and network in the recent times, well known ones include Facebook, MySpace, Twitter, Google+, etc. Twitter, in specific, has rapidly grown in the recent years, reaching 460,000 average number of new users per day in the month of March 2011. These numbers have in turn played a crucial role to increase the number of tweets from 65 million to 200 million[3] in the past year. This proves that the interested users are therefore facing the problem of information overload. Filtering uninteresting posts for users is a necessity and plays a crucial role [8] to handle the information overload problem on Twitter.

On Twitter it is necessary to *follow* another user in order to receive his/her tweets. The user who receives the tweets is called a *follower* and the user who generates the tweet is called a *followee*. However, they receive all the tweets from the users that are also not of their interests. Twitter by itself provides features such as keyword/hashtag search as a naïve solution for the information overload problem, but these filters are not sufficient to provide complete personalized information for a user. Although Twarql [6] improved the filtering mechanism

[3] `http://blog.twitter.com/2011/08/your-world-more-connected.html`

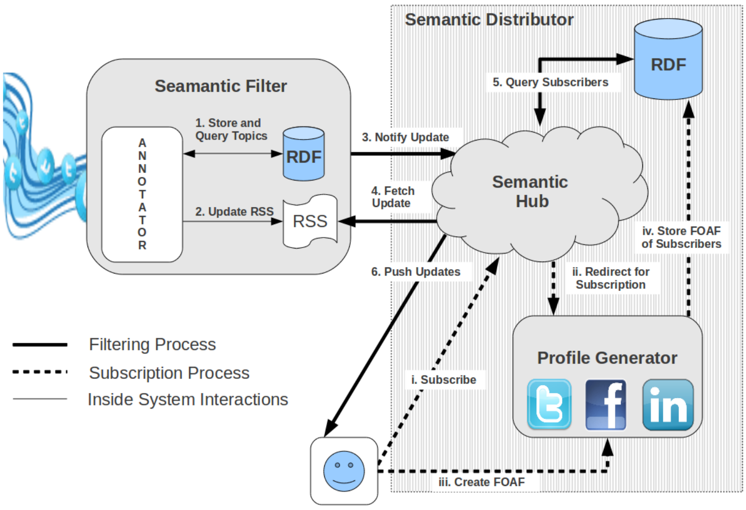

Fig. 1. System Architecture

for Twitter by leveraging Semantic Web technologies, the user still needs to track information by manual selection or formulation of SPARQL Query using Twarql's interface. So far applications such as TweetTopic [1] and “Post Post”[4] focus on filtering the stream of tweets generated from the people who are followed by the user. Instead of limiting the user experience only to his/her personal stream we propose a Semantic Web approach to deliver interesting tweets to the user from the entire public Twitter stream. This helps filtering tweets that the user is not interested in, which in turn reduces the information overload.

Our contributions include (1) automatic generation of user profiles (primarily interests) based on the user's activities on multiple social networks (Twitter, Facebook, Linkedin). This is achieved by retrieving users' interests, some implicit (analyzing user generated content) and some explicit (interests mentioned by the user in his/her SN profile). (2) Collecting tweets from the Twitter stream and mapping (annotating) each tweet to its corresponding topics from Linked Open Data. (3) Delivering the annotated tweets to users with appropriate interests in (near) real-time.

2 Architecture

Our architecture can be separated into three modules (1) Semantic Filter (**SF**) (2) Profile Generator (**PG**) (3) Semantic Hub (**SemHub**) as illustrated in Fig-

[4] http://postpo.st/

ure 1. In this section we first explain the interaction between the three modules, later each one is explained in detail.

In the above architecture two processes run in parallel (a) Filtering of tweets (b) Subscription to the System. The sequence for each process is represented by different types of arrows in Figure 1. The *Subscription to the system* is included in the Semantic Distributor. The Semantic Distributor (**SD**) comprises of both SH and PG. Once the user requests for the subscription (Seq. *i* in Figure 1) he/she is redirected to the PG (Seq. *ii*). PG generates the profiles based on the the user's activities on multiple social networks (Seq. *iii*). These profiles are stored in the SemHubs' RDF store (Seq. *iv*) using PuSH vocabulary [5]. On the other hand, *Filtering of tweets* is performed by annotating tweets from Twitter stream in SF. The annotations are further transformed to a representation of groups (SPARQL queries) of users who have interests corresponding to the tweet (Seq. *1*). These SPARQL Queries are termed as Semantic Groups (**SG**) in this paper. The tweet with its SG is updated as an RSS feed (Seq. *2*) and notified to SemHub (Seq. *3*). SemHub then fetches the updates (Seq. *4*) and retrieves the list of subscribers whose interests match the group representation of the tweet (Seq. *5*). Further the tweet is pushed to the filtered subscribers (Seq. *6*).

2.1 Semantic Filter

Semantic Filter (Figure 1), primarily performs two functions: (1) Representing tweets as RDF (2) Forming interested groups of users for the tweet.

First, information about the tweet is collected to represent the tweet in RDF. Twitter provides information of the tweet such as author, location, time, "reply-to", etc. via its streaming API. Including this, extraction of entities from the tweet content (content-dependent metadata) is performed using the same technique used in Twarql. The extraction technique is dictionary-based, which provides flexibility to use any dictionary for extraction. In our system the dictionary used to annotate the tweet is a set of concepts[6] from the Linked Open Data [2] (LOD)[7]. The same set is also used to create profiles, as described in the next Section 2.2. After the extraction of entities, the tweets are represented in RDF using lightweight vocabularies such as FOAF, SIOC, OPO and MOAT. This transforms the unstructured tweet to a structured representation using popular ontologies. The triples (RDF) of the tweet are temporarily stored in an RDF store.

The annotated entities represent the topic of the tweet. These topics act as the key in filtering the subset of users who receive the tweet. Topics are queried from the RDF store to be included in SGs that are created to act as the filter. The SG once executed at the Semantic Hub fetches all the users whose interests match to the topic of the tweet. If there are multiple topics for the tweet then the SG is created to fetch the union of users who are interested in at least one topic of the tweet.

[5] http://vocab.deri.ie/push

[6] Topic and concept are used interchangeably.

[7] http://richard.cyganiak.de/2007/10/lod/

2.2 User Profile Generator

The extraction and generation of user profiles from social networking websites is composed of two basic parts: (1) data extraction and (2) generation of application-dependent user profiles. After this phase other important steps for our work involve the representation of the user models using popular ontologies, and then, finally, the aggregation of the distributed profiles.

```
<foaf:topic_interest rdf:resource="http://dbpedia.org/resource/Semantic_Web" />
<wi:preference>
  <wi:WeightedInterest>
    <wi:topic rdf:resource="http://dbpedia.org/resource/Semantic_Web" />
    <rdfs:label>Semantic Web</rdfs:label>
    <wo:weight>
      <wo:Weight>
  <wo:weight_value rdf:datatype="http://www.w3.org/2001/XMLSchema#double">0.5</wo:
       weight_value>
  <wo:scale rdf:resource="http://example.org/01Scale" />
      </wo:Weight>
    </wo:weight>
    <opm:wasDerivedFrom rdf:resource="http://www.twitter.com/BadmotorF" />
    <opm:wasDerivedFrom rdf:resource="http://www.linkedin.com/in/fabriziorlandi" />
  </wi:WeightedInterest>
</wi:preference>
 [...]
<wo:Scale rdf:about="http://example.org/01Scale">
  <wo:max_weight rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal">1.0</wo:
       max_weight>
  <wo:min_weight rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal">0.0</wo:
       min_weight>
</wo:Scale>
```

Fig. 2. Representing an interest (*Semantic Web*) and its weight (*0.5*) found in two sources (Twitter and LinkedIn)

First, in order to collect private data about users on social websites it is necessary to have access granted to the data by the users. Then, once the authentication step is accomplished, the two most common ways to fetch the profile data is by using an API provided by the system or by parsing the Web pages. Once the data is retrieved the next step is the data modeling using standard ontologies. In this case, a possible way to model profile data is to generate RDF-based profiles described using the FOAF vocabulary [4]. We then extend FOAF with the SIOC ontology [3] to represent more precisely online accounts of the person on the Social Web. Additional personal information about users' affiliation, education, and job experiences can be modeled using the DOAC vocabulary[8]. This allows us to represent the past working experiences of the users and their cultural background. Another important part of a user profile is represented by the user's interests. In Figure 2 we display an example of an interest about "Semantic Web" with a weight of 0.5 on a specific scale (from 0 to 1) using the Weighted IntListingerests Vocabulary (WI)[9] and the Weighting Ontology (WO)[10]. In order to compute the weights for the interests common approaches are based on the number of occurrences of the entities, their frequency, etc.

[8] DOAC Specification: http://ramonantonio.net/doac/0.1/

[9] WI Specification: http://purl.org/ontology/wi/core#

[10] WO Specification: http://purl.org/ontology/wo/core#

Finally, the phase that follows the modeling of the FOAF-based user profiles and the computation of the weights for the interests is the aggregation of the distributed user profiles. When merging user profiles it is necessary to avoid duplicate statements (and this is done automatically by a triplestore during the insertion of the statements). Furthermore, as in the case of the interests, if the same interest is present on two different profiles it is necessary to: represent the interest only once, recalculate its weight, and update the provenance of the interest keeping track of the source where the interest was derived from. As regards the provenance of the interest, as showed in Figure 2, we use the property `wasDerivedFrom` from the Open Provenance Model[11] (OPM) to state that the interest was originated by a specific website.

As regards the computation of the aggregated global weight for the interest generated by multiple sources, we propose a simple generic formula that can be adopted for merging the interest values of many different sources. The formula is as follows:

$$G_i = \sum_s w_s * w_i \tag{1}$$

Where: G_i is the global weight for interest i; w_s is the weight associated to the source s; w_i is the weight for the interest i in source s.

2.3 Semantic Hub

The Semantic Distributor module comprises of Semantic Hub [5] and Profile Generator. Semantic Hub (SemHub) is an extension of Google's PubSubHubbub (PuSH) using Semantic Web technologies to provide publisher-controlled real-time notifications. PuSH is a decentralized publish-subscribe protocol which extends Atom and RSS to enable real-time streams. It allows parties understanding it to get near-instant notifications of the content they are subscribed to, as PuSH immediately *pushes* new data from publisher to subscriber(s) where traditional RSS readers periodically *pull* new data. The PuSH ecosystem consists of a few hubs, many publishers, and a large number of subscribers. Hubs enable (1) publishers to offload the task of broadcasting new data to subscribers; and (2) subscribers to avoid constantly polling for new data, as the hub pushes the data updates to the subscribers. In addition, the PuSH protocol is designed to handle all the complexity in the communication easing the tasks of publishers and subscribers.

The extension from PuSH protocol to Semantic Hub is described in [5]. In our work, SemHub performs the functionality of distributing the tweets to its interested users corresponding to the Semantic Groups generated by SF. The SemHub will have only one publisher as shown in Figure 1 which is the SF, and there can be multiple subscribers. SemHub, as in our previous work, does not focus on creating a social graph of the publisher, the PG is responsible to store the subscribers's FOAF profile in the RDF store accesssed by the SemHub.

[11] OPM Specification: `http://openprovenance.org/`

3 Implementation

In this section we provide the implementation details for each module in the architecture. Firstly to collect tweets we use the *twitter4j Streaming API* [12]. Starting with SF, the entity extraction of tweets is dictionary-based similar to the extraction technique used in *Twarql* [7]. This technique is opted due to performance requirements for real-time notifications. A set of 3.5 million entities[13] from DBpedia is built as an in-memory representation for time-efficient and longest sub-string matching. The in-memory representation is known as ternary interval search tree (Trie) and the longest sub-string match using trie is performed at time complexity of *O(LT)* where L is the number of characters and T is the number of tokens in the tweet.

```
<http://twitter.com/rob/statuses/123456789>
  rdf:type   sioct:MicroblogPost ;
  sioc:content   What is the over/under on the Kim Kardashian / Kris Humphries
      Hollywood wedding lasting more than 5 years? #fb
  sioc:has_creator   <http://twitter.com/rob> ;
  foaf:maker   <http://example.org/rob> ;
  moat:taggedWith   dbpedia:Kim_Kardashian ;
  moat:taggedWith   dbpedia:Kris_Humphries ;
  moat:taggedWith   dbpedia:Hollywood .

<http://twitter.com/rob/statuses/123456789#presence>
  rdf:type   opo:OnlinePresence ;
  opo:startTime     2010-03-20T17:55:42+00:00      ;
  opo:customMessage <http://twitter.com/rob/statuses/123456789> .

<http://twitter.com/rob> geonames:locatedIn Dbpedia:Ohio .
 [...]
```

Fig. 3. Representing Tweet in RDF

As mentioned in section 2.1, tweets are transformed into RDF using some lightweight vocabularies, see Figure 3 for an example. The RDF is then stored in an RDF store using SPARQL Update via HTTP. For performance issues it is preferable to have the RDF Store on the same server. However, architecturally it can be located anywhere on the Web and accessed via HTTP and the SPARQL Protocol for RDF. Presently, this RDF generated for each tweet is stored in a temporary graph and topics/concepts of the tweet are queried. These concepts are then used to formulate the SPARQL representation of the group (SG) of users who are interested in the tweet. The RSS is updated as per the format specified in [5] with the SG and the Semantic Hub is notified. The SG for the tweet in Figure 3 will retrieve all the users who are interested in at least one of the extracted interests (*dbpedia:Kim_Kardashian, dbpedia:Kris_Humphries, dbpedia:Hollywood*).

The Semantic Hub used for our implementation is hosted at `http://semantichub.appspot.com`. The SemHub executes the SG on the graph

[12] `http://stream.twitter.com`

[13] `http://wiki.dbpedia.org/About` (July 2011)

that contains the FOAF profiles of subscribers generated by PG. The corresponding tweets are *pushed* to the resulting users.

Profile Generator considers three different social networking sites: Twitter, LinkedIn and Facebook for generating user profiles. In order to collect user data from each of those platforms, we developed three different types of applications. For Twitter and Facebook we implemented similar PHP scripts that makes use of the respective query API publicly accessible on the Web. For LinkedIn we use a XSLT script that parses the LinkedIn user profile page and generates an XML file containing all the attributes found on the page. The user information collected from Twitter is the publicly available data posted by the user, *i.e.* his/her latest 500 microblog posts. The technique used for entity recognition in the tweets of the user is the same one used in SF for annotating the tweets. The extracted concepts are then ranked and weighted using their frequency of occurrences. A similar approach is described in [9].

While on Twitter we create profiles with implicitly inferred interests, on LinkedIn and Facebook we collect not only interests that have been explicitly stated by the users, but also their personal details such as contacts, workplace and education. The user personal data is fetched through the Facebook Graph API as well as the interests (*likes*) that are then mapped to the related Facebook pages representing the entities. We represent the entities/concepts on which the user is interested in using both DBpedia and Facebook resources.

The weights for the interests are calculated in two different ways depending on whether or not the interest has been *implicitly* inferred by the entity extraction algorithm (the Twitter case) or *explicitly* recorded by the user (the LinkedIn and Facebook cases). In the first case, the weight of the interest is calculated dividing the number of occurrences of the entity in the latest 500 tweets by the total number of entities identified in the same 500 tweets. In the second case, since the interest has been manually set by the user, we assume that the weight for that source (or social networking site) is 1 (on a scale from 0 to 1). So we give the maximum possible value to the interest if it has been explicitly set by the user.

Our approach as regards the computation of the new weights as a result of the aggregation of the profiles is straightforward. We consider every social website equally in terms of relevance, hence we multiply each of the three weights by a constant of 1/3 (approximately 0.33) and then we sum the results. According to the previously described formula (1) in this case we use the following values: $w_s = 1/3 . \forall s$.

4 Conclusion and Future Work

In this paper we described an architecture for filtering the public Twitter stream and delivering the interesting tweets directly to the users according to their multi-domain user profile of interests. We explained how we generate comprehensive user profiles of interests by fetching and aggregating user information from different sources (*i.e.* Twitter, Facebook and LinkedIn). Then, we detailed how we extract entities and interests from tweets, how we model them using Semantic Web technologies, and how is possible to automatically create dynamic groups of users related to the extracted interests. According to the user groups the tweets are then "pushed" to the users using the Semantic Hub architecture.

In future, we want to extend our work to handle social streams in general (not only Twitter). Also, leveraging inferencing (category - subcategory relationships) on LOD, rather than just filtering based on concepts. Our extention would also include users not only subscribe to concepts from LOD as interests but also subscribe to a SPARQL Query as in Twarql. We are also working on providing interesting information and ranking based on the user's social graph.

5 Acknowledgements

This work is funded by (1) Science Foundation Ireland under grant number SFI/08/CE/I1380 (Líon 2) and by an IRCSET scholarship supported by Cisco Systems (2) Social Media Enhanced Organizational Sensemaking in Emergency Response, National Science Foundation under award IIS-1111182, 09/01/2011 - 08/31/2014.

References

1. M.S. Bernstein, Bongwon Suh, Lichan Hong, Jilin Chen, Sanjay Kairam, and E.H. Chi. Eddi: interactive topic-based browsing of social status streams. In *The 23rd annual ACM symposium on User interface software and technology*, 2010.
2. Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked data - the story so far. *Int. J. Semantic Web Inf. Syst.*, 5(3):1–22, 2009.
3. John Breslin, Uldis Bojars, Alexandre Passant, Sergio Fernàndez, and Stefan Decker. SIOC: Content Exchange and Semantic Interoperability Between Social Networks. In *W3C Workshop on the Future of Social Networking*, January 2009.
4. Dan Brickley and Libby Miller. FOAF Vocabulary Specification 0.98. Namespace Document 9 August 2010 - Marco Polo Edition. `http://xmlns.com/foaf/spec/`, 2010.
5. Pavan Kapanipathi, Julia Anaya, Amit Sheth, Brett Slatkin, and Alexandre Passant. Privacy-Aware and Scalable Content Dissemination in Distributed Social Networks. In *ISWC 2011 - Semantic Web In Use*, 2011.
6. Pablo N. Mendes, Alexandre Passant, and Pavan Kapanipathi. Twarql: tapping into the wisdom of the crowd. I-SEMANTICS '10, 2010.
7. Pablo N. Mendes, Alexandre Passant, Pavan Kapanipathi, and Amit P. Sheth. Linked Open Social Signals. In *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, 2010.
8. D. Ramage, S. Dumais, and D. Liebling. Characterizing microblogs with topic models. In *ICWSM*, 2010.
9. Ke Tao, Fabian Abel, Qi Gao, and G.J. Houben. TUMS: Twitter-based User Modeling Service. In *Workshop on User Profile Data on the Social Semantic Web (UWeb), ESWC 2011*, 2011.