S. Boll, S. Maaß & R. Malaka (Hrsg.): Mensch & Computer 2013
München: Oldenbourg Verlag, 2013, S. 349– 352

Tangible Twitter Search: A Multiuser Multitouch Tabletop Interface

Andreas Lingnau[1], Eva Hornecker[2,3], Kenneth Coyle[3]

Catholic University Eichstätt-Ingolstadt, Germany[1]
Bauhaus-Universität Weimar, Fakultät Medien, Germany[2]
University of Strathclyde, Dept. of Computer & Information Sciences, Glasgow, Scotland[3]

Abstract

Social Media has become an integral part of our society that is no longer limited to private exchange and contacts but has also grown into business and commercial application and even culture and learning. Nevertheless, the representation of information is a crucial part and is currently dominated by chronological representations that can be found in Twitter, Facebook, Google+ and the like. Although these interfaces are usually offering dedicated filtering mechanisms and sorting, there is a potential for using information from social media platforms in various contexts that have different requirements than a chronological order, such as reviewing posts about social events (e.g. concerts, museum exhibitions or festivals) which can be distributed over a longer period. In this paper we present a prototype and some use case examples of a search term based, multi-user enabled visualisation of twitter postings.

1 Introduction

While social networks like Facebook or Google+ are imitating the access of third party applications to their content, Twitter has a more open access model allowing users to e.g. integrate timelines into websites and filtering only relevant information represented by keywords that are used in a tweet, marked by a so called hash tag (#). Thus, a chronological list of postings about a special event like a conference can be presented by filtering all tweets that are using the dedicated hash tag (e.g. #delfi2013).

The problem with chronological representations of such information is that there is no indicator on how useful or important a tweet may be and whether or not it is worth being notified. Even mechanisms like re-tweeting usually only reach a closed circle of followers of the user who re-tweets something. Furthermore, sharing information in open spaces or public places such as a museum or tourist information centre is difficult if multiple users should be able to work collaboratively or access information collaboratively. There are examples of public twitter displays, e.g. in train stations, as a kind of real-time FAQ board, but here again the problem of a chronological order arises, causing tweets to disappear from the screen after a certain time when newer tweets come in so that important information vanishes.

Tabletop interfaces are considered well suited for supporting collaborative interaction. They allow users to surround the table and share documents. Touch interaction is highly visible for others, while providing a casual and 'natural' means of interaction. Tangible input on tables further benefits from the immediacy of touching and moving objects on a surface. Supporting collaborative search for multiple users via tabletop devices is an interesting but challenging approach. Since users around the table in open places like museums do not necessarily know each other, the interface has to provide the possibility to start individual queries that can be joined from case to case. Furthermore, users around the tabletop normally act asynchronously, which makes it technically challenging for a search interface to provide results within an acceptable time. In PuppyIR we have tackled some of these problems when we introduced a tabletop device for collaborative search in a museum (Lingnau, et.al, 2010). Although the target group were children, the requirements for a search interface in an open space are similar. The interface will be used by users with different technical background over random periods of time. It has to be self-explaining and robust and the workflow should not be interrupted by a need for reset or restart when a user leaves the tabletop.

2 A Tabletop Twitter Interface

As mentioned, most existing twitter interfaces follow a sequential, chronological approach. After the user has logged in he/she can view his/her timeline with all tweets from users he/she follows. Additionally, most twitter interfaces provide a search interface where tweets can be filtered and tweets from other users are displayed according to search terms. The tweets found by a search query will also be displayed in chronological order, usually starting with the most recent and allowing to scroll back on the timeline. Following the ideas developed in PuppyIR for the design of search interfaces in open spaces, we have developed a query based twitter interface for a multi-touch tabletop device to help users search for topics of interest, pick out relevant results and share information with other users present.

The interface is implemented using Java standard libraries, such as MT4J, Twitter4J and some smaller helper libraries. It consists of 3 independent modules: Twitter Stream Collector (using Twitter4J), Data Storage (standard Java and helper libraries), and Interface (using MT4J). Coupled with the standard application structure that MT4J supports and partially enforces, we augmented a logical approach to fiducial handling and animation control. This can be seen in the source we uploaded to Github[1]. MT4J was a good choice for implementing this system as a demonstrator prototype – however further work may involve abstracting away from it. Twitter4J was used to handle communication with Twitter, abstracting away the idiosyncrasies of this API. At all stages, there was an understanding that this system may be used to integrate with similar platforms to twitter – namely Identi.ca and Facebook.

[1] https://github.com/automatical/MultiTouch

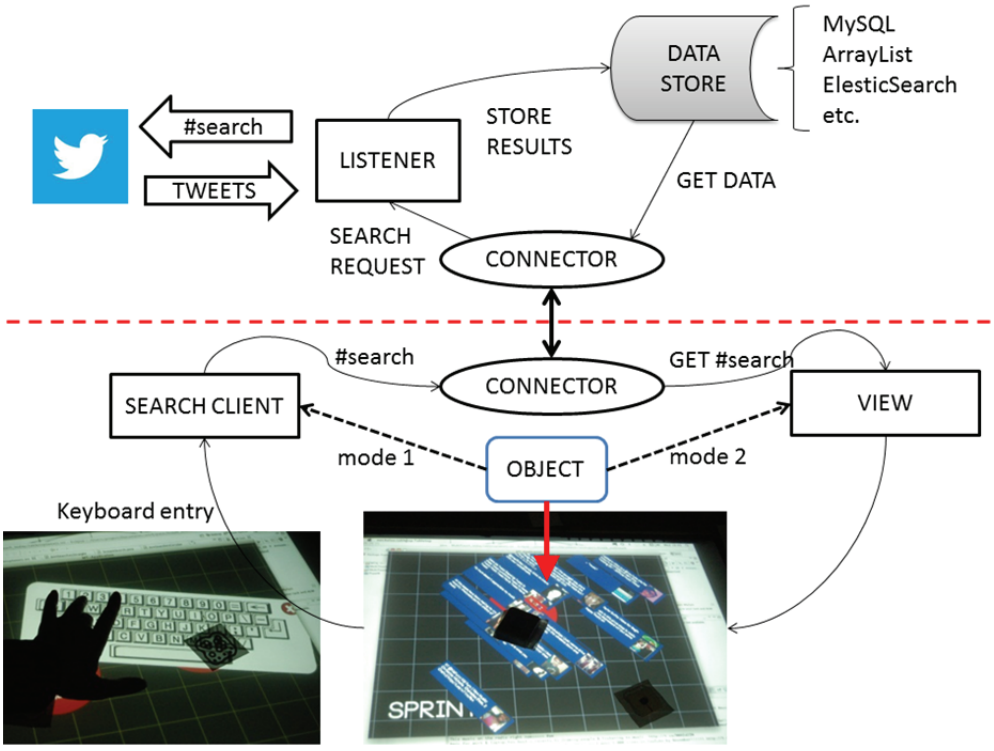

Figure 1: Architecture of the Twitter tabletop search interface with screenshots of the implementation

Fig. 1 shows the architecture with an information cycle. The upper half visualises the tweet collection architecture where requests are sent to the Twitter API as search queries. The results are collected by a listener and buffered in a data store. In the implementation of the architecture the data storage is a flexible plug-in module that can easily be changed. We are currently using a simple implementation that makes use of the Java ArrayList library and a custom collection logic. There has been some experimentation and semi-implementation of this using a MySQL database. Since initial implementation we have discovered that using Elasticsearch as a database and search mechanism will improve both speed and accuracy of data storage and retrieval. Currently we are using a MySQL database but it could also be an ArrayList or ElesticSearch. The lower half of figure 1 shows the architecture of the user interface implementation at the tabletop device. It consists of a search client and a view. When a user enters a search query at the table, the search client will send a search request that will be processed and results will be collected as described above. From the data store results will then be sent back to the table and visualised as objects in the user interface.

The user interface is deliberately simple. Given the technical limitations of our hardware, we focused on tangibles, rather than physical touch. We thus limited interaction to the following actions: add tangible token, remove token, and rotate or move token. To this end, we built a workflow for searching and filtering. A user would choose which type of search they wished to perform, and from this they would select the correct token. Multiple tokens are available for different searches: stream search (for messages that can be collected live), static search (for messages that have already been sent), people search (search for tweets from a certain

individual). Tweets are not necessarily filtered by hash tags, but hash tags get a better weighting in this case. Once a user has selected a token and placed it on the table, a keyboard appears. Using this keyboard, the user can enter a search term and confirm by closing the keyboard. The search or filter is now active. Tweets will appear, circling the tangible token. This can be repeated multiple times on the same table. To combine two queries, the tokens can be moved closer together and on getting closer together, they will define a combined search. We borrowed the metaphor of a water stream from previous work done on 'facet streams' (Jetter et al 2011) and developed two versions of representing results: waterfall (tweets related to both search terms run towards the user originating between the markers) and fountain (tweets originate between markers and then circle around both markers). The waterfall model was easier to understand for test users, but has obvious disadvantages in terms of screen estate, as well as tweets disappearing once they hit the screen border.

3 Discussion and Outlook

The interface is designed to be used mainly for retrieving "on the fly", without the need of a configuration. A possible use case could e.g. be a tourist information centre where people are using physical objects that represent tourist attractions, activities or events. By placing objects related to the activity or event they are interested in, results, i.e. twitter messages, will immediately be shown. This setting would also allow for multi-user search activities (probably on a bigger table). Be selecting tweets that are interesting for the visitor the other users around the table may be inspired while searching for "things to do". In a next step a 'join' of queries could lead to either collaboration between people around the table or better results that help individuals. In a next step, we would like to introduce objects that represent standard filtering, e.g. to find tweets from people of a certain age group, and also tokens with dynamically assigned functionality, depending on the particular situation.

References

Jetter, H-C., Gerken, J., Zöllner, M., Reiterer, H., Milic-Frayling, N. Materializing the query with facet-streams. Proc. of CHI '11, 3013–3022, New ACM NY 2011.

Lingnau, A., Ruthven, I., Landoni, M., & Van der Sluis, F. (2010). *Interactive Search Interfaces for Young Children - The PuppyIR Approach*. 10th IEEE International Conference on Advanced Learning Technologies (pp. 389–390). IEEE. doi:10.1109/ICALT.2010.111

Contact

Dr. Andreas Lingnau, Katholische Universität Eichstätt-Ingolstadt, Mathematisch-Geographische Fakultät, Abteilung Informatik, Ostenstr. 14, 85072 Eichstätt, Germany. lingnau@ieee.org

Professor Eva Hornecker, Bauhaus-Universität Weimar, Fak. Medien, Bauhausstr. 11, D-99423 Weimar. eva.hornecker@uni-weimar.de