Microblog Hot Event Detection Based on Restart Random Walk and Modularity

Xiaohong Li, Jiheng Gong, Yuyin Ma, Huifang Ma, Na Qin

Microblog Hot Event Detection based on Restart Random Walk and Modularity

XiaoHong Li, JiHeng Gong, Yuyin Ma, HuiFang Ma, Na Qin

College of Computer Science and Engineering Northwest Normal University
730070 China
xiaohongli@nwnu.edu.cn

Abstract. Using traditional method to extract semantic relations between words hardly applied to micro-blog, which make finding hot event not sensitive. We propose a new method based on restart random walk and Modularity. The semantic relation between items is calculated by conducting the restart random walk iteratively on graph, and then the semantic correlation matrix is constructed. Next, the idea of Modularity is introduced to design algorithm for word clustering, which make a series of micro-blog hot events obtain. The experimental results show that our method has a higher accuracy compared with the kindred method, and hot events could be detected effectively.

Keywords: restart random walk, hot degree, hot event detection, Modularity.

1 Introduction

As an up-to-date information media, Micro-blog, where folks speak out their opinions on social events, has been an important place in which hot issues are born and discussed for its shortness, rich content, relatively low barrier to entry, and fast propagation velocity. Especially, users' following, reposting and commenting usually help micro-blog events propagate. It's been hard for users to find information they are interested in because of much valuable data being flooded caused by information explosion. Therefore, studying how to get valuable information out of a vast number of micro-blog data becomes a hot spot in computer science area. Meanwhile, detection of micro-blog hot event, known as an important branch of web public sentiment monitor, concerned both domestic and foreign academia, showing its huge research value.

By now, researchers have done numerous research about micro-blog hot event detection which could be divided into two following categories[1]: 1) Methods focused on texts, specifically, micro-blogs are clustered into several clusters in order to identify hot events. For example, Shi[2] propose a hot event evolution model to discover the user interest distribution, but also a hot event filtering algorithm is developed to detect important events. Yang[3], who's been devoted to clustering hot topics based on timing characteristic, presented K_SC algorithm based on hotness tendency of topics. However, data sparsity problems caused by shortness of micro-blog and lots of noise

data in it, make a relatively low efficiency on identifying burst words after clustering. 2) Approach focused on burst features, that is, burst features are extracted and divided into different groups, and unexpected events are identified by feature groups. Yang[4] detected hot events through changes of amount of emotionally key words. Chen[5] detected burst features by analysis method based on analysis of timing windows, then utilize Affinity Propagation algorithm to cluster burst features. Similarly, Zhao[6] propose a novel real-time event detection method by generating an intermediate semantic level from social multimedia data, named MC, which is able to explore the high correlations among different microblogs. Aforementioned methods only theoretically improve effectiveness of event detection when detecting burst events, not achieve satisfying result in real-life applications. The most fundamental cause is that topics drift will appear with time changing during event detection.

In order to improve accuracy of micro-blog hot event detection and reduce complexity, we proposes a microblog hot event detection algorithm based on restart random walk model and modularity, which divide micro-blog hot event detection into two phases, Phase 1: to know hidden semantic relations among terms through restart random walk algorithm. Phase 2: to cluster terms with the idea of modularity based on the former result, and find hot events. The main contributions of this work are as follows:

1. To know hidden semantic relations among terms, we construct an undirected weighted graph and run restart random walk algorithm on it.
2. We apply the idea of modularity as clustering for detecting hot event, and achieve the goal of corresponding between hot words and hot events.
3. We use three experiments on two datasets to verify effectiveness of hot event detection algorithm, which demonstrate promising results compared to the kindred methods.

The remaining of this paper is as follows. Section 2 discusses the preliminary knowledge of our proposed method. Construct graph and acquire association relationship between words in Section 3. Find Hot Event based on Restart Random Walk and Modularity in Section4. Experimental results are discussed in Section 5, and conclusions are drawn in Section 6.

2 Preliminary Knowledge

2.1 Hot Degree

Micro-blog's sensitivity towards hot events makes it able to reflect hot events. Popular micro-blog, whose number of comments and reposts gradually increases, spreads very soon which is why we need a metric to measure how much the microblogs concerned us[7].

Assuming that user u_i had posted a micro-blog mb, then it was posted by user u_j in time Δt, then the repost value ret(mb, u_j) of the latter user on this micro-blog is defined as ret(mb,u_j):

$$ret(mb,u_j)=\begin{cases}1 & otherwise\\ 1-sim(u_i,u_j) & if\ u_i\ is\ similar\ to\ u_j\end{cases} \quad (1)$$

As the same, comment value com(*mb*, u_j) of u_j towards this micro-blog is defined as follows:

$$com(mb,u_j)=\begin{cases}1 & otherwise\\ 1-sim(u_i,u_j) & if\ u_i\ is\ similar\ to\ u_j\end{cases} \quad (2)$$

sim(u_i, u_j) represents similarity between users, we calculate it by user similarity method as[8]. $sim(u_i,u_j)=\frac{F(u_i)\bigcap F(u_j)}{F(u_i)\bigcup F(u_j)}$. Where $F(u_i)$ denote collection composed of user that u_i is attentioned.

Then we got the definition of hot degree based on formula (1) and (2).

Definition: Hot degree of a micro-blog ***Hot*(*mb*)** equals the weighted sum of its re-post value ret(mb_i,u_j) and comment value com(mb_i,u_j). After normalization it is:

$$Hot(mb)=\frac{\lambda\sum_{j=1}^{l}ret(mb,u_j)+(1-\lambda)\sum_{j=1}^{h}com(mb,u_j)}{l+h} \quad (3)$$

Where, λ is the adjustment parameter, $0<\lambda<1$, l is number of reposts, h is number of comments. By definition of the hot degree, the hot event should be directly related to hot degree and not the content itself of micro-blog.

2.2 Co-occurrence Degree between Words

Given a micro-blog *mb*, co-occurrence degree of term t_i and t_j is denoted as $c(t_i, t_j)$, which is as follows[9]:

$$c(t_i,t_j)=e^{-dist(t_i,t_j)} \quad if\ t_i\in mb\ \ and\ \ t_j\in mb \quad (4)$$

dist(t_i, t_j) is co-occurrence distance between t_i and t_j, whose value is number of words between t_i and t_i in micro-blog mb. Co-occurrence degree $c(t_i, t_j)$ reflects that two words are correlated if they often appear in the same micro-blog.

3 Acquire Association Relationship between Words

$MB=\{mb_1,mb_2,\cdots,mb_N\}$ is micro-blogs set, and $mb_i=\left\{t_{i1},t_{i2},...,t_{i|mb_i|}\right\}$ is the *i-th* micro-blog, and candidate item set is $MT=\{t_1,t_2,\cdots,t_m\}$,where *m* represents the size of the dictionary.

3.1 Construct Graph Model

We construct an undirected weighted graph $G = (V, E)$, where $V=\{ v_1,v_2,\ldots,v_M \}$ is vertex set, M is the number of the rest of the vertices, v_i corresponds to candidate item in the *MT*. Then we connect any two vertexes in the set V if they're from the same micro-blog, so edges set $E= \{(v_i,v_j)|\ v_i \in mb$ and $v_j \in mb\}$. Notes: in the rest of the paper v_i represents the vertex word t_i corresponds.

$$A' = \begin{bmatrix} w_{11} & w_{12} & \ldots & w_{1m} \\ w_{21} & w_{22} & \ldots & w_{2m} \\ \ldots & \ldots & \ldots & \ldots \\ w_{m1} & w_{m2} & \ldots & w_{mm} \end{bmatrix} \qquad A = \begin{bmatrix} c_{11} & c_{12} & \ldots & c_{1m} \\ c_{21} & c_{22} & \ldots & c_{2m} \\ \ldots & \ldots & \ldots & \ldots \\ c_{m1} & c_{m2} & \ldots & c_{mm} \end{bmatrix}$$

Fig. 1. The weighted adjacency matrix **Fig. 2.** The weighted matrix after normalization

First, to get weight matrix as Figure 1 shows. In A', element $w_{ij}=w(v_{i,}, v_j)$, $w(v_{i,}, v_j)$ represents weight on the edge$(v_{i,}, v_j)$, which is defined as the sum of cooccurrence degrees of terms $v_{i,}$and v_j in micro-blog set. As equation (5) shows:

$$w(v_i, v_j) = \begin{cases} \sum_{mb \in MB} c(v_i, v_j) & (v_i, v_j) \in E \\ 0 & otherwise \end{cases} \tag{5}$$

Afterwards, run normalization and asymmetric operation on matrix A' to obtain matrix A[10]. Value of element c_{ij} is calculated through the formula (6).

$$c_{ij} = \frac{w_{ij}}{n_j + 0.01} \tag{6}$$

Where $0 \le c_{ij} \le 1$ and $\sum_j c_{ij} = 1$, $n_j = \sum_i w_{ij}$ represents the sum of elements in *j-th* column in matrix A'. The rest of this paper is developed based on graph *G*.

3.2 Restart Random Walk on Graph

Random walk model[11] means to traverse a graph beginning with one vertex or a series of vertexes. At any vertical, traverser randomly selects an edge connecting the vertex at a certain possibility, then randomly jumps to the next vertex along the edge or jumps back to the starting point at a certain possibility. Mathematic expression of it is:

$$r^{(t+1)} = (1-\alpha) * C * r^{(t)} + \alpha * d \tag{7}$$

Where C is transition probability matrix. $r^{(t)}$ represents possibility assignment at the t-th time. d is restart vector, which is possibility assignments jumping into every

vertex when jumps happening. α is an adjusting factor which controls reliance degrees among terms.

First, assuming that it starts random walk from v in graph *G*. The closer between *v* and v_j, the more possibly that *v* walks to v_j. Matrix *A* represents co-occurrence relations between any two words, which is consistent with tendency of walking. Therefore, matrix *A* is selected as transition probability matrix, i.e. *C=A*.

Next, determining the value of the initial vector $r^{(0)}$, it's value are shown in formula (8). Assuming that *h=index*(*v*) can locate the index of vertex *v* in G, it can be seen from the formula that value of $r^{(0)}$ is transposition of the *h-th* row vector in matrix *A* actually.

$$r^{(0)}(j)=\begin{cases}0 & (v,v_j)\notin E\\ c_{hj} & (v,v_j)\in E\end{cases} \tag{8}$$

Finally, the paper hypothesizes that starting point is equally randomly selected, so initial possibility assignment $d=\left[\frac{1}{m},\frac{1}{m},.....\frac{1}{m}\right]_m$.

So far, all parameters formula (7) needs are determined. Put them in formula (6) and calculate iteratively, till r falls in a stable condition. Ultimately, vector r describes the comprehensively semantic relations between vertex v and other vertexes. Let each of the vertexes in graph as start point in turn and repeat aforementioned process so Matrix reflects semantic relations among all term pairs will be obtained, which is represented as P.

4 Find Hot Event Using Modularity

In this section, we'll introduce the idea of modularity to reach the aim of word clustering use Matrix P, so hot events are found by filtering.

4.1 Modularity

With the further research on web, researchers find many large complex net made up of lots of communities, nodes are connected very firmly within each community, while connections among communities are relatively sparse. Modularity[12] is a common metric to evaluate quality of community partition in complex web, whose formula is as follows:

$$Q=\sum_{i=1}^{k}(e_{ii}-a_i^2) \tag{9}$$

e_{ii} represents the fraction of total edges that connect to vertices in community C_i, $a_i=\sum_j e_{ij}$ i≠j, which represent the fraction of edges that link vertices in community C_i

to vertices in community C_j, k is the number of communities. Newman points out that it is difficult to recognize whether Q has reached the maximum or not. Therefore, modularity increment is introduced to determine if communities are partitioned properly in this paper, and decide when the partition terminated. Modularity increment is defined as:

$$\Delta Q = 2(e_{ij} - a_i * a_j) \tag{10}$$

4.2 Hot Event Detection Algorithm

In this paper, we adopts the idea of modularity, and take graph as partition object, and we also use the correlation matrix P in Section3.2 as prior information. The results of graph partitioning are hot events. Prerequisites of the algorithm is that we see each node in graph as an independent clustering. First, the initial operating is find the maximum in matrix P, assume that $\max(P)=p_{ij}$, two vertex, vi and vj, are merged into the same cluster. Continue searching the maximum in P and calculate modularity increment ΔQ. If $\Delta Q>0$, then merging, Otherwise, partition's over. Pseudocode of the algorithm is as follow:

Input: P, correlation matrix, parameter β
Output: a community

1. find the maximum in matrix P, assume that $max(P)=p_{ij}$,
2. set stack's initial value is: $nodestack$ ={(v_i,v_j)}, and let p_{ij}=0;
3. Let variable $processed$ save the vertices in the discovered community, its initial value is:
 $processed$={ v_i,v_j}, similarly $v_{adj}=\Phi$;
4. While $nodestack \neq \Phi$ do:
  4.1 v_c =$pop(nodestack)$;
  4.2 $processed$= $processed \cup v_c$;
  4.3 for each node v in the c-th row;
      if($P(v_c, v)>\beta$) then $v_{adj} = v_{adj} \cup v$;
  4.4 delete nodes in the set processed:
      $v_{adj} = v_{adj} - processed$;
  4.5 *For each* v_{temp} *in* v_{adj} *do*
    4.5.1 Divide v_{temp} into community $processed$,
      and compute the value of ΔQ;
    4.5.2 if ($\Delta Q>0$), push($nodestack$, v_{temp})
  4.6 empty set v_{adj}: $v_{adj} =\Phi$
5. return processed //end

5 Experimental Results and Analysis

5.1 Experimental Data

Data Set 1: Messages posted in Sina Weibo from January to June in 2016 are sampled manually as experimental data. To ensure being in accordance with real events at best, enormous noise data are added during manual sampling which turns out a complete noise-contained data set including 2541 Weibo posts of 8 hot events in all. Of these micro-blog, 1749 are events-describing and 792 are noise data. Then, pre-processing methods including word segmentation and stop words removal are launched, and isolated word filtering is conducted according to affinities among terms. Finally, 12000 terms remain.

Data Set 2: In total, titles of 3755 essays in 6 categories of data mining are drawn from DBLP to run experiments. They are: Text clustering(614),Text classification(484), Video processing(516), Speech recognition(685), Image processing(960), Graphical model(496). After pre-processing work like removal of stop words and HTML tags, we get final experimental data sets.

5.2 Comparative Analysis on the Results

Three experiments are designed in this paper to verify effectiveness of hot event detection algorithm. Experiment 1 utilizes dataset 1 to extract hot topics. Experiment 2 adjusts important parameters in our algorithm to observe the influence on hot event results. Experiment 3 is to compare our method with the existing methods in the similar manner. In this paper, we adopt NMI and ARI [13, 14] as the evaluation criterion.

Table 1. Comparison of real hot events and hot words detected by our method

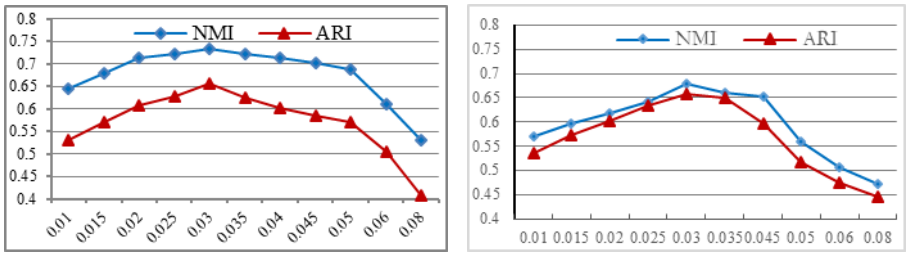
real hot events (reposts/comments)	hot words detected by out method
baidu isn't a internet company no longer (526/1566)	baidu internet company AI LiYanHong
it is cotton in shredded dried meat pies, but not meat. (957/3121)	shredded dried meat, pie, cotton, soaked, burn
Huangbo can perform comic dialogue sufficiently(6862/1314)	humor, perform, comic dialogue Huangbo, YueYue, EQ(emotional quotient)
Andy's two boyfriends in 《Ode to Joy》. (5448/2175)	Ode to Joy, Andy's boyfriend, like, love, difference, little boss Bao
big secret is behind ErKang pharmaceutical sales. (212/558)	Erkang, sale, secret, dealer pharma companies
The son get into the gambling and the drug taking. (336/2026)	addiction to drugs, song, sadness, parents, sellers, repay a debt, gambling
Kejie rate the AlphaGo highly (489/951)	chess, KeJie, AlphaGo, appraise,defeated

We use the same parameters for restart random walk model as work [10] in Experiment 1, that is $\alpha = 0.15$. And constructing matrix representation for micro-blog based

on data set 1. Hot events are obtained through algorithm proposed in this paper. Final experimental result selects illustrated in the following Table 1. Some key terms with stronger correlations are selected to describe hot events, and take hot events published by authorized institutes for comparison, which show good agreements with actual network hot event results.

Experiment 2: There are two parameters: λ and β, λ balances the dedications to hotness from number of reposts and comments, and β takes dedications to hot words extraction results with affinities among words into consideration. We research the influence on topic words extraction by setting different values for them. λ is set 0.5,0.55and 0.6, still β ranges from 0.01 to 0.08.

Experimental results are illustrated (a) and (b) in Fig.3, it can be seen that the effect of repost value of micro-blog on results is slightly higher than comment value. And the worst performance is when λ=0.6, so comparison diagram is omitted. It also can be observed that NMI and ARI are on the rise slowly before β=0.03, and they reach the maximum When β is 0.03. But with value of β keeping rising and reaching the maximum values allowed by theory, effectiveness start decreasing. Especially, NMI and ARI fell quickly after 0.05.



(a) λ=0.55 (b) λ=0.5

Fig. 3. Effect of parameters on the result of hot event detection

Experiment 3: Select DPSO algorithm in literature [15] and MCF method proposed in work [16] with our method to comparing on two data sets. Through mining mutual information between words and Internal/External correlative information, DPSO finds micro-blog hot events in the best angle. MCF proposes using topic model for extracting micro-blog themes, and word activation force model is introduced to generate hot events. The experimental comparative result among method of this paper and other two methods is illustrated in (a) and (b) of Fig.4.

We can see from Fig.4 that our method has a little higher NMI and ARI than the other two methods. Possible cause is that method in the paper mines surface and hidden semantic relations among terms as well as possible, which makes micro-blog semantic expression clear. While drawbacks of the other methods, such as noise information, have result in a lot of low-quality feature items and small numbers of thematic words. Result in Fig.4 also shows the superiority of method in this paper.

Meanwhile, since data set 2 is cleaner and it brings less distribution, obtained results are higher than those from dataset 1.

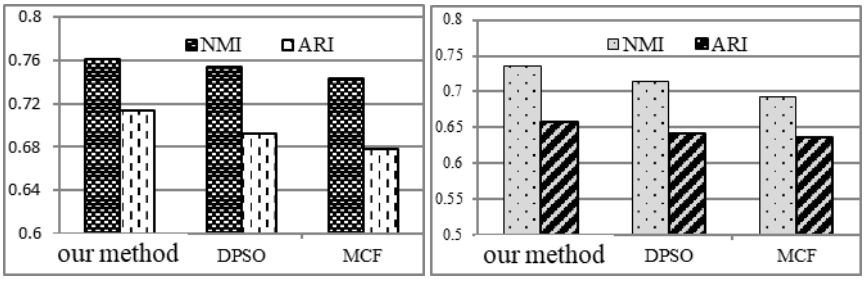


Fig. 2. Effect of three different method on clustering results on different datasets (a on dataset 1, and b on dataset 2)

6 Conclusions

The paper proposes a hot event detection method based on restart random walk model and community partition. Main design idea is to calculate shown and hidden semantic relations among lexical items by conducting restart random walk algorithm iteratively on graph and construct a semantic correlation matrix. Meanwhile, the idea of community partition is introduced. An algorithm performing word clustering with the semantic correlation matrix is designed in order to obtaining the set of hot events. The experimental result points out that hot events found are consistent with real-time events, so the effectiveness of detection is outstanding. From now on, researches about reducing the outliers in feature word sets, initialization of random walk model metrics and judging standards of convergent conditions in community partition can be performed, even trying to introduce expert dictionaries or lexicons themselves, to raise accuracy of hot event detection.

Acknowledgments

The work is supported in part by the Natural Science Foundation for Young Scientists of Gansu Province, (No.1606RJYA269), and Youth Teacher Scientific Capability Promoting Project of NWNU(No. NWNU-LKQN-16-20), and the National Natural Science Foundation of China (No. 61762078).

References

1. Diao Q, Jiang J, Zhu F. Finding Bursty Topics from Microblogs[C]//Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1. Association for Computational Linguistics, 2012: 536-544.

2. Shi L L, Liu L, Wu Y, et al. Event Detection and User Interest Discovering in Social Media Data Streams[J]. IEEE Access, 2017, 5(99):20953-20964.
3. Yang J, Leskovec J. Patterns of Temporal Variation in Online Media[C]//Proceedings of the fourth ACM international conference on Web search and data mining. ACM, 2011: 177-186.
4. Yang L, Lin Y, Lin H F. Microblog Hot Events Detection based on Emotion Distribution[J]. Journal of Chinese Information Processing, 2012, 26(1): 84-91.
5. Chen H, Chen W. Analyzing Bursty Feature for Event Detection [J]. Application Research of Computers, 2011, 1: 30-33.
6. Zhao S, Gao Y, Ding G, et al. Real-Time Multimedia Social Event Detection in Microblog[J]. IEEE Transactions on Cybernetics, 2017, PP(99):1-14.
7. Liu Y Z, Du Y N, Jiang Y C. Trend Prediction for Microblog Based on Classification Modeling of Heat Curves [J]. Pattern Recognition and Artificial Intelligence, 2015, 28(1): 27-34.
8. ZhiYun Z, ChunYuan J,ZhenFei W. Computing Research of User Similarity Based on Microblog [J]. Computer Science, 2017,44(2):262-266.
9. Hua W, Wang Z, Wang H, et al. Short Text Understanding through Lexical-Semantic Analysis[C]//2015 IEEE 31st International Conference on Data Engineering. IEEE, 2015: 495-506.
10. Pan J Y, Yang H J, Fallouts C. Automatic Multimedia Cross-Modal Correlation Discovery[C]//Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2004: 653-658.
11. Fu B, Wang Z, Xu G, et al. Multi-label Learning based on Iterative Label Propagation over Graph[J]. Pattern Recognition Letters, 2014, 42: 85-90.
12. Newman M E J, Girvan M. Finding and evaluating community structure in networks [J]. Physical review E, 2004, 69(2): 026113.
13. Rand W M. Objective Criteria for the Evaluation of Clustering Method [J]. Journal of the American Statistical association, 1971, 66(336): 846-850.
14. Fahad A, Alshatri N, Tari Z, et al. A Survey of Clustering Algorithms for Big Data: Taxonomy and Empirical Analysis [J]. IEEE transactions on emerging topics in computing, 2014, 2(3): 267-279.
15. Ma H F, Ji Y G, Li X H, Zhou R N. A Microblog Hot Topic Detection Algorithm based on Discrete Particle Swarm Optimization[C] //Proceedings of the 14th Pacific Rim International Conference on Artificial Intelligence, Phuket, Thailand. August 22-26 2016.
16. Dai T, Wu Y, Lei D J. Hot Topic Summarization on Microblog Generated by Model Combination [J]. Application Research of Computers, 2016, 33(7):2026-2029.