# TRENDSPOTTER DETECTION SYSTEM FOR TWITTER

Wataru Shirakihara
*Graduate School of Information Science and Electrical Engineering, Kyushu University, Fukuoka City, Japan*

Tetsuya Oishi[†], Ryuzo Hasegawa[‡], Hiroshi Hujita[‡], Miyuki Koshimura[‡]
[†]*Research Institute for Information Technology, Kyushu University, Fukuoka City, Japan*
[‡]*Faculty of Information Science and Electrical Engineering, Kyushu University, Fukuoka City, Japan*

Keywords: Twitter, Data stream algorithms, Information recommendation.

Abstract: It is too difficult for us to find out trends with search engines. Twitter, a popular microblogging tool, has seen a lot of growth since it launched in October, 2006. Information about the trends is posted by many twitterers. If we find out trendspotters from twitterers, and follow them, we can get it more easily. Our trendspotter detection system uses the burst detection algorithm, and we verified its effectiveness for Twitter's posts. We succeeded in detecting the 24 trendspotters by 5277 users.



## 1 INTRODUCTION

Over the past few decades Internet has developed rapidly. Many people use Internet when they want to get some pieces of information, but information we find on the Internet is out of date unless it is renewed. Therefore it is too difficult for us to find out trends with search engines.

Twitter, a popular microblogging tool, has seen a lot of growth since it launched in October, 2006(Java et al., 2007). According to Netratings, in Japan, Twitter had about 4,730,000 users as of January, 2010[1]. However, the figures include only the users who use Twitter from Twitter's website[2], that is to say, the users from mobile phone, or Twitter client are not included. To sum up, actually, there are much more users(Kanda, 2009).

Twitter has a lot of differentiating good factors from other SNS or blogging, and is used by many users as an area of exchange of information.

Users (Twitterers) can broadcast an unlimited amount of messages (tweets) to a group of other Twitterers who have opted to subscribe to these broadcasts (followers). Twitterers also recieve broadcasts from other users. Individual tweets are made within a limit of 140 characters(Starbird and Palen, 2010).

Twitter has four main characteristics. The first is that twitterers can 'retweet' someone else's post by copying the post and the person's username. The retweeted post is shared with all of their followers.

The second is that the hashtag convention (#[hashtag term]) is used inline to call out user-chosen keywords. Hashtags tag or markup a tweet to help others understand the content context, as well as support keyword term-searching.

The third is that the posts may be directed to a particular person by putting an @username at the biginning of the post. Even though the post is directed to a person, others can still view it(Ehrlich and Shami, 2010).

The fourth is that twitterers can use Twitter's API. Many client tools were invented with API. API allows other web services to integrate with Twitter. Buzztter[3], one of the Twitter's web services, shows buzz terms in Twitter.

In Twitter, because of its characteristics, information about the trends is posted by many twitterers. If we find out trendspotters from twitterers, and follow them, we can get it more easily. The purpose of this paper is to develop a system that detects the trendspotters.

For detecting trendspotters, our system uses burst detection algorithm(Fujiki et al., 2004). In blogging or bulletin board, a particular phrase appears fre-

[1]http://www.netratings.co.jp/New_news/News02242010.htm

[2]http://twitter.com

[3]http://buzztter.com

Shirakihara W., Oishi T., Hasegawa R., Hujita H. and Koshimura M..
TRENDSPOTTER DETECTION SYSTEM FOR TWITTER.
DOI: 10.5220/0003183806250628
In *Proceedings of the 3rd International Conference on Agents and Artificial Intelligence* (ICAART-2011), pages 625-628
ISBN: 978-989-8425-40-9

quently when certain topic is focused. The reason for this phenomenon is that a relevant proper name in the topic comes up in many users posts. The burst detection algorithm, proposed by Kleinberg, the posts are treated as the document stream (Kleinberg, 2003).

The document stream is a set of documents with the time it's posted, such as newspaper articles. This algorithm detects the time when the number of documents are sharply increased.

The paper is organized as follows: in section 2 , we discribe the practical move of this system. Next, in Section 3 we discribe the result the system gets, and consider it. Additionally, we determine that information about the trends will still be posted by the trendspotters even after they were found out. Finally, we conclude and discribe the future view in Section 4.

## 2 SYSTEM OVERVIEW

Our system is overviewed as follows.

1. The system gets a buzz word *s* from Buzztter. Buzztter shows different kinds of buzz words, in which some everyday talks such as "Good morning" or "lunch" are included. However, these everyday talks actually are not buzz words. Therefore, we ignore them and pick out the other buzz words.
2. Search the posts which include the buzz word *s* with Twitter's API, and get usernames and posted time.
3. Detect the time when the posts including buzz word *s* are sharply increased with the burst detection algorithm.
4. Extract the earliest 20 posts in detected burst, and the twitterers who posted them are recognized as the trendspotters for buzz word *s*.
5. Repeat this sequence and pick out the trendspotters who posted a significant number of buzz words.

## 3 EXPERIMENTS

We evaluate our system by three experiments as follows.

1. Verification of Effectiveness (section 3.1)
   We verify whether the burst detection algorithm is effective for Twitter's posts.

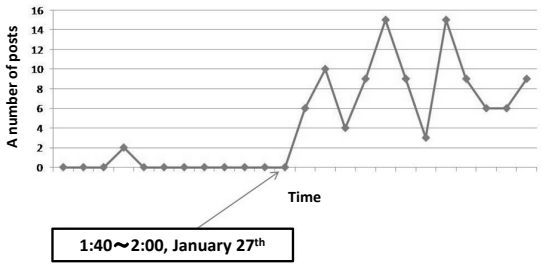


Figure 1: The number of posts including the buzz word 1.

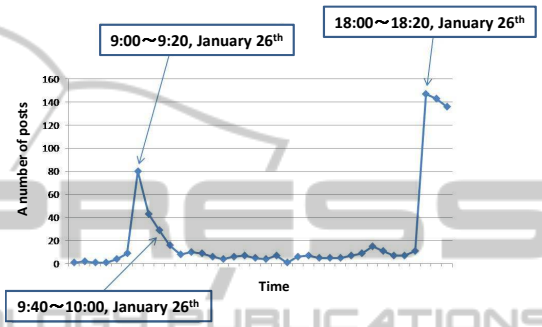


Figure 2: The number of posts including the buzz word 2.

2. Detecting Trendspotters (section 3.2)
   We examine whether our system surely gets trendspotters who posted a significant number of buzz words.
3. Trendspotter's Potency (section 3.3)
   We verify whether the trendspotters we got in section 3.2 will still be trendspotters even after they are found out.

### 3.1 Verification of Effectivness

To verify the effectiveness of burst detection, we choose three example buzz words and discribe the results that the burst detection algorithm outputs.

Figure 1 - figure 3 shows a graph with a number of posts including the buzz word on the y-axis, and the time on the x-axis.

Our system's outputs are as follows.

- In figure 1, the time after 2:00, January 27th was burst.
- In figure 2, there are two distinct bursts. The first is seen from 9:00 to 10:00, and the second is after 18:00.
- In figure 3, there are also two bursts. The first is seen from 21:20 January 31th to 1:40 February 1st, and the second is after 7:00 February 1st.

The conclusion of these experiments is as follows.

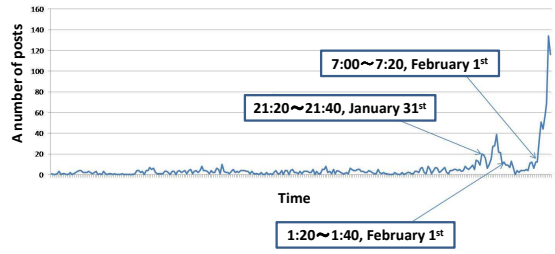


Figure 3: The number of posts including the buzz word 3.

- The burst detection algorithm recognizes the sharp increase of the posts as a burst.
- It does not recognize the little increase of the posts as a burst.
- When the number of posts increased and decreased in a short time, it recognizes them as the same burst.
- It can detect the bursts from data in the long time period.

From these considerations, it is clear that the burst detection algorithm is effective for Twitter's posts.

## 3.2 Detecting Trendspotters

We examine whether our system surely gets trendspotters who posted a significant number of buzz words. This experiment's procedure is as follows.

- Pick out 200 buzz words from Buzztter.
- Search the posts which include buzz words with Twitter's API, and get usernames and posted time, up to 1500 for each buzz word.
- Experimental period is from January, 2010 to February, 2010.
- Detect the bursts for these buzz words by burst detection algorithm. Then extract the twitterers who posted the earliest 20 posts in detected burst.
- We examine how many buzz words each trendspotter posted.

Our system detected 5277 trendspotters. Table 1 shows a number of trendspotters for *N* buzz words and percentage of 5277 trendspotters.

This table shows that trendspotters for more than three buzz words were 1.52 of the total. Thus, our system detected the trendspotters for many buzz words by a lot of twitterers.

Table 1: The number of trendspotters for *N* buzz words and percentage of total.

| buzz words(*N*) | number | percentage(%) |
|---|---|---|
| 1 | 4734 | 89.71 |
| 2 | 463 | 8.77 |
| 3 | 56 | 1.06 |
| 4 | 12 | 0.23 |
| 5 | 9 | 0.17 |
| 6 | 3 | 0.06 |

## 3.3 Trendspotter's Potency

Our system detected the trendspotters. Now we verify whether they will still be the trendspotters even after they are found out.

- Pick out 110 buzz words from Buzztter.
- Search the posts which include buzz words with Twitter's API, and get usernames and posted time, up to 1500 for each buzz words.
- Experimental period is from February 5th, 2010 to February 7th, 2010.
- Detect the bursts for these buzz words by the burst detection algorithm. Then extract the twitterers who posted in detected bursts.
- We examine how many buzz words each twitterer posted in detected bursts, and compare the average of all twitterers with the average of the top 24 trendspotters in experiment2.

Our system detected 34381 twitterers. Table 2 shows the number of twitterers who posted *N* buzz words in bursts, and percentage of 34381 twitterers.

Table 2: The number of users who posted *N* buzz words in bursts and percentage of total.

| buzz words(*N*) | number | percentage(%) |
|---|---|---|
| ‘2 | 31174 | 90.67 |
| 3‘5 | 2916 | 8.48 |
| 6‘8 | 227 | 0.66 |
| 9‘11 | 46 | 0.13 |
| 12‘ | 18 | 0.05 |

Table 3 shows the number of top 24 twitterers who posted *N* buzz words in bursts, and percentage of 24 trendspotters.

Comparing the table 2 with the table 3, top 24 trendspotters posted more buzz words than the other twitterers. In all twitterers, the percentage of twitterers who posted more than 6 buzz words is less than 1%. On the other hand, in the top 24 trendspotters, it is 33.4%. Additionally, the average buzz words

Table 3: The number of the top 24 trendspotters who posted *N* buzz words in bursts and percentage of total.

| buzz words(*N*) | number | percentage(%) |
|---|---|---|
| ‘2 | 8 | 33.3 |
| 3‘5 | 8 | 33.3 |
| 6‘8 | 4 | 16.7 |
| 9‘11 | 3 | 12.5 |
| 12‘ | 1 | 4.2 |

per all twitterers who posted the buzz words in the bursts is 1.43, while the average buzz words per top 24 trendspotters is 4.54. Thus, the trendspotters our system detected still be the trendspotters even after they are found out.

## 4 CONCLUSIONS

In this paper, we have presented a system that detects the trendspotters, and enables us to easily find out the trends with Twitter. We found out that the burst detection algorithm is applicable to Twitter’s posts. In addition, it appropriately recognized a sharp increase of the posts as a burst. Therefore, the burst detection algorithm is suitable to detect the trendspotters. Moreover, the trendspotters our system detected still post more buzz words than the other twitterers even after they are found out. Thus, our system is effective to find out the trends.

In future research, we are going to develop the method of categorizing the trendspotters. Twitterers are divided into the clusters such as politics, art, and sports. Then, the system detects the trendspotters in each clusters. However, the rules of clustering are needed.

For finding these rules, we will work out Twitter’s text mining. In particular, we take into account the following 6 points.

- Connection of ”follow”
- Reply (the posts directed to a particular person by putting an @username)
- Hashtags
- Posted time
- Distance between the sentenses in different posts.
- Bursts

Figure 4 shows an overview of the system. For calculating them, we use Hadoop, a framework for distributed processing. Then, we use HBase and Cassandra for storing calculated results.

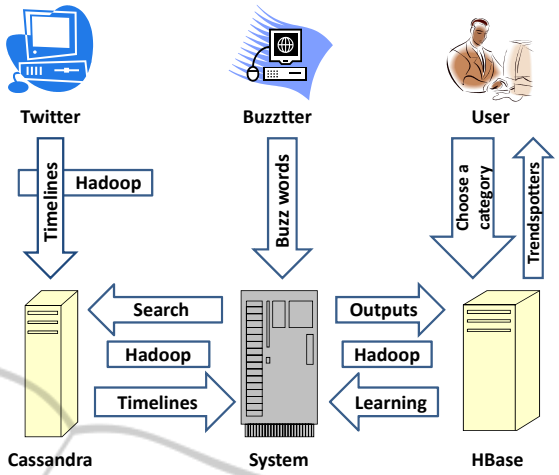


Figure 4: Image of future system.

## ACKNOWLEDGEMENTS

This work was supported by JSPS KAKENHI (21500102).

## REFERENCES

Ehrlich, K. and Shami, N. (2010). Microblogging Inside and Outside the Workplace.

Fujiki, T., Nanno, T., Suzuki, Y., and Okumura, M. (2004). Identification of bursts in a document stream. In *First International Workshop on Knowledge Discovery in Data Streams*, pages 55–64. Citeseer.

Java, A., Song, X., Finin, T., and Tseng, B. (2007). Why we twitter: understanding microblogging usage and communities. In *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*, pages 56–65. ACM.

Kanda, T. (2009). *Twitter Revolution*. Softbank.

Kleinberg, J. (2003). Bursty and hierarchical structure in streams. *Data Mining and Knowledge Discovery*, 7(4):373–397.

Starbird, K. and Palen, L. (2010). Pass It On?: Retweeting in Mass Emergency. In *Proceedings of the 7th International ISCRAM Conference–Seattle*, volume 1.