Real-time #SemanticWeb in <= 140 chars

Joshua Shinavier
Tetherless World Constellation
Rensselaer Polytechnic Institute
Troy, NY 12180, USA
josh@fortytwo.net

ABSTRACT

Instant publication, along with the prospect of widespread geotagging in microblog posts, presents new opportunities for real-time and location-based services. Much as these services are changing the nature of search on the World Wide Web, so the Semantic Web is certain to be both challenged and enhanced by real-time content. This paper introduces a semantic data aggregator which brings together a collection of compact formats for structured microblog content with Semantic Web vocabularies and best practices in order to augment the Semantic Web with real-time, user-driven data. Emerging formats, modeling issues, publication and data ownership of microblogging content, and basic techniques for real-time, real-place semantic search are discussed.

1. INTRODUCTION

Compared to the World Wide Web, the Semantic Web is lacking in user-driven content. Large, user-curated data sets such as those of DBPedia[1] and Freebase[2] notwithstanding, recent analyses [5][9] of the Linked Data cloud indicate that it does not exhibit the power-law distributions or strong connectivity typical of naturally-evolving networks.

Meanwhile, Web 2.0 services channel large amounts of potentially valuable user-driven data every minute. Semantic wikis and the Microformats[3] community aim to bridge this gap by enabling users to add small amounts of semantic data to their content, while much of the work on *semantic microblogging* thus far focuses on representing users, microblogs and microblog posts in the Semantic Web: essentially, on doing for microblogs what SIOC[4] has done for blogs. The work described in this paper takes the complementary approach of harvesting semantic data *embedded in* the content of microblog posts, or of doing for microblogs what microformats do for Web pages. Its contribution to the emerging semantic microblogging ecosystem includes:

- a set of syntax conventions for embedding various structured content in microblog posts
- an information schema for user-driven data and associated metadata
- a technique for translating microblog streams into RDF streams in real-time
- a way of publishing user-driven data in the web of Linked Data while being fair to microblog authors
- an open-source semantic data aggregator, called TwitLogic, which implements the above ideas

In addition, a simple technique for scoring microblog content based on recency and proximity is presented.

2. NANOFORMATS FOR THE SEMANTIC WEB

A number of compact formats, variously called *nanoformats*[5], *picoformats*,[6] or *microsyntax*,[7] have been proposed to allow users to express structured content or issue service-specific commands in microblog posts. Examples in widespread use include @usernames for addressing or mentioning a particular user, and #hashtags for generic concepts. So-called *triple tags* even allow the expression of something like an RDF triple. These formats are subject to a tradeoff between simplicity and expressivity which heavily impacts community uptake.

2.1 Twitter Data

Twitter Data [4] is an open proposal for embedding structured data in Twitter messages. According to its FAQ, "the purpose of Twitter Data is to enable community-driven efforts to arrive at conventions for common pieces of data that are embeddable in Twitter by formal means". To kick-start this process, Twitter Data introduces a concrete syntax based on key/value pairs. For instance,

I love the #twitterdata proposal! $vote +1

RDF-like triples are possible using explicit subjects:

@romeo $foaf>loves @juliet

2.2 MicroTurtle

MicroTurtle[8] is a particularly compact serialization format for RDF, capable of embedding general-purpose RDF data in

[1] http://dbpedia.org/
[2] http://www.freebase.com/
[3] http://microformats.org/
[4] http://rdfs.org/sioc/spec/

WWW2010, April 26-30, 2010, Raleigh, North Carolina.

[5] http://microformats.org/wiki/microblogging-nanoformats
[6] http://microformats.org/wiki/picoformats
[7] http://www.microsyntax.org/
[8] http://buzzword.org.uk/2009/microturtle/spec

microblog posts. It makes use of hard-coded CURIE[9] prefixes as well as keywords for terms in common vocabularies such as FOAF,[10] Dublin Core,[11] and OpenVocab.[12] For example:

Wow! Great band! #mttl #music <#me>
♥ [→<http://theholdsteady.com/> #altrock] .

This expresses, in a named graph tagged "music", that the author of the post likes a band, tagged "altrock", with the given homepage.

2.3 smesher

Smesher[13] is a semantic microblogging platform which collects structured data from microblog posts in a local RDF store. Its syntax includes key/value pairs similar to Twitter Data's which are readily translated into RDF statements:

RT @sue: I can #offer a #ride to the #mbc09 #from=Berlin #to=Hamburg

Smesher users can query their data using SPARQL and filter it to create customized data streams.

2.4 TwitLogic

TwitLogic currently supports a near-natural-language format which is intended to be particularly memorable and unobtrusive. Structured content is expressed by annotating a user name, hashtag or URL with a parenthetical "afterthought" resembling a relative clause. For example:

Great convo with @lidingpku (creator of #swoogle) about ranking algos.

This approach makes the assumption that hashtags such as #swoogle are semantically stronger than "ordinary" tags, in that microblog users who really want to refer their readers to a specific concept tend to avoid ambiguous tags. Ideally, the syntax should be natural enough so as not to distract the reader, yet contrived enough to minimize false positives:

#sioclog (see http://bit.ly/2uAWo2) makes Linked Data from IRC logs.

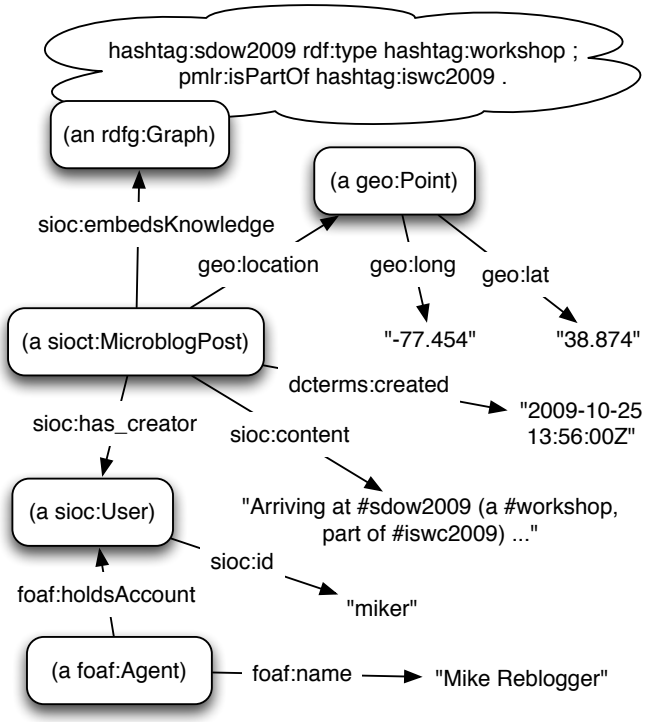
Some afterthoughts are represented with more than one RDF statement. For example, the following produces a minimal review of a movie in terms of the RDF Review Vocabulary:[14]

Who would have guessed such a funny movie as #Zombieland (3/4) could be made around zombies?

There are several other "review" formats, such as LouderTweets,[15] which could be handled in exactly the same way; all that is needed is an appropriate parser for the format.

TwitLogic will eventually take advantage of some of the other formats described above, as well as pre-existing conventions such as "tag++". Probably the best approach to the chicken-and-egg problem of semantic nanoformats is to promote and build tools to support a variety of formats, see what "sticks", and then take steps to keep up with any community-driven conventions which may arise.

Figure 1: Embedded data and its metadata



3. A USER-DRIVEN SEMANTIC WEB KNOWLEDGE BASE

There are several kinds of structured content which can be gathered from a microblogging service such as Twitter:

1. authoritative information about microblogging *accounts* and the *people* who hold them. SemanticTweet[16] is an example of a service which publishes the social network information provided by Twitter on the Semantic Web.

2. authoritative information about microblog *feeds* and individual *posts*. The SMOB semantic microblogging system [7], for one, represents microblog content at this level.

3. user-created information *embedded* in the text of a microblog post. Gathering such user-driven "statements about the world" and using them to populate the Semantic Web is the main goal of TwitLogic. People, accounts, and microblog posts are included in the knowledge base only as contextual metadata to enhance information discovery and provide author attribution for the embedded data.

[9] http://www.w3.org/TR/curie/
[10] http://xmlns.com/foaf/spec/
[11] http://dublincore.org/documents/dcmi-terms/
[12] http://open.vocab.org/terms/
[13] http://smesher.org/
[14] http://vocab.org/review/terms
[15] http://www.loudervoice.com/2007/06/13/loudervoice-twitter-mash-up/
[16] http://semantictweet.com/

3.1 Representing microblog content in RDF

The schema used for TwitLogic's knowledge base draws upon a discussion[17] on the Semantic Web mailing list about RDF vocabularies for modeling microblogging resources. In particular, it makes use of a collection of terms from the FOAF, SIOC, Dublin Core, Named Graphs[18] and Basic Geo[19] vocabularies.

The sioc:embedsKnowledge property, which has been proposed in connection with UfoWiki [8], serves to associate a microblog post with any structured data that has been extracted from it, in the form of a named graph containing the extracted RDF statements. This link not only provides source metadata for those statements, but it also connects them with a *timestamp* and, potentially, a *placestamp* which are useful in searching and filtering. The use of geo:location as depicted is convenient, although its domain of geo:SpatialThing arguably does not include microblog posts.

3.2 Publishing the knowledge base as Linked Data

From the moment it is added to the TwitLogic knowledge base, the embedded data and contextual metadata of a microblog post are made available in accordance with best practices for publishing Linked Data on the Web [3]. Also available are a voiD [1] description of the data set as a whole, periodically generated RDF dumps of the data set, and owl:sameAs links into related data sets (currently, SemanticTweet's). In order to serve all of the information about a resource against the same dereferenceable URI (regardless of how many user-driven named graphs the resource is described in), TwitLogic provides unconventional TriX and TriG serializations[20] of the data alongside the more common RDF formats.

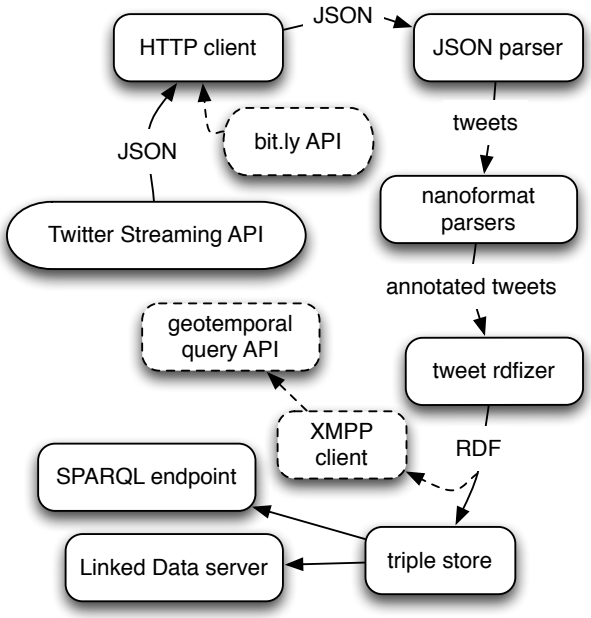
3.3 Data ownership

According to Twitter's terms of service,[21] "you own your own content", although that content may be freely copied, modified, and redistributed by Twitter and its partners. The data model described above supports authors' rights by providing attribution metadata for all user-driven content: the text of a microblog post is always associated with its author, and the RDF statements from embedded data in a post are always contained in a named graph which is associated with that post. Although TwitLogic does not rdfize every microblog post that passes through the system, it does maintain an RDF description of every tweet from which it has extracted content, so that attribution metadata is guaranteed, independently of the availability of the Linking Open Data[22] data sets into which TwitLogic links.

On account of its diverse authorship, the TwitLogic knowledge base as a whole is published under the Open Data Commons [6] Public Domain Dedication and License (PDDL).

[17] http://lists.w3.org/Archives/Public/semantic-web/2009Sep/0174.html
[18] http://www.w3.org/2004/03/trix/
[19] http://www.w3.org/2003/01/geo/
[20] http://wiki.github.com/joshsh/twitlogic/twitlogic-linked-data
[21] http://twitter.com/tos
[22] http://esw.w3.org/topic/SweoIG/TaskForces/CommunityProjects/LinkingOpenData/

Figure 2: Data flow in TwitLogic



4. ARCHITECTURE AND IMPLEMENTATION

The essential components of TwitLogic are an HTTP client, a collection of nanoformat parsers, and an rdfizer component which translates microblogging artifacts and annotations from TwitLogic's internal object model to equivalent RDF representations. The HTTP client maintains a connection with Twitter's streaming API,[23] receiving status updates, or JSON-formatted tweets, as they become available. When the client receives a tweet, it passes it into a parser which creates an intermediate Java object for the tweet, making it easier to work with at the lower levels. This tweet object is then passed into the top level of a hierarchy of parsers which combine BNF grammars with procedural code to match expressions in any of the supported nanoformats. If the tweet is successfully matched by the parser for a specific syntax convention, the parser will attach an annotation in RDF which the tweet carries with it into the rdfizer. The rdfizer, in turn, mints a URI for the graph into which it places the annotation and maps the tweet and the rest of the metadata into RDF according to the schema in Figure 1.

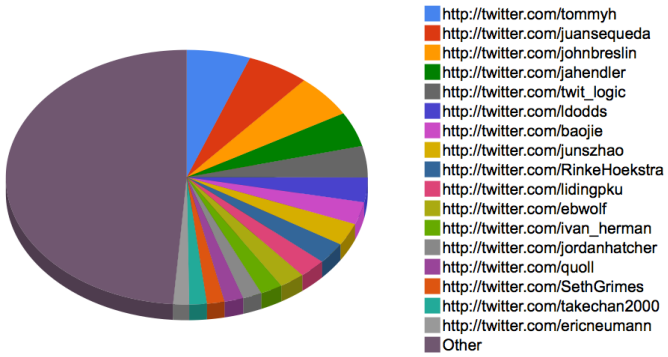
At this point, the tweet passes into an RDF stream and from there into a built-in Sesame triple store. This triple store is exposed via a SPARQL endpoint as well as the Linked Data server. The application is configurable with respect to the underlying triple store and to the set of users and Twitter Lists it "listens to" through Twitter's rate-limited API.

The open-source TwitLogic implementation[24] is available online. The TwitLogic home page can be found at:

http://twitlogic.fortytwo.net/

[23] http://apiwiki.twitter.com/Streaming-API-Documentation
[24] http://github.com/joshsh/twitlogic

Figure 3: Who's tweeting about ISWC 2009?



4.1 Demo application

A Linked Data mashup[25] featuring TwitLogic was deployed at the 8th International Semantic Web Conference.[26] The mashup, called Linking Open Conference Tweets, gathered statistics about conference-related Twitter posts to generate Google visualizations, at regular intervals, as the conference proceeded. A tweet was considered to be conference-related if it contained the hashtag, #iswc2009, of the conference, or the hashtag of any of the subevents of the conference, such as #sdow2009. This goes beyond the search functionality provided by the Twitter API. The #iswc2009 hashtag was used as a starting point for the statistics, but the application had no other built-in knowledge of the conference. Instead, it executed SPARQL queries over data provided by TwitLogic and the Semantic Web Conference Corpus[27] to discover sub-events on-the-fly. The Conference Corpus provided the subevent-superevent relationships, while a few individuals very deliberately tweeted TwitLogic-formatted owl:sameAs statements to link hashtags to Conference Corpus resources. Two dozen structured tweets were far more than enough to categorize hundreds of tweets by individuals with no knowledge of the syntax, demonstrating that "a little semantics goes a long way".

The author plans to make an enhanced version of the Linking Open Conference Tweets service available for the 2010 World Wide Web Conference.[28]

5. TOWARDS REAL-TIME, REAL-PLACE SEMANTIC SEARCH

Apart from collecting and publishing user-driven semantic data, we would also like to search and reason on it The mashup described above uses SPARQL to query over the data, and other Semantic Web query and reasoning techniques are equally possible. In this environment, however, every user-driven statement is associated with a time-stamped and potentially[29] place-stamped microblog entry. We would like to take advantage of this metadata in order to score search results based on nearness in time and location to the context of the query.

At least for some types of query, a simple but effective approach is to keep a record of the *influence* of a named graph on intermediate results during query execution and to combine this with a measure of the *significance* of the graph, in terms of space and time, to produce a score for each query result. Overall significance of a graph is computed as the product of its significance in time and space. To illustrate, when the (abbreviated) SPARQL query

```
SELECT ?w WHERE { ?w rdf:type hashtag:workshop }
```

is evaluated against a knowledge base containing the data in Figure 1, the triple pattern will match one of the embedded statements which were tweeted in Chantilly, Virginia at a certain time in October, 2009. Since there's only one triple pattern, the influence on the corresponding result w of the graph containing that statement is 1, whereas the significance of the graph depends on the query context. One might imagine a SPARQL engine augmented with an appropriate system of provenance-aware bindings, such that a user in Chantilly at the time of the workshop would find hashtag:sdow2009 first in a list of scored solutions, whereas a user in Germany, or the following day in Chantilly, would find other workshops first, due to the different significance of graphs in different contexts. No such query engine has been designed or built, although current work in progress uses the significance function in an analogous manner to control activation-spreading.

5.1 Time-based significance

Clearly, the significance of a graph decreases over time in a query environment which favors recency. That is, there is an inverse relationship between the significance of a graph and the positive difference between the current (or reference) time and the timestamp of the graph. Specifically, the time-based significance, S_{time}, of a statement should have a value of 1 when there is no difference in time, and should approach a designated baseline value, b_{time}, as the difference goes to infinity. A modified exponential decay function has this behavior:

$$S_{time}(t) = b_{time} + (1 - b_{time}) \cdot 2^{-\frac{t}{t_h}}, \quad (1)$$

where t_h is the amount of time it takes for the significance of a statement to drop to half of its original value, disregarding the influence of b_{time}.

The resulting preference for the *most recently acquired* information is fitting for microblogging environments, in which it is generally not possible to "take back" statements which have been made in the past:[30] instead, one simply makes new statements.

The constant b_{time} should be given a value greater than zero if it is not desirable for old statements to "disappear" entirely. For example, a statement of a person's gender is usually as valid years from now as it is today. However, as new information becomes available, it will tend to supplant older information. This "freshness" is particularly advantageous for properties such as foaf:based_near whose value is subject to frequent change.

[25] http://tw.rpi.edu/portal/Linking_Open_Conference_Tweets
[26] http://iswc2009.semanticweb.org/
[27] http://data.semanticweb.org/
[28] http://www2010.org/
[29] http://blog.twitter.com/2010/03/whats-happeningand-where.html
[30] Some microblogging services, including Twitter, do allow users to delete their posts. Nonetheless, once a post is "out there", it is potentially out there for good, both in computer systems and human memory.

5.2 Location-based significance

Our requirements for location-based significance are the same as those for time-based significance, except that geo-distance varies over a finite interval, whereas time-distance varies over an infinite one. S_{loc} should have a value of 1 at the current (or reference) location, and a baseline value of b_{loc} at the maximum distance d_{max}:

$$S_{loc}(d) = 1 + (b_{loc} - 1) \cdot \frac{d}{d_{max}} \quad (2)$$

where d is the great-circle distance from the reference location. Note that only distance, as opposed to actual position on the global map, is considered here.

5.3 Closing the world of time and space

It is necessary to impose a baseline significance of b_{time} or b_{loc} on graphs with no time or place metadata. Equivalently, one can think of such graphs as occupying a time and place long, long ago or at the other end of the world. All of the data from the *rest* of the Semantic Web which we might like to search and reason on, resides here. For example, if b_{time} and b_{loc} are both 1/4, then static data will not fall to less than $1/16^{th}$ the significance of any new data.

5.4 What is real-time?

TwitLogic provides real-time queries in that it enables querying on real-time data: a few milliseconds after a microblog post is received from Twitter's streaming API, it is ready to participate in query answering. The data loses significance over time, but it remains available for all future queries unless it is removed by external means. This approach is distinct from continuous query techniques, such as C-SPARQL [2], in which a query is first defined, then allowed to match data as it becomes available. On the other hand, TwitLogic produces an RDF stream which C-SPARQL could query natively, given a suitable transport protocol.

6. WORK IN PROGRESS, AND FUTURE WORK

The dashed lines in Figure 2 indicate TwitLogic components currently under development. Apart from merely adding to the triple store, the RDF stream is additionally broadcast by an XMPP client using the Publish-Subscribe[31] extension, as has been done in other contexts.[32] An AllegroGraph[33] instance subscribes to the stream in order to execute time- and location-based queries using the scoring system described above. Thus decoupling the query environment from the aggregator not only allows it to take advantage of AllegroGraph's built-in geotemporal features; this is also a step towards a more modular architecture in which a service like TwitLogic merely produces an RDF stream in a well-understood way, for consumption by external services.

For now, the relative newness of this domain means that TwitLogic has to play a number of roles in order to enable its data to be put to a variety of uses.

As an incentive to use semantic nanoformats effectively, Twitter users will be able to access the query API by tweeting at the @twit_logic user, which responds to correctly-formatted queries with a bit.ly URL for a query results page. In order to get better results, it will be in users' best interest to tweet relevant information. Since tweet-based queries are answered in real-time and contain a placestamp for those users who have opted into Twitter's geolocation functionality, no extra syntax is required to make queries time- and location-sensitive.

Apart from more and more real-time Semantic Web mashups, possibilities for future work with TwitLogic include continuous queries, support for a greater variety of semantic nanoformats, and a trust-based significance factor for query evaluation.

[31] `http://xmpp.org/extensions/xep-0060.html`
[32] `http://vimeo.com/992973`
[33] `http://www.franz.com/agraph/allegrograph/`

7. ACKNOWLEDGEMENTS

This project has been supported by Franz Inc. as well as RPI's Tetherless World Constellation. Special thanks go to Jans Aasman, James A. Hendler, Deborah L. McGuinness and Steve Haflich for their positive influence on the concept and implementation of TwitLogic, to Li Ding and Zhenning Shangguan for their contribution to the Linking Open Conference Tweets mashup, and to Jie Bao, Tom Heath, Marko A. Rodriguez, Alvaro Graves, Gregory Todd Williams, Jesse Weaver and Xixi Luo for their helpful comments and feedback.

8. REFERENCES

[1] K. Alexander, R. Cyganiak, M. Hausenblas, and J. Zhao. Describing linked datasets: On the design and usage of voiD, the "Vocabulary of Interlinked Datasets". In *2nd International Workshop on Linked Data on the Web*, Madrid, Spain, April 2009.

[2] D. Barbieri, D. Braga, S. Ceri, E. Della Valle, and M. Grossniklaus. C-SPARQL: SPARQL for continuous querying. In *Proceedings of the 18th international conference on World Wide Web*, pages 1061–1062, Madrid, Spain, April 2009. ACM.

[3] C. Bizer, R. Cyganiak, and T. Heath. How to publish Linked Data on the Web. http://www4.wiwiss.fu-berlin.de/bizer/pub/LinkedDataTutorial/.

[4] T. Fast and J. Kopsa. Twitter Data – a simple, open proposal for embedding data in Twitter messages. http://twitterdata.org/, May 2009.

[5] H. Halpin. A query-driven characterization of Linked Data. In *2nd International Workshop on Linked Data on the Web*, Madrid, Spain, April 2009.

[6] P. Miller, R. Styles, and T. Heath. Open Data Commons, a license for open data. In *1st International Workshop on Linked Data on the Web*, Beijing, China, April 2008.

[7] A. Passant, T. Hastrup, U. Bojars, and J. Breslin. Microblogging: a Semantic Web and distributed approach. In *Proceedings of the 4th Workshop on Scripting for the Semantic Web*, June 2008.

[8] A. Passant and P. Laublet. Towards an interlinked semantic wiki farm. In *3rd Semantic Wiki Workshop*, 2008.

[9] M. Rodriguez. A graph analysis of the Linked Data cloud. KRS-2009-01, February 2009.