Pulse: A Social Media Analysis Toolkit

http://demos.terrier.org/pulse/

Paul Holmes
paul@synae.co.uk

Craig Macdonald
craigm@dcs.gla.ac.uk

Iadh Ounis
ounis@dcs.gla.ac.uk

School of Computing Science
University of Glasgow
Glasgow, G12 8QQ, UK

ABSTRACT

An explosion in the usage of social media such as Twitter has created a wealth of data well-suited to mining and analysis. With the correct tools, this data can be analysed and interpreted in a meaningful way — identifying trends, establishing opinions and understanding usage. In this demonstration, we describe *Pulse*, a social media analysis toolkit. It is capable of processing tweets streamed directly from Twitter, managing their storage and indexing, analysing each in order to aggregate geographical, temporal and magnitudinal statistics about Twitter usage and presenting the results in a browser-based application. Users can search using temporal or subject-based queries, and are presented with components that display information relevant to a particular time, country or subject. The results themselves are presented through a number of modular components: as text, and graphically using chart visualisations and maps.

Categories and Subject Descriptors: H.3.3 [Information Storage & Retrieval]: Information Search & Retrieval

General Terms: Design, Human Factors

Keywords: Twitter, Trends

1. INTRODUCTION

The volume of data being created by the users of Twitter is incredibly large, and access to archived data is limited. Additionally, few utilities exist that provide the ability to analyse Twitter posts *en masse* in great depth across multiple variables — including tweet location, time, language, sentiment and volume. Being able to mine mining Twitter to understand trends and topics of note at a particular time would be useful in many commercial and academic fields [4]. For instance, a business may wish to monitor the perception of their product in social media [5], or measure the success of its interactions with customers on Twitter. Other searching of Twitter may encapsulate current news or people [7, 8].

Pulse provides three key elements to help in such scenarios. Critically, it can collect a wealth of data to allow analysis of business / brand name / person & usage through either hashtags or plain text within tweets. Building upon this data is a comprehensive suite of tools, which enable users to understand, analyse and manipulate query results. This achieved through many various 'components', which are a series of widgets designed to answer specific questions, such as: "where are people talking about this brand?", "how does our presence compare to our competitors?" and "what effect has a given event had on our social media presence, relative to before?". Finally, it includes an intuitive, customisable application through allowing users to easily access and manipulate results.

Social media, and in particular Twitter, provides organisations with an unprecedented opportunity to immediately and directly communicate with their stakeholders. Pulse allows organisations to better evaluate the effect of these ad-hoc, qualitative, one-to-one conversations through data aggregation, mapping, sentiment and textual analysis. In addition to marketing and business potential, Pulse is also a tool for exploration and discovery. Tweets are also semi-structured in nature (text, links, user data and 'retweets'). Being able to browse the structure by topic or time, in either a directed or undirected fashion, provides an interesting and unique way to view the collected thoughts, ideas and opinions of the Internet at large.

Several systems already exist that provide some level of insight into tweet trends. Trendistic [9] allows users to review and compare term volumes against each other over specific time periods. Whilst this is useful to see when and by how much particular terms 'spike', it gives little surrounding context, e.g. the geographical nature of spikes, their underlying sentiment, or which other terms simultaneously experienced a surge in usage. Pulse remedies this by bringing together numerous data views as components – self-contained units of functionality that provide a single perspective on a retrieved Twitter data set.

Indeed, it is a weakness of most existing systems in that they each consider only one particular aspect of the Twitter domain, providing data out of context, and focusing specifically on the individual. However, there are existing systems that provide more than a single piece of functionality. Trendsmap [10] is one of these. While it is primarily focused on illustrating top trending terms, tags and people across the globe, it also supplies updates in real-time and extensive filter options. However, with all these features, it is challenging to interpret the overwhelming amount of data presented by Trendsmap in a useful way. Indeed, whilst aesthetically impressive, the visualisation techniques used are not necessarily effective in aiding users in extracting meaning. In contrast, Pulse has a intuitive & clear user interface which encourages exploration and discovery as much as directed investigation.

To summarise, the main aims of Pulse are: (i) to allow users to move from understanding tweets on a micro-level in an unstructured fashion, to extrapolating information of importance on a macro-level through data aggregation and analysis; (ii) to encourage users

CIKM 2011, October 24-28, Glasgow, UK.

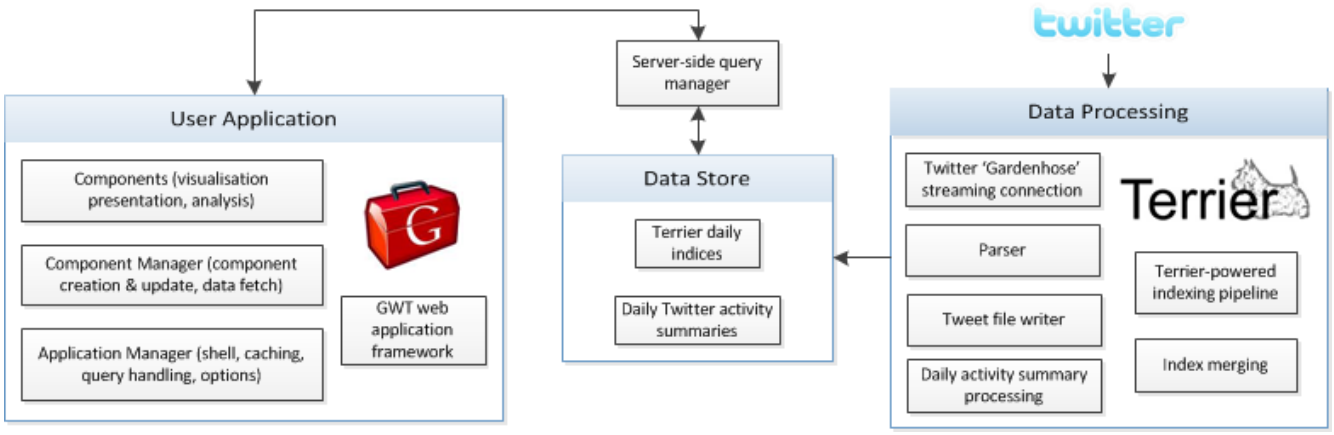

Figure 1: System architecture of Pulse.

to browse and explore Twitter data sets out of curiosity, or for pleasure, as users do when reading their own news feeds on Twitter; and (iii) to make it easy for advanced users and developers to further enhance the system by providing a standard method to add new components featuring additional functionalities.

The structure of the remainder of this paper is as follows: Section 2 details the design of the Pulse architecture; Section 3 describes examples of components that are built into Pulse; in Section 4, we provide an outline plan for the demonstration of Pulse at CIKM 2011; finally, concluding remarks follow in Section 5.

2. PULSE SYSTEM DESIGN

Pulse consists of two major components: a server-side, 'always on' *data processing* system which handles the gathering of tweets streamed from Twitter as well as indexing and storage into appropriate data structures; and a browser-based *user application*, which provides components for viewing the aggregated Twitter data in various ways. While the data processing system builds upon indexing and querying functionalities provided by the Terrier platform [6][1], the user application comprises of a modern interface built upon the Google Web Toolkit framework [2][2]. The overall system architecture of Pulse is shown in Figure 1.

2.1 Data Processing

Underlying the Pulse user application is the data processing system, which streams, parses, stores and indexes every tweet obtained from the Twitter Gardenhose data stream for later retrieval. The system aims to achieve reliability through three tenets: robustness – to recover from errors; scalability – to ensure both short-term bursts and long-term increases in tweet volume can be handled; and efficiency – to ensure that processing is carried out with assurances that the system cannot 'fall behind' with respect to incoming tweets. To achieve these three tenets, the core of the data processing system (right hand side of Figure 1) is a multi-threaded program that uses triggers to schedule activities at certain points in time.

Tweet indexing is achieved through integration with the open source Terrier platform [6], which provides efficient and highly compressed index data structures, querying functionality, and compressed metadata storage. Indices are created regularly (e.g. hourly) by triggers, and occasionally merged into daily indices, as well as additional daily summaries of Twitter activity. These indices permit querying by full text or hashtag only, with high time resolution.

2.2 User Application

The requirement of the user application focus on usability, comprehensive functionality and extensibility. As Pulse aims to allow directed and undirected exploration, it must be usable by both casual users to fulfil an unspecified information need, as well as expert users (e.g. marketing specialists) with an explicit information need modelled in terms of topics or themes. This affected not only the aesthetic design and layout, but also the choice of platform.

To best meet these needs we opted to develop Pulse as a browser-based application, choosing to use an Asynchronous JavaScript and XML (AJAX) framework as its foundation. After reviewing several such frameworks, we chose Google Web Toolkit (GWT), a development toolkit for building complex browser-based applications. GWT simplifies the creation of web applications by allowing Java development of JavaScript applications, along with an RPC mechanism for development of communication with traditional server-side services. At deployment time, the Java code is compiled to JavaScript and deployed like any other website [2]. Building Pulse with GWT has numerous advantages: a rapid build-deploy-test cycle; the ability to use mature and familiar Java tools; a framework for building and integrating the server-side components necessary for querying the tweet indices, with customised versions of the application being loaded dependent on the user's browser. With GWT at its foundation, the rest of Pulse was designed as a stack (see left hand side of Figure 1) with three layers: the application framework, component manager, and the components themselves.

The first of these, the application layer, provides system-wide functionality. It structures the shell of the application interface, handles cross-query features (caching, navigation and comparison) and manages user options. The shell was designed to be clear and intuitive — only a search bar and date picker are visible at first, with advanced options and features initially concealed to avoid overwhelming the user. However, advanced users may customise components through optional parameters, and developers can easily include more by adding additional menu panes to the shell.

Built upon the application layer is the component manager, responsible for the creation, update and removal of components, requesting data from the server, propagating this data to each component and managing the query cache.

Finally, the topmost layer of the Pulse user application stack is the components themselves. Each is unique, but they do share cer-

[1] `http://terrier.org`
[2] `http://code.google.com/webtoolkit/`

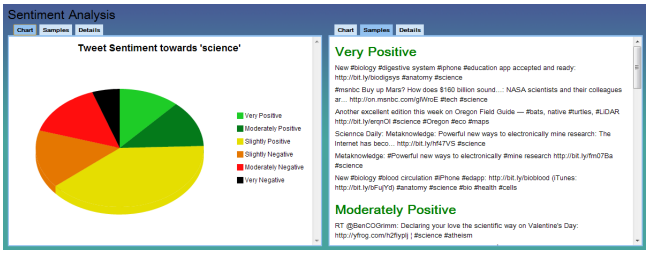

Figure 3: Pulse's sentiment analysis component.

tain attributes: modularity; loose coupling; and support for query comparison and regeneration from the cache. Components in Pulse provide the most important end user functionalities, from geographical plots to word clouds, detailed manipulable graphs to sentiment analysis. All components are built from a common template with standard methods to interact with the component manager, making Pulse easily extensible.

2.3 Query Response Flow

The user application allows users to query, analyse and manipulate Twitter data stored in Terrier indices in an intuitive, usable and detailed manner. Various query types are available within Pulse – for example: a *hashtag search query* obtains and analyses the appropriate Terrier index for tweets matching a given hashtag; a *temporal query* contains no query terms, but allows users to see an overview of Twitter activity, usage and trends across a period of time. All queries are initiated through a simple search bar at the top of screen. The query type is implicitly derived from the fields completed by the user. Entering a query term and selecting a date will begin a hashtag search query for that term on that date, whilst leaving the term field blank and selecting a date will start a temporal for that date.

For each query, requests are sent to a server-side service with access to the Terrier indices. The service ensures data exists for the requested day, queries the appropriate index, parses the results into an array, and returns this to the *component manager*. The component manager identifies and initialises the appropriate components for the type of the query. It is of note that the results of all queries can be stored, compared and quickly regenerated to avoid unnecessary re-querying. Figure 2 shows the Pulse user interface for the query 'apple' with various components displayed. In the following section, we describe a few example components of Pulse.

3. USER APPLICATION COMPONENTS

3.1 Hashtag Query Components

The components generated for hashtag queries aim to provide a comprehensive overview of the importance, context and frequency of tag usage. For example: the *Tagged With Component* illustrates co-occurring tags as a word cloud, as well as comparison of tagging between two queries; the *Location Component* maps tweeting by tag globally, establishing user location through a web geolocation service; the *In Context Component* performs additional textual analysis of each tweet, showing what proportion of the retrieved tweets were questions, retweets, featured mentions or were from unique users; while the *Sentiment Analysis Component* categorises tweets into one of six sentiment groups, from 'very positive' to 'very negative'. In the interest of brevity, we explain only one of these in detail: the sentiment analysis component (shown in Figure 3). This Pulse component can be used to understand popular opinion (by theme or over time), aid discovery of new interests and inform decision making.

Sentiment analysis is performed by a server-side service. In particular, a sentiment score is obtained for a tweet by using the Stanford Part of Speech (POS) tagger [3] to obtain the POS of every token, before looking up the sentiment of the tokens in SentiWordNet [1]. Moreover, given typical tweet length (140 characters or fewer) and the challenges presented by their content (e.g. slang, punctuation, sarcasm), we also apply Twitter-specific sentiment features. For example, the score of tweets, which feature emoticons (such as ':D' or ':(') is adjusted to reflect their presence, in turn affecting the overall score of the tweet. The final breakdown of tweet sentiments matching the user's query are returned to the user application and displayed within the sentiment analysis component. This offers three tabbed views of the results: as an interactive pie chart, a list of sample tweets, and a data table of the raw results (shown in Figure 3). Whilst this is a fairly simple method of sentiment analysis, the extensible nature of Pulse and its services mean that other sentiment analysis techniques (e.g. those described in [4]) can easily be integrated.

Another feature of Pulse is double hashtag queries (e.g. '#obama #teaparty'), which generate a specialised set of components that not only show results for each tag, but also comparisons between the two result sets. For example, for a double hashtag query, the 'Tagged With' component will show tags common to both queries, as well as tags unique to one or the other query as a separate pane within the component. Finally, queries are cached, allowing users to quickly select past queries from the 'Tag Query History' pane on the left-hand side of the application (see Figure 2), without additional server requests.

3.2 Temporal Query Components

Components generated in response to temporal queries aim to highlight trends and summarise day-by-day usage of Twitter over periods of time. This was achieved through the development of various components, including: the Trends Component, which provides a highly customisable and powerful data visualisation showing tag usage across several variables (including time, volume and location) in a number of different ways, with advanced filtering and display options; the *Place / Device Component*, which shows the volume of tweets posted by software / platform, as well as by geographic location; and the *Retweet Paths Component*, which highlights the most popular retweets of the day in a visually compelling way, with mapping to illustrate each retweet's propagation path, and on-the-fly translation of non-English tweets.

Again, for brevity, we focus only on one widget — namely the 'Retweet Paths' component (see Figure 4). Like all temporal query components it uses a data generated every 24 hours, which summarises Twitter activity across an entire day, including the most prolific tags (along with usage metadata such as location and user details), links, retweets and tweeting platforms used.

The 'Retweet Paths' component has three key features: a data visualisation showing the top retweets of the day by volume, a translation function to convert non-English tweets to English, and a mapping function to show a propagation trail, illustrating how a tweet spreads across the twittersphere. The primary purpose of the component is to display the day's top retweets in a visually compelling ordered list, with colour used to represent relative volume differences between each retweet. Should Pulse detect that any retweet is not English, then a translation is performed by invoking an existing commercial translation tool. Moreover, should a retweet set include sufficient location data a mapping pane becomes available to the user. This maps the trail of a retweet from first appearance to final appearance, using a web service to establish latitude / longitude from unstructured location information, for each retweet occurrence. In doing so, Pulse provides an interesting

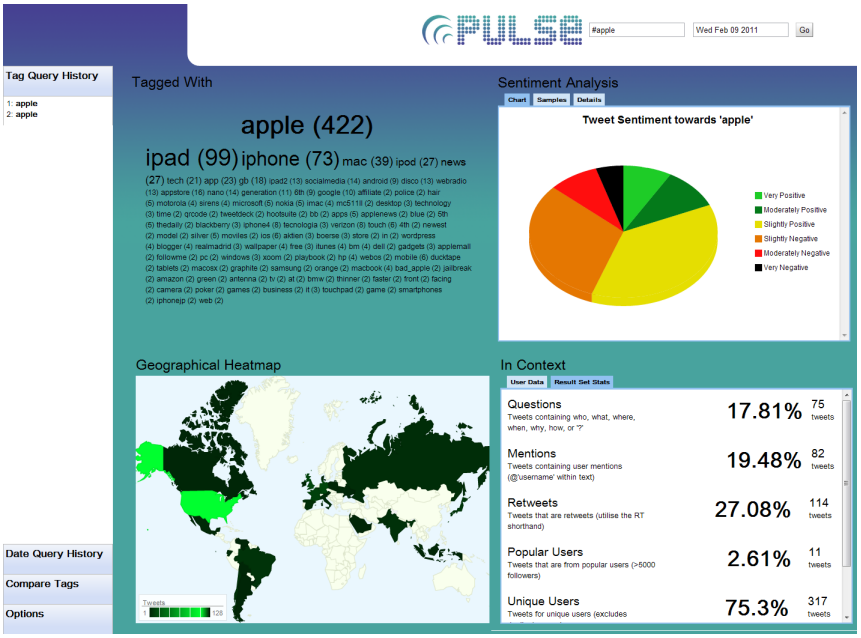

Figure 2: The Pulse user application.

Figure 4: Pulse's Retweet Path component.

perspective on the locality of retweets, and highlights what kind of tweets are likely to have country, continent or global reach.

4. DEMONSTRATION PLANS

In the brief live plenary demonstration we will provide an overview of the motivations of the Pulse system, and demonstrate one or two interesting timely queries to the system, showcasing particular components and query types. For instance, we will show how Pulse can initially be used to explore Twitter trends across time periods, then used to drill down on specific themes. For the demonstration session, it is envisaged that a standard set of queries will be presented to attendees showing various query types and components, before allowing users to explore Pulse for their own interests by themselves. We will provide a laptop, however, a wireless Internet connection will be required. A large flat-screen monitor would be appreciated to facilitate showing the demonstration to several attendees at once.

5. CONCLUSIONS

In this paper, we detailed our prototype system, Pulse, for storing, indexing and analysing Twitter data in a comprehensive and robust manner. Through our demonstration we hope to garner additional feedback to inform future development, and demonstrate both the scope of potential applications, and value of such applications which use Twitter and social media data.

6. REFERENCES

[1] A. E. S. Baccianella and F. Sebastiani. SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proc. LREC 2010*.

[2] P. Chaganti. *Google Web Toolkit: GWT Java Ajax Programming*. Packt Publishing, Birmingham, UK, 2007.

[3] K. T. Dan, D. Klein, C. D. Manning, and Y. Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proc. HLT-NAACL 2003*, pages 252–259.

[4] C. Macdonald, R. L. Santos, I. Ounis, and I. Soboroff. Blog track research at TREC. *SIGIR Forum*, 44:58–75, 2010.

[5] G. Mishne and M. de Rijke. A study of blog search. In *Proc. ECIR 2006*, pages 289–301.

[6] I. Ounis, G. Amati, V. Plachouras, B. He, C. Macdonald, and C. Lioma. Terrier: A high performance & scalable IR platform. In *Proc. OSIR 2006*, pages 18–25.

[7] S. Petrović, M. Osborne, and V. Lavrenko. Streaming first story detection with application to Twitter. In *Proc. HLT-NAACL 2010*, pages 181–189.

[8] J. Teevan, D. Ramage, and M. R. Morris. #twittersearch: a comparison of microblog search and web search. In *Proc. WSDM 2011*, pages 35–44.

[9] Trendistic.com, 2011.

[10] Trendsmap.com, 2011.