Hindawi Publishing Corporation
Discrete Dynamics in Nature and Society
Volume 2013, Article ID 903765, 8 pages
http://dx.doi.org/10.1155/2013/903765

Research Article

Two Applications of Clustering Techniques to Twitter: Community Detection and Issue Extraction

Yong-Hyuk Kim,[1] Sehoon Seo,[2] Yong-Ho Ha,[3] Seongwon Lim,[1] and Yourim Yoon[4]

[1] *Department of Computer Science and Engineering, Kwangwoon University, 20 Kwangwoon-ro, Nowon-gu, Seoul 139-701, Republic of Korea*
[2] *Tmaxsoft, Bundang-gu, Seongnam-si, Gyeonggi-do 463-824, Republic of Korea*
[3] *SK Telecom, Jung-gu, Seoul 100-999, Republic of Korea*
[4] *Future IT R&D Laboratory, LG Electronics Umyeon R&D Campus, 38 Baumoe-ro, Seocho-gu, Seoul 137-724, Republic of Korea*

Correspondence should be addressed to Yourim Yoon; yryoon@soar.snu.ac.kr

Received 25 July 2013; Revised 25 October 2013; Accepted 31 October 2013

Academic Editor: Daniele Fournier-Prunaret

Twitter's recent growth in the number of users has redefined its status from a simple social media service to a mass media. We deal with clustering techniques applied to Twitter network and Twitter trend analysis. When we divide and cluster Twitter network, we can find a group of users with similar inclination, called a "Community." In this regard, we introduce the Louvain algorithm and advance a partitioned Louvain algorithm as its improved variant. In the result of the experiment based on actual Twitter data, the partitioned Louvain algorithm supplemented the performance decline and shortened the execution time. Also, we use clustering techniques for trend analysis. We use nonnegative matrix factorization (NMF), which is a convenient method to intuitively interpret and extract issues on various time scales. By cross-verifying the results using NFM, we found that it has clear correlation with the actual main issue.

1. Introduction

Twitter, which is causing a worldwide craze, is spreading information faster than the existing broadcasting media. The flight emergency landing incident in the Hudson River through New York, Iran's rigged election incident, China's Uighur bloodshed incident, Gangnam Financial fire incident, Asiana plane crash at San Francisco airport, and other incidents are a few cases in which Twitter delivered news faster, and more accurately, than traditional media.

As the importance of Twitter rises, the associated research field is also developing. Twitter is widely studied in the field of information science, while corporate, civil society sectors, and education organizations are exploring the various applications of Twitter in each field.

We use clustering techniques to study two fields related to Twitter. First, we focus on finding communities on Twitter using clustering. Using the results of network partition/clustering, we are able to find a strong-interacting user group "community" that shares a similar spending habit or political inclination. We explore many studies that deal with existing clustering problems and examine these partition methods.

Second, we focus on effective trend analysis using clustering. The existing trend analysis has two weaknesses. The existing analysis is based on word statistics, which extract sporadic group of words, rather than meaning-based subject extraction. Also, it fails to respond with composure when extracting subjects from various aspects, such as timely, daily, and weekly time scales. This is because it is designed to passively detect words that pass over a specific threshold that is designed in advance. We use a nonnegative matrix factorization (NMF) clustering method to overcome these weaknesses [1].

This paper is composed as follows. We introduce previous Twitter-related researches in Section 2. In Section 3, we introduce a clustering method that searches communities and suggest a partitioned Louvain algorithm. In Section 4, we investigate NFM clustering, which is a Tweet content-based clustering method, for trend analysis. In Section 5.1, we give a partitioned Louvain algorithm and examine its performance.

In Section 5.2, we analyze Tweets using NMF and examine the results. We present the conclusion of the paper in Section 6.

2. Related Work

Twitter is a microblogging service that was released in autumn 2008. On Twitter, users can post or subscribe a 140-word short sentence. Compared to other social networks, which require acceptance between users before making a friend relationship, Twitter allows users to browse or reply, without acceptance of the other party. Also, the user can forward others' content to its subscribers, by using the Retweet function. These functions build a space in which information and public sentiment can freely and rapidly spread through users.

Nowadays, Twitter is highly rated as a new shape of media and utilized in various fields, such as corporate marketing, education, and broadcasting. Hughes and Palen [2] analyzed how Twitter users react and spread information on political and social issues. Also, they found out that these big events become factors that attract new users to Twitter. Diakopoulos and Shamma [3] collected Tweets about the 2008 US presidential election candidate debate. By using this data, they studied a method to analyze public emotional reaction, visual expression, and so on. Kwak et al. [4] compiled statistics and analyzed a great amount of Twitter dialogues and user relationships.

On the web (WWW), a webpage can be thought of as a node and a hyperlink as a directed edge. From this, the Web can be seen as a huge network. There have been many studies that analyzed network features [5–10] from this aspect. In this regard, it is possible to analyze the network, considering each Twitter user as a node and a friend relation as an edge. Java et al. [11] divided users into various groups according to user intentions, investigated each group's ratio, and analyzed the network by dividing various clusters. There was a study that newly defined a network's edge [12]. While existing research considered "following" as a groundwork in network formation, this study defined a meaningful relationship, which is an actual interaction, as two or more replies from one user to another.

On the other hand, there are practical restrictions in collecting and processing data because social networks contain a massive amount of data. In consequence, most experiments extract only a part of the total data. Choudhury et al. [13] compared various methods to extract samples that can well contain the characteristic of the network on Twitter.

Lerman and Ghosh [14] studied information transmission on Twitter. The study found out that Twitter has an important role in information diffusion, and the network structure influenced information flow. Sarma et al. [15] designed mechanisms to rank individual posts, such as Tweets in Twitter, in Twitter-like forums, based on user reviews. They compared mechanisms where forum users rate individual posts and also compared mechanisms where the users are asked to perform pairwise comparisons and give a statement which one is better. They implemented a system, *shout-velocity* (see http://shoutvelocity.com/), which is a Twitter-like forum, but individual posts are rated by pairwise comparisons.

The use of Twitter is limitless. Twitter is much anticipated in the field of corporate promotion and advertisement. A classic example is domestic major companies operating Twitter [16–18]. They are using Twitter not only for promotional purposes but also for multiple uses, such as a service communication tool with customers. Also, Twitter is used between employees, to communicate with each other within a corporate boundary. Corporates use *Yammer* [19], a corporate version of Twitter, or develop their exclusive applications. Zhao and Rosson [20] considered the microblog a new communication channel, which spreads nonshared information through traditional channels.

3. Community Detection by Partitioned Louvain Algorithm

The community detection algorithm has been actively studied in the 21st century, and today many fast algorithms have been developed. Among them, the Louvain algorithm is one of the fastest existing algorithms. Because of its low time complexity and sequential access feature, the Louvain algorithm shows good performance [21].

However, most clustering algorithms, including the Louvain algorithm, assume that the entire data can be uploaded into memory. Considering today's tendency, in which the enlargement of memory size cannot follow the explosive growth of graph size, the assumption is gradually becoming invalid. In practice, when the Louvain algorithm performs with a small-size memory, page swapping frequently occurs, and that means failure in performance. So, we developed a partitioned Louvain algorithm, which adds input/output minimization technique to the traditional Louvain algorithm, to maintain good performance with a huge amount of data.

3.1. Graph Clustering Algorithm. The graph clustering algorithm has been studied in graph theory for a long time, and numerous techniques have been suggested. The techniques are largely divided into hierarchical clustering, divisive clustering, agglomerative clustering, and spectral clustering [22].

By the rapid increase of graph size, there is recently a tendency towards giving importance to speed in the speed/accuracy tradeoff. Because most evaluation functions for graph clustering optimization are classified as NP-hard problems, it is not easy to detect the optimal solutions on massive graphs. Therefore, the greedy algorithm forms the mainstream, and various heuristics are being suggested.

Recently suggested graph clustering and community detecting algorithms usually use greedy techniques [21, 23, 24]. Among them, the Louvain algorithm performed well in practical time with massive graphs that carried over millions of nodes, with respect to speed and accuracy measurement [25].

3.2. Louvain Algorithm. The Louvain algorithm is a hierarchical greedy algorithm. It largely improves both the modularity and computational complexity and it is the only existing algorithm to be applicable to large networks of more than

```
// G: the initial network
repeat
   Put each node of G in its own community;
   while some nodes are moved do
      for each node v of G do
         Place v in its neighboring community including
            its own which maximizes the modularity gain;
      end for
   end while
   if the new modularity is higher than the initial then
      G ← the network between communities of G;
   else
      return;
   end if
end repeat
```

Algorithm 1: Pseudocode of the Louvain algorithm [26].

10 million nodes [25]. The whole process of the Louvain algorithm is described in Algorithm 1. It is mainly composed of three loops. The inner loop moves a node to a nearby community that maximizes the quality criterion "modularity." The middle loop repeats the inner loop, until all nodes fall into the optimal community at the present. In other words, it repeats the inner loop, until modularity increase stops, by moving nodes to other communities. From the result of the middle loop, a given graph converts into a set of communities and their intervening links. The outer loop regards each community as a node and repeats the inner and middle loop on the newly created graph (see Figure 1).

3.3. Weakness of the Louvain Algorithm. The Louvain algorithm has a structure that approaches the entire data sequentially and frequently. If the entire data can be uploaded in memory, the approach is efficient because it uses the disk's and chase's prefetch. However, if the data size outruns the memory size, the approach becomes inefficient. If memory shortage occurs in the run of data scan, it swaps out partial input data. Additional input/output takes place because the swap-out data is again inputted in the next scan process. This process repeats in each scan, which practically reads and writes the whole data. In the Louvain algorithm, if the memory is smaller than the data, it will cause performance degradation, because of its overlapping loop, and the whole data scan on each loop.

3.4. Partitioned Louvain Algorithm. The partitioned Louvain algorithm method is derived from a quite simple idea. Since the Louvain algorithm causes swapout with repeated scan on larger data than the memory, the partitioned Louvain algorithm divides the data to fit the memory and processes the divided data appropriately. While the traditional Louvain algorithm reads all data in the first scan, the partitioned Louvain algorithm reads data that is partitioned smaller than the memory size, and then it performs the Louvain algorithm on a single partition. To increase modularity, the Louvain algorithm is performed until the first partition stops, and then the interim result is saved on the disk. Afterwards, it reads the next partition and repeats the same process. If it is processed as a partition unit, it can prevent additional disk input/output caused by swapout because the partition can be loaded in memory.

We can materialize a partitioned Louvain algorithm, by modifying and supplementing the Louvain algorithm, for which the developer named Blondel has released the code to the public (see http://perso.uclouvain.be/vincent.blondel/research/louvain.html). In this regard, the modified parts of the algorithm are as follows:

(i) before processing: divide given data in the ground of the first data scan,

(ii) add a partition processing loop in between the existing algorithm's outer loop and the middle loop,

(iii) after processing: repeat the Louvain algorithm on each detected community in the partition, and search for community, in respect of the entire network.

4. Issue Extraction by Tweet Content-Based Clustering

To analyze Twitter trends, instead of measuring simple word appearance, we used a clustering method. Clustering is frequently used as a technique to extract subjects in document information. Techniques to detect hidden subjects in listed information are regarded as latent semantic analysis (LSA), and singular vector decomposition (SVD) is a widely-used technique of LSA. SVD is a method that articulates data in a vector, dissembles this data, and detects features. In SVD, feature vectors must be orthogonal to each other. Since the original data should be represented by the sum of the orthogonal vectors, feature vectors obtained from SVD may have negative terms or negative weights. Since it is hard to directly interpret the derived vectors, the second process of them is necessary to extract the meaning.

We used a nonnegative matrix factorization (NMF) technique. Compared to the traditional document clustering method, such as eigenvector-based clustering or SVD, NMF shows similar performance in clustering accuracy. However, NFM is more intuitive, and easier to interpret the results, compared to the others [27, 28].

NMF produces a feature vector that is free from "orthogonality." Therefore, it articulates a feature vector's entire property in the nonnegative, and, also, each weighted value is composed in the positive. This makes it possible to express established data with the addition of the weighted value. It matches the intuition that the entire feature vector is nonnegative. It is intuitive to say that one document is correlated to one subject in a positive number. However, it is nonintuitive, and many times signifies nothing, to say how noncorrelated it is in a negative number. In other words, NFM has the great advantage that people can easily interpret the generated feature.

Also, NFM is suited to the objective of "subject extraction in various time scales." Existing studies had to measure the average frequencies in all words because it is based on the

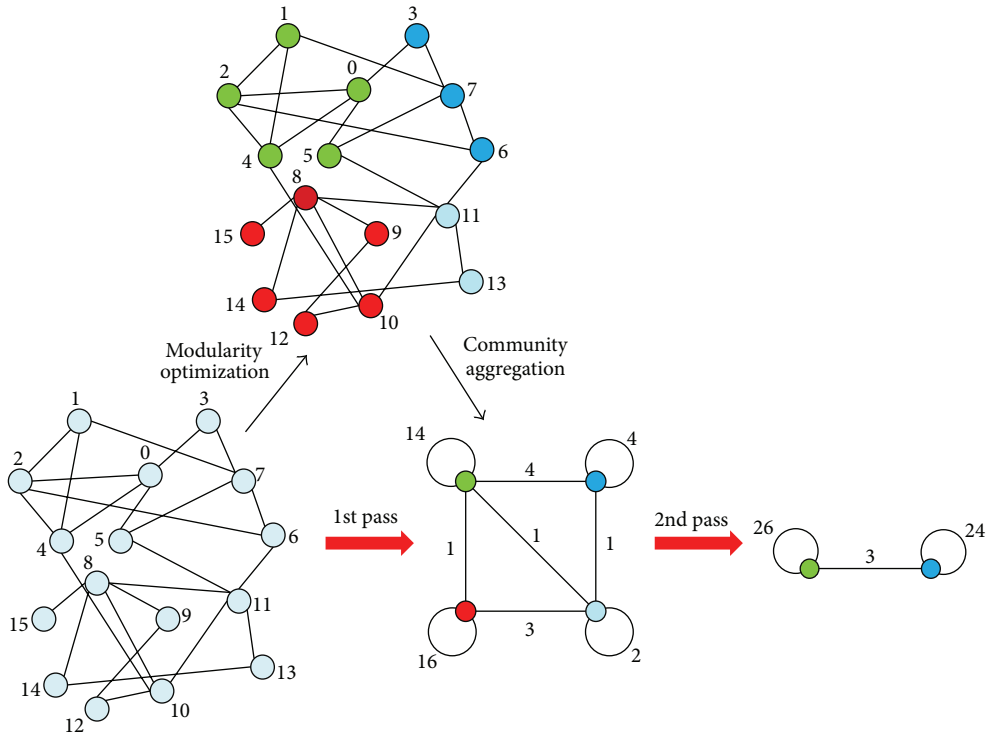


Figure 1: Process of the Louvain algorithm [21].

Table 1: Comparison between data sizes.

	Twitter	*Twitter_small*
Number of nodes	approximately 41 M	approximately 2.6 M
Number of links	approximately 1.2 G	approximately 10 M
Size (binary system)	approximately 9.8 GB	approximately 83 MB

Table 2: Experimental results on *Twitter* data.

	Original Louvain algorithm	Partitioned Louvain algorithm
Processing Time	N/A (>72 hours)	16 hours 38 minutes
Modularity	N/A	0.343

average word frequencies. Considerable effort is needed to trace them, and it makes it harder to analyze in various time scales. However, since NMF uses a fixed time interval to analyze, it is not necessary to trace the average word frequencies. Therefore, the user can choose the time interval that he/she wants. NMF has merit in that it can extract subjects in various time scales, such as the moment when an issue occurs, and the most popular subject within an hour, a day, a week, a month, and so on.

5. Experiments

5.1. Community Detection. Our experiments used specific Twitter network data (below *Twitter*) and reduced data (below *Twitter_small*) that are obtained by extracting partial nodes from the whole data [4]. The sizes of both data are in Table 1. The computing machine used in our experiments is equipped with 8 GB memory and an Intel Quad-core i5-760 CPU at 2.8 GHz with 8 MB cache memory. All programs for community detection were written in *C++* and ran on Linux operating system of *Ubuntu 8.04*. They were compiled using GNU's *g++ 4.2* compiler.

The environment of the experiment is suited for the partitioned Louvain algorithm to make a practical effect because, in the experiment using *Twitter* data, the data size is larger than the machine's memory. In the experiment on *Twitter_small* data, the partitioned Louvain algorithm has no effect because the data size of *Twitter_small* is smaller than the memory. We made additional experiments, to investigate how it affects modularity.

In the experiment on *Twitter* data, a single partition's size was set to be 2 GB, and the entire data was divided into 5 partitions. In the case of the existing Louvain algorithm, the experiment was processed over the course of 72 hours but failed to get a result because of performance degradation caused by page swapping (see Table 2). In the experiment on *Twitter_small* data, the partition size was set to be 64 MB, and it was divided into 2 partitions (see Table 3).

In the experiment on *Twitter* data, we investigated the performance degradation caused by page swapping. In reality, we inevitably had to stop the experiment because it could not pass the early stage of the algorithm after four days. As a result, we did not obtain a modularity value. While

Table 3: Experimental results on *Twitter_small* data.

	Original Louvain algorithm	Partitioned Louvain algorithm
Processing Time	15 minutes 10 seconds	10 minutes 32 seconds
Modularity	0.534	0.492

the existing Louvain algorithm consists of frequent sectional scans, random node access still exists on the inside. We guess that random disk access occurred in this part, causing decrease in performance. On the other hand, the partitioned Louvain algorithm ended after approximately 16 hours 38 minutes.

In the experiment on *Twitter_small* data, the computing time has little difference because the data is smaller than the memory. Our experimental results showed that the partitioned Louvain algorithm performed faster than the existing algorithm. However, this result is not from the minimization of disk input/output, which is the primary purpose of the algorithm. This seems to be a subsidiary result caused by the shifted community condition. In reality, the modularity value decreased from 0.534 to 0.492.

As we can investigate from the result, the partitioned Louvain algorithm does not assure the same quality as the existing algorithm. It cannot correctly calculate communities because partition division is unrelated to community structure, and the algorithm works randomly. In most cases, it is expected to be modified in the postprocessing level, where it detects actual communities. However, this is also not guaranteed. It seems that approximately 0.04 decrease in modularity is caused by this "boundary effect." Although modularity, which is a quality criterion, does not match the existing algorithm, the existing algorithm also uses a greedy approach that sacrifices quality a little to obtain fast results. In this regard, it is reasonable to choose an approach that gains fast results, by sacrificing qualities.

5.2. Issue Extraction. We executed crawling on 15,000 users that used Koreans and collected data from them for about two months (from June 30th, 2011, to Aug 28th, 2011). Out of these, in order to sort out users that are so-called Twitter trend leaders, we mostly used 2,600 users that had posted more than 1,000 Tweets for about two months.

The computing machine used in our experiments is equipped with 8 GB memory and an Intel Quad-core i5-760 CPU at 2.8 GHz with 8 MB cache memory. All programs for crawling and issue extraction were written in *Python 2.6* and ran on Linux operating system of *Ubuntu 8.04*.

We segmented their Tweets daily and weekly and selected daily trends and weekly ones. To input them on NMF, first we extracted words by using natural language processing on their Tweets. For example, if the user tweeted "today's baseball game was interesting," it would extract words such as [today, baseball, interest].

After preprocessing, we constructed a word matrix. In the constructed matrix, a row exhibits each individual user, and a column exhibits the frequency of each word. The matrix has the form of a massive sparse matrix [29] (see Table 4).

Table 4: Example of word matrix used by each user.

	Word_1	Word_2	Word_3	$\cdots$	Word_n
User_1	0	2	4		0
User_2	·				·
User_3	·				·
⋮	·				·
User_m	1	0	0		0

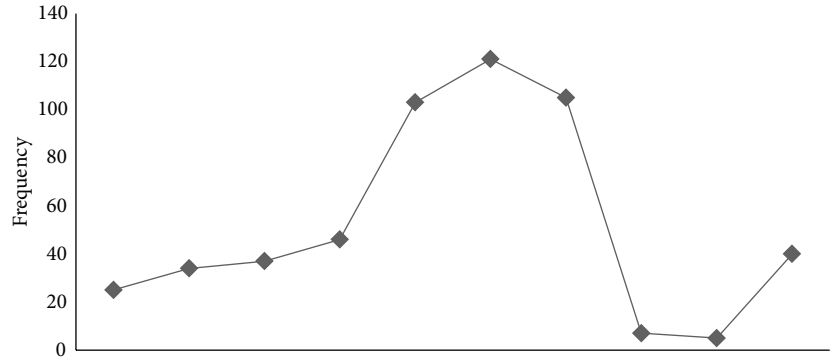
Table 5: Examples of the seven highest ranked issues on June 30th, 2011.

(1) People, today, thought, now, we, love, recently
(2) Today, just, now, from now, really, seriously, alone, company
(3) KBS, police, reporter, wiretap, the public, Democratic party, the prosecution, congressman, parliament, Hanjin, Grand national party, bus
(4) The prosecution, thought, worker, discontent, resignation, group, prosecutor
(5) Baseball, Lotte, KBO, match, record, KIA, SK, Hanwha Corporation, today, Sajik, Munhak, Garcia, Samsung, Doosan, rain, called, cancellation, LG
(6) Lotte, one-piece, department store, today, gift card, purchase, period, T-shirts, Tweet, free
(7) Today, Seoul, bus, Hanjin, violation of the constitution, blockade, square, combat police

If we apply NMF, it is factorized into two vectors W and H. At this time, H is a feature vector that exhibits an $f \times n$ sized vector, in which f is the number of subjects given in advance, and n is the number of words. W exhibits a weighted-value vector of $m \times f$, which informs how much users possess the components of individual features. In other words, it expresses how much interest (weight) m users gave to subject f subjects.

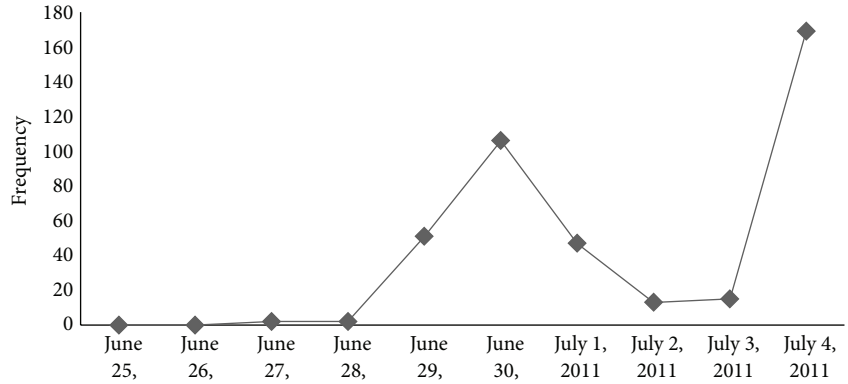
From the application of NMF, we could deduce the number of f subjects at the given period and how much weight value an individual user gave to each subject. We selected the subject that each user gave the highest weighted value as the main interest of that user. In this way, by counting each user's interests, we could deduce the feature that received the highest weighted value among users.

Table 5 shows an example of the highest ranked subjects that most users selected for June 30. If we examine the result, the 1st and 2nd ranked words, such as "people, today, just, alone" compose the subject. From this, we can see that users most frequently deal with their everyday life and small talk in Twitter. In the 3rd and 4th rank, "prosecution" and "wiretap" relating to political subjects appeared. The "Baseball"-related subject was the 5th most mentioned, and then for the 6th rank, we can estimate from words such as "Lotte," "gift card," "Tweet," and "free," that a "Lotte"-related event was popular. For the 7th rank, we can see that a "Hanjin"-related political issue was present.



June 25, 2011	June 26, 2011	June 27, 2011	June 28, 2011	June 29, 2011	June 30, 2011	July 1, 2011	July 2, 2011	July 3, 2011	July 4, 2011
25	34	37	46	103	121	105	7	5	40

Figure 2: News frequencies for ten days of "wiretap" and "KBS" (Subject 3).

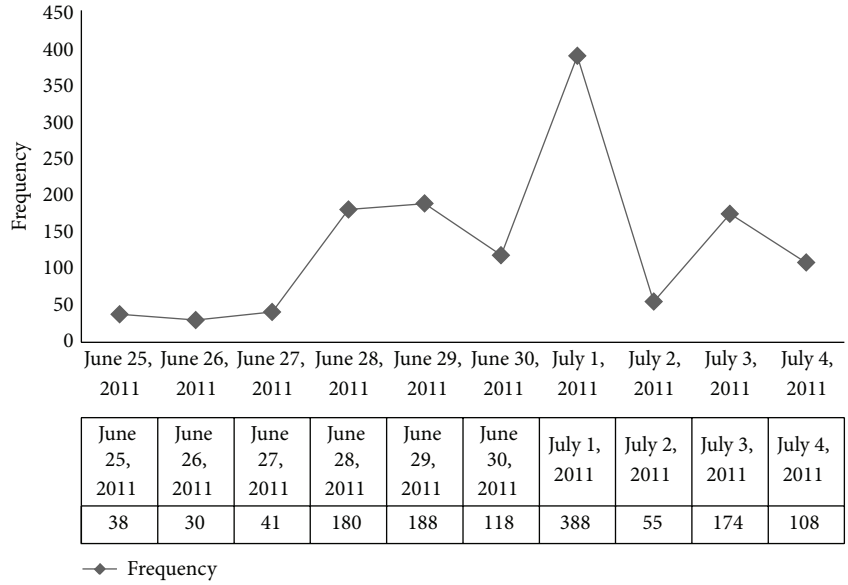


June 25, 2011	June 26, 2011	June 27, 2011	June 28, 2011	June 29, 2011	June 30, 2011	July 1, 2011	July 2, 2011	July 3, 2011	July 4, 2011
0	0	2	2	51	106	47	13	15	169

Figure 3: News frequencies for ten days of "prosecution" and "resignation" (Subject 4).

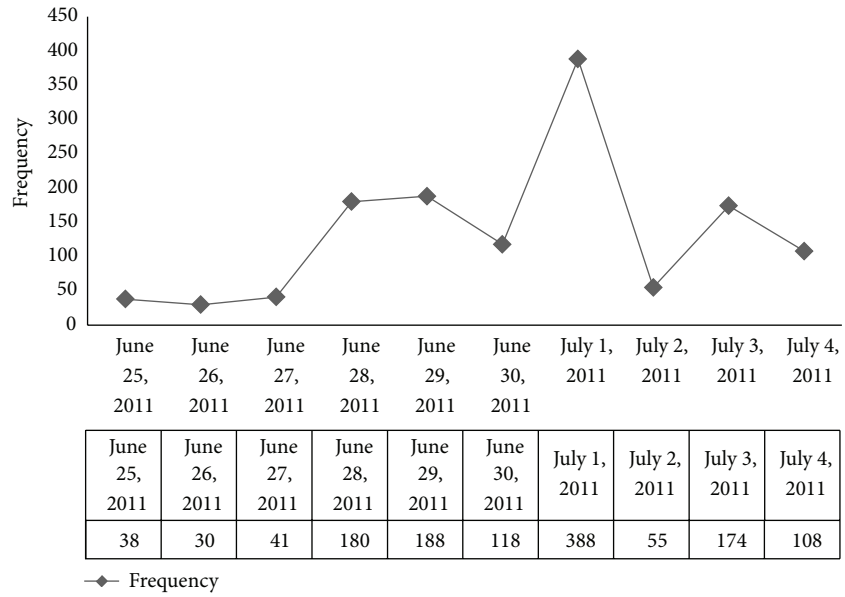
From this result, we specifically need to pay attention to the fact that by only using NFM, related words composed a congregation, without using additional heuristic. The existing method, which counted word appearance frequency, could give a sequence of words as a result but failed to congregate words by subject. By using NMF, words that composed the subject automatically congregated and presented meaningfully. Therefore, the interpretation of this can be very intuitive. To compare the pertinent issue with the actual issue in that period, we used *Naver*'s news search (see http://news.naver.com/). We assumed that if the pertinent issue was actually popular, many related news articles would have been created. In this regard, we investigated the number of search results about the pertinent subject.

We daily investigated the number of articles on *Naver*'s news search, excluding everyday subjects related to Subjects 1 and 2, and commercial events of Subject 6. We could discover that a correlation between the detected subjects for June 30 using NMF and the main subjects in the daily newspaper clearly existed. In the case of Subject 7 "Seoul square," we could observe a phenomenon in which it slightly preceded the next day July 1 on daily comparison (see Figures 2, 3, 4, and 5). We could see the possibility of Twitter as an effective tool that can detect trends more quickly than other media.



June 25, 2011	June 26, 2011	June 27, 2011	June 28, 2011	June 29, 2011	June 30, 2011	July 1, 2011	July 2, 2011	July 3, 2011	July 4, 2011
38	30	41	180	188	118	388	55	174	108

Figure 4: News frequencies for ten days of "baseball" and "called" (Subject 5).



June 25, 2011	June 26, 2011	June 27, 2011	June 28, 2011	June 29, 2011	June 30, 2011	July 1, 2011	July 2, 2011	July 3, 2011	July 4, 2011
38	30	41	180	188	118	388	55	174	108

Figure 5: News frequencies for ten days of "Seoul" and "square" (Subject 7).

6. Concluding Remarks

We applied clustering techniques to analyze a Twitter network and obtain trends on Twitter. In the field of network analysis, recent community analysis algorithms have showed a change from accurately analyzing small networks to roughly but quickly analyzing large networks. This is because network data, including social networks, have been experiencing substantial growth. When the data size is bigger than the memory, the existing Louvain algorithm has the possibility of incurring sharp performance degradation. In this paper, we designed a partitioned Louvain algorithm that can avoid this performance degradation and showed performance improvement through experiments.

Meanwhile in detecting trends, the existing studies have drawbacks, in that they are fragmentary and sporadic, because they only detect many words that appeared frequently. To improve them, we used an NFM clustering technique and analyzed trends correctly. From this, we could obtain a congregation of intuitive issues that people can easily understand. Also, after cross-verifying the results, we could show that it had clear correlation with the actual main issue.

Conflicts of Interest

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors would like to thank Mr. Inwook Hwang for his valuable suggestions in improving this paper. This research was supported by the Basic Science Research Program, through the National Research Foundation of Korea (NRF), funded by the Ministry of Education, Science and Technology (2012-0001855).

References

[1] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Advances in Neural Information Processing Systems*, vol. 13, pp. 556–562, MIT Press, 2001.

[2] A. L. Hughes and L. Palen, "Twitter adoption and use in mass convergence and emergency events," *International Journal of Emergency Management*, vol. 6, no. 3-4, pp. 248–260, 2009.

[3] N. A. Diakopoulos and D. A. Shamma, "Characterizing debate performance via aggregated twitter sentiment," in *Proceedings of the 28th International Conference on Human Factors in Computing Systems (CHI '10)*, pp. 1195–1198, April 2010.

[4] H. Kwak, C. Lee, H. Park, and S. Moon, "What is Twitter, a social network or a news media?" in *Proceedings of the 19th International World Wide Web Conference (WWW '10)*, pp. 591–600, April 2010.

[5] A. Broder, R. Kumar, F. Maghoul et al., "Graph structure in the web," *Computer Networks*, vol. 33, no. 1–6, pp. 309–320, 2000.

[6] R. Kumar, P. Raghavan, S. Rajagopalan, D. Sivakumar, A. Tomkins, and E. Upfal, "The web as a graph," in *Proceedings of the 19th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pp. 1–10, May 2000.

[7] R. Albert, H. Jeong, and A. Barabási, "Diameter of the world-wide web," *Nature*, vol. 401, no. 6749, pp. 130–131, 1999.

[8] S. Chakrabarti, B. E. Dom, S. R. Kumar et al., "Mining the web's link structure," *IEEE Computer*, vol. 32, no. 8, pp. 60–67, 1999.

[9] K. Bharat, B. Chang, M. Henzinger, and M. Ruhl, "Who links to whom: mining linkage between web sites," in *Proceedings of the 1st IEEE International Conference on Data Mining (ICDM '01)*, pp. 51–58, December 2001.

[10] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank citation ranking: bringing order to the web," in *Proceedings of the 61st Annual Meeting of the American Society for Information Science (ASIS '98)*, pp. 161–172, October 1998.

[11] A. Java, X. Song, T. Finin, and B. Tseng, "Why we twitter: understanding microblogging usage and communities," in *Proceedings of the Joint 9th WebKDD and 1st SNA-KDD Workshop on Web Mining and Social Network Analysis*, pp. 56–65, August 2007.

[12] B. A. Huberman, D. M. Romero, and F. Wu, "Social networks that matter: Twitter under the microscope," CoRR abs/0812. 1045, 2008.

[13] M. D. Choudhury, Y. R. Lin, H. Sundaram, K. S. Candan, L. Xie, and A. Kelliher, "How does the data sampling strategy impact the discovery of information diffusion in social media?" in *Proceedings of the 4th International AAAI Conference on Weblogs and Social Media*, 2010.

[14] K. Lerman and R. Ghosh, "Information contagion: an empirical study of the spread of news on Digg and Twitter social networks," in *Proceedings of the 4th International AAAI Conference on Weblogs and Social Media*, 2010.

[15] A. D. Sarma, A. D. Sarma, R. Panigraphy, and S. Gollapudi, "Ranking mechanisms in twitter-like forums," in *Proceedings of the 3rd ACM International Conference on Web Search and Data Mining*, pp. 21–30, February 2010.

[16] Account @samsungin in Twitter, http://twitter.com/samsungin.

[17] Account @lg_theblog in Twitter, http://twitter.com/lg_theblog.

[18] Account @olleh in Twitter, http://twitter.com/ollehkt.

[19] Web site of Yammer, http://www.yammer.com.

[20] D. Zhao and M. B. Rosson, "How and why people Twitter: the role thatmicro-blogging plays in informal communication at work," in *Proceedings of the ACM International Conference on Supporting Group Work*, pp. 243–252, May 2009.

[21] V. D. Blondel, J. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics*, vol. 2008, no. 10, Article ID P10008, 2008.

[22] S. E. Schaeffer, "Graph clustering," *Computer Science Review*, vol. 1, no. 1, pp. 27–64, 2007.

[23] A. Clauset, M. E. J. Newman, and C. Moore, "Finding community structure in very large networks," *Physical Review E*, vol. 70, no. 6, Article ID 066111, 6 pages, 2004.

[24] K. Wakita and T. Tsurumi, "Finding community structure in mega-scale social networks," in *Proceedings of the 16th International Conference on World Wide Web (WWW '07)*, pp. 1275–1276, May 2007.

[25] H. Kwak, Y. Choi, Y. Eom, H. Jeong, and S. Moon, "Mining communities in networks: a solution for consistency and its evaluation," in *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement Conference*, pp. 301–314, 2009.

[26] T. Aynaud and J. Guillaume, "Static community detection algorithms for evolving networks," in *Proceedings of the 8th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt '10)*, pp. 513–519, June 2010.

[27] W. Xu, X. Liu, and Y. Gong, "Document clustering based on non-negative matrix factorization," in *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, pp. 267–273, 2003.

[28] P. Rakesh, G. Shivapratap, G. Divya, and K. P. Soman, "Evaluation of SVD and NMF methods for latent semantic analysis," *International Journal of Recent Trends in Engineering*, vol. 1, no. 3, pp. 308–310, 2009.

[29] P. O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *Journal of Machine Learning Research*, vol. 5, pp. 1457–1469, 2004.