Linked Open Social Signals

Pablo N. Mendes*, Alexandre Passant†, Pavan Kapanipathi* and Amit P. Sheth*
*Kno.e.sis Center, CSE Department
Wright State University, Dayton, OH - USA
Email: firstname@knoesis.org

†Digital Enterprise Research Institute
National University of Ireland, Galway
Email: firstname.lastname@deri.org

***Abstract*—In this paper we discuss the collection, semantic annotation and analysis of real-time social signals from microblogging data. We focus on users interested in analyzing social signals collectively for sensemaking. Our proposal enables flexibility in selecting subsets for analysis, alleviating information overload. We define an architecture that is based on state-of-the-art Semantic Web technologies and a distributed publish-subscribe protocol for real time communication. In addition, we discuss our method and application in a scenario related to the health care reform in the United States.**

***Keywords*-Semantic Web, Real-time systems, Microblogging, Linked Data**

I. Introduction

Microblogging services such as Twitter.com and Friendfeed.com are Web platforms for exchanging short updates, generally of 140 characters or fewer — called "tweets" in the case of Twitter. Microblogs have emerged as a medium where people are increasingly "playing an active role in the process of collecting, reporting, analyzing and disseminating news and information."[1] Microbloggers can be seen as **citizen sensors** from which **social signals** can be collected and analyzed for the understanding of events [10], gathering of opinions and many more. One of the most visible uses of citizen-sensors occurred during the Mumbai terrorist attacks in November 2008, when tweets and Flickr feeds by citizens armed with mobile phones reported observations of events in real time, often well before traditional media reports could do so[2].

However, analyzing these numerous social signals can be extremely challenging. Microblog posts are streamed in large quantities every second, in textual format, creating significant **information overload** for the user interested on making sense of the information around a topic of interest. As an attempt to alleviate the problem, Twitter users adopted hashtags[3] for tracking topics such as live events, communities, or breaking news. However, hashtags have several limitations such as their ambiguity (*#apple*) and heterogeneity (*#realtime*, *#rt*), as well as their lack of organization [6] [11] . In addition, they have to be explicitly included by the creator of the microblog post. Some users may forget or choose not to include them e.g. due to post length constraints.

In order to make more sense of such data, there is a need for more robust ways to (i) aggregate, (ii) organize and (iii) collectively analyze the wealth of social signals. Several advancements in terms of the Linked Open Data project [4] and Semantic Web technologies have been made to facilitate global access, interconnecting and organizing information. Particularly we can mention the use of common representation languages, domain models (ontologies), sharing of knowledge bases on the Web and semantic annotation of all types of data to extract spatial, temporal, and thematic metadata [14] [10]. Through the use of a standard representation framework (e.g. RDF(S)/OWL) and associated query languages (e.g. SPARQL), it is possible to flexibly select and meaningfully analyze subsets of data from heterogeneous sources in order to alleviate the information overload problem.

So far, such selection and analysis capabilities have not been explored to their full extent in the context of streaming social data. Common data acquisition approaches are based on *pull* models (e.g. requesting RSS feeds from Twitter at regular intervals). While a pull-style communication is suitable for transient needs (navigation searches, research searches), the *push* model is particularly useful in cases where real time situational awareness plays a key role. In microblogs, there is an ubiquitous feed of social signals around the globe that may or may not find its way to an interested user. Imagine, for example, a person interested in following the coverage on the Haiti earthquake, or the discussions around the Iran elections. How can a system track a stream of social signals and facilitate the collective analysis of social signals pertinent to those topics? What are the interesting facts? How does the public perception vary between regions? In order to address these questions, there is

[1] http://en.wikipedia.org/wiki/Citizenjournalism

[2] http://www.forbes.com/2008/11/28/mumbai-twitter-sms-tech-internet-cx_bc_kn_1128mumbai.html

[3] Hashtags are short character strings that start with the character '#' and are inserted in a tweet by users as an attempt to explicitly associate it with a topic.

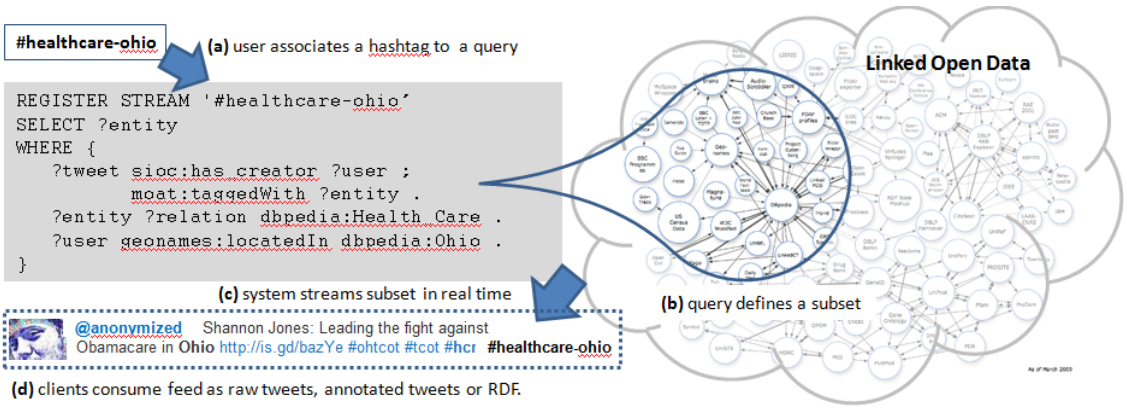

Figure 1. Concept feeds are named by a hashtag and define a subset of tweets through a query.

an opportunity to combine the advantages of real-time push models for information delivery and Semantic Web querying and analysis of microblog content.

In this work we describe an approach for bringing social signals to Linked Open Data. We propose a **Linked Open Social Signals** architecture that gathers, annotates, filters and delivers to interested parties the social signals that are relevant to a given topic. As the word "sensor" suggests, it represents an instrument that receives stimuli (social signals) and responds with an action (collection, filtering, syndication). Our approach complements traditional Web search systems by offering a an alternative request/delivery method more suitable for real time sensemaking. Although our architecture naturally supports regular queries, it offers an alternative *push-style persistent search*, where the system stores a user-provided topic definition and proactively responds with a *stream of real time updates* relevant to that topic.

We summarize our main contributions as follows:

Semantic Annotation: We describe and implement a method for acquisition, processing and annotation of social signals (Twitter messages in our case) in real time. We use a set of standard RDF(S)/OWL formats for these annotations that enable easy reuse across Semantic Web based applications (Section IV). Our system supports the SPARQL Protocol and the Linked Open Data principles, allowing our social signals to be accessed, browsed, and queried through W3C standard technologies.

Concept Feeds: We implement an aggregation service of social signals that helps users to cope with information overload. Through our approach, users can subscribe to more flexible "concepts" (Figure I) instead of only hashtags or users. We adapt the definition of concept from Tom Mitchell [8]. Each such concept can be viewed as describing some subset of objects or events defined over a larger set (e.g. subset of tweets that mention *Obama*'s birth place), or alternatively, each concept can be thought of as a boolean-valued function defined over this larger set (e.g., a function defined over all tweets, whose value is true for tweets containing *Obama*'s birth place and false for all others). In our work, concepts are defined by SPARQL queries through user-friendly interfaces that do not require any knowledge of query language from the users. The system then publishes the evolving relevant content as an annotated web feed, proactively delivered through a push model, providing a simple interface for users to follow topic-relevant information in real-time (Section V).

Software Architecture: In order to provide a scalable and efficient realization of the aforementioned principles, we provide a loosely coupled architecture for distributing concept feeds of semantically annotated tweets. The implementation relies on existing standards and protocols, is entirely HTTP-based and can be easily deployed in any environment. This allows interested parties to deploy their own system without having to rely on a centralized authority to index or distribute their feeds.

The remainder of the paper discusses an illustrative scenario (Section II), our main contributions (Sections III, IV, V), related work (Section VII) and concludes the discussion with final comments and future work (Section VIII).

II. Scenario

In order to illustrate our approach in this work, we focus on the Health Care Reform in the United States.

The health care reform topic has created much controversy due to its reach into the lives of patients, doctors, politicians, and health insurance companies. Mainstream news agencies have aggregated information about the health care reform[4] aiming at providing comprehensive coverage of all aspects of the discussion. We propose a complementary approach to such services that automatically aggregates information from relevant microblog posts generated by regular web users (microbloggers).

[4]http://voices.washingtonpost.com/health-care-reform/

In fact, this proposal is very much aligned with discussions in mainstream media companies. The director of BBC Global News Peter Horrocks says that "for BBC news editors, Twitter and RSS readers are to become essential tools." In an interview to The Guardian in February 2010, he says: "Aggregating and curating content with attribution should become part of a BBC journalist's assignment; and BBC's journalists have to integrate and listen to feedback for a better understanding of how the audience is relating to the BBC brand." [7]

We consider the use case of a journalist, Joe, interested in selecting one of many topics for publication. He wants something that is stirring up discussion today, but was not popular yesterday. Through the use of our concept feeds, Joe can just use a Web interface to formulate a query that instructs the system to "select entities from microposts that mention any topic related to health care in Ohio on the relevant dates." Performing this task on Twitter would mean searching for keywords such as "health care", hashtags such as *#hcr* and *#health-care*, copying and pasting selected microposts, reading one by one and comparing them to last night's.

This use case illustrates the need for aggregation services that alleviate the information overload and support real time analytical interfaces for the aggregation of social perceptions. Moreover, users like Joe have different needs and see the same data from different perspectives. Flexibility in "slicing" the data is an important requirement: there is a need to enhance **raw data** so that it can be consumed from different perspectives thanks to the various **dimensions** that have been extracted: topic, location, temporal aspects, etc.

III. Architecture

The driving engineering requirements in our system are: scalability and (near) real time delivery of semantically annotated information. In order to address those requirements, our architecture separates concerns, and includes decoupled implementations for collection, processing, persistence, subscription and delivery components. The coarser components of our architecture are: (i) Social Sensor Server, (ii) Semantic Publisher, (iii) Distribution Hub and (iv) Application Server.

The **sequence of interactions** in our approach is conducted as follows. From the client side, users only need regular Web browsers in order to use our service. Query formulation, subscription requests, data visualization and analytical interfaces run on the client side (e.g. JavaScript-enabled Web browser) and communicate with the Web through the Application Server, all communications being done through HTTP. Upon the user request for a query, the Application Server relays the request to the Semantic Publisher, that passes the results collected from the Social Sensor Server onto Distribution Hubs for delivery.

Within the **Social Sensor Server** component, several modules realize the collection and data processing. Our collection module uses the Twitter Streaming API[5]. The Twitter Streaming API allows near-realtime access to various subsets of Twitter public statuses. The collector opens an HTTP connection to Twitter and receives updates as they are made available by the server. As each micropost arrives, it is sent to the processing module and the next micropost is parsed. The processing module then releases the collector and starts the **Extraction Pipeline**. The pipeline performs a series of extraction methods for annotating microposts with entities, hashtags and URLs that are mentioned in those microposts. More details on the information extraction approach and micropost annotation are given on Section IV.

Once the information is transformed to RDF, it is sent to a **Semantic Publisher** using SPARQL Update [13] via HTTP. Although it is desirable, for performance issues, to have the Semantic Publisher on the same server, architecturally it can be located anywhere on the Web and accessed via an abstraction layer achieved via HTTP and the SPARQL Protocol for RDF. In this implementation, our Semantic Publisher uses the open-source edition of OpenLink Virtuoso[6] for RDF storage. Storing incoming microposts is necessary for use cases requiring queries on past data. The Semantic Publisher can be configured with storage policies that determine how long microposts will be stored for.

We support three types of distribution, as described in Section V. The first two implementations have the objective of supporting archive queries: selecting past microposts that match a certain criteria. The Twitter Search-like API provides backwards compatibility with regular Twitter clients, and the SPARQL protocol supports Semantic Web queries. The third type of distribution is a stream provider implementing a PubSubHubBub (PuSH) Publisher for supporting real-time updates. This implementation delivers social signals to clients through **Distribution Hubs**. The job of the Distribution Hubs is to provide a layer of separation to relieve the Semantic Publisher from the heavy load of clients polling for more results to their queries. Since our publisher implements the PuSH protocol, we are able to simply reuse any existing PuSH hub on the Web, including the public hub provided by Google (http://pubsubhubbub.appspot.com/).

The decoupled, distributed character of this architecture helps to distribute the load between servers, while its topical nature can help to reduce the information overload problem for users.

IV. Semantic Annotation of Social Signals

In order to provide users with more powerful tools for understanding the topics and events being discussed in microblog posts, we connect the real-time collection of signals to a processing pipeline for the extraction of content descriptors and subsequent transformation into structured information.

[5]http://api.twitter.com

[6]http://virtuoso.openlinksw.com/

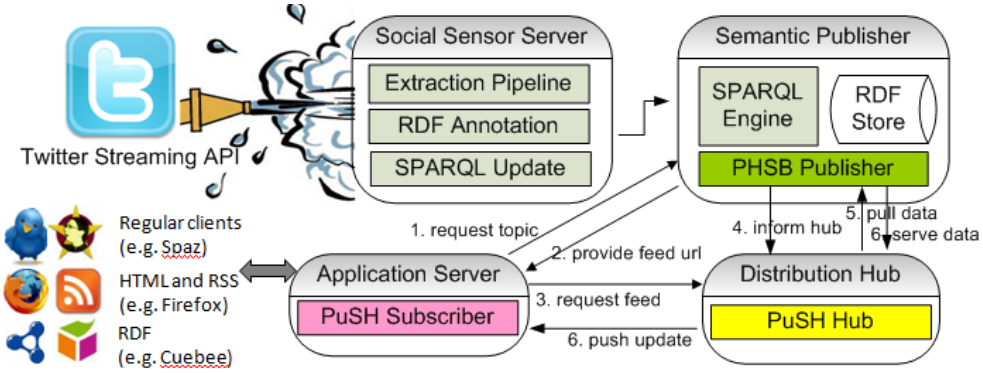

Figure 2. The Architecture of a Topical Social Sensor in interaction with Hubs and Clients.

A. *Extracting semantic descriptors*

Besides the limitations presented in Section I, hashtags are important topical descriptors of microposts, since they indicate the intention of the author to categorize a given micropost under a given topic. For that reason, our information extraction pipeline includes hashtag extractors. Hashtag extraction can be done through simple regular expressions. With the objective of maximizing the information conveyed by the hashtags, we also perform what we call "hashtag resolution". We take advantage of socially maintained hashtag glossaries to obtain a definition for each hashtag extracted. Through those services, Twitter users create new hashtags, propose descriptions and vote on the descriptions they judge as the best for a given tag. Some of the services available include Tagdef.com, Tagal.us and WTHashTag.com. At the moment our pipeline uses Tagdef.com for our hashtag resolution. For each extracted hashtag we perform a lookup through the Tagdef.com API and pick the highest voted description for that hashtag. The description is then appended to the tweet text as an extension of its content. As an example, consider the tags *#hcr* and *#killthebill*. Although there is no string overlap between the two hashtags, both definitions contain the phrase 'Health Care Reform'. Thus, adding hashtag definitions to microposts can bring important information for describing micropost content.

While hashtags aim at representing categories that coarsely describe microblog posts, the textual content of each micropost carries good hints of its underlying topic. Entities such as *Obama*, *Senate* and *Health Care Bill* are mentioned within the text in microposts and represent finer grained semantic units that can be extracted. The task of Named Entity Recognition has been studied in casual text [5] and in more general form following both unsupervised and supervised machine learning approaches [9]. The best performing systems achieve up to 90.8 F_1 score [12] through supervised approaches, i.e. require hand labeling of examples, an infeasible approach for Web scale entities from LOD. However, for such focused tasks, it has been reported that a simple dictionary lookup of the entities which appeared in the training data achieves 71.91 F_1 score on the test set [15]. Due to our strong performance requirements, as well as the open nature and scale of the LOD, we focus on the more simplistic dictionary-based entity extraction approach. We obtain a list of known 'surface forms' (entity labels) from a wide coverage Linking Open Data (LOD) subset and build an in-memory representation optimized for string matching. In our current implementation we use a set of about 2M entities from DBpedia, an open database of user-generated data in RDF extracted from Wikipedia [1]. However, our framework is flexible enough to be adapted to other datasets from the LOD cloud, e.g. GeoNames for geolocation purposes. We load the entity set as a trie (prefix tree) in memory and perform longest common substring match at time complexity $O(LT)$ where L is the number of characters and T is the number of tokens in the sentence provided as input.

At the start of the information extraction process, a micropost contains its original text, author, time and geography information. After the extraction is completed, the micropost also contains a collection of entities, hashtags, hashtag definitions and URLs that help to expand the description of its content. The next step is to encode the original information of the micropost together with the extracted information in a common representation format for delivery to users.

B. *Semantic annotation of microblog posts*

Semantic annotation transforms unstructured data into a structured representation that enables applications to better search, analyze, and aggregate information. We use common RDF(S)/OWL data formats for this modeling in order to provide easy reuse across Semantic Web based applications, notably by using SPARQL for querying.

Particularly, we rely on the following models for micropost annotations: (1) **FOAF** (foaf-project.org) — Friend of a Friend — is used to represent users, as it provides a simple way to describe people, their main attributes and their social

acquaintances; (2) **SIOC** (sioc-project.org) — Semantically-Interlinked Online Communities — and its Types module are used to model microblog updates themselves, as it is now a standard vocabulary for expressing social data in RDF; (3) **OPO** (online-presence.net) — Online Presence Ontology — for describing a user's presence as well as their context that can give better insight into their current situation, such as the current geographical location; (4) **MOAT** (moat-project.org) — Meaning Of A Tag [11] — to model semantic tagging capabilities, i.e. linking tagged microposts to meaningful resources on the Linked Open Data Cloud.

The combination of these ontologies form a complete stack to represent various elements involved in microblogging applications. As an example of these ontologies in-use, the following snippet of code exemplifies a post about the Health Care reform in the US, written by a user based in Dayton, as mentioned in our journalist use-case[7].

```
<http://twitter.com/bob/statuses/123456789>
  rdf:type sioct:MicroblogPost ;
  sioc:content
   "Fingers crossed for the upcoming #hcr vote"
  sioc:has_creator <http://twitter.com/bob> ;
  foaf:maker <http://example.org/bob> ;
  moat:taggedWith dbpedia:Healthcare_reform .

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

<http://twitter.com/bob> geonames:locatedIn
    dbpedia:Ohio .
```

Figure 3. Example of a Twitter messages represented in RDF

We used known entities from LOD sources during the extraction of meaningful units of description for microposts. This choice has important ramifications on the use of those annotated microposts. Background knowledge changes the way you can look at the information, because it puts the information in context. This is especially important for tweets because they are short, and therefore individually lack volume of information that provides an informative context. The use of annotated microposts together with background knowledge obtained from Linked Open Data will offer important capabilities: (1) We benefit from information in these knowledge bases to enable different granularity levels in information retrieval. For example, thanks to the links provided (as RDF) between Dayton and Ohio in GeoNames, all the information mapped only to the city of Dayton could be retrieved in queries regarding Ohio or USA. Similar effects can be observed on the thematic dimension (Figure 4) (2) Links to the LOD allow discovery of microposts from the entity annotations. For instance, through search engines such as Sindice, users can search for *Dayton* and see all links to that entity, including all microblog posts mentioning Dayton. (3) Users can benefit from the annotations when 'slicing' the data to specify more sophisticated constraints such as the Concept Feeds, as described on Section V.

[7]Due to space limitations we have omitted the definition of prefixes.

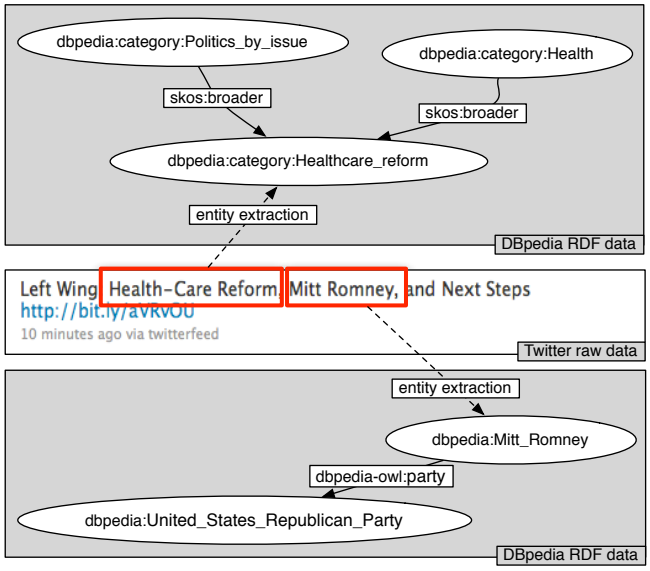

Figure 4. Benefits of interlinking to LOD for content discovery.

V. Delivering Annotated Social Signals

In this section we describe in more detail how we connect users to the stream of social signals being collected by our Linked Open Social Signals architecture.

A. Subscribing to Concept Feeds

When Twitter users subscribe to a hashtag, only tweets that were explicitly tagged with that hashtag will be delivered to the user. We are able to provide more sophisticated subscription mechanisms since our Social Sensor extracts semantic information beyond hashtags. We propose Concept Feeds as substitutes for hashtag-based topic subscription. Through the use of SPARQL, users can specify RDF graph patterns as constraints to select a subset of data that matches their need.

In our implementation, no technical knowledge is required for creating SPARQL queries. Users can interact with applications that guide the query formulation through interface components as if they were filling a Web form. We use Cuebee (cuebee.sf.net), a knowledge-driven query formulation tool. Cuebee uses information about classes and relationships from (ontology) schemata to guide the user in creating a query.

Figure V-A shows the SPARQL representation of the query from our scenario.

```
SELECT ?entity
WHERE {
  ?tweet sioc:has_creator ?user ;
    moat:taggedWith ?entity .
  ?entity ?relation dbpedia:Health_Care .
  ?user geonames:locatedIn dbpedia:Ohio .
}
```

Figure 5. SPARQL representation of a question from our scenario.

The use of SPARQL queries for defining the users' information need provides for better control over the ability to select subsets of data. In the example of Joe's question, it can be noted that he was able to narrow data thematically (health care) and geographically (Ohio), while expanding his thematic reach by including other entities explicitly related to Health Care.

Upon receiving a query, our system creates a stream that users can subscribe to. Users can optionally name Concept Feeds with a hashtag. Providing names to Concept Feeds is an easy way to enable backward compatibility with Twitter clients. By subscribing through our architecture to a hashtag that is associated with a Concept Feed, a user will trigger a subscription to the query represented by that hashtag. As a result, before delivering the tweets our implementation will append the hashtag to the tweet text, allowing for transparent consumption by existing Twitter clients.

B. Distributing updates to simple Twitter clients

Hashtags are an established mechanism among Twitter users for following topics. Widespread support is provided by tools like Spaz (code.google.com/p/spaz) and others. Through the use of the same signatures and result formats as the Twitter API, our Semantic Publisher provides backward compatibility with regular Twitter clients. Any Twitter client can subscribe to a hashtag that identifies a given Concept Feed and receive updates matching that query from our Topical Social Sensor. However, in comparison to the use of regular hashtags, the queries being executed in the background provide much higher expressiveness in filtering out uninteresting signals. In comparison to regular hashtags, another feature of our concept feeds is that the context changes dynamically. Recall the journalist use case presented in Section II. If a new relationship links the Health Care Reform to a newly approved Health Care Law, Joe will automatically get updated with tweets mentioning that law. Even if the tweet that mentions the law did not include a *#healthcare* hashtag.

C. Distributing updates to Semantic Web clients

Since the Semantic Publisher also includes a SPARQL Protocol compliant service, any standard Semantic Web client can send SPARQL queries through HTTP and obtain RDF data in response. In addition, we follow the Linked Open Data principles and expose our data through *dereferenceable URIs*. In other words, client applications may send an HTTP request to the URI that identifies a tweet within our server, and an RDF description of that tweet will be provided, using the ontology stack that we previously detailed. Since the information extraction pipeline uses a list of known LOD entities for annotating tweets, the RDF description for those tweets also includes links to other LOD datasets on the Web.

D. Pushing Signals: Enabling Real-time Notification

In order to enable real-time notification of social signals, we enable a *push* approach, where information is delivered to the user as soon as it becomes available. Our approach relies on the pubsubhubbub (PuSH) protocol combined with RSS-enabled SPARQL query results and dynamic triggers in RDF stores, as follows: (1) When a user defines a concept using a SPARQL query, our system creates an RSS feed with the expected information, and stores the mapping between the SPARQL query and the RSS feed; (2) This feed contains a link to a PuSH hub. Based on the PuSH specification, the client then automatically subscribes to this hub. (3) Each time new data is added to the RDF store, the queries known by the system (corresponding to the RSS feeds) are run. If the data matches a query, the system notifies the PuSH hub of the update of these feeds. The hub, in turn, broadcasts the information to corresponding subscribers.

VI. Discussion and Applications

We start our discussion by assessing our ability to accurately capture social signals from microblog posts through the semantic annotation with entities from Linked Open Data. Further we briefly discuss our processing time performance for real time delivery, and demonstrate applications that empirically support the value of the system.

Capturing Social Signals. In order to obtain a gold standard of event descriptors, we collected **1,242** article abstracts from the New York Times from August 1st to October 10th, 2009. We collected **755,294** tweets on the health care topic using the crawling technique described by Nagarajan et al. [10]. We performed entity extraction on both datasets using the techniques described in this work. Figure VI shows a series of entity frequencies per date. Each curve represents an entity **(obama, senate)**. Each point on the X axis represents a date, starting from 1=**2009-08-01** until 73=**2009-10-30**. On the Y axis you see the frequency of occurrence of the given entity on the corresponding date. Although some shifts and mismatches can be naturally expected, the overall trend seems to be consistent. Peeks of entities obtained from the tweets seem to correspond to peeks in the gold standard. That initial assessment provides encouragement for more in depth evaluations that will be performed in future work.

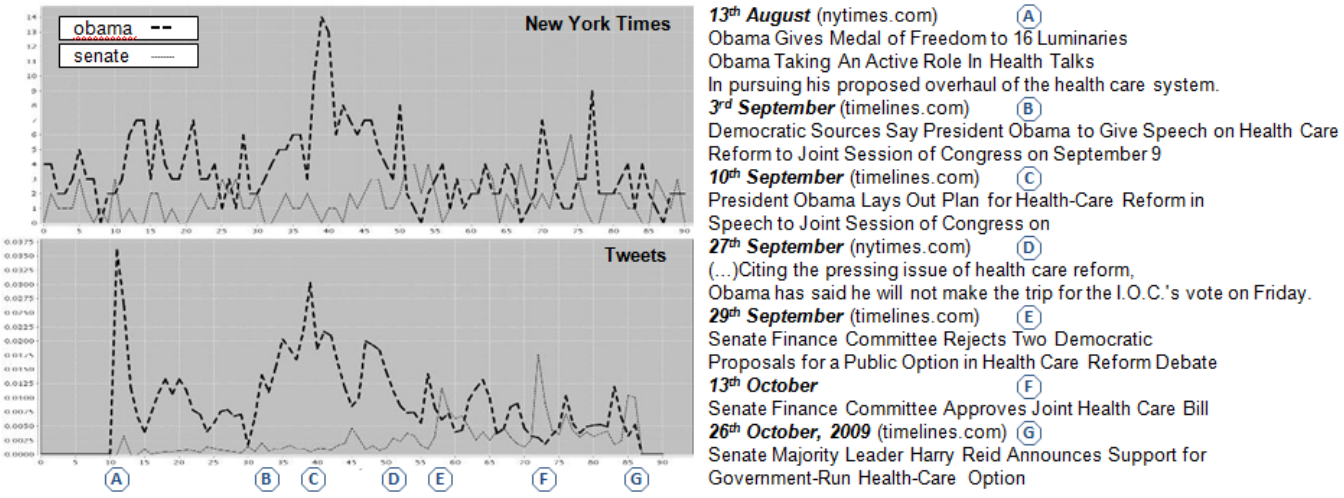

Figure 6. Entities frequently occurring in the New York Times (top) and in tweets (bottom) for the health care dataset. The X axis represents dates and the Y axis represents term frequency.

Time Performance. Twitter does not provide an official estimate for the outgoing rate of tweets per minute. Our preliminary tests for the health care scenario measured an incoming rate of about 1,000 microposts per minute. For a hashtag focused stream, this rate may drop to about less than 5 microposts per minute[8]. The Streaming API Quality of Service (QoS) states that microposts may be missing from the delivered stream, may arrive in any order and in near real time. The QoS of our implementation is upper-bounded on that of Twitter's API. The hashtag extraction and entity extraction steps take 0.00214 seconds per micropost. Serializing and storing a micropost takes 0.0137 sec. The entire annotation pipeline takes 0.454 seconds for an incoming rate of 20 tweets per second. PuSH broadcast of the RSS feed from the Semantic Publisher to the clients, via the PuSH hub, takes less that one second, when relying on the public Google PuSH hub.

Applications. The Linked Open Social Signals approach proposed in this work impacts a number of applications where real-time acquisition of social perceptions is needed, information overload is an issue and flexibility in selecting subsets for analysis is important. Twitris[9] is an analytical web application that provides spatial-temporal-thematic (STT) exploration of aggregated social signals. The central thesis behind Twitris is that citizen sensor observations are inherently multi-dimensional in nature and taking these dimensions into account will provide useful organization and consumption principles. In addition to what is being said about an event (theme), where (spatial) and when (temporal) are integral components to the analysis of social signals. Our approach extends Twitris to provide semantically annotated content with links to the LOD cloud, as well as to enable real-time streaming of annotated microblog posts.

[8]Stream limited to posts containing *#hcr*
[9]http://twitris.org

VII. Related Work

In mid-2008, we provided one of the first Semantic Web enabled microblogging frameworks with SMOB — Semantic MicrOBlogging. Among other functionality, SMOB provides an interface where users can annotate their content directly with URIs from DBpedia (and other KBs), by suggesting resources when typing hashtags. Each micropost generated through SMOB is immediately available as RDF data on the Web, and made publicly available for further reuse and mashups. Since then, several systems have been build to enhance Twitter with Semantic Web technologies. Other related microblogging approaches comprise Identi.ca, Status.net, SemanticTweet.com and Smesher.org. These approaches differ greatly from ours as they do not focus on extracting semantic descriptors of microposts. Another class of approaches aims at parsing specialized syntaxes (nanosyntaxes) for adding triples from tweets, including HyperTwitter (semantictwitter.appspot.com) and TweetLogic. In contrast to those approaches, we focus on extracting information from regular tweets from casual users.

Considering the real time notification of updates of LOD sources, our system is marginally related to Semantic Web search engines[10][11] and update notification vocabularies [12] [13] [14]. Those search engines or update notification vocabularies

[10]http://pingthesemanticweb.com/
[11]http://kmi-web05.open.ac.uk/WatsonWUI/
[12]http://vocab.org/changeset/schema.html
[13]http://triplify.org/vocabulary/update
[14]http://vocab.deri.ie/dady

allow users to retrieve any new links to a particular entity (*e.g.* a topic or a location) during a particular interval. Those changes could include any links, not only microblog posts. Moreover, the use of those approaches implies a *pull* model and all its disadvantages for real-time delivery. Other approaches have proposed streaming of SPARQL results. C-SPARQL [3] is an extension of SPARQL to support continuous queries over RDF data streams. C-SPARQL is defined by orthogonal extensions to the standard SPARQL grammar. As future work we plan to also support queries encoded in the C-SPARQL dialect. The authors also recently published an execution environment for C-SPARQL[2] that can be tested as a substitute RDF Store to the implementation used in our current prototype.

VIII. Conclusion

In this work we described a comprehensive approach for collecting and distributing social signals in real time. The use of Linked Open Data principles for sharing social signals will facilitate global information access for the collective analysis of social signals.

We introduced the idea of concept feeds for aggregating microblog posts. We proposed a realization of such concepts as SPARQL queries that have the ability to express compositions of entities, classes and attributes of interest - on the thematic, spatial and temporal dimensions. This goes beyond the use of hashtags. Concept feeds provide more sophisticated and flexible mechanisms for selecting subsets of interest. The streaming of concept feeds was engineered through the PuSH protocol, a scalable distributed solution for real-time delivery of information. Our approach represents a novel contribution with a positive impact on several applications that employ the collective analysis social signals for situational awareness.

The implementation is shared as an open source implementation available at http://code.google.com/p/rtsw/ under license New BSD License[15]. A project page with updated links, instructions and results is kept under http://wiki.knoesis.org/index.php/LinkedOpenSocialSignals.

Acknowledgements

The work presented in this paper has been funded in part by Science Foundation Ireland under Grant No. SFI/08/CE/I1380 (Líon 2). Thanks to T.K. Prasad, Harshal Patni and the Sensor Web team for allowing us to use their servers for the development of the software used in this project.

References

[1] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. DBpedia: A nucleus for a web of open data. In *Proceedings of the 6th International Semantic Web Conference (ISWC)*, volume 4825 of *Lecture Notes in Computer Science*, pages 722–735. Springer, 2008.

[2] D. F. Barbieri, D. Braga, S. Ceri, and M. Grossniklaus. An execution environment for c-sparql queries. In *EDBT*, pages 441–452, 2010.

[3] D. F. Barbieri, D. Braga, S. Ceri, E. D. Valle, and M. Grossniklaus. C-sparql: Sparql for continuous querying. In *WWW*, pages 1061–1062, 2009.

[4] C. Bizer, T. Heath, and T. Berners-Lee. Linked data - the story so far. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 2009.

[5] D. Gruhl, M. Nagarajan, J. Pieper, C. Robson, and A. P. Sheth. Context and domain knowledge enhanced entity spotting in informal text. In *International Semantic Web Conference*, pages 260–276, 2009.

[6] A. Mathes. Folksonomies: Cooperative Classification and Communication Through Shared Metadata, December 2004.

[7] Mercedes Bunz. . *The Guardian, PDA: The Digital Content Blog*, February 2010.

[8] T. M. Mitchell. *Machine Learning*. McGraw-Hill, New York, 1997.

[9] D. Nadeau and S. Sekine. A survey of named entity recognition and classification. *Linguisticae Investigationes*, 30(1):3–26, January 2007. Publisher: John Benjamins Publishing Company.

[10] M. Nagarajan, K. Gomadam, A. P. Sheth, A. Ranabahu, R. Mutharaju, and A. Jadhav. Spatio-temporal-thematic analysis of citizen sensor data: Challenges and experiences. In *WISE*, pages 539–553, 2009.

[11] A. Passant, P. Laublet, J. G. Breslin, and S. Decker. A URI is Worth a Thousand Tags: From Tagging to Linked Data with MOAT. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 5(3):71–94, 2009.

[12] L. Ratinov and D. Roth. Design challenges and misconceptions in named entity recognition. In *CoNLL '09: Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, pages 147–155, Morristown, NJ, USA, 2009. Association for Computational Linguistics.

[13] A. Seaborne, G. Manjunath, C. Bizer, J. G. Breslin, S. Das, I. Davis, S. Harris, K. Idehen, O. Corby, K. Kjernsmo, and B. Nowack. SPARQL Update – A language for updating RDF graphs. W3C Member Submission 15 July 2008, World Wide Web Consortium, 2008. http://www.w3.org/Submission/2008/SUBM-SPARQL-Update-20080715/.

[14] A. Sheth and M. Perry. Traveling the semantic web through space, time, and theme. *IEEE Internet Computing*, 12(2):81–86, 2008.

[15] E. F. Tjong Kim Sang and F. De Meulder. Introduction to the conll-2003 shared task: language-independent named entity recognition. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003*, pages 142–147, Morristown, NJ, USA, 2003. Association for Computational Linguistics.

[15] http://www.opensource.org/licenses/bsd-license.php