First Monday

Volume 17, Number 4 - 2 April 2012

Table of Contents

Book Reviews

A Great Cities Initiative of the University of Illinois at Chicago University Library.

OPEN JOURNAL SYSTEMS

Journal Help

USER
Username
Password
Remember me
Log In

JOURNAL CONTENT
Search
All
Search

Browse
- By Issue
- By Author
- By Title
- Other Journals

INFORMATION
- For Readers
- For Authors
- For Librarians

First Monday, Volume 17, Number 4 - 2 April 2012

first monday
PEER-REVIEWED JOURNAL ON THE INTERNET

Tools and methods for capturing Twitter data during natural disasters

by Axel Bruns and Yuxian Eugene Liang

Abstract

During the course of several natural disasters in recent years, Twitter has been found to play an important role as an additional medium for many–to–many crisis communication. Emergency services are successfully using Twitter to inform the public about current developments, and are increasingly also attempting to source first–hand situational information from Twitter feeds (such as relevant hashtags). The further study of the uses of Twitter during natural disasters relies on the development of flexible and reliable research infrastructure for tracking and analysing Twitter feeds at scale and in close to real time, however. This article outlines two approaches to the development of such infrastructure: one which builds on the readily available open source platform yourTwapperkeeper to provide a low–cost, simple, and basic solution; and, one which establishes a more powerful and flexible framework by drawing on highly scaleable, state–of–the–art technology.

Contents

Introduction

The role played by social media in the coverage of natural disasters as well as in the mobilisation of affected locals and volunteers is increasingly being recognised (*e.g.*, Liu, 2009; Liu, *et al.*, 2008; Mark and Semaan, 2008; Mendoza, *et al.*, 2010; Shklovski, *et al.*, 2008; Sutton, *et al.*, 2008). Twitter, in particular, lends itself well to these tasks, due to its flat and flexible communicative structures: users interested in specific topics can easily find one another through the rapid and ad hoc establishment of shared hashtags related to the crisis event (keywords, prefixed with the hash symbol '#', which users can include in their tweets to make these messages visible to others following the hashtag). Such hashtags provide a mechanism for conversation and update threads between users even if these users are not already 'following' one another in the social network; indeed, hashtag streams may even be followed by visitors to the Twitter Web site who are not themselves registered Twitter users (Bruns and Burgess, 2011a). Additionally, the comparatively simple network structure of the Twitter platform (where accounts are either 'public' (visible to all, and even to non–registered visitors) or 'private' (visible only to followers approved by the author) means that topically relevant tweets from public accounts can be found and reshared very widely — for the purposes of crisis communication, this compares favourably for example with the communicative structures of Facebook, where more complex visibility permissions mean that messages will not normally travel far beyond a user's immediate circle of friends, or friends of friends.

This significant suitability of Twitter as a flat and open communication medium for crisis communication has led to its playing an important role in a number of recent natural and human–made crises and disasters, ranging from the 2011 floods in the Australian state of Queensland (Bruns, *et al.*, 2012) through the three major earthquakes in Christchurch, New Zealand, during 2010 and 2011 (Bruns and Burgess, 2011b) to the massive earthquake and tsunami in Japan in March 2011; earlier uses during previous crises and disasters have also been observed (Hughes and Palen, 2009; Mendoza, *et al.*, 2010; Palen, *et al.*, 2010; Starbird and Palen, 2010). Additionally, while claims of 'Twitter revolutions' are likely to overstate its importance, the use of Twitter for public information and coordination has also been noted for the uprisings of the 2011 'Arab Spring' as well as, more controversially, for the 2011 riots in London and the overall United Kingdom.

Research into the use of social media (in general) and Twitter (in particular) during these and similar crises, disasters, and other acute events (Burgess and Crawford, 2011) has proceeded from a number of disciplinary and methodological bases. To develop a more comprehensive and reliable foundation for such research activities in the future, however, and to improve the comparability of their findings, it is necessary to share the emerging tools and methods for systematic Twitter research more widely and more openly than has previously been the case. Due to the recency of Twitter and other social media platforms, and the relative novelty of mixed–method, interdisciplinary approaches for the qualitative and quantitative study of 'big data' datasets drawn from social media platforms (boyd and Crawford, 2011), many extant studies employ custom–made research tools which are discussed only in passing and remain unavailable to other researchers; this undermines the replicatability and translatability of such studies to other, similar contexts. This paper, therefore, aims to begin a more systematic, open and ongoing conversation about Twitter research tools and methods (especially for the study of natural disasters and similar crises) by outlining the research approaches employed in a collaborative project involving Australian and Taiwanese researchers: by putting our cards on the table in this way, we hope to provide more detailed methodological background to our published and forthcoming work (see esp. Bruns, *et al.*, 2012; Bruns, 2011a; Bruns and Burgess, 2011a/b/c), and to thereby also enable other researchers to apply these methods to their own areas of research, to generate comparable datasets, and to replicate or challenge our findings.

The following discussion outlines two main approaches, then: first, we discuss a more limited (and thus more easily replicatable) method for the tracking and analysis of hashtag–based Twitter activities which builds on the open source tool yourTwapperkeeper (2011) and uses a number of additional tools to process and visualise Twitter activities; in a further section, we then describe a more comprehensive (but also more complex) method for capturing Twitter content which requires the development of custom–designed tracking and analysis tools.

Tracking Twitter through yourTwapperkeeper

The first challenge in doing research on the use of Twitter for crisis communication is to capture a comprehensive (or at least representative) sample of tweets which relate to the crisis event under investigation. One relatively simple and straightforward approach to this challenge is to focus on tweets which contain the relevant topical hashtag (or hashtags) related to the crisis: for the 2011 Queensland floods, for example, this was #qldfloods (with additional, adjunctive and sometimes overlapping discussion also taking place using #thebigwet; Bruns, *et al.*, 2012); for the Christchurch earthquakes, #eqnz (Bruns and Burgess, 2011b); for discussion of the Arab spring uprisings, hashtags referring to the countries in question (#egypt, #libya) or to specific events (*e.g.*, #25Jan — referring to 25 January 2011, the date commonly seen as marking the start of the Egyptian uprising) were common.

By tracking topical hashtags and capturing hashtagged tweets, we may assume to establish a dataset of the most visible tweets relating to the event in question, since it is the purpose of topical hashtags to aid the visibility and discoverability of Twitter messages, and since this is especially important in a crisis context (in this we distinguish topical hashtags such as #eqnz from other hashtag uses — *e.g.*, from emotive hashtags such as #facepalm or #fail; *cf.*, Bruns, 2011a). This does not mean that we are able to capture all messages relating to the crisis event or its implications, however; it cannot be ruled out (indeed, it is virtually guaranteed) that some users tweeting about the crisis will be unaware of the existence of the central hashtag, using a different hashtag variant, or even unfamiliar with the concept of hashtags altogether. (Some of these limitations may be addressed by tracking a wider range of relevant hashtags or other keywords, of course.) Additionally, anecdotal evidence also suggests that while hashtags may be used for the sharing of key information and opinion about the event, follow–on @reply conversations between participating users may well take place outside of the hashtagged stream of tweets (unless users specifically choose to again hashtag their public responses to one another, in order to give these messages greater visibility as well); further, of course, follow–on communication through private, direct Twitter messages or other communication media will also remain outside the scope of any research which can be conducted using the methods outlined here.

Twitter provides access to public tweets through two key elements of its Application Programming Interface (API): the search API and the streaming API. Of these, the former can be used to retrieve past tweets according to a range of criteria (including keywords/hashtags, senders, location, etc.), within set limits: in the first place, the search API will only return a limited number of tweets, and therefore cannot be used to retrieve a comprehensive archive of past tweets containing specific hashtags, for example; further, there are in–built limits on how many keywords or users can be queried at any one time or within set timeframes. Where the search API is focussed on past content, the streaming API, by contrast, can be used to subscribe to a continuing stream of new tweets containing specific keywords or originating from specific users or locations; here, too, however, significant limits on the number of users or keywords which can be followed do apply. (It should be noted that some such limits can be overcome, at a cost, by accessing the Twitter API through one of a number of third–party resellers of Twitter content.)

Given these limitations of the Twitter API, any research method which seeks to establish a reasonably comprehensive dataset of tweets related to a specific crisis event will need to begin tracking the event as it happens (that is, when keywords or hashtags relevant to the event first appear on Twitter), or otherwise it will risk missing these early tweets as they will eventually no longer be retrievable using the search API. Further, follow–up tweets must be captured either by using the streaming API to subscribe to an ongoing update feed of relevant tweets, or by regularly retrieving the latest past tweets through the search API. Even such retrieval methods cannot guarantee a comprehensive capture of Twitter data, however: outages on the side of server or client, or transmission problems between them, cannot be ruled out altogether, and may result in message loss. Further, there are very few reliable means of comprehensively cross–checking the dataset for its veracity, since the Twitter API constitutes the only point of access to the Twitter stream which is available to researchers. No dataset captured by using the Twitter API is guaranteed to be entirely comprehensive, therefore; especially where research focusses on identifying broad patterns in Twitter activity from a large dataset, however, such research nonetheless remains valid and important.

One solution for tracking hashtags and other keywords on Twitter in the manner described above is the open–source tool yourTwapperkeeper (2011). Building on PHP and MySQL, it draws mainly on the Twitter streaming API to track a number of keywords selected by its user, using the search API to fill any gaps which may exist in the data received from the streaming API. Data captured through the tool can be exported in a number of formats, and for each tweet contains the following data points retrieved from the Twitter API:

- **archivesource:** API source of the tweet (twitter–search or twitter–stream)
- **text:** contents of the tweet itself, in 140 characters or less
- **to_user_id:** numerical ID of the tweet recipient (for @replies)
  (not always set, even for tweets containing @replies)
- **from_user:** screen name of the tweet sender
- **id:** numerical ID of the tweet itself
- **from_user_id:** numerical ID of the tweet sender
- **iso_language_code:** code (e.g. en, de, fr, …) of the sender's default language
  (not necessarily matching the language of the tweet itself)
- **source:** name or URL of the tool used for tweeting (*e.g.*, Tweetdeck, …)
- **profile_image_url:** URL of the tweet sender's profile picture
- **geo_type:** form in which the sender's geographical coordinates are provided
- **geo_coordinates_0:** first element of the geographical coordinates
- **geo_coordinates_1:** second element of the geographical coordinates
- **created_at:** tweet timestamp in human–readable format
  (set by the tweeting client — inconsistent formatting)
- **time:** tweet timestamp as a numerical Unix timestamp

yourTwapperkeeper is the open source version of a platform previously made available at Twapperkeeper.com, to enable researchers to track, archive, and share datasets of tweets relating to various keywords. Following an intervention by Twitter, that platform functionality is now no longer publicly available, but Twapperkeeper's data format — which did not include the 'archivesource' data point — has become a quasi-standard for tweet datasets. Bruns (2011b) provides an extension of yourTwapperkeeper which enables it to export Twapperkeeper–compatible datasets in comma– and tab–separated value formats (CSV/TSV).

In itself, however, yourTwapperkeeper only provides the means to capture tweet datasets on specific topics; any analysis of these datasets must rely on additional tools. Here, we may distinguish between three broad areas of further analysis: general statistical analysis and activity metrics, network analysis, and textual analysis. Different tools must be used for each of these areas.

Tweet statistics and activity metrics

The calculation of statistics and metrics describing the Twitter activities captured in a given dataset relies mainly on processing these datasets to count and compare specific communicative patterns; further filtering of datasets for specific timeframes, users, or keywords may also be necessary. Building on the data tables which may be exported from yourTwapperkeeper in various formats, such processing can be achieved using a variety of tools (such as the statistical processing language R, or to some extent even through standard spreadsheet software); our own approach has utilised the open–source command–line tool Gawk (2011), which uses a simple but flexible scripting language that can be used to process CSV/TSV–format files (a package of Gawk scripts is available at Bruns and Burgess, 2011d; these can easily be translated into R or other processing languages). Finally, the results of such data processing may be visualised in common spreadsheet software, or through other tools which enable the generation of standard chart types.

While a detailed discussion of possible Twitter data metrics which can be obtained through this approach would be well beyond the scope of this article, we provide a brief overview of the range of metrics which are possible here:

- time–based series:

  - overall volume of tweets over time
  - volume of different types of tweets over time (original tweets, @replies, unedited retweets, edited retweets, tweets containing URLs, etc.)
  - volume of specific keywords (or keyword bundles over time)
  - number of users active during any one time period (day, hour, minute)
  - average number of tweets per user during any one time period

- user–based metrics:

  - distribution of activity across the userbase, from heavy or lead users to casual and random participants (often a 'long tail'–style distribution)
  - activity by specific users or user groups over time (also separated into different tweet types)
  - activity profile for specific users or user groups (*e.g.*, percentage of different tweet types amongst their total output)
  - distribution of user visibility, as measured by the number of @replies and/or retweets received (also often a 'long tail'–style distribution)

- other content metrics:

  - most prominent keywords
  - most prominent URLs (full URLs, and/or domains only)

Further, more complex combinations between these metrics can also be developed, of course; for example, it would be possible to calculate, for individual users or larger groups of users, what most keywords or URLs are most prominent in their tweets. For groups of users identified through network analysis (discussed below), or for known groups of 'official' or otherwise notable accounts, this may reveal important differences in their information sources, language, or communicative style. Additionally, it may also be useful to group users by their total number of contributions into lead users, active users, and less active participants (following to the widely used 1/9/90 distribution), and to examine the tweeting patterns of these three groups to explore any differences in their Twitter usage.

Network analysis

Data processing tools such as Gawk may also be used to extract network data from the Twitter dataset. Here, too, a number of different networks, which we outline below, may be distinguished; additionally, due to the time–bound nature of Twitter datasets, for any such networks it is also possible to generate network analyses and visualisations which take into account the changeability of these networks over time (see *e.g.*, Bruns, 2011a, for a discussion of how to generate and visualise the dynamics of network data on @reply and retweet interactions). To analyse and visualise network data, a number of further software tools are readily available; often such tools also implement a range of different visualisation algorithms. A discussion of the relative merits of these tools and algorithms is well beyond the scope of this article, but we do stress that it is important for researchers to consider their choices in these matters, rather than to treat the network visualisation tool as a simple and unproblematic 'black box' technology; what specific choices are made in visualising network data can have substantial impact on the eventual output, and on the interpretations of that output. Our own work in this area has largely employed the powerful and flexible open source network visualisation software Gephi (2011), but we acknowledge that many other alternatives exist.

Network analysis approaches can similarly be separated into a number of different approaches (and as noted, for each of these networks, further distinctions between static and dynamic analyses and visualisations can be made, but are not listed separately here):

- homogenous networks:

  - user–to–user messaging networks (aggregate, or for specific tweet types: @replies or retweets only)
  - keyword co–occurrence networks (which keywords commonly occur together in tweets)

- heterogeneous, hybrid networks:

  - user–and–URL networks (which users share which URLs, at full URL or at domain level)
  - user–and–keyword networks (for a select list of keywords: which users refer to which keywords)
  - user–and–hashtag networks (for multi–hashtag datasets: which users participate in which hashtags)

Further, even more complex hybrid networks can also be developed, depending on the specific focus of the Twitter dataset under investigation; for any such network, whether simple or complex, a wide range of further analytical tools are also available to describe the network properties of specific nodes or groups of nodes, of course. So, for example, depending on the exact nature of the map, node activity or visibility may be measured by identifying the nodes (*e.g.*, users) with the most outbound (*e.g.*, sent @replies) or inbound connections (*e.g.*, received retweets); node importance may be described by calculating various betweenness or centrality measures; separate communities of users or themes of discussion may be determined by identifying clusters and divisions in the network.

Content analysis

Finally, another important analytical approach focuses specifically on the textual content of the tweets. While at a maximum length of 140 characters, tweets necessarily represent a highly compressed textual format, they nonetheless contain enough information for researchers to be able to extract a significant amount of valid information; some of that information also provides input to the analytical approaches outlined in the previous two sections, in fact.

Content analysis of tweets proceeds mainly by counting the key words, terms, and phrases used in those tweets (variously focussing on the complete dataset, or on tweets made during specific time periods or by particular users or groups of users); additionally, it is also possible to track the extent to which any such terms or phrases occur together (either in the same tweet, or in tweets by the same user). Common 'stop words' — generic terms such as 'and', 'for', 'if', etc. — are usually ignored in such analyses; where the dataset is defined in the first place by the presence of a specific hashtag or keyword, that keyword itself must also be ignored, of course. On the basis of such counting and tracking, a number of observations can then be made:

- overall distribution of keywords:
  - most used keywords or phrases overall
  - frequent keyword patterns for specific users or user groups
  - frequent keyword patterns for specific time periods (*e.g.*, each day or hour)
- occurrence over time:
  - rise and fall of keywords or keyword bundles over time
  - rise and fall of keywords or keyword bundles over time, per user or group
- co–occurrence:
  - interrelationships between keywords or phrases (may also be used to determine keyword bundles to be tracked in more detail)

In this context, it is also important to consider the impact which retweeting practices, in particular, may have on these analyses. A widely retweeted message will necessarily result in the words which constitute that tweet occurring together more frequently; especially for very prominent retweets, such patterns may come to overshadow all other co–occurrence patterns, so that any analysis which takes retweets into account will do little more than highlight the most retweeted messages.

It may be necessary, therefore, to consider only original tweets and @replies in such content analysis, ignoring retweeted content altogether. At the same time, retweets are prominent for a reason, and to ignore them completely may end up undercounting recurring connections between keywords, of course: if retweets are dismissed from the analysis, an obscure connection between two terms which are used together only once in the entire dataset (say, 'bad climate') is now accorded the same weight as a connection between two terms which occur in a prominent retweet (say, 'climate change'). Ultimately, an acceptable solution may require a compromise which weights co–occurrence through retweets less strongly than mere counting of each retweeted instance would do, but still more strongly than if retweets were ignored altogether.

More generally, these considerations also highlight the fact that especially in the context of content analysis, quantitative approaches alone are often merely a useful starting point. Especially where the content of tweets is concerned, further qualitative analysis and interpretation, and possibly also a formal coding of tweets according to their tone, theme, tenor, or other factors which cannot easily be identified by automated means alone, is likely to be necessary.

Other analytical approaches

In addition to these three major areas of analysis, it should also be noted that our discussion above has focussed mainly on the most important data points available from Twitter; it would also be possible, of course, to add to the analysis elements such as the Twitter client used for each tweet (*e.g.*, the Twitter Web interface, or a specific mobile of desktop client), the geolocation provided (if any; anecdotal evidence suggests that only a very small percentage of users provide such details with their tweets), or the language code which Twitter users have chosen (this is set in the user profile, however, and does not change with each tweet). Such data points may well be important especially in crisis communication–related research: for example, it may be interesting to distinguish tweets made from mobile devices, or to separate out tweets made by speakers of a language other than that used in the immediate disaster area.

Further approaches could also combine the data available immediately from yourTwapperkeeper with other data sources, of course, and explore further avenues for hybrid analysis (taking into account information about follower/followee networks on Twitter, for example). A discussion of such more complex, multi–source approaches is well beyond the scope of the present paper, however, especially also because these additional sources will usually be highly idiosyncratic and project–specific.

Beyond Twapperkeeper

The approaches we have discussed so far are valid and useful especially for the retrospective study of single–hashtag (or more broadly, single–keyword) datasets, and the methods used to conduct such analyses are well within the grasp of most media and communication researchers. However, for more sophisticated research programmes, and for the tracking and study of larger–scale datasets over longer time periods, more advanced and usually custom–made tools and methods are required. In the following discussion, we therefore sketch out the features of a more comprehensive Twitter tracking mechanism which advances well beyond what out–of–the–box solutions such as Twapperkeeper and yourTwapperkeeper are capable of.

In general, the research issues faced in the development of more advanced, custom tools for capturing and analysing Twitter data fall into three broad categories:

1. Dealing with the Twitter API
2. Scalability issues
3. Timeliness

Dealing with the Twitter API

While Twitter provides a comparatively open API for developers, using the Twitter API requires us to overcome various issues, including:

1. Throttling and data limitation issues
2. Historical data issues
3. Geographical data issues

Throttling and data limitation issues

Twitter controls third party developers' access by providing them with a personalised API key, through which the company tracks the usage of its API. In addition, Twitter also throttles access to its API per IP address: repeated, authenticated API requests coming from the same IP address may face throttling issues if they reach the API limit of 350 requests per hour (Twitter, 2011a); non–authenticated API access is limited to 150 requests per hour. Connections to any of Twitter's API endpoints are counted towards these API requests (Twitter, 2011b). yourTwapperkeeper, too, is subject to these limitations.

As noted above, Twitter provides two main APIs through which tweets may be retrieved: a search and a streaming API. Of these, the streaming API (Twitter, 2011c) enables applications to retrieve tweets in close to real time. Various API access methods may be used to retrieve tweets through the streaming API: the yourTwapperkeeper approach outlined above, for example, uses the 'statuses/filter' method which retrieves all tweets matching a given set of keywords provided by the application. Other approaches, which would enable us to move beyond this keyword filtering approach and capture a greater range and volume of tweets, include 'statuses/sample' (which returns a random sample of one percent — at 'Spritzer' level — or 10 percent — at 'Gardenhose' level — of all tweets being made; see (Twitter, 2011d) and 'statuses/firehose' (which returns a full feed of all tweets). However, the 'Spritzer' or 'Gardenhose' samples contain only a very rough and potentially unrepresentative sample of total current Twitter activity, while the 'Firehose' is "not a generally available resource" (see Twitter, 2012b), ruling our both of them for our purposes.

A different option, therefore, is to utilise Twitter's search API (Twitter, 2012a), due to its flexibility and predictability. The search API allows us to retrieve both recent and mixed (*i.e.*, recent as well as popular) results, which may be more useful in a study of natural disasters. In addition, compared to the higher–volume streaming API methods, the search API provides better control over the amount of data to be retrieved. Most importantly, the rate limit of 350 (or 150) requests per hour does not apply to search API requests; it is rate–limited by the IP address of the requesting client (Twitter, 2011e). The rate limit of 350 requests are governed by an API provided by Twitter; all requests coming through this particular API are counted towards the rate regardless of the IP address. However, all search API requests are anonymous and do not require API credentials.

Historical data issues

The Twitter API does not provide any reliable means to comprehensively retrieve historical tweets. The search API does provide access to past tweets, but reaches back to cover only between six to nine days' worth of tweets, at the point of writing (Twitter, 2011e). In practice, it should also be noted, searching for historical tweets often leads to less than satisfactory results. For the researcher, this means that on the onset of a natural disaster, it is necessary to respond almost immediately to track the related tweets; otherwise, data on the early hours of a crisis may be missing from the dataset.

Geographical data issues

Twitter does not allow applications to retrieve tweets from a specific geographic location on the basis of the stated location or geo–IP of a user (*e.g.*, tweets from Australian or Taiwanese users); the only mechanism it provides for retrieving geographically relevant tweets is to specify latitudes, longitudes and radius parameters in search requests. However, evidence from our research to date suggests that only a very small percentage of tweets are encoded with geographic metadata; this means that only a small (and likely highly unrepresentative) sample of tweets from the target geographic region will be retrieved using this method.

This means that — in the absence of reliable means for limiting data retrieval to specific geographic areas — tweet datasets cannot be easily confined to certain geographic areas. Even more elaborate methods for retrieving tweets through a combination of various approaches may be able to be developed — but such more complex approaches in turn suffer from scalability issues in storing and computing the data.

Scalability issues

Scalability issues result from the relatively large amounts of data that we need to collect and compute, and from the limited resources available for doing so. Such issues emerge in two areas: storage space and computing power.

Storage space

Even in spite of the very limited size of tweets themselves, at a maximum of 140 characters, once the attendant metadata are added, Twitter datasets comprising several hundreds of thousands (or even millions) of tweets can quickly reach significant volume. Further, while we may often think of storage space as a static resource, in the context of tracking social media activities we will eventually face a decline in available storage space as we collect data on a continuous basis; disk space will fill up as the amount of data grows. Therefore, we need to design our tools such that they are able to continuously increase their storage space as the need for space increases. Further, modes of storage must also be considered: yourTwapperkeeper, which draws on a basic MySQL database platform, does not scale especially well as it inherits the limitations of MySQL itself; storing, retrieving, and exporting selected data from MySQL databases several gigabytes in size can be a very time–consuming process (Cattell, 2010).

Computing power

Similarly, we also require our tools to scale computationally as our dataset increase in size; as we analyse our datasets, the greater the amount of data, the greater the amount of computing power is required.

Our solution to such scalability issues is through cloud computing and the use of NoSQL databases (Stonebraker, 2010), which are designed to scale horizontally (increasing our storage space and computing power) as and when we need it. Most importantly, by drawing on such technologies, there are minimal disruptions to our research infrastructure as we scale the storage space and the computing power required.

Timeliness issues

Further, as we aim to provide tools which can provide timely reports even while the crisis event is still unfolding, we need to be able to aggregate and analyse data as quickly as possible. The approach outlined above, using yourTwapperkeeper, Gawk, and Gephi as tools for data capture and analysis, continues to rely on analytical processes which are driven by the researcher, who must manually download and process the datasets gathered by yourTwapperkeeper as and when they feel it is appropriate to do so. A simple solution for generating reasonably up–to–date analytics using this approach is to aggregate the results of the data analysis on a regular (*e.g.*, daily) basis; this does mean, however, that there will be a time lag between capturing social media data and disseminating the results of the analysis.

While such lags may be acceptable in many contexts, especially during rapidly unfolding crisis events it would be preferable to aggregate and analyse data automatically and in real time, and to disseminate outcomes of the analysis as soon as they come to hand. True real–time processing may be costly given limited resources, however. For now, therefore, our solution to this issue is to deal with incoming data automatically, in predefined batches of material.

An advanced system for analysing tweets

To address these challenges, the following sections outline the overall structure of a system for capturing and analysing thematically relevant tweets in close to real time, which we have developed. The system is designed as a Web–based tool, enabling users to track and analyse data in an interactive fashion. End users log in to the Web application and enter their desired thematic keywords or phrases; once the application receives the request, it begins to capture relevant tweets using Twitter's search API at regular intervals specified by the user. Tweets are saved to a Data Store; for every set amount of tweets collected, the system's Analytics Engine processes these collections of tweets using a specified range of algorithms. Results are similarly stored in the Data Store, and published by the system.

The operation of the system can be divided into three main phases, then:

1. Data collection
2. Data analysis
3. Results publication

Data collection

Inputs from an operator are required to initiate the data collection process. Required inputs from an end user include:

- keywords/phrase, *e.g.*, "earthquake"
- language, *e.g.*, "en–US"
- result type: recent, mixed ("mixed" returns both popular and recent tweets)
- frequency of data collection, *e.g.*, every 15 minutes

Using the parameters given as examples above, for example, the system would perform a search for tweets containing the keyword "earthquake" every 15 minutes. Only tweets from users who set the language code of "en–US" will be captured, and only unique tweets will be collected per keyword. Unique tweets are identified by their tweet IDs.

For example, assuming we are collecting the keywords "earthquake" and "japan" as separate keywords during a Japanese earthquake, it is likely that we will collect tweets that contain both terms; such tweets will be included in both collections.

Data analysis

Data are analysed incrementally by the system: tweets that have already been analysed are marked, while new tweets will be processed for analytical purposes. This helps us deal with scalability issues in terms of computing power, as tweets that has already been analysed will not be processed again; the results from each analysis will be aggregated to the results. This also helps us update results in a much predictable and stable manner.

The following key metrics are extracted from tweet datasets:

- Frequency over time:

  - tweets
  - users
  - keywords
  - replies
  - retweets

- Changes of Interest over time:

  - changes in the prominent use of different keywords or phrases

Results publication

Especially in the context of natural disasters and similar crises, rapid results publication will often be necessary. The system is designed so that graphs presenting the results of the analysis can be retrieved speedily from the Results Database in the Data Store, using a Web interface.

System architecture

A system as outlined above can be built cost effectively on the basis of several open source technologies:

- Server

- Ubuntu Server 10.04
  (http://releases.ubuntu.com/lucid)

- Database

- MongoDB
  (http://www.mongodb.org/)

- Programming/Scripting languages:

- Python
  (http://www.python.org/)
- JavaScript/HTML/CSS

- Other packages/libraries

- Natural Language Toolkit
  (http://www.nltk.org/)
- Matplotlib
  (http://matplotlib.sourceforge.net/)
- NetworkX
  (http://networkx.lanl.gov/)
- Tornado Web Framework
  (http://www.tornadoweb.org/)

Further, the entire system is built on top of Amazon Web Services (and related services) for ease of scalability. The high level architecture of our tool is as follows:

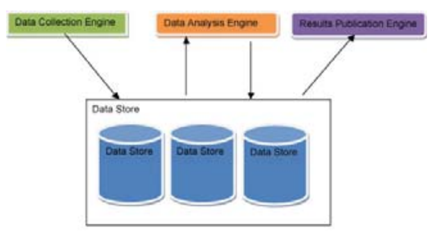

Figure 1: High–level architecture of the Twitter capture tool.

The system is divided into four major components, matching the major elements outlined above:

1. Data collection engine: retrieves data from Twitter
2. Data store: stores tweet datasets, and analysis results
3. Data analysis engine: analyses tweet datasets to generate key metrics.
4. Results publication engine: publishes analysis results from data store as graphs

Data store

The data store is designed to scale horizontally to deal with increasing amounts of data. What makes the data store scalable is its use of cloud computing infrastructure such as Amazon Web Services; we have used the Elastic Compute Cloud solution (Amazon Web Services, 2012). Further, the system avoids traditional database solutions (such as MySQL and PostgreSQL) and instead uses NoSQL databases: a class of database solutions which are defined by their superior scalability.

Compared to traditional database solutions, NoSQL databases enable us to avoid extra development work; we selected MongoDB due to its ease of use in terms of scalability and features (Stonebraker, 2010). MongoDB supports auto–sharding, which means data can be stored in different physical servers with minimal additional programming work. We are able to add in new physical servers to store more data without significant effort. Traditional database solutions — such as MySQL — would require substantially more software development and advanced planning in order to store data across different physical servers.

Setup of the data store

Following the recommendations made by MongoDB (2011b), the data store is designed to distribute data across multiple servers. The following diagram illustrates the set up of the data store:

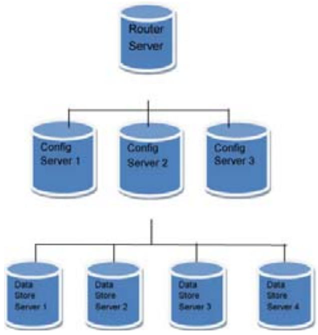

Figure 2: Data store architecture.

Each of the object above represents a server: in our case, an Amazon EC2 Server instance running on Ubuntu 10.04 Server Edition.

Scaling up the data store

Assuming we are running out of disk space, we can scale up the data store by simply adding in new EC2 instances (with persistent storage):

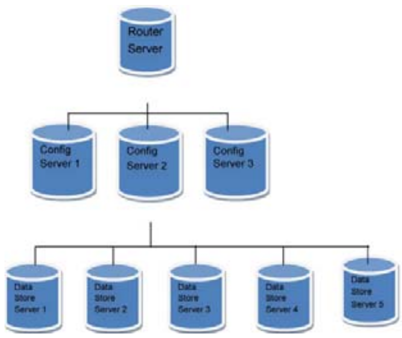

Figure 3: On–the–fly expansion of the data store architecture.

In the diagram above, our data store is scaled horizontally by adding a new data store server; after adding the IP address of the new server at the router server, data collected will be distributed to the new server as well.

Data analysis engine

Intuitively, it may seem more logical to collect all currently available data before performing any data analysis. However, if new data are added subsequently, the whole process would then need to be performed again for the entire dataset (including both old and new data). This results in a waste of computing resources, though, since analysis which had already been performed on the old data would now be performed again.

Our approach to data analysis, on the other hand, is designed to minimise the use of computing resources by processing data in batches: only tweets that have not yet been analysed will be processed. This process is repeated for every new batch of tweets, and the results of each step in the data analysis are saved to the results store as collections in a MongoDB database (MongoDB, 2011a). Collections are similar to a "table" in a database; they constitute a named grouping of documents.

The collections are organized as follows:

DataFreqPerSecondDB	— overall volume of tweets, per second
DataFreqPerMinuteDB	— overall volume of tweets, per minute
DataFreqPerHourDB	— overall volume of tweets, per hour
DataFreqPerDayDB	— overall volume of tweets, per day
UserFreqPerSecondDB	— overall volume of tweets by a specific user, per second
UserFreqPerMinuteDB	— overall volume of tweets by a specific user, per minute
UserFreqPerHourDB	— overall volume of tweets by a specific user, per hour
UserFreqPerDayDB	— overall volume of tweets by a specific user, per day
RTFreqPerSecondDB	— overall volume of retweets, per second
RTFreqPerMinuteDB	— overall volume of retweets, per minute
RTFreqPerHourDB	— overall volume of retweets, per hour
RTFreqPerDayDB	— overall volume of retweets, per day
TweetsLinkFreqPerSecondDB	— overall volume of tweets with links, per second
TweetsLinkFreqPerMinuteDB	— overall volume of tweets with links, per minute
TweetsLinkFreqPerHourDB	— overall volume of tweets with links, per hour
TweetsLinkFreqPerDayDB	— overall volume of tweets with links, per day

KeywordFreqPerSecondDB	— overall volume of a specific keyword, per second
KeywordFreqPerMinuteDB	— overall volume of a specific keyword, per minute
KeywordFreqPerHourDB	— overall volume of a specific keyword, per hour
KeywordFreqPerDayDB	— overall volume of a specific keyword, per day

Automatic generation of activity metrics

On the basis of these collections, a range of metrics which describe the incoming Twitter data may be generated through automatic analysis. Overall, these address two major areas: volume patterns and content patterns.

Tweet volumes

Metrics describing tweet volumes indicate the overall frequency of Twitter updates, across a range of categories. Frequencies are calculated for tweets, retweets, users, @replies, and tweets containing URLs.

As the timestamp of each tweet field encoded as a Python DateTime Object, calculating tweeting frequencies for any given period of time (seconds, minutes, hours, days) becomes straightforward when using MongoDB as the database solution. MongoDB supports a feature known as upserts (MongoDB, 2012), which means that a given ID's count is incremented when an exact replica for a certain field is found in the database. In other words, when two tweets share the same timestamp, the count of tweets for that timestamp is incremented automatically; when two tweets share the same originating user, the count of tweets for that user is incremented automatically; etc. For any set of tweets, then, it becomes relatively simple to generate frequency indices for each of the metrics outlined above.

Keyword volumes

Similar metrics can also be generated to describe the content of tweets. Similar to the frequency indices, keyword indices are based on the occurrence of keywords over time. They may be generated by processing incoming tweets as follows:

1. Punctuation marks are removed from each tweet.
2. Each tweet is split into its constituent words, using the Natural Language Toolkit (NLTK).
3. Stopwords are removed, using NLTK's stopword library.
4. All keywords are converted to lowercase.
5. Timestamps from the original tweet are assigned to each keyword.
6. Frequency indices are built for each of the keywords.

A tweet such as "OMG, Earthquake in Japan again!" would thus undergo the following transformations:

OMG Earthquake in Japan again	— punctuation removed
OMG, Earthquake, in, Japan, again	— split into keywords
OMG, Earthquake, Japan	— stopwords removed
omg, earthquake, japan	— conversion to lowercase

Finally for each of the keywords, a JSON data structure is created for insertion into the results database:


```
{
_id:1234
task_id:989,
keyword:"omg",
created_time:13 Mar 2011 22:44:43
}

{
_id:1234
task_id:989,
keyword:"earthquake",
created_time:13 Mar 2011 22:44:43
}

{
_id:1234
task_id:989,
keyword:"japan",
created_time:13 Mar 2011 22:44:43
}
```


As with the tweet volume metrics, it now becomes possible to draw in built–in MongoDB functionality to automatically generate keyword volume metrics which track the occurrence of keywords over time or per user. These data are saved into the result store.

Client and reporting interface

The client and reporting interface essentially provides access to graphical visualisations of the analyses contained in the results store. These graphs are updated after every new batch of tweets are analysed. Such visualisations can be performed by a range of available libraries, scripts or softwares. Since our tool is Web–based, we use JavaScript libraries for visualising the results. In addition, since bulk of our results is frequency–based, we most often use time series, line charts to visualise them.

Using this approach, for example, a visualisation of the volume of tweets on an hourly basis would proceed by accessing data from the DataFreqPerHourDB collection and outputting it to a Web page; the task of the visualisation library is to build the line charts from the data and display them to the user. We find that JavaScript libraries such as Flot (http://people.iola.dk/olau/flot/examples/) and HighCharts (http://www.highcharts.com/) work well for our purposes, due to their ease of use.

Conclusion

In this paper, we have presented two different approaches to the tracking and analysis of Twitter user activities, designed especially to be utilised in the study of the uses of social media during natural disasters, but applicable also in a much broader range of research projects. 'Big data' research into social media activities (on Twitter and elsewhere) constitutes a growing field of scholarly endeavour, and early results from this work have managed to generate a substantial amount of academic and general interest already. Detailed discussions of the methods and methodologies of such research projects still remain few and far between, however, and data gathering and analysis tools, to the extent that they are readily available at all, are all too often treated uncritically as mere 'black box' tools which do the necessary job but require no further discussion.

We offer this paper as a contribution to the urgent task of exploring available (and potential) methodological solutions to the study of Twitter in general as well as in the specific context of acute crisis events, and of problematising the data capture and analysis toolkits currently available to researchers. The two approaches we have outlined here — using our–of–the–box solutions such as yourTwapperkeeper, Gawk, and Gephi, or custom–made data capture and analysis infrastructure which builds on available open source platforms and technologies — are by no means perfect or universally applicable, but already do enable and support a wide range of important and innovative research projects. Further extension of these approaches and technologies, or their replacement with new, more advanced, and ideally open source research tools, remains necessary, and we hope that our own work in this area may encourage others to take up the challenge as well.

In closing, however, it should also be noted that as third–party researchers with no special relationship to Twitter itself, we continually operate in a precarious space which remains outside our control. Any change to the Twitter API, other relevant infrastructure, or the platform's terms and conditions may undermine or invalidate our work, requiring significant elements of our research tools and technologies to be redeveloped (or indeed ruling out specific approaches which had been possible previously). For example, Twitter's move to provide exceptions to its API access rate limits — previously available on request and granted on a case–by–case basis — only through third–party resellers such as Gnip, at a price point beyond the budgets available to most publicly funded researchers (see Melanson, 2011), served to stifle a substantial number of highly innovative research projects. The loss of such projects is Twitter's as much as it is the individual researchers', however: as is especially obvious in the context of research into crisis communication, where many recent studies have demonstrated the value of social media in informing affected populations and providing them with a platform to organise relief and recovery (see *e.g.*, Earle, *et al.*, 2010; Goolsby, 2010; Guy, *et al.*, 2010; Hughes and Palen, 2009; Mark and Semaan, 2008; Palen, *et al.*, 2010; Vieweg, *et al.*, 2010), research into the uses of Twitter frequently concludes by pointing to the significant public utility of the platform. By making such research more difficult in its push to extract revenue from its users, Twitter only ends up alienating some of its most visible allies, and reduces the number of good–news stories the company is able to tell about the service it provides; in thus preventing researchers from documenting how Twitter is used, the company is cutting off its nose to spite its face. But that is a discussion to be had in another paper.

About the authors

Dr. **Axel Bruns** is an Associate Professor in the Creative Industries Faculty at Queensland University of Technology in Brisbane, Australia, and a Chief Investigator in the ARC Centre of Excellence for Creative Industries and Innovation (http://cci.edu.au/). He is the author of *Blogs, Wikipedia, Second Life and beyond: From production to produsage* (2008) and *Gatewatching: Collaborative online news production* (2005), and the editor of *Uses of blogs* with Joanne Jacobs (2006; all released by Peter Lang, New York). Bruns is an expert on the impact of user–led content creation, or produsage, and his current work focuses especially on the study of user participation in social media spaces such as Twitter, especially in the context of acute events. His research blog is at http://snurb.info/, and he tweets at @snurb_dot_info. See http://mappingonlinepublics.net/ for more details on his current social media research.
E–mail: a [dot] bruns [at] qut [dot] edu [dot] au

Yuxian Eugene Liang enjoys solving business/social science problems with computer science. He is currently doing research at National Cheng Chi University, Taipei Taiwan. His research interests include social computing, data mining, machine learning, human computer interaction and social media marketing. He can be reached at http://www.liangeugene.com/.
E–mail: ye [dot] eugene [at] gmail [dot] com

References

Amazon Web Services, 2012. "Amazon Elastic Compute Cloud (Amazon EC2)," at http://aws.amazon.com/ec2/, accessed 10 January 2012.

danah boyd and Kate Crawford, 2011. "Six provocations for big data," paper presented at *A Decade in Internet Time: Symposium on the Dynamics of the Internet and Society*, Oxford Internet Institute (21 September), at http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1926431, accessed 20 March 2012.

Axel Bruns, 2011a. "How long is a tweet? Mapping dynamic conversation networks on Twitter using Gawk and Gephi," *Information, Communication & Society* (17 November), at http://www.tandfonline.com/doi/abs/10.1080/1369118X.2011.635214, accessed 20 March 2012.

Axel Bruns, 2011b. "Switching from Twapperkeeper to yourTwapperkeeper," *Mapping Online Publics* (21 June), at http://www.mappingonlinepublics.net/2011/06/21/switching-from-twapperkeeper-to-yourtwapperkeeper/, accessed 1 August 2011.

Axel Bruns and Jean Burgess, 2011a. "The use of Twitter hashtags in the formation of *ad hoc* publics," paper presented at the European Consortium for Political Research conference, Reykjavik (25–27 August), at http://snurb.info/node/1533, accessed 20 March 2012.

Axel Bruns and Jean Burgess, 2011b. "Local and global responses to disaster: #eqnz and the Christchurch earthquake," paper presented at the Association of Internet Researchers conference, Seattle (12 October), at http://snurb.info/node/1569, accessed 20 March 2012.

Axel Bruns and Jean Burgess, 2011c. "New methodologies for researching news discussion on Twitter," paper presented at the Future of Journalism conference, Cardiff (8-9 September), at http://snurb.info/node/1535, accessed 20 March 2012.

Axel Bruns and Jean Burgess, 2011d. "Gawk scripts for Twitter processing," v1.0, *Mapping Online Publics* (22 June), at http://mappingonlinepublics.net/resources/, accessed 4 January 2012.

Axel Bruns, Jean Burgess, Kate Crawford, and Frances Shaw, 2012. "#qldfloods and @QPSMedia: Crisis communication on Twitter in the 2011 south east Queensland floods," Brisbane: ARC Centre of Excellence for Creative Industries and Innovation, at http://cci.edu.au/floodsreport.pdf, accessed 12 January 2012.

Jean Burgess and Kate Crawford, 2011. "Social media and the theory of the acute event," paper presented at Internet Research 12.0 — Performance and Participation, Seattle (12 October).

Rick Cattell, 2010. "Scalable SQL and NoSQL data stores," *ACM SIGMOD Record*, volume 39, number 4, pp. 12–27, and at http://www.sigmod.org/publications/sigmod-record/1012/index.html, accessed 20 March 2012.

Paul Earle, Michelle Guy, Richard Buckmaster, Chris Ostrum, Scott Horvath, and Amy Vaughan, 2010. "OMG earthquake! Can Twitter improve earthquake response?" *Seismological Research Letters*, volume 81, number 2, pp. 246–251, at http://www.seismosoc.org/publications/SRL/SRL_81/srl_81-2_es/, accessed 20 March 2012.

Gawk, 2011. "Gawk," at http://www.gnu.org/software/gawk/, accessed 1 April 2011.

Gephi, 2011. "Gephi," at http://gephi.org/, accessed 1 April 2011.

Rebecca Goolsby, 2010. "Social media as crisis platform: The future of community maps/crisis maps," *ACM Transactions on Intelligent Systems and Technology (TIST)*, volume 1, number 1, at http://doi.acm.org/10.1145/1858948.1858955, accessed 4 January 2012.

Michelle Guy, Paul Earle, Chris Ostrum, Kenny Gruchalla and Scott Horvath, 2010. "Integration and dissemination of citizen reported and seismically derived earthquake information via social network technologies," In: Paul R. Cohen, Niall M. Adams, Michael R. Berthold (editors). *Advances in Intelligent Data Analysis IX. Lecture Notes in Computer Science*, number 6065. Berlin: Springer, pp. 42–53.

Amanda Lee Hughes and Leysia Palen, 2009. "Twitter adoption and use in mass convergence and emergency events," *International Journal of Emergency Management*, volume 6, numbers 3–4, pp. 248–260.

Sophia B. Liu, 2009. "Informing design of next generation social media to support crisis–related grassroots heritage, *Ph.D. Colloquium of the 6th International ISCRAM Conference* (Gothenburg), at http://sophiabliu.com/Publications.html, accessed 20 March 2012.

Sophia B. Liu, Leysia Palen, Jeannette Sutton, Amanda L. Hughes, and Sarah Vieweg, 2008. "In search of the bigger picture: The emergent role of on–line photo sharing in times of disaster," *Proceedings of the 5th International ISCRAM Conference*, at http://www.cs.colorado.edu/~palen/Papers/iscram08/OnlinePhotoSharingISCRAM08.pdf, accessed 20 March 2012.

Gloria Mark and Bryan Semaan, 2008. "Resilience in collaboration: Technology as a resource for new patterns of action," *CSCW '08: Proceedings of the ACM 2008 Conference on Computer Supported Cooperative Work*, p. 137.

Mike Melanson, 2011. "Twitter kills the API whitelist: What it means for developers & innovation," *ReadWriteWeb* (11 February), at http://www.readwriteweb.com/archives/twitter_kills_the_api_whitelist_what_it_means_for.php, accessed 4 January 2012.

Marcelo Mendoza, Barbara Poblete, and Carlos Castillo, 2010. "Twitter under crisis: Can we trust what we RT?" *SOMA '10: 1st Workshop on Social Media Analytics*, at http://research.yahoo.com/pub/3255, accessed 20 March 2012.

MongoDB, 2012. "Updating: Upserts with modifiers," at http://www.mongodb.org/display/DOCS/Updating#Updating-UpsertswithModifiers, accessed 10 January 2012.

MongoDB, 2011a. "Collections," at http://www.mongodb.org/display/DOCS/Collections, accessed 10 January 2012.

MongoDB, 2011b. "Sharding," at http://www.mongodb.org/display/DOCS/Sharding, accessed 10 January 2012.

Leysia Palen, Kate Starbird, Sarah Vieweg, and Amanda Hughes, 2010. "Twitter–based information distribution during the 2009 Red River Valley flood Ttreat," *Bulletin of the American Society for Information Science and Technology*, volume 36, number 5, pp. 13–17.

Irina Shklovski, Leysia Palen, and Jeannette Sutton, 2008. "Finding community through information and communication technology in disaster response," *CSCW '08: Proceedings of the ACM 2008 Conference on Computer Supported Cooperative Work*, p. 127.

Kate Starbird and Leysia Palen, 2010. "Pass it on? Retweeting in mass emergency," *Proceedings of the 7th International ISCRAM Conference*, at http://www.cs.colorado.edu/~palen/starbirdpaleniscramretweet.pdf, accessed 20 March 2012.

Michael Stonebraker, 2010. "SQL Databases v. NoSQL Databases," *Communications of the ACM*, volume 53, number 4, pp. 10–11.

Jeannette Sutton, Leysia Palen, and Irina Shklovski, 2008. "Backchannels on the front lines: Emergent uses of social media in the 2007 southern California wildfires," *Proceedings of the 5th International ISCRAM Conference*, at http://www.cs.colorado.edu/~palen/Papers/iscram08/BackchannelsISCRAM08.pdf, accessed 20 March 2012.

Twitter, 2011a. "Rate Limiting" (15 July), at https://dev.twitter.com/docs/rate-limiting, accessed 10 January 2012.

Twitter, 2011b. "REST API Resources," at https://dev.twitter.com/docs/api, accessed 10 January 2012.

Twitter, 2011c. "Streaming API" (1 November), at https://dev.twitter.com/docs/streaming-api, accessed 10 January 2012.

Twitter, 2011d. "Streaming API Concepts: Sampling" (14 November), at https://dev.twitter.com/docs/streaming-api/concepts#sampling, accessed 10 January 2012.

Twitter, 2011e, "Using the Twitter Search API" (2 December), at https://dev.twitter.com/docs/using-search, accessed 10 January 2012.

Twitter, 2012a. "GET search" (3 January), at https://dev.twitter.com/docs/api/1/get/search, accessed 10 January 2012.

Twitter, 2012b. "Streaming API Methods" (20 January), at https://dev.twitter.com/docs/streaming-api/methods, accessed 10 January 2012.

Sarah Vieweg, Amanda L. Hughes, Kate Starbird, and Leysia Palen, 2010. "Microblogging during two natural hazard events," *CHI '10: Proceedings of the 28th International Conference on Human Factors in Computing Systems*, p. 1,079.

yourTwapperkeeper, 2011. "yourTwapperkeeper," at https://github.com/jobrieniii

/yourTwapperKeeper, accessed 1 April 2011.

Editorial history

Received 26 January 2012; accepted 18 March 2012.

Tools and methods for capturing Twitter data during natural disasters
by Axel Bruns and Yuxian Eugene Liang
First Monday, Volume 17, Number 4 - 2 April 2012
http://www.uic.edu/htbin/cgiwrap/bin/ojs/index.php/fm/article/viewArticle/3937/3193

FM
Volume 17, Number 4 - 2 April 2012

TABLE OF CONTENTS

Reading Tools

Tools and methods...
Bruns, Liang

Abstract
Review policy
About the author
How to cite item
Indexing metadata
Print version
Notify colleague*
Email the author*

RELATED ITEMS
Author's work
Government policy
Book
Book reviews
Dissertations
Online forums
Quotations
Resources
Media reports
Web search

SEARCH JOURNAL

All

Search

CLOSE

* *Requires registration*