Twarql: Tapping Into the Wisdom of the Crowd*

Pablo N. Mendes
Kno.e.sis Center, Wright State University
Dayton, OH, USA

Alexandre Passant
Digital Enterprise Research Institute
National University of Ireland, Galway, Ireland

Pavan Kapanipathi
Kno.e.sis Center, Wright State University
Dayton, OH, USA

ABSTRACT

Twarql is an infrastructure translating microblog posts from Twitter as Linked Open Data in real-time. The approach employed in Twarql can be summarized as follows: (1) extract content (*e.g.* entity mentions, hashtags and URLs) from microposts streamed from Twitter; (2) encode content in RDF using shared and well-known vocabularies (FOAF, SIOC, MOAT, etc.); (3) enable structured querying of microposts with SPARQL; (4) enable subscription to a stream of microposts that match a given query; and (5) enable scalable real-time delivery of streaming annotated data using sparqlPuSH. In this paper we use a brand tracking scenario to demonstrate how Twarql enables flexibility in handling the information overload of those interested in collectively analyzing microblog data for sensemaking. The dataset produced is shared as Linked Data. Twarql is available as open source and can be easily deployed or extended for monitoring Twitter data in various contexts such as brand tracking, disaster relief management, stock exchange monitoring, etc.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

Design, Management, Experimentation

Keywords

Social Media, RDF, SPARQL, Streaming, Twitter

1. INTRODUCTION

Every day, Web users are using Twitter to simultaneously publish millions of microblog posts (microposts or "tweets") with opinions, observations and suggestions that may represent invaluable information for businesses and researchers around the world[1]. Taking advantage from this "wisdom of the crowd" — which refers to "the process of taking into account the collective opinion of a group of individuals rather than a single expert to answer a question"[2] —, Twitter data has been successfully used, for example, to forecast box-office revenues for movies [1] or to manage earthquakes detection [6]. However, analyzing the vast amount of microblog data published each second can be extremely challenging, especially in situations where it has to be done in real-time.

Twarql encodes information from microblog posts as Linked Open Data in order to enable expressive queries and flexible analysis of microblog data for sensemaking tasks. Instead of requiring the use of keywords or custom software for filtering information, Twarql leverages a full fledged query language (SPARQL) that is much more expressive than keywords in selecting subsets for analysis. One obvious example of such queries would be a query for a stream of microposts that match given criteria. However, Twarql allows us to approach the data from different perspectives, pivoting the analysis on different aspects — *e.g.* users, topic, time, location of tweets. This functionality is enabled by the annotation of Tweets using different models and knowledge bases (each of them for a specific purpose).

In this paper we demonstrate Twarql's flexible support for expressive queries through a brand tracking scenario related to the recently released Apple IPad™. Twarql is available at `http://twarql.sf.net` as open source and can be easily extended and deployed to enable Twitter monitoring systems that can be used in various contexts: brand tracking, disaster relief management, stock exchange monitoring, etc. A screencast demonstration is available from the aforementioned project website.

2. SYSTEM DESCRIPTION

Twarql collects, annotates, filters and delivers to the interested party the data that are relevant to a given query. Twarql's architecture (Figure 2)[3] is (1) loosely coupled, (2) relies on existing W3C standards and protocols, (3) entirely HTTP-based and (4) can be easily deployed in many platforms. This allows interested parties to deploy their own system without having to rely on a centralized authority to index or distribute their feeds.

*This work has been funded in part by Science Foundation Ireland under Grant No. SFI/08/CE/I1380 (Líon 2).

I-SEMANTICS 2010 September 1-3, 2010, Graz, Austria.

[1]See recent statistics from Twitter at `http://blog.twitter.com/2010/02/measuring-tweets.html`.
[2]From Wikipedia, see `http://en.wikipedia.org/wiki/Wisdom_of_the_crowd`.
[3]Animated version of the figure is available at `http://wiki.knoesis.org/index.php/Twarql`

Data collection is performed through the use of the Twitter Streaming API[4]. As new tweets are acquired, they are sent through the information extraction and annotation modules. The information extraction module performs a series of processing steps for annotating microposts based on the content in those microposts. 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 sentiment annotation, plus a collection of DBpedia entities [2], hashtags, hashtag definitions and URLs that help to expand the description of its content. More information about the extraction module itself is provided in [4].

After the information extraction is complete, tweets are encoded in RDF. Particularly, we rely on the following ontology stack: (1) **FOAF** (`http://foaf-project.org`) — Friend of a Friend — to represent users and their social network; (2) **SIOC** (`http://sioc-project.org`) — Semantically-Interlinked Online Communities — to model microblog updates themselves; (3) **OPO** (`http://online-presence.net`) — Online Presence Ontology — to describe 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** (`http://moat-project.org`) — Meaning Of A Tag — and **CommonTag** (`http://commontag.org`) to model semantic tagging capabilities, i.e. linking tagged microposts to meaningful resources on the Linked Open Data Cloud. Once the information is transformed to RDF, it is sent to a publisher using SPARQL Update[5] via HTTP. Twarql offers a programmatic API and a Web-based user interface that allow users to register queries and request a stream for a registered query using the sparqlPuSH approach [5]. Twarql also offers a SPARQL Protocol compliant interface that allows users to query archived streamed data. In order to facilitate non-computer expert user interaction, Twarql uses Cuebee (http://cuebee.sf.net) as user-interface for query formulation. Cuebee is able to encode and execute SPARQL queries on any SPARQL Protocol-compliant server.

3. SCENARIO: BRAND TRACKING

Social media has become an ubiquitous platform for Internet users to quickly and openly voice their opinions. The Nielsen Global Online Consumer Survey (July 2009) reports that 90% of people trust recommendation from their social network and 70% trust recommendations posted online [8]. Product managers, marketers and investors interested in monitoring the health and the value of their brands have social media as a rich source to engage in brand tracking: monitoring and analyzing the reputation of a brand.

Twarql allows users to encode questions as SPARQL queries that have the ability to narrow the incoming stream of information to a specific subset of interest. We list below a few example use cases to illustrate the usefulness of Twarql[6].

Use Case 1: Location. "Give me a stream of locations where my product is being mentioned right now."

By registering this query with Twarql, every time a tweet that matches this query is streamed, the system will update the user with a new location. On the client side a user may choose to show that on a map, or create statistics of popularity of the product across the country.

```
SELECT ?location
WHERE {
  ?tweet moat:taggedWith dbpedia:IPad .
  ?presence opo:currentLocation ?
      location .
  ?presence opo:customMessage ?tweet .
}
```

Figure 2: SPARQL query for Use Case 1.

Use Case 2: Sentiment. "Give me all people that have said negative things about my product."

```
SELECT ?user
WHERE {
  ?tweet sioc:has_creator ?user .
  ?tweet moat:taggedWith dbpedia:IPad .
  ?tweet twarql:sentiment twarql:
      Negative .
}
```

Figure 3: SPARQL query for Use Case 2.

Sentiment analysis is an open research problem that we do not attempt to solve in this project. Our focus is rather to demonstrate the use of sentiment annotations in the context of this tool. We employed a naive sentiment annotator that is based on dictionaries of positive and negative words to generate example data for this demonstration.

Use Case 3: Content suggestion. "Give me all URLs that people recommend with relation to my product."

```
SELECT ?url
WHERE {
   ?tweet moat:taggedWith dbpedia:IPad .
   ?tweet sioc:links_to ?url .
}
```

Figure 4: SPARQL query for Use Case 3.

Twarql uses regular expressions to extract URLs from tweets. The URLs are resolved (in case they are short URLs) and added as annotations to a tweet. That allows users to directly request content (pages) that is suggested in tweets. On the client side the user may decide to crawl those pages or show links to content that is relevant to their product.

Use Case 4: Related entities. "What competitors are being mentioned with my product?"

This use case requires merging streaming data with background knowledge information (*e.g.* from DBpedia). Examples of *?category* include *category:Wi-Fi_devices* and *category:Touchscreen_portable_media_players* amongst others. As a result, without having to elicit all products of interest as keywords to filter a stream, a user is able to leverage relationships in background knowledge to more effectively narrow down the stream of tweets to a subset of interest.

[4] `http://apiwiki.twitter.com/Streaming-API-Documentation`
[5] `http://www.w3.org/TR/sparql11-update/`
[6] Prefixes have been omitted for conciseness.

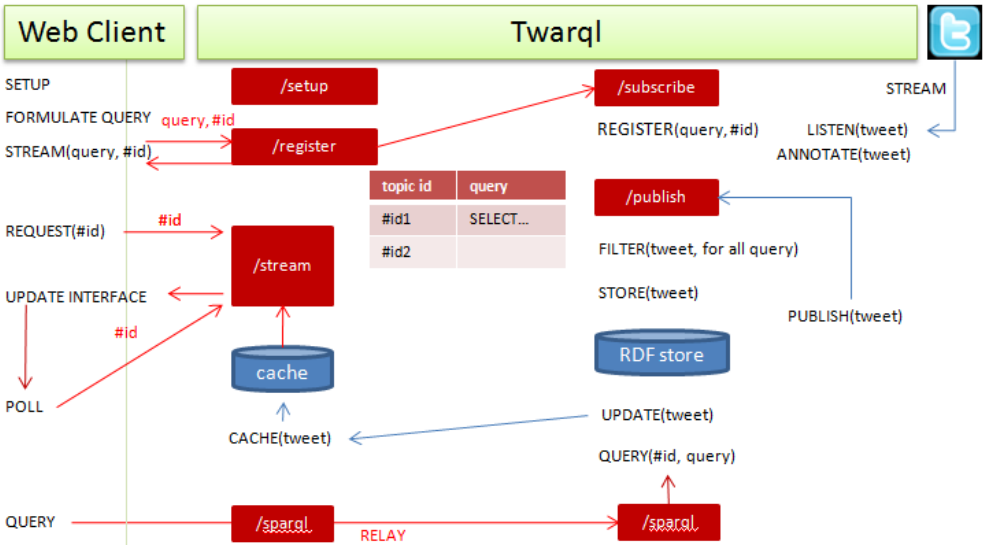

Figure 1: Flow of control through Twarql's architecture.

```
SELECT ?competitor
WHERE {
  dbpedia:IPad skos:subject ?category .
  ?competitor skos:subject ?category .
  ?tweet moat:taggedWith ?competitor .
}
```

Figure 5: SPARQL query for Use Case 4.

Note that all four use cases focus on retrieving items that are not tweets. The information extraction and annotation from microposts enables data aggregation in different dimensions, opening numerous possibilities for analysis.

In order to demonstrate Twarql capabilities, we streamed tweets for iPad between June 3rd and June 8th 2010. We collected a total of 511,147 tweets that were encoded in 4,479,631 triples. The dataset is available from `http://wiki.knoesis.org/index.php/Twarql`.

4. CONCLUSION

One of the bottlenecks for successfully generating a Web of Data is easy to install/configure/extend tools to generate triples. Twitter has the potential to generate many triples with user opinions, and other observations that are useful to many use cases. But as more and more triples are generated, it becomes obvious the need to control information overload. Twarql offers focused streams based on SPARQL queries as a solution to this problem.

Twarql integrates with various of our contributions in the Social Semantic Web [3] realm and in the Citizen Sensing [7] area. Particularly, Twarql relies on: (1) an ontology stack for representing microblogging information, built in the context of the SMOB microblogging platform; (2) sparqlPuSH [5] a way to push the results of SPARQL queries matching new data loaded in the triple store; (3) Cuebee (`http://cuebee.sf.net`), an interface for knowledge guided query formulation.

In this paper we focused on demonstrating how Twarql can help in analyzing subsets of micropost feeds through flexible and expressive querying. We discussed Twarql in the context of brand tracking. Datasets for other scenarios can be easily generated, other extractors can be added to the pipeline and suitable SPARQL queries can be dynamically registered through Twarql's API as needed for a given use case. Twarql is available as open-source at `http://twarql.sf.net`

5. REFERENCES

[1] S. Asur and B. A. Huberman. Predicting the future with social media. *CoRR*, abs/1003.5699, 2010.

[2] S. Auer, C. Bizer, J. Lehmann, G. Kobilarov, R. Cyganiak, and Z. Ives. Dbpedia: A nucleus for a web of open data. In *Proceedings of the 6th International Semantic Web Conference and 2nd Asian Semantic Web Conference (ISWC/ASWC2007)*, volume 4825 of *Lecture Notes in Computer Science*, pages 715–728. Springer, 2007.

[3] John G. Breslin and Alexandre Passant and Stefan Decker. *The Social Semantic Web*. Springer.

[4] P. N. Mendes, A. Passant, P. Kapanipathi, and A. P. Sheth. Linked open social signals. In *Web Intelligence and Intelligent Agent Technology, 2010. WI-IAT '10. IEEE/WIC/ACM International Conference on*, 2010.

[5] A. Passant and P. N. Mendes. sparqlPuSH: Proactive notification of data updates in RDF stores using PubSubHubbub. In *Scripting for the Semantic Web Workshop (SFSW2010) at ESWC2010*, 2010.

[6] T. Sakaki, M. Okazaki, and Y. Matsuo. Earthquake shakes twitter users: Real-time event detection by social sensors. In *WWW2010*. ACM, 2010.

[7] A. Sheth. Citizen Sensing, Social Signals, and Enriching Human Experience. *IEEE Internet Computing*, 13(4):87–92, 2009.

[8] The Nielsen Company. Global Advertising: Consumers Trust Real Friends and Virtual Strangers the Most.