Detecting Trend and Bursty Keywords Using Characteristics of Twitter Stream Data

Daehoon Kim[1], Daeyong Kim[1], Seungmin Rho[2] and Eenjun Hwang[1,*]

[1] School of Electrical Engineering, Korea University, Seoul, 135-701, South Korea
[2]Division of Information and Communication, Baekseok University, Chonan, 330-704, South Korea
[1]{kdh812, ritgd05, ehwang04}@korea.ac.kr, [2]smrho@bu.ac.kr

Abstract

Twitter is a very popular online social networking and micro-blogging service that enables its users to post and share text-based messages called tweets. The numbers of active users and tweets generated daily are enormous and hence they, collectively, can give crucial clues to several interesting problems such as public opinion analysis and hot trend detection. Especially, to find out hot issues and trends from tweets, detection of popular keywords is very important. In this paper, we propose a new scheme for detecting trend and bursty keywords from Twitter stream data. Our scheme is very robust in that it can handle typical usages such as various abbreviations, minor typing errors and spacing errors that occur very frequently when writing tweets on various mobile devices. We implemented a prototype system and performed various experiments to show the effectiveness of our scheme.

Keywords: *Bursty Keyword Detection, Twitter, SNS, Keyword*

1. Introduction

Recently, one of the world’s hottest IT market issues has been the smartphone. The smart phone boom has made various social network services such as Facebook and Twitter social phenomena. Twitter is an online social networking and micro-blogging service that enables its users to post and share text-based messages of up to 140 characters, known as "Tweets". It has gained worldwide popularity, with over 500 million active users (as of 2012) generating over 340 million tweets daily. Due to this popularity, it is often described as "the SMS of the Internet." Unregistered users can read tweets, while registered users can post tweets through the website interface or a range of apps for mobile devices [1]. Since Twitter enables people to share information in real time without any constraint on time or location, it has attracted a lot of attraction as a new means for social communication.

On the other hand, we can observe several characteristics from typical tweets, which are mainly due to the small size screen of most mobile devices and the maximum length limitation of tweets. Such characteristics include various word variants such as abbreviations, typing errors and spacing errors. So, we need to handle them very carefully for more accurate detection of trend and bursty keywords.

In this paper, we propose a robust scheme for detecting trend and bursty keywords from Twitter stream data. To detect trend and bursty keywords, we first select candidate keywords from tweets by performing simple syntactic feature–based filtering. And then, we merge various keyword variants using several heuristics and select bursty keywords based on the term frequency. By tracing the popularity transition of such trend keywords, we determine

[*] Corresponding Author

bursty keywords. We implemented a prototype system and performed various experiments to show the effectiveness of the scheme. We report some of the results.

The remainder of the paper is organized as follows. In Section 2, we discuss the background of this study and related work. In Section 3, we describe our scheme for detecting keywords in detail. Section 4 presents experimental results. The conclusion is briefly discussed in Section 5.

2. Related Work

Extraction of keywords from tweets is a little different from traditional keyword extraction in the information retrieval arena. This is because tweets have different characteristics compared to traditional documents. Usually, tweets are short in length, possibly a few short sentences, but the numbers of active users and tweets generated are enormous. Also, due to the inherent size limitation of tweet, acronym and other types of abbreviations are used very frequently. In addition, compared to traditional keyword mining, trend and bursty keyword detection from tweets has different characteristics. Generally, a keyword of a document represents one of possibly multiple main topics in the document. However, bursty or trend keyword detection requires popularity analysis over some time period. Thus, detecting bursty keywords requires time domain analysis. That is, to detect bursty and trend keywords, we need to solve two problems: One is to extract popular or trend keywords from tweets, and the other is to determine which trend keywords are bursty keywords by temporal transition analysis.

2.1. Detecting Trend Keywords

SNS messages can be treated as traditional text document or as a streaming data or time sequence data. Some of popular methods for treating SNS messages as traditional text are as follows. Latent Dirichlet allocation (LDA) [2], a generative probabilistic model for collections of discrete data such as text corpora, has shown a good performance in general text mining. Ramage et al. presented a scalable implementation of a partially supervised learning model that maps the content of Twitter feeds into dimensions [3]. They characterized users and tweets using this model, and presented their results on two information consumption oriented tasks.

SNS messages can be viewed as time sequence data or streaming data from the point of view of the time domain. Many studies have proposed methods for analyzing various types of SNS messages. Kumaran, *et al.*, proposed a text classification method for detecting new events [4]. They improved the detection performance by using text classification techniques and named entities in a new way. Their scheme involves the application of hierarchical and non-hierarchical document clustering algorithms that focus on the exploitation of both content and temporal information, and the use of a single pass clustering algorithm and a novel thresholding model that incorporates the properties of events as a major component. Sayyadi, *et al.*, built a network of keywords based on their co-occurrence in documents [5]. They proposed a new event detection algorithm that creates a keyword graph and uses community detection methods similar to those used for social network analysis to discover and describe events. Lastly, Google Trends [6] shows the total search-volume of certain keywords based on Google Search. This service provides daily trend keywords and specific trends based on time or location. In particular, it provides an interface for comparing some trend keywords. Still, these methods are not suitable for real-time processing and focus on providing static trend summary.

2.2. Detecting Bursty Keywords

Recently, several works have been proposed for extracting trends in real time. Mathioudakis et al. proposed TwitterMonitor for the detection of trends over Twitter streams [7]. To do that, TwitterMonitor first identifies emerging topics on Twitter in real time and provides meaningful analytics for accurate description of each topic. The method discovers topic trends by detecting bursty single tags. However, it is difficult to get information about various events using only single tags. Thus, Alvanaki et al. presented the enBlogue system, an approach for automatically detecting emergent topics by detecting shifts in tag correlations as they dynamically arise [8, 9].

In addition, several works have been done to provide bursty keywords from text streams that arrive continuously over time. Fang, *et al.*, proposed a bursty keyword discovery scheme using co-occurrence and time information of words [10]. They generate pairs of co-occurring words by applying OpenNLP, and extract bursty keywords by analyzing word clusters within a specified time range. Du, *et al.*, introduced an unsupervised learning method for detecting bursty topics based on user relationship [11]. They detected topics by calculating term weights that correlate with user weight and account information and user weight using improved PageRank algorithm to model the user's life cycle. Then, they perform machine learning for them. Kleinberg proposed a text data mining method for detecting topics that suddenly rise in frequency [12]. The approach uses infinite-state automaton on modeling text stream to analogize models from queuing theory. It creates hierarchical structure of the set of bursts on overall streams. Platakis, *et al.*, introduced a method for discovering hot topics in the blogosphere [13]. They specified domain as short contents from user created blogs, and found bursty terms based on time stamp produced by blogosphere and applied a burst model algorithm. Takahashi et al. proposed a burst model for detecting bursty topics in a traditional topic model [14]. Based on the burst model that is for burst topics but not for whole topics, they applied the burst model to the topics estimated by the traditional dynamic topic model. Vakali, *et al.*, proposed a framework for detecting bursty keywords called Cloud4Trends [15]. The framework contains collection of user generated content through micro-blogging and blogging applications and trend analysis via the detection of bursty keywords on clusters divided by topics. Overall, aforementioned methods provide bursty keywords pretty well. However, they usually require significant amount of time and have difficulty in providing their temporal characteristics.

3. System Architecture

Figure 1 shows overall architecture of our proposed system. The first task is to collect user tweets via the Twitter Streaming API and extract candidate keywords from them by calculating their term frequency (TF). Then, various words variants are considered to identify and merge semantically same keywords. Finally, trend keywords are determined based on their rank and bursty keywords are selected from them based on the temporal pattern of their popularity.

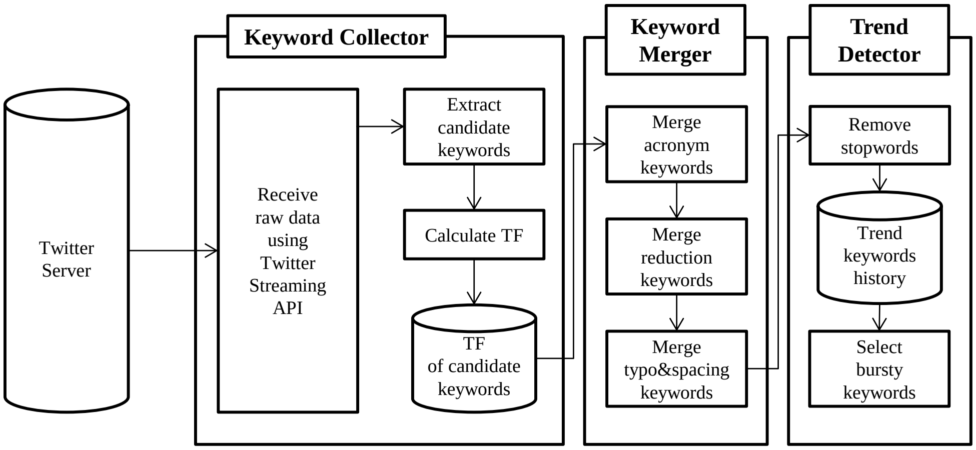
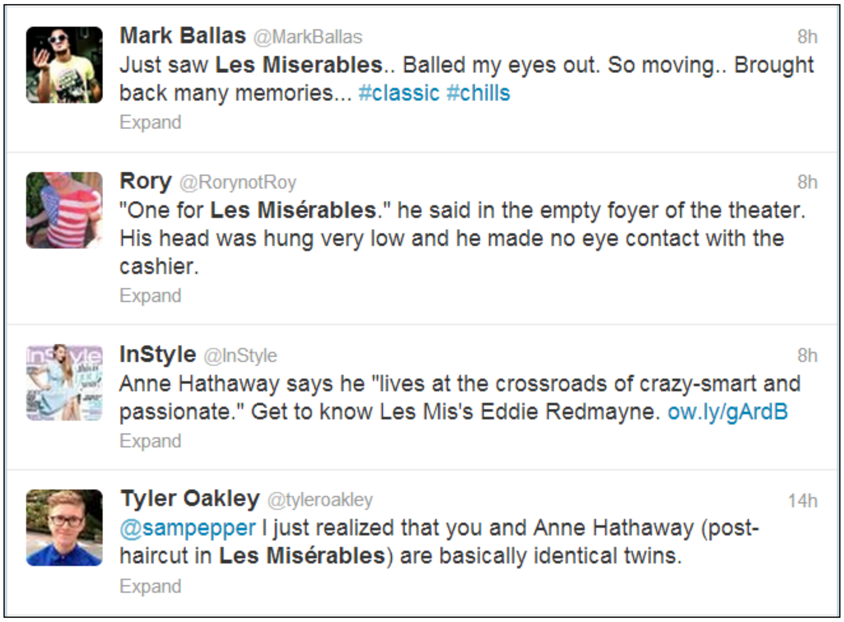


Figure 1. Overall System Architecture

3.1. Collecting Candidate Keywords

For fast and effective keyword detection, we investigated a large number of user tweets and observed that most keywords are composed of words starting with a capital letter or enclosed by a pair of quotation marks. For instance, the tweets in Figure 2 contain several candidate keywords such as “Les Miserables,” “Anne Hathaway,” and “Eddie Redmayne.” As a consequence, in this paper, we define a candidate keyword as a series of words starting with a capital letter or enclosed by a pair of quotation marks. In addition, we add three more special cases. The first case is when the first letter is small but the second letter is capitalized, as in “iPhone” and “jQuery.”



Mark Ballas @MarkBallas 8h
Just saw **Les Miserables**.. Balled my eyes out. So moving.. Brought back many memories... #classic #chills
Expand

Rory @RorynotRoy 8h
"One for **Les Misérables**." he said in the empty foyer of the theater. His head was hung very low and he made no eye contact with the cashier.
Expand

InStyle @InStyle 8h
Anne Hathaway says he "lives at the crossroads of crazy-smart and passionate." Get to know Les Mis's Eddie Redmayne. ow.ly/gArdB
Expand

Tyler Oakley @tyleroakley 14h
@sampepper I just realized that you and Anne Hathaway (post-haircut in **Les Misérables**) are basically identical twins.
Expand

Figure 2. Example of Tweets

The second case is when a number precedes or follows some candidate keyword, in which case we consider them as one keyword. In many cases, the number is meaningful. For example, in the tweet, “If anybody is selling iPhone 5, iPhone 4, a blackberry or an iPad in kennishead flats contact the police. Stolen from my house, please RT.” “iPhone 5” and “iPhone 4” would be more meaningful as candidate keywords than “iPhone.” The third case is when a word consists of a number and three and more characters following the number. For example, “49ers rookie @LaMichaelJames was forced to grow up fast during his senior year of high school. http://bit.ly/RChIHX.” Here, “49ers” is a part of the NFL football team, “San Francisco 49ers.” In many tweets, this type of abbreviation is frequently used instead of its full name.

At this point, we note that Twitter has a special feature called hash tag for categorization and search purposes, which we exclude as a candidate keyword because, in many cases, it does not any trend in spite of its frequent appearance in tweets. To avoid ambiguity and complexity caused by case-sensitiveness, we capitalize all the candidate keywords for further processing. Also, in this work, we perform the collection of user tweets and their analysis for candidate keywords every hour.

3.2. Merging Keywords

In this step, we merge possible variants of candidate keywords and adjust their term frequencies accordingly. The variants we consider in this paper include acronyms, typos, spacing and word expansion. When merging variants of candidate keywords, we also adjust their term frequencies accordingly. In this way, we can evaluate the term frequency of each candidate keyword more accurately. Table 1 shows the keyword variants we considered and their examples, which were observed in user tweets between 1/3/2013 and 1/6/2013. In the table, we indicate all the word variants in bold. We merge variants into their full keyword and accumulate their term frequencies onto that of the full keyword.

Table 1. Keyword Variants

Variant type	Full keyword	Variant
Acronym	NEW YEARS EVE CALL OF DUTY NEW YORK CITY DESIGNATED DRUNK DRIVER	**NYE** **COD** **NYC** **DDD**
Reduction	CHRIS BROWN WATERLOO ROAD STEVEN GERRARD LAS VEGAS	CHRIS WATERLOO GERRARD VEGAS
Typo	HIP HOP CO**L**LIN KLEIN	HIP **P**OP COLIN KLEIN
Spacing	KSTATE KEVINPRINCE BOATENG	KSTATE KEVINPRINCE BOATENG

For effective merging, we have to consider the variants mentioned above. The first variant is an acronym. Due to the size limitation of a tweet, acronyms are used quite frequently in tweets. For example, “New York City,” a famous city in US, is often written as “NYC” in tweets. To handle acronyms, we first classify candidate keywords by the number of words in the keyword. For all keywords comprising at least three words, we construct their acronym and compare them with one-word keywords. For example, for the keyword “English Premier League,” we construct “EPL” as its acronym. If the acronym appears in the TF matrix, then we have to add the term frequency of the acronym onto that of “English Premier League.”

The second variant type—keyword reduction—can be processed based on the number of words. A typical example is a person’s name, which is composed of first name, possibly middle name, and family name. In the real world, several different formats are used for a person’s name. For example, for “Steven George Gerrard,” a famous singer, “Gerrard,” “Steven Gerrard,” *etc.*, are also used. These names can be considered to indicate the same name. Sometimes, we may have an ambiguous situation in this step. For example, “Manchester” appears in both “Manchester United” and “Manchester City.” Since the two keywords are not related, it is not clear where to merge “Manchester.” To eliminate this ambiguity, we assume that words of the same meaning will appear at a similar frequency and merge the keyword onto the full keyword that has a similar frequency.

The third and final variant type can be handled by considering the length and character histogram of keywords. Due to the limited input facility of mobile devices, typos and spacing errors occur very frequently in tweets. For accurate trend keyword extraction, these types of errors are resolved in the merge step. To do this, we first create a histogram for the characters in the keywords and compare them to determine typos. For example, “RHIANNA” is a common typing error for “RIHANNA.” In this case, they give the same character histogram, which is [A:2, H:1, I:1, N:2, R:1]. Another example is “BIEBER” and “BLEBER” where their character histograms are [B:2, E:2, I:1, R:1] and [B:2, E:2, L:1, R:1], respectively. Spacing errors can be detected in a similar way. Keywords with a spacing error give the same character histogram but their lengths are different. For instance, “X FACTOR” and its variant with spacing error, “XFACTOR” give the same character histogram and their lengths differ by one. We calculate the differences in the number of characters and lengths of two words, and use their sum as criterion on the typo-spacing error. The reason for this is that in many cases, typo and spacing errors occur simultaneously. Extensive experiments show that threshold three for the sum gives the best F-measure value.

3.3. Detecting and Selecting Keywords

After merging keywords, we are now ready to detect a set of trend keywords which is a superset of bursty keywords. In Section 3.1, we assumed that all candidate keywords start with a capital letter. Candidate keywords were merged together according to their variant types for the correct calculation of term frequency. Bursty and trend keywords are selected from those candidate keywords. Our simple assumption eliminates many unnecessary words very quickly. Also, we used predefined 315 stopwords to remove frequently used but semantically meaningless words. Even though this number is much smaller compared traditional approach, it turns out to be enough since we consider words starting with a capital letter. After removing stopwords, remaining candidate keywords are sorted by the term frequency and their first top k keywords are selected as trend keywords. This means that these keywords are the most popular in that time period. In addition, rookie keywords that attract user attention abruptly are select as bursty keywords.

To select the bursty keywords, we define a new metric called *busty ratio* as follows:.

$$Ratio_{bursty}(K,t) = \frac{TF_K(t) - TF_K(t-1)}{TF_K(t)} \qquad (1)$$

Here, K is the target keyword, and $TF_K(t)$and $TF_K(t\text{-}1)$ are the TF value of the keyword K on time t and t -1, respectively. Namely, $Ratio_{bursty}(K\ ,t)$is the measure on the change of term frequencies on time t for the keyword K. For example, for a keyword W, if $TF_W(t)$ is 300 and $TF_W(t\text{-}1)$ is 270, then the busty ratio will be 0.1. In this paper, we consider trend keywords whose bursty ratio is greater than 0.5 as bursty keywords.

4. Results and Discussion

We implemented a prototype system based on our scheme and conducted a variety of experiments to evaluate its performance. Our system was implemented using MathWorks MATLAB 2011b. The experiment was performed on a desktop PC with an Intel Core 2 Quad 2.67 GHz processor, 8 GB of RAM and the Windows 7 Enterprise operating system.

4.1. Collecting Candidate Keywords

In the first experiment, we measured how accurate our scheme was in selecting keywords from user tweets. For the comparison, we used 212 keywords from Google trends [6] between 12/7/2012 and 1/6/2013. Among them, our scheme missed 17 keywords and detected remaining 195 keywords, which gives it a 92.0% accuracy rate. However, for nine of the missing keywords, our scheme still gave their partial keywords. Table 2 shows the 17 missing keywords.

Table 2. Keywords Missing from the Google Trends Result

Missing keywords	
Alaska earthquake Fiscal cliff deal Fiscal cliff 2013 weather after Christmas sales Twas the Night Before Christmas champs End of the world	December 21 2012 12/21/12 2012 meteor shower earthquake 12 12 12 right to work Man of Steel

Some of the keywords in the table start with a small letter or number. Even though our scheme missed such keywords, still the remaining keywords that start with a capital letter appeared in our result. For example, “Man of Steel” appeared as “Man” and “Steel.” Moreover, for the keyword “Alaska earthquake,” “Alaska” only appeared in our trend result.

4.2. Merging Keywords

In this experiment, we measured how accurate our merge scheme is using 232,549 tweets from 1 am to 2 am on 1/4/2013. Table 3shows the result.

Table 3. Example for Trend Keyword Extraction

Initial state			After removing stopwords			After handling acronyms			After handling reduction			After handling typo and spacing		
1	FACEBOOK	157	1	MILAN	105	1	MILAN	105	1	KEVIN PRINCE BOATENG	133	1	KEVIN PRINCE BOATENG	183
2	TWITTER	115	2	CHRISTMAS	103	2	CHRISTMAS	103	2	AC MILAN	126	2	NEW YEAR	127
3	MILAN	105	3	NEW YEAR	98	3	NEW YEAR	98	3	JUSTIN BIEBER	105	3	AC MILAN	126
4	CHRISTMAS	103	4	JUSTIN	78	4	KEVIN PRINCE BOATENG	90	4	CHRISTMAS	103	4	JUSTIN BIEBER	105
5	NEY YEAR	98	5	KEVIN PRINCE BOATENG	64	5	JUSTIN	78	5	NEW YEAR	98	5	CHRISTMAS	103
7	JUSTIN	78	19	NEW YEARS	29	18	AMERICA	29	6	CRAIC	50	6	AMERICAN	62
10	KEVIN PRINCE BOATENG	64	26	JUSTIN BIEBER	27	18	NEW YEARS	29	7	KEVINPRINCE BOATENG	50	7	CRAIC	50
38	NEW YEARS	29	28	AC MILAN	26	26	JUSTIN BIEBER	27	17	AMERICAN	33	8	TAYLOR SWIFT	48
48	JUSTIN BIEBER	27	28	KPB	26	28	AC MILAN	26	22	AMERICA	29	9	SANDY HOOK	47
51	AC MILAN	26	38	KEVINPRINCE BOATENG	22	37	KEVINPRINCE BOATENG	22	22	NEW YEARS	29	10	RAY LEWIS	46

Removing stopwords	Handling acronyms	Handlingreduction	Handling typo and spacing
FACEBOOK TWITTER …	KEVIN PRINCE BOATENG(64) + KPB(26)	KEVIN PRINCE BOATENG(90) + PRINCE BOATENG(43)	NEW YEAR(98) + NEW YEARS(29)
		KEVINPRINCE BOATENG(22) + BOATENG(28)	KEVIN PRINCE BOATENG(133) + KEVINPRINCE BOATENG(50)
		JUSTIN BIEBER(27) + JUSTIN(78)	

		AC MILAN(26)
		+
		MILAN(105)
			AMERICAN(33)
			+
			AMERICA(29)

The upper part of the table shows top five trend keywords, and the middle part shows keywords that will eventually appear in the list of top five keywords. For each keyword, its rank and term frequency at each stage are shown on the left and at the bottom, respectively. Additionally, the lower part shows all the affected keywords with their rank at each merge step. As can be seen in the table, our scheme removes stopwords and handles all the word variants effectively. Hence, compared to naive trend keyword extraction based on the term frequency, our method shows much better result.

4.3. Typo-spacing Threshold

The threshold for judging typo-spacing error in the keyword has a significant influence on the merge result. In this experiment, we measure the effect of different typo-spacing thresholds. We used 4,179,273 user tweets for this experiment.

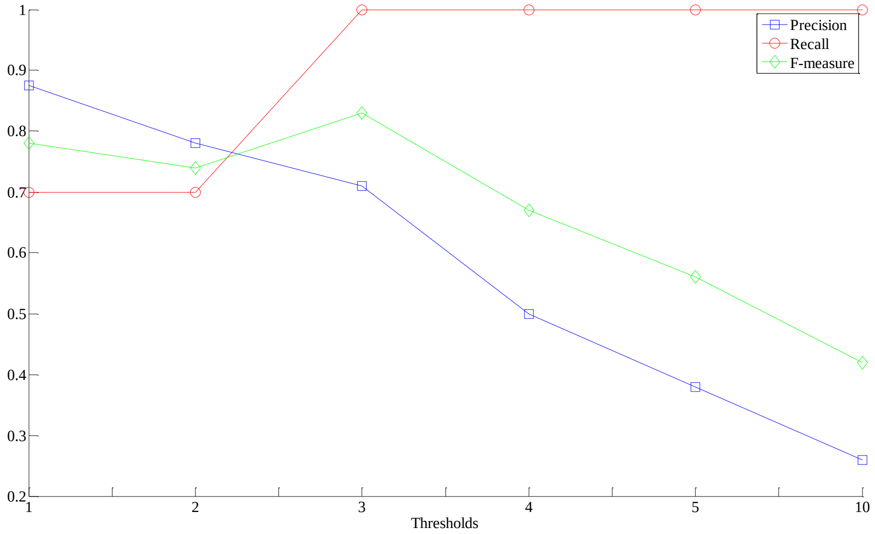


Figure 3. Recall, Precision and F-measure for Various Typo-spacing Thresholds

Figure 3 Figureshows the recall, precision and F-measure of different thresholds. As we can see in the figure, the precision decreases as we increase the threshold. On the other hand, the recall increases as we increase thresholds. This is because with low typo-spacing thresholds, many keywords could not be merged even though they have to be merged. Empirically, the threshold value 3 gives the highest F-measure and hence we use it as typo-spacing threshold in this paper.

4.4. Selecting Bursty Keywords

Figure 4 shows the temporal TF distribution of four sample keywords. According to the graph, “JUSTIN BIEBER” shows high term frequency all the time. So this keyword can be considered as a steady trend keyword. On the other hand, “DANIEL STURRIDGE” shows a sharp increase on 1/2/2013 12 pm in the term frequency. In this case, this term becomes a trend keyword and at the same time bursty keyword. In fact, “Daniel Sturridge” was announced to leave Chelsea to join Liverpool at that time. Soon, this keyword showed very low rank and eliminated from the trend keyword list. Using our scheme, we can easily see temporal trend of top keywords in terms of popularity.

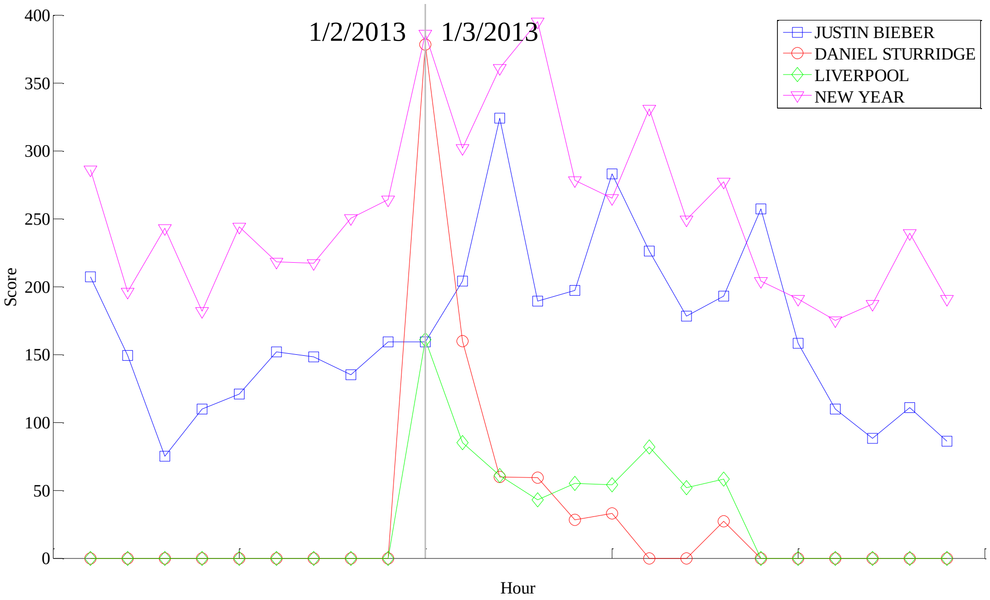


Figure 4. Temporal TF Distribution of four Keywords

5. Conclusion

In this paper, we proposed a trend and bursty keyword detection scheme based on characteristics of Twitter stream data. Our scheme is very robust in that it can handle typical usages such as a variety of abbreviations and spacing and typing errors that occur very frequently when writing tweets on mobile devices. To do this, we first selected candidate keywords from collected user tweets by using simple syntactic feature–based filtering. Then, we performed several merge steps for word variants to calculate their term frequencies correctly. By analyzing temporal transition of their popularity, we can determine which trend keywords are bursty keywords. We implemented a prototype system and demonstrated via extensive experiments that our scheme can achieve satisfactory result.

Acknowledgements

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology(2012-0007202)and the MKE(Ministry of Knowledge Economy), Korea, under the ITRC(Information Technology Research Center) support program supervised by the NIPA(National IT Industry Promotion Agency) (NIPA-2013-H0301-12-3006).

References

[1] Twitter, http://en.wikipedia.org/wiki/Twitter.
[2] D. M. Blei, A. Y. Ng and M. I. Jordan, “Latent dirichlet allocation”, J. Mach. Learn. Res., vol. 3, **(2003)** March, pp. 993–1022.
[3] D. Ramage, S. Dumais and D. Liebling, “Characterizing microblogs with topic models”, in In Proceedings of the 4th International AAAI Conference on Weblogs and Social Media, **(2010)**, pp. 130–137.

[4] G. Kumaran and J. Allan, "Text classification and named entities for new event detection", in Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval, New York, NY, USA, **(2004)**, pp. 297–304.

[5] H. Sayyadi, M. Hurst and A. Maykov, "Event detection and tracking in social streams", in In Proceedings of the International Conference on Weblogs and Social Media, **(2009)**.

[6] Google Trends, http://trends.google.com.

[7] M. Mathioudakis and N. Koudas, "TwitterMonitor: trend detection over the twitter stream", in Proceedings of the 2010 ACM SIGMOD International Conference on Management of data, New York, NY, USA **(2010)**, pp. 1155–1158.

[8] F. Alvanaki, M. Sebastian, K. Ramamritham and G. Weikum, "EnBlogue: emergent topic detection in web 2.0 streams", in Proceedings of the 2011 ACM SIGMOD International Conference on Management of data, New York, NY, USA **(2011)**, pp. 1271–1274.

[9] F. Alvanaki, S. Michel, K. Ramamritham and G. Weikum, "See what's enBlogue: real-time emergent topic identification in social media", in Proceedings of the 15th International Conference on Extending Database Technology, New York, NY, USA **(2012)**, pp. 336–347.

[10] F. Fang, N. Pervin, A. Datta, K. Dutta and D. VanderMeer, "Detecting Twitter Trends in Real-Time", Proceedings of the 21st Workshop on Information Technologies and System (WITS), **(2011)**.

[11] Y. Du, Y. He, Y. Tian, Q. Chen and L. Lin, "Microblog bursty topic detection based on user relationship", in Information Technology and Artificial Intelligence Conference (ITAIC), 2011 6th IEEE Joint International, vol. 1, **(2011)**, pp. 260 –263.

[12] J. Kleinberg, "Bursty and hierarchical structure in streams", presented at the KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, **(2002)**, pp. 91–101.

[13] M. Platakis, D. Kotsakos and D. Gunopulos, "Discovering hot topics in the blogosphere", Proceedings of the 18th Int. World Wide Web Conference, **(2009)**, pp. 1225–1226.

[14] Y. Takahashi, T. Utsuro, M. Yoshioka, N. Kando, T. Fukuhara, H. Nakagawa and Y. Kiyota, "Applying a Burst Model to Detect Bursty Topics in a Topic Model", in Advances in Natural Language Processing, Springer Berlin Heidelberg, **(2012)**, pp. 239–249.

[15] A. Vakali, M. Giatsoglou and S. Antaris, "Social networking trends and dynamics detection via a cloud-based framework design", in Proceedings of the 21st international conference companion on World Wide Web, New York, NY, USA **(2012)**, pp. 1213–1220.