New-Web Search with Microblog Annotations

Tom Rowlands
Dept. of Computer Science
Australian National University
and CSIRO ICT Centre
tom.rowlands@ieee.org

David Hawking
Funnelback Pty. Ltd.
Canberra
Australia
david.hawking@acm.org

Ramesh Sankaranarayana
Dept. of Computer Science
Australian National University
ramesh@cs.anu.edu.au

ABSTRACT

Web search engines discover indexable documents by recursively 'crawling' from a seed URL. Their rankings take into account link popularity. While this works well, it introduces biases towards older documents. Older documents are more likely to be the target of links, while new documents with few, or no, incoming links are unlikely to rank highly in search results.

We describe a novel system for 'new-Web' search based on links retrieved from the Twitter micro-blogging service. The Twitter service allows individuals, organisations and governments to rapidly disseminate very short messages to a wide variety of interested parties. When a Twitter message contains a URL, we use the Twitter message as a description of the URL's target. As Twitter is frequently used for discussion of current events, these messages offer useful, up-to-date annotations and instantaneous popularity readings for a small, but timely, portion of the Web.

Our working system is simple and fast and we believe may offer a significant advantage in revealing new information on the Web that would otherwise be hidden from searchers. Beyond the basic system, we anticipate the Twitter messages may add supplementary terms for a URL, or add weight to existing terms, and that the reputation or authority of each message sender may serve to weight both annotations and query-independent popularity.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval; H.3.4 [**Information Storage and Retrieval**]: Systems and Software

Keywords

search, information retrieval, Web search, Twitter, microblogging, demonstration

General Terms

Algorithms, Experimentation, Performance

1. INTRODUCTION

The Web continues to grow with new information. Social networks, such as Flickr and Twitter, make it easy to quickly and frequently make available thought, audio and video from their part of the world and to comment and contribute to the Web at large. These contributions offer a possible new source of evidence for search on recent events on the Web.

Twitter is a *microblogging* service. It allows users, who can be private individuals, companies or government agencies, to post messages up to one hundred and forty characters long, which are displayed at the top of the posting user's twitter page (`twitter.com/`*username*) for anyone to see. Here, they persist for a time but, by their nature, rapidly stale. Twitter messages must be text only, but *can* contain URLs.

In this demo paper, we present an experimental new system utilising Twitter data as annotations for Web content. Rather than crawling the Web, we use the URLs mentioned within each message of a microblogging stream to discover pages. We then use the surrounding message as a description of the URL's content. Though not a replacement for a typical whole-of-Web engine, we are using this system to investigate search regarding current events.

2. BACKGROUND AND RELATED WORK

The major whole-of-Web search engines index many billions of documents. The Web has no central list of documents, however, so these are discovered through the use of a 'crawler', also known as a 'robot' or 'spider'. The process is bootstrapped with a seed list of URLs. As each page from the list of URLs is retrieved, it is analysed, new URLs in the page are added to a 'frontier' and the process is repeated [**?**].

The crawling approach is not without problems. Pages change, so re-crawling is required to keep the collection fresh. Some pages may not exist, or have no incoming links, at crawl time and given the size of the Web it will never be completely crawled. These potential problems can be partly addressed using heuristics to assign a crawling priority [**?**].

The link information gathered from the documents is not used exclusively for the discovery of new documents. Based on the idea that documents to which there are more links are more interesting, it is possible to infer a page's authority, or usefulness [**?**]. Further, the text of each link can be seen to describe the target's content, even if the target itself is not textual or otherwise parsable [**?**].

Online microblogging services, such as Twitter, are so popular and quick to respond to events that they have been preemptively blamed for the 'slow and lingering death' of the conventional media [**?**]. Java et al. have investigated the reasons *why* people use tools such as Twitter [**?**]. They found that users enjoyed sharing information and that one of the

WWW 2010, April 26–30, 2010, Raleigh, North Carolina, USA.
ACM 978-1-60558-799-8/10/04.

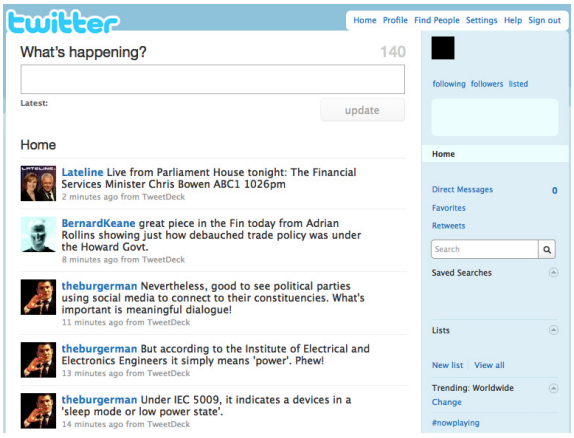

Figure 1: Twitter's Web interface

primary types of sharing was reporting or commenting on current events or news. Thirteen per cent of their collection contained URLs. Jansen et al. investigated how microblogging was used to distribute word of mouth branding [**?**]. They found microblog monitoring was a useful source of fast feedback from existing and potential customers.

3. MICROBLOGGING AND TWITTER

As already discussed, Twitter is a microblogging service allowing people to, often publicly, discuss 'What's happening?'. The tweets[1] and the posting service are hosted by Twitter at `twitter.com`. Before a user registers or logs in, visiting `twitter.com` shows a search page which displays 'popular topics' of varying period (minute, day, week). Examples at the time of writing were 'iPad' and 'Google Buzz'.

Twitter requires users to register, after which they may tweet as often as they wish. The Web interface of a user logged in is shown in Figure 1. A user can opt to 'follow' other users (perhaps friends or organisations of interest). By following another Twitter user, the followed user's tweets will appear for the following user on their Twitter page. By following many users, a constant 'stream' of information is generated, showing what the followed users are doing or thinking.[2] The screenshot shows several messages from people the user has followed. It is possible to reply to a tweet from another user by prefixing their tweet with `@`*`target-username`*. Unofficially, users discuss topics without all following each other by inserting 'hashtags' into their messages. A search on the Twitter service will then find all tweets with a particular hashtag. For example, during the recent Iranian election, by searching for `#iranelection`, Twitter users were able to discover otherwise difficult to find information [**?**]. Twitter offer a Web API, making the search function, and others, available for tools such as dedicated Twitter clients so a user need not use the Web to post and receive messages.

Beyond the standard Web service API intended for individuals, Twitter offer a 'Streaming' API.[3] While, at the time of writing, under 'alpha test', this API facilitates access to a subset of the full 'firehose' of Twitter data, updates from all users with the exception of private postings. Twitter do not allow general, public access to the firehose. Instead, subsets of the firehose are available through a 'sample' and 'filter' methods.

[1]In Twitter terminology, a 'tweet' is a Twitter message.
[2]A user may choose to hold their tweets 'private', in which case they will not appear in all their followers' streams.
[3]`http://apiwiki.twitter.com/Streaming-API-Documentation`

4. RATIONALE

Pages that have been on the Web for a while are more likely to have links pointing to them than new pages. New pages, which we assume to be of particular interest for recent event searches, are likely to have fewer, or no, incoming links. This may harm recent event search in several ways. Having more links may lead to old pages being crawled at the expense of new pages, through the chance of happening upon a link, and because crawlers may use link-count based methods such as indegree to prioritise their crawl [**?**]. For the same reason—higher indegree—old pages may be ranked above more relevant new pages when a search is performed. Old links may also feature out of date anchortext, giving a description of the page's previous content, or many anchors containing a particular query term, artificially inflating relevance.

There are certainly old pages with information relevant to new discussion—a news page may be at a fixed location but have its content updated, or a pertinent encyclopedia entry—but such pages will also be mentioned in discussion. If the page's content is relevant, it is more likely to be mentioned, regardless of its age. It is less likely to be the target of many links if it is new, however, resulting in the biases described above.

Our demonstration collects tweets that contain URLs using the filter method. We hope to use our tool to investigate whether such microblogging evidence defeats the above effects for new-Web search; search of the Web that is new or recently changed. An important distinction must be made between this demonstration (and our intended study) and existing Twitter search tools, such as the one available from `twitter.com` itself. We are not searching for tweets as an end goal. Our tool searches the Web using tweets as supporting evidence.

5. ARCHITECTURE

In order to experiment with the use of tweeted URLs, we first ran a static recording of tweets followed by a manual retrieval of URLs. With this information we designed and constructed a 'live' indexer. Both are described below.

5.1 Static investigation

Our initial recording took place over eight days and used Twitter's 'filter' Streaming API method. We used '`http`' as the keyword to target only tweets containing links. From these, URLs were extracted using a simple regular expression. The distribution of hosts in the initial URL set are illustrated in Figure 2. The figure shows a substantial difference between regularly mentioned hosts, such as `bit.ly` and `tinyurl.com`, and the less frequently ranked hosts. These are hosts extracted directly from the URLs in the tweets, before any possible redirection.

5.1.1 URL shortening

Twitter users frequently shorten the URLs they post to fit both a message and a URL into the one hundred and

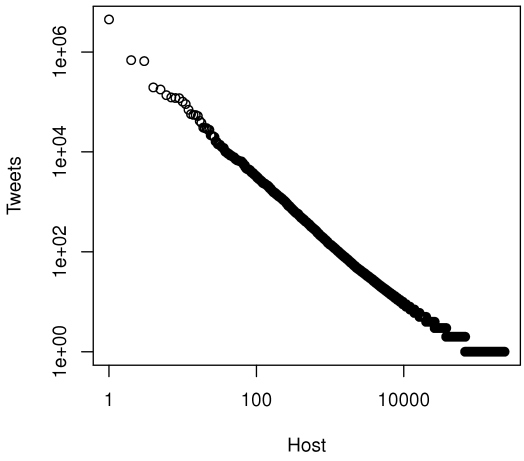

Figure 2: This graph shows the frequency in which each host in the 'crawl' is mentioned. The first few hosts are the dominant URL shortening services.

forty character limit. Another motivation is to avoid possible mangling of unusually long or complicated URLs by Twitter clients. This is achieved by a 'URL shortening service' that redirects a request for the shortened URL to the real URL. Twitter currently automatically re-writes URLs it receives via the Web using `bit.ly`, yielding URLs similar to `http://bit.ly/AbCdE`. The prominence of shortening services in Figure 2 demonstrates the importance of the URL shortening to Twitter users and the importance of resolving URLs, if we wish to see interesting results.

After resolution by the shortening service, it is possible that two, distinct, short URLs target the same page. Similarly, shortened URLs may appear in more than one tweet.

5.2 Live search

While the static recording and retrieval of pages was useful as an initial experiment, it became clear that a more compelling experiment could be conducted with a quicker turnaround between the tweet and availability in the index. To this end, we have constructed a 'live' tweet annotation search system comprising a tool to collect the data and a typical indexer-query processor. A slightly simplified diagram of the data collection tool appears in Figure 3. As it is, in a sense, much like a typical Web crawler without the recursive component, we call it a 'wriggler.' The wriggler was constructed using Perl.

Again, the Twitter Streaming 'filter' API is used to gather tweets including the text '`http`'. These are passed through a Unix FIFO to the *enqueuer* which is responsible for the removal of duplicates and the recording of the annotations along with the associated URL. The use of a FIFO enabled convenient repeated debugging with the same test set. After recording, the URLs are passed to the queue q^*.

The *disseminator* removes URLs from q^* and hands them on to individual host queues. For the vast majority of hosts, there is only one host process, such as p_i. This allows simple rate limiting, maintaining TCP connections with 'Keep-Alive' and adhering to `robots.txt`. There is a strict limit on the number of requesting processes and new host process is only created when another empties its queue.

The static investigation revealed that there are likely to be many URLs from a small number of high-performance hosts (such as `bit.ly`). These services are easily capable of responding to more than one concurrent request. Many of the most popular hosts are URL shortening services and if URLs from these were strictly serial the retrieval of many 'real' URLs would be greatly slowed. For these reasons, a small number of hosts are permitted a number of threads. In the diagram, p_1 and p_2 are both reading from the one queue.

After downloading, the HTTP responses, including documents, are delivered through another queue, q_β, to a WARC file [**?**]. If a URL leads to a redirection, the redirection itself is recorded through another queue, q_γ. (Processes responsible for writing these queues to disc are not shown.) The files are named to make rolling over to new data as easy as possible. Redirections are also passed back to the enqueuer, where they may be dropped if they have already been retrieved.

5.2.1 *Indexing and evidence*

In the demonstration there is a reasonably constant indexing process. With an initial delay of thirty seconds (which is arbitrarily defined so as there is at least one document) the tweet recording file's redirects are resolved (this is particularly important, given the prevalence of URL shortening systems), and the WARC and tweet annotation files passed to an indexer.

The indexer considers the tweets as anchors. Terms in tweets indicate relevance to the target document and the more tweets to a document, the more relevant it is, all else being equal. Once complete, the new index is atomically introduced as the 'current' index, and another indexing run is begun.

After a pre-determined period, old tweets, redirects and WARC files are 'rolled out' of the collection. The period is still subject to experimentation.

5.2.2 *Query interface*

The Web based query interface is shown in Figure 4.

The demonstration system has shown useful results for a variety of ad hoc queries. Results in the system are always Web pages. The significance tweet evidence relative to content is subject to experimentation. Consequently, in this demonstration, the evidence used to find those pages can be varied. It is possible to separately search over the pages' content, the tweets that point to the Web page, or both. Each delivers different results, much like searching over Web page content compared to anchortext. Similarly, the significance of many pages pointing to the one Web page (analogous to indegree) is yet to be fully explored.

6. CONCLUSIONS AND FUTURE WORK

With our demonstration we hope to derive useful search over websites discussed on a microblogging service. There are many unanswered questions and ideas for future work.

We have presumed parallels between tweets and anchortext; similarly, between indegree and URL mentions. Both of these could also be used by the wiggler to better prioritise the downloading of pages. URLs mentioned more often

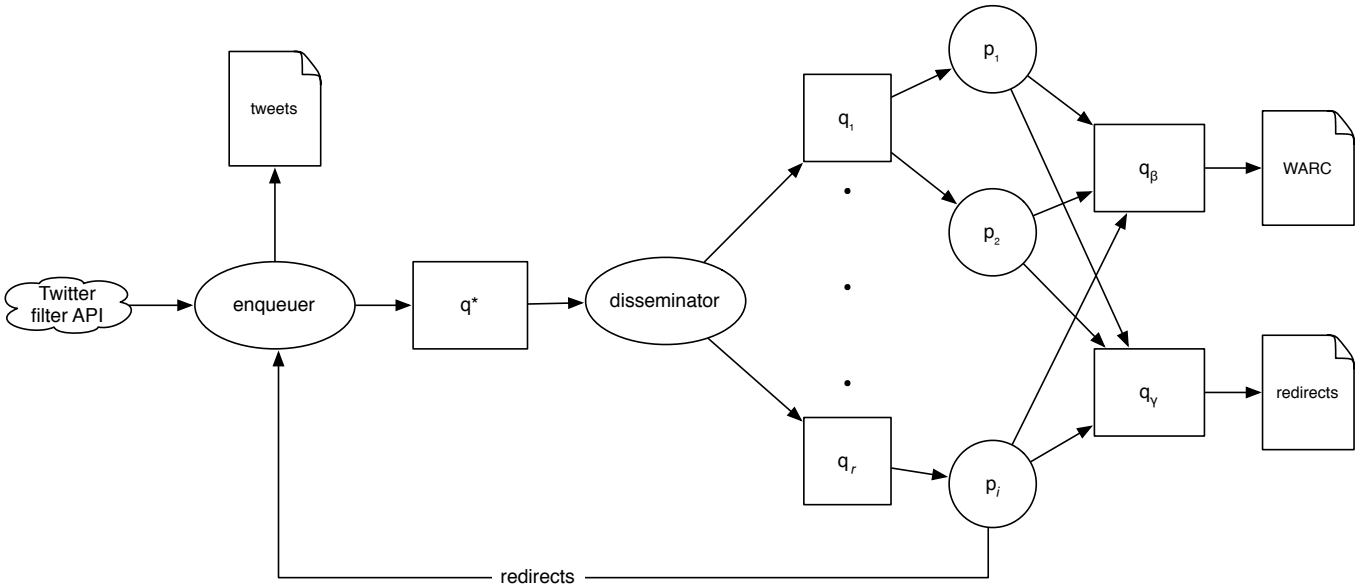

Figure 3: This diagram shows the general construction of the wriggler. Circular icons represent processes and rectangular icons represent queues.

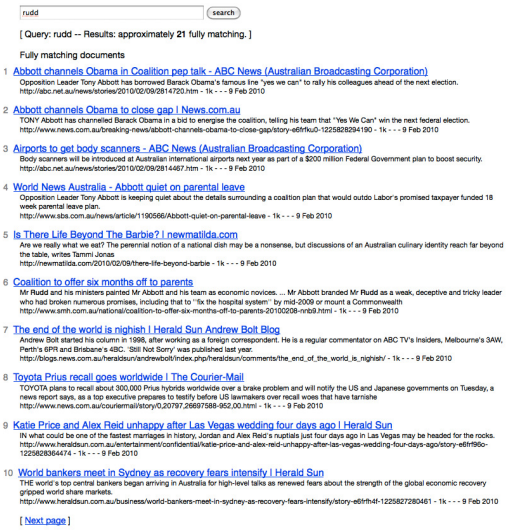

Figure 4: This screenshot shows the Web-based query interface for the live search.

should probably be downloaded with a higher priority.

Spam was was not unusual in the Twitter stream, although exactly how much or what proportion was not measured; useful results were attainable regardless. Removing spam may improve results further.

The nature of Twitter's 'filter' API and sampling process is not known. There is a chance that it will change and that its output will be hard to reproduce.

Acknowledgements. Thanks to Francis Crimmins and Funnelback for their help and bandwidth.