Microblogging Macrochallenges for Repositories

Adam Field & Leslie Carr, University of Southampton

Social Media, Social Science and Repositories

The Web offers social scientists and other researchers unprecedented data resources for primary empirical research and for the development of analytical and methodological skills. The ubiquity of confessional interviews across the mass media offers instant access to people's everyday lives and thoughts. Reflecting on these developments, Savage and Burrows (2007) insist that academic researchers should engage seriously with the digital data that are produced outside the academy and that this data poses fundamental challenges for sociological practice[1].

The need for repositories to support researchers by facilitating long-term access to otherwise ephemeral social data can be clearly seen in the following cases:

- A digital humanities researcher describes her wishes to archive a number of 'Facebook walls' in order to capture evidence of labour movement activism in Egypt[2]. "Beyond taking screen shots and saving as a PDF, this is difficult to do, and it is risky to assume that the material you want to look at will still be available in six months' time, let alone years hence."
- A criminologist studies the causes of civil disturbance from the participants in a riot who post Youtube videos taken from their own cameras, but who then remove them for fear of police action[3]. "But the fourth, and by far the most comprehensive resource, has been video material posted to You Tube. There has been a remarkable wealth of such material, although some of it is now being removed out of fears it will be used to prosecute participants."

At the University of Southampton, the newly formed interdisciplinary Web Science Doctoral Training Centre[4] brings Web technologists and social scientists together to analyse the impact of the Web on society. Many students have identified Twitter and other social media channels as useful sources of data for their initial investigations, but lack either the technical knowledge or the time to create their own software to download and store data from the services' respective APIs.

This paper describes our attempt to adapt a repository to capture real-time data from a popular social media service (Twitter) in order to facilitate student projects. Twitter (twitter.com) is a popular microblogging service, first established in 2006, that enables subscribers to send and receive status updates limited to 140 characters. Its web API allows query access to its database, but the scale of data and the rate that it is updated (approximately 290 million tweets per day) means that API access is deliberately throttled (and that query access to individual status updates (tweets) is limited to the most recent week.

In this paper we describe the implementation of two designs for Tweet ingest, both of which are shown to be flawed with respect to the scale of the data capture problem. The work described in this paper was undertaken on an EPrints version 3.3 installation, but the lessons learned are applicable to any digital object store or repository platform. The wider aim of this work is (a) to improve the ingest and curation of streams of rich media items (not limited to Twitter) and (b) to embed the repository at the very start of the research process and workflow, improving the relationship between the researcher and the repository by increasing the value of the repository as a research enabler.

Document-Centric Design

The first version used standard repository publication objects to store the Twitter data (figure 1a). The user creates a new repository record with bibliographic metadata as normal, but 'uploaded' a

[1] Savage, M., & Burrows, R., (2007) 'The coming crisis of empirical sociology' Sociology, 41(5) pp.885-899

[2] DCC case study http://www.dcc.ac.uk/news/challenges-managing-social-media-research-data-researchers-perspective

[3] Steve Reicher and Cliff Stott (2011) *Mad Mobs and Englishmen? Myths and Realities of the 2011 Riots* Constable & Robinson: London.

[4] www.dtc.webscience.ecs.soton.ac.uk

document to the record whose source method is marked specially as 'Twitter' (with an appropriate hashtag for searching and a date to cease the twitter search). On depositing the newly created record into the repository, a background harvesting process performs a periodic Twitter query and attaches any new results to the document as an additional XML file. A separate rendering process periodically batches up the results to create a set of human-readable HTML pages. This included:

- Clickable links, referencing the ultimate destination of the short URL.
- Clickable #hashtags and @names, referencing the Twitter pages for each.
- The datestamp, tweet ID and tweeter (username and profile image).

Each single HTML page contained 1000 tweets and an index page was provided to browse through the pages of content. The XML files were available for download for further processing.

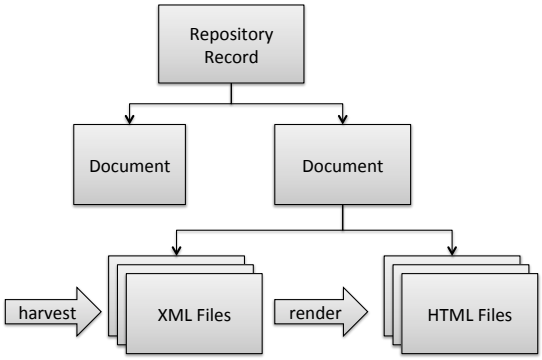

Figure 1a: Initial Repository Design for Tweet Management

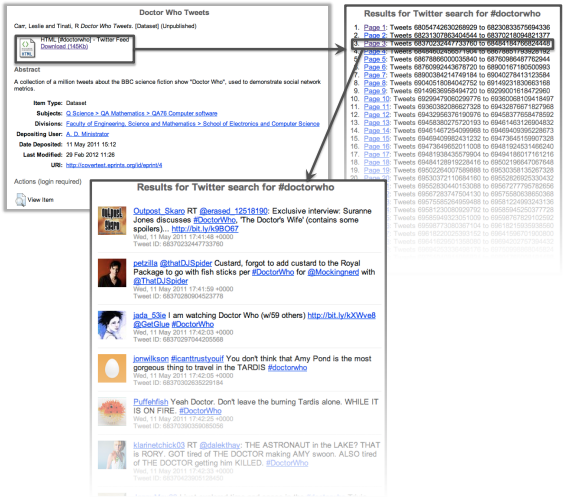

Figure 1b: Rendered Twitter Records

Object-Centric Design

Our second design leveraged EPrints support for bespoke first-class objects *i.e.* data objects that aren't publications (figure 2). A TweetStream class was designed to store the search terms and act as an interface to a collection of tweets and a Tweet class stored all metadata associated with an individual tweet. Each Tweet object contained a reference to each TweetStream to which it belonged (note that some tweets appear in more than one stream).

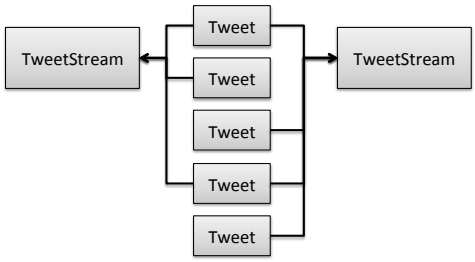

Figure 2: Tweet / TweetStream Data Objects

EPrints' built-in 'Manage Records' infrastructure, which was introduced for the management of bespoke system objects, enabled much of the required functionality (e.g. Tweet modelling, rendering, search and transformation) to be gained with minimal development time. The Tweet object was designed around the metadata values returned by the Twitter API. The text of the tweet was parsed for significant data including URLs, #hashtags and usernames, which were stored in metadata fields. All shortened URLs were followed and their ultimate destinations were stored to preserve the address of the linked resource.

The Tweetstream object acted an interface for a collection of tweets with a set of search terms in common. Two key metadata fields controlled the harvesting – a set of search terms and a date at which to stop harvesting. Once harvesting started, aggregate data from the tweets belonging to the tweetstream would be stored for rendering on the tweetstream's 'abstract page' (figure 3). This included:

- The total number of tweets
- A selection of the tweets
- A list of the top tweeters, #hashtags, @names and links
- A frequency graph of tweets over time

This aggregation of tweet data was performed nightly. These simple additions serve to give an overview of the data to the researcher – not to try to take the place of their bespoke data analysis software, but simply to help understand some of the major features of the data as it is being collected.

Figure 3: Tweetstream Render

EPrints' export plugin infrastructure made development of export formats for tweetstreams simple. The system currently exports Twitter streams as:

- JSON
- HTML (human readable)
- CSV

The CSV export has proved extremely popular with the social sciences students who want to make use of this data with external systems.

Repository Issues

The key issues that we encountered in this work all derive from the fact that EPrints is designed around assumptions of scale that do not hold true for unbounded, large-scale, crowd-created datasets.

The key technical problem in the first version was that EPrints checks all files associated with a document before displaying them in the EPrints workflow, which was used as the interface through which the XML files were downloaded. Harvesting was done hourly, and so a new XML file was created every hour. Every 1000 tweets would result in an additional rendered HTML file. After several months, when there may be several thousand files in a tweetstream document, EPrints became unresponsive to the user and processor-intensive on the server when the upload stage of the workflow was requested. These problems started to happen when there were half a million tweets in half a dozen separate Twitter documents before the system became problematic.

The second design iteration fared well for another order of magnitude. Problems started when there were about 5 million tweets in around two dozen tweet streams. The issue was a core design assumption that the EPrints search API would always be able to return a list of the all IDs of the objects in the search results and *that list of IDs would always fit in the available memory*. In the test system (a virtual machine running with 'normal' repository resources) this turned out not to be the case.

A functional search was implemented using 'offset' and 'count' parameters to page through the results, but performance remains less than adequate. More development is required to create an efficient, quick, memory-safe search. We are currently discussing this with the EPrints Development team.

Ingest Performance Issues

Each tweet needs to be processed on ingest. Trivially, a number of important metadata fields are read from the JSON source and stored in the tweet object. Significantly, a number of regular expressions are run on the text of the tweet to extract #hashtags, @names and URLs. Finaly, each extracted URL is followed to enable the preservation of the forwarding addresses (sometimes though multiple redirects). This last step takes a significant amount of time.

On our prototype, the logs of the nightly job that processes URLs and performs tweet enrichment show that a rate of 10,000 per hour is rarely exceeded. We consider this rate to be far too slow. Our pilot system is currently harvesting 100,000 tweets per day, and it takes about 10 hours to process the results every night. However, we consider our pilot repository to be extremely small-scale. It is not hard to imagine a production system capturing millions of tweets every day[5]. We are currently investigating the possibility of a light-weight concurrent process outside of EPrints to handle URL processing and cacheing, and its likely further optimisations will need to be engineered into the system.

Summarising Streams

Aggregation of tweet data is performed nightly. The first version of this process iterated across all tweet objects, but that approach was found not to scale well. We replaced this with a set of direct queries to the database. We do not expect to have to further optimise this until we reach a tipping point at a much higher order of magnitude due to the limitations of the database itself.

Twitter API

We are currently using the Twitter search API while we continue exploring the issues surround the storage of large-scale sets of fine-grain objects in repositories. The search API is not guaranteed to return all results for a set of search terms. Furthermore, peaks in tweet frequency (e.g. during the oscars) can make it impossible to collect all tweets within a given time period. To collect more complete datasets, we intend to use the Twitter stream API, which will require a process constantly receiving data from Twitter and creating EPrints objects in real-time. We expect this to require significantly more software engineering than our current system, but we anticipate an interesting challenge exploring the issues that will surround this.

Conclusions

Apart from direct use of the API, alternative Twitter capture services are available, but are often aimed at commercial social analytics customers (Datasift, TwapperKeeper/Hootsuite). Our aim in this work is to provide a sufficiently useful platform to facilitate the research of small to moderate users (e.g. postgraduate students) who do not have the technical knowhow, the time or the funding to obtain a more comprehensive service. It is also our intention to extend the Twitter facilities to enable ongoing harvesting of data from other social services (e.g. Flickr, YouTube, Wordpress, Blogger etc), to make the repository an acknowledged source of research data and research utility, as well as the ultimate sink of research publications.

[5] http://blog.Twitter.com/2011/08/your-world-more-connected.html