

TWinChat: A Twitter and Web User Interactive Chat System

Yuanyuan Wang
Kyoto Sangyo University
603-8555 Japan
yuanw@cc.kyoto-su.ac.jp

Yukiko Kawai
Kyoto Sangyo University
603-8555 Japan
kawai@cc.kyoto-su.ac.jp

Gouki Yasui
Kyoto Sangyo University
603-8555 Japan
i1458085@cse.kyoto-
su.ac.jp

Toyokazu Akiyama
Kyoto Sangyo University
603-8555 Japan
akiyama@cse.kyoto-
su.ac.jp

Yuji Hosokawa
Kyoto Sangyo University
603-8555 Japan
i1358103@cc.kyoto-
su.ac.jp

Kazutoshi Sumiya
University of Hyogo
670-0092 Japan
sumiya@shse.u-
hyogo.ac.jp

ABSTRACT

This paper presents *TWinChat*, a Twitter and Web user interactive chat system to support simultaneous communication between microbloggers and Web users in real time through both the contents of microblogs and Web pages. *TWinChat* provides a question answering interface attached to Web pages, which allows Web users to chat with Twitter users in real time while presenting tweets that are associated with Web pages, i.e., simultaneous cross-media communication. In order to map heterogeneous media, the system extracts relationship between tweets and Web pages by generating queries based on location names. Thus, our system can effectively present messages from Web users to help Twitter users immediately obtain useful information or knowledge, and it also can effectively present tweets from the Twitter users to help the Web users easily grasp the current situation in real time.

Categories and Subject Descriptors

H.3.4 [Systems and Software]: Question-answering (fact retrieval) systems; H.3.5 [Online Information Services]: Web-based services

Keywords

location-based microblogs; cross-media user communication

1. INTRODUCTION

In recent years, microblogging services such as Twitter¹, Tumblr², and Plurk³, are now one of the most frequently used tools for online communication. For instance, Twitter users can broadcast and share information about their activities, opinions, and statuses

¹<https://twitter.com>

²<https://www.tumblr.com/>

³<http://www.plurk.com/top/>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

CIKM '14 Nov 03-07 2014, Shanghai, China

ACM 978-1-4503-2598-1/14/11.

<http://dx.doi.org/10.1145/2661829.2661844>.

in short messages (tweets), up to 140 characters long, using smartphones at anytime and anywhere. Despite the useful information on Twitter, there still exists a lack of Twitter users' requirements. That is, tweet senders are difficult to obtain useful information (e.g., bus timetables, sightseeing maps, etc.) about their current location, since they are often posted on Web pages, the tweet senders will have to access each Web page using smartphones. Meanwhile, Twitter users can gain information about current scenarios from tweet senders; however, Web users cannot simultaneously obtain this information through tweets while they browse Web pages. Furthermore, Twitter supports only limited communication between its users (i.e., followees and followers). Web users cannot access the most recent information from Web pages of popular places (e.g., tourist spots, shopping centers, etc.) because the Web pages are not updated in real time. Therefore, it is important to focus on associating Twitter feed and Web pages to facilitate simultaneous communication for Twitter and Web users.

Although several techniques on social communication have been studied [4, 6, 3], these studies have focused on content-based communication between users who browse the similar Web pages, they do not solve the mentioned issues without microblogging. We have proposed *TWinChat*, a novel simultaneous communication system between Twitter users and Web users in real time through both the contents of tweets and Web pages. To achieve this, we first acquired geo-tagged tweets based on content analysis and region selection. Therefore, our method can detect tweets if they are related to, or nearby, a target location, even though that they do not include the location names, or it can also detect tweets from Twitter users who are not in the target location. Furthermore, the proposed method can filter out tweets from the target location for which the content is not related to the target location. The system then maps the acquired tweets and Web pages by generating queries based on location names is explained in Section 3. As depicted in Figure 1, *TWinChat* has two features: 1) streaming tweets are associated with each Web page that are presented to Web users based on the common locations; 2) a chat box is attached to each Web page so that Web users can communicate with the Twitter users who follow an account of our system in real time.

2. SYSTEM OVERVIEW

To use *TWinChat*, which is on the basis of existing Web services, Twitter users first required to follow an account⁴ of our system, as

⁴<https://Twitter.com/@RtQAService>

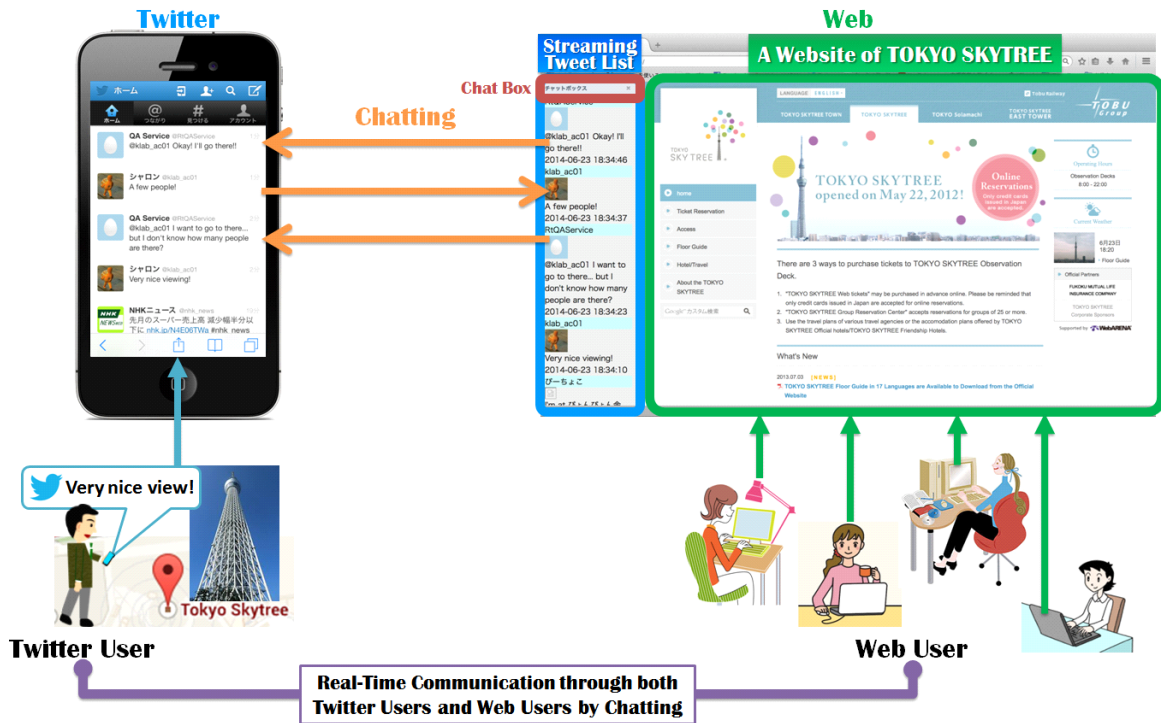


Figure 1: System snapshot of *TWInChat*

followers of our service, and Web users first required to simply install a toolbar (a Firefox add-on). Once a Web user browses a Web page, the system records the information into a server database, which is used for mapping tweets from Twitter users and the Web page, and he can chat with the Twitter users and other Web users. In our system, anonymous of all messages (tweets) can be maintained through a server.

The flow of *TWInChat* is described as follows:

1. A Web user selects a Web page to browse, the system then returns a streaming tweet list, in which tweets from Twitter users with respect to the Web page are presented in the Web browser in real time.
2. When the Web user submits a message in the chat box, the system presents the message in the streaming tweet list, all other users who browse the same Web page, or the Twitter users whose tweets are presented in the streaming tweet list, can receive the message.
3. When the Twitter user replies the message of the Web user through Twitter service and the server of the system; the system presents the reply that are related to the Web page in the streaming tweet list in real time.

An example is shown in Figure 1, which depicts a Web user browsing an official Website of Tokyo Skytree in the Web browser of our system. Streaming tweets, e.g., “Very nice view!” located in the Tokyo Skytree, are associated with the Web page by generating a query based on a common location name, “Tokyo Skytree,” even though the tweets do not include the location name. This allows the Web user to gain insight into the congestion level or gain impressions of the Tokyo Skytree from the presented tweets in real time. On the other hand, this allows the Twitter user to obtain useful information or knowledge such as observation deck or tallest tower

rank of the Tokyo Skytree from the Web users by chatting through our system.

3. MAPPING FUNCTION

3.1 Acquisition of Streaming Tweets

A conventional method based on a content analysis of tweets and hashtag search [5], it can detect the tweets of Twitter users who are not in the locations in the real world. Also, tweets can be acquired by identifying the location information from geo-tagged tweets [2]. However, many detected tweets are still not related to the locations of them, it is difficult to report the current situation from the detected tweets. Therefore, we first obtain geo-tagged tweets except duplicates from a specified region by using Streaming APIs (left part of Figure 2), and we analyze the content of the tweets focusing on nouns by a morphological analysis. The specified region is determined by a northeast point and a southwest point, then, we can obtain streaming tweets in a rectangular region surrounding these two points. Next, we detect location names within a radius d of a region by using Google Places API v3⁵, from the latitude and longitude information of the obtained streaming tweets, and we empirically set $d = 5m$ by considering the movement of Twitter users. Then, our server database manages {Twitter user ID, icon URL, latitude, longitude, location name, tweet, word set, acquisition time} in a certain time (central part of Figure 2).

3.2 Mapping of Tweets and Web Pages

For selecting tweets related to the detected location names, we extract high frequency words from the tweets as feature terms of the locations. Then, we selected the tweets that contain many feature

⁵<https://developers.google.com/place>

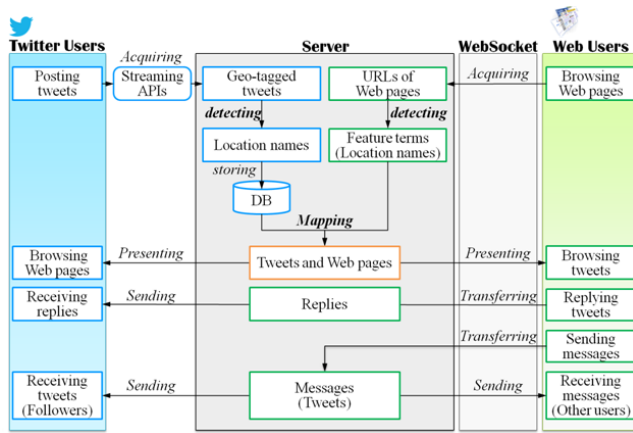


Figure 2: System configuration diagram

terms. For this, we obtain a total amount n of tweets in a certain time within a radius d of a region based on a given location of the acquired tweet t . The average of the frequency of each word i that appears in t is then calculated by the following equation:

$$\sum_{i=1}^q \frac{\# \text{tweets with } i}{n} \times \frac{1}{q} \quad (1)$$

Here, q denotes the total number of words that appear in tweet t . If a value of Eq. (1) is more than the empirical threshold value, t related to the location of it is selected.

To extract location names from Web pages, we acquire URLs of the Web pages that Web users are browsing (right part of Figure 2). Then, we extract high frequency words from snippets of the acquired URLs as feature terms of the Web pages and detect location names of the Web pages by using a morphological analyzer.

Based on the above, when Twitter users post tweets and a Web user browses a Web page, the system can obtain and present the tweets that are relevant based on a common location name from the Web page and the tweets. In this case, the server database of the system stores the obtained tweets, the obtained Web pages, and the detected common location names (central part of Figure 2).

4. COMMUNICATION FUNCTION

4.1 Server Side

The server side is built using Apache httpd 2.4, Java, PHP 5.5. The servlet can perform parallel processing for multiple requests, since it makes a “user thread” for each request. When the user thread receives the requests from the clients, it parses them, performs the corresponding action, returns information to the requesting user, and sends information to other users. The browsing, acquiring, and communication logs are stored in the server database using SQLite 3.8.5.

4.2 Client Side

The client side is implemented as a toolbar (browsing plug-in). The question answering interface is programmed using XUL (extensible user interface language) and the development is programmed using JavaScript. A Twitter user is connected to the server using Twitter service, and a Web user is connected to the server using an asynchronous communication program running in a WebSocket⁶.

⁶<http://gihyo.jp/dev/feature/01/websocket/0001>

5. DEMONSTRATION

With *TWinChat*, Twitter users and Web users can communicate with each other in real time by chatting (see Figure 1), and a demonstration video is shown in [1]. *TWinChat* could present tweets from the Twitter users and messages from the Web users within 2 seconds, respectively; and the Web users could reply the tweets to the Twitter users within 3 seconds.

5.1 Tweet Presentation into a Web Page

Web users can browse a Web page and simultaneously obtain a list of streaming tweets that with respect to the Web page in real time. The server database of our system stores the acquired tweets, the most recently viewed Web pages, and the detected common location names from them.

Since *TWinChat* presents tweets into a Web page that Web users are browsing, the Web users can gain the current situation in real time, not only through the presented tweets on the Web page but also through Twitter users of them.

5.2 Real-Time Cross-Media Communication

Twitter users and Web users can chat with each other in real time. *TWinChat* allows the Web users to send messages in the chat box to the Twitter users of the presented tweets or other Web users who browse the same Web page. If the Web user has a question about the current situation while browsing a Web page, he can immediately ask the Twitter users by entering the question in the chat box. The Twitter users, who follow the account of *TWinChat*, can receive the question and reply it to the Web user through Twitter service. The replies from the Twitter users can be simultaneously presented on Twitter and streaming tweet list of our system.

Meanwhile, if the Twitter user has a question about something on the Web page, he can immediately ask the Web users by posting the question on Twitter. The Web users can receive the question and send messages as tweets, reply to the Twitter user through the chat box. The messages from the Web users can be simultaneously presented on streaming tweet list of our system and Twitter.

6. ACKNOWLEDGMENTS

This work was partially supported by SCOPE (the Ministry of Internal Affairs and Communications of Japan), and a Grant-in-Aid for Scientific Research (B)(2) 26280042 (the Ministry of Education, Culture, Sports, Science, and Technology of Japan).

7. REFERENCES

- [1] Demonstration Video Page of Developed *TWinChat*. <http://klab.kyoto-su.ac.jp/~yasui/TWINChat.html>.
- [2] G. Ference, M. Ye, and W.-C. Lee. Location recommendation for out-of-town users in location-based social networks. In *Proc. of CIKM 2013*, pages 721–726, 2013.
- [3] S. Fragoso, R. R. Rebs, and D. L. Barth. Interface affordances and social practices in online communication systems. In *Proc. of AVI 2012*, pages 50–57, 2012.
- [4] Y. Shiraishi, J. Zhang, Y. Kawai, and T. Akiyama. Simultaneous realization of page-centric communication and search. In *Proc. of CIKM 2012*, pages 2719–2721, 2012.
- [5] S. Tajima and T. Ushijima. A method for composing ad-hoc following networks on twitter for sharing information among event participants. *Journal of ADADA*, 17(4):119–124, 2013.
- [6] B. Wang, C. Wang, J. Bu, C. Chen, W. V. Zhang, D. Cai, and X. He. Whom to mention: Expand the diffusion of tweets by @ recommendation on micro-blogging systems. In *Proc. of WWW 2013*, pages 1331–1340, 2013.