IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 13, 2020, accepted February 27, 2020, date of publication March 2, 2020, date of current version March 12, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2977424

Method to Predict Bursty Hot Events on Twitter Based on User Relationship Network

XICHAN NIE[1], WANSHAN ZHANG[1,2], YANG ZHANG[1], AND DUNHUI YU[1,2]
[1]College of Computer and Information Engineering, Hubei University, Wuhan 430062, China
[2]Education Informationization Engineering and Technology Center, Wuhan 430062, China

Corresponding author: Wanshan Zhang (1197813795@qq.com)

This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFB1400602 and Grant 2016YFB0800401, in part by the National Natural Science Foundation of China under Grant 61572371 and Grant 61702377, and in part by the Technology Innovation Special Program of Hubei Province under Grant 2018ACA13.

ABSTRACT In the field of hot event prediction on online social networks, not considering user information leads to poor prediction effect. In this paper, a novel method that considers the behaviors and characteristics of users is proposed to identify and predict suspected bursty hot events. First, the keywords in each tweet are extracted and divided into different sets according to part of speech, and then similar topics are clustered according to semantic similarity. Second, the growth rates of topics are monitored in the sliding timestamp and the suspected bursty hot events are marked. Then, a user relationship network is constructed based on the information of the registered users on Twitter. Finally, according to the propagation trend of suspected bursty hot events in the network, the quasi-burst hot events are marked and sorted in descending order. Experimental results show that only using the historical re-tweeting behavior of users as the judgment basis to predict the current re-tweeting probability of users will lead to the phenomenon of error cascading, while taking the information of users into account can effectively improve the prediction performance. Compared with the existing methods, the proposed method improves the prediction precision rate by 27.38%, accuracy rate by 23.49%, and recall rate by 20.16%, demonstrating that it can predict bursty hot events effectively.

INDEX TERMS Hot event prediction, suspected bursty hot events, semantic similarity, user relationship network.

I. INTRODUCTION

Bursty hot events in news reports spread rapidly through the Internet [1] and have a great impact on society [2]. As a mainstream social media network, Twitter is used worldwide. The number of active Twitter users reached 645 million according to a study by the Statistic Brain Research Institute in July 2014 (http://www. statisticbrain.com). Therefore, it is desirable to identify suspected bursty hot events on Twitter and predict their spread trend, which can be used to inform the government and enterprise departments in a timely manner to help them take steps for scientific and effective control and guidance, so as to contribute to social stability and people's well-being [3].

A key problem in predicting bursty hot events on Twitter is to predict the spreading path of tweets. The current mainstream academic community focuses on the information cascade, classification, and game theory models for prediction. The information cascade model mainly includes the independent cascade model (ICM) [4], [5] and linear threshold model (LTM) [6], [7]. This kind of model assumes that a node has active and inactive states at a given time, and an active node activates an inactive node with an independent activation probability. Dichkens *et al.* used an independent cascade model to predict the re-tweet path of messages on Twitter and then used a Markov chain Monte Carlo method to estimate its parameters [8]. This model assumes that the probability of activation between nodes follows a β distribution of users' historical re-tweets. Most of these methods consider that the decisions of individuals can only be influenced by decisions of their direct neighbors, but in the case of iterative propagation (e.g., secondary re-tweeting), the influence of the central node is not considered. Kempe *et al.* proposed an LTM that assumes that the propagation behavior of an individual depends on whether the sum of the effects on it of all neighbor nodes exceeds an activation threshold [9]. However, in reality, the propagation behavior of individuals is affected by their neighbor nodes independently. Saito *et al.* introduced

The associate editor coordinating the review of this manuscript and approving it for publication was Zhe Xiao.

a continuous time axis and added a time-delay parameter to each edge of the graph of information transmission [10], [11] to extend the ICM and LTM to asynchronous independent cascade (AsIC) and asynchronous linear threshold (AslT) models. However, such models assume that neighbor nodes have the same or similar influence over each other, regardless of the degree of intimacy between them. The prediction method based on a classification model is a special form of ICM that assumes whether a user is activated or not depends on a number of factors [12], and this kind of method relies on the accuracy of predicting the user's historical re-tweeting behavior. Prediction based on the game theory model assumes that every user plays a game of interest when receiving certain information and takes the most profitable strategy [13]–[16]. This approach, however, does not consider a user's interaction with information. In addition, Sanda *et al.* analyzed link prediction based on local similarity measures to find the optimal measurement method for predicting the linking of words and tags in future tweets [17]. Mahdi *et al.* studied link prediction in a two-tier social network, using the same person's connections on two social networks: Twitter (directed network) and Foursquare (undirected network), and using the information layer (Twitter and Foursquare) structure to predict links in the Foursquare network [18]. These two methods consider the judgment and recognition of the social relationship between people, and do not consider the prediction of the propagation of unexpected events.

Above all, to improve the accuracy of bursty hot event prediction on Twitter, a prediction method based on the platform's structure and content information is proposed.

(1) The first step is to identify suspected bursty hot event (*SE*) on Twitter. First, the keywords of tweets in the current time stamp are extracted and similar topics clustered based on the web ontology language(OWL), and then *SE* is captured by monitoring the growth rate of topics. Finally, *SE* is placed in the suspected bursty hot event list (*SL*).

(2) The second step is to predict the propagation path of *SE* on the user relationship network. First, by analyzing users' behavior and social relationships between them, the Twitter registered user relationship network can be constructed to predict the propagation path of *SE*. *SE*s are then labeled as quasi-burst hot events (*PE*) and put into a quasi-burst hot event list (*L*).The heat of *PE* (*HEAT*(*PE*)) in the next time stamp can be predicted based on the propagation path, and the *PE* with low *HEAT*(*PE*) can be sifted to update *L*.

(3) At the same time, the growth rate of *SE* in *SL* is monitored and those events with decreasing or stable growth rate in the current time stamp removed from *SL*.

Experimental results show that the proposed method is effective and feasible. It can improve the accuracy of hot event prediction effectively and provide a new idea for bursty hot event prediction.

II. OVERALL SOLUTION

In order to solve the problem of bursty hot event prediction on Twitter, a method based on OWL is proposed to identify and monitor suspected bursty hot events (IMSB). A method is then proposed to predict bursty hot events based on a user relationship network (PBUR). Compared with the traditional prediction method, the proposed method features the following improvements. (1) When calculating the probability of activation, it considers the influence of inactive nodes' neighbors and mentions @ (calling grammar on Twitter); furthermore, the intimacy between users and the degree of users' interest in information are introduced as influencing factors at the same time. (2) The influence of the central node is introduced as the influencing factor in the iterative propagation. (3) When judging an individual's propagation behavior, the activation behavior of the active node relative to the inactive node is regarded as an independent event, and the event independence is used to calculate the re-tweeting probability of the inactive node. The overall system solution is shown in Figure 1:

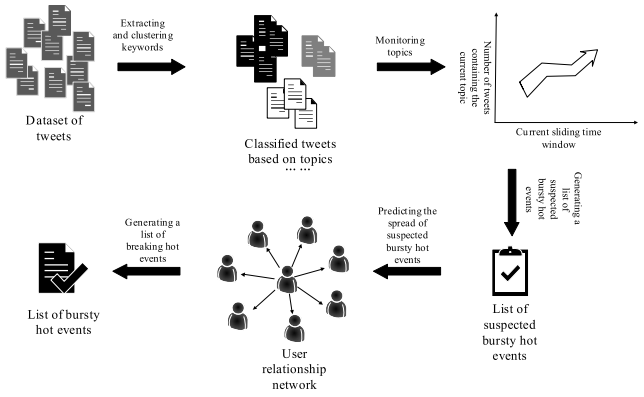


FIGURE 1. Overall solution of bursty hot event prediction.

III. OWL-BASED METHOD TO IDENTIFY AND MONITOR SUSPECTED BURSTY HOT EVENTS

To identify and monitor suspected bursty hot events on Twitter, extraction of the keywords of each tweet based on the TF-IDF algorithm is first performed, and then the semantic similarity between the two topics based on OWL are calculated, thus obtaining the semantic similarity of different topics of tweets, and similar topics are clustered based on a similarity matrix. Finally, the growth rate of topics in a sliding timestamp with timespan p is monitored. Events with growth rates that exceed a certain threshold are marked as *SE* and placed in the *SL*.

A. KEYWORD EXTRACTION BASED ON TF-IDF

Keywords of tweets based on the TF-IDF algorithm [19] are extracted as follows.

(1) Calculate the frequency of a word *TF* [19]:

$$TF = \frac{C_t}{C_{\max}}, \tag{1}$$

where C_t is the number of times the word t appears in tweets, and $C_{\max}$ is the number of times of those words that appear most frequently in tweets.

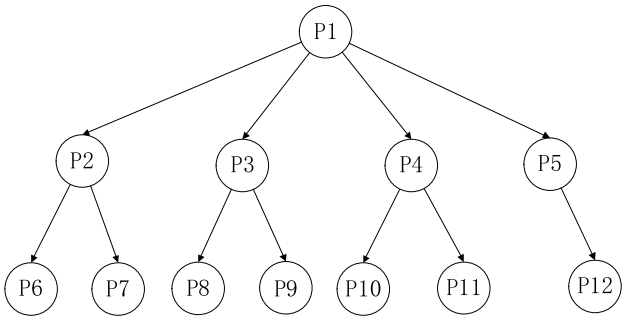


FIGURE 2. Domain ontology node diagram.

(2) Calculate the inverse document frequency IDF [19]:

$$IDF = \log \frac{N}{N_t + 1}, \tag{2}$$

where N is the total number of tweets in the Twitter corpus, and N_t is the number of tweets including the word t.

(3) Calculate $TF - IDF$ and extract keywords [16]:

$$TF - IDF = TF \times IDF. \tag{3}$$

Finally, $TF - IDF$ is calculated for each word and sorted in descending order. The $Top - k$ words are selected, and the nouns are put into the noun set η_n and the verbs into the verb set ζ_n.

B. TOPIC CLUSTERING BASED ON OWL

Since different keywords may be used interchangeably in different contexts without changing the syntactic and semantic structure of the text, similar topics were clustered based on OWL. The topics on different tweets were labeled as $T_i = \{\eta_i, \zeta_i\}$.

where $\eta_i = \langle n_1^i, n_2^i, \cdots n_s^i \rangle$, $\zeta_i = \langle v_1^i, v_2^i, \cdots v_t^i \rangle$; $i = 1, 2 \cdots, n$; $s + t \leq Top - k$.

To calculate the semantic similarity between the two topics T_1 and T_2, the semantic similarities between the noun sets and verb sets of the two topics must be calculated first. Therefore, the similarity between the keywords in the two sets were calculated, the matching between the keywords realized, and the semantic similarity between the words then recorded.

A word is treated as a node in domain ontology as shown in Figure 2. Assuming that $C(P1)$ is the number of upper common nodes that node $P1$ traces back to the root node in the domain ontology, $C(P1) \cap C(P2)$ can represent the semantic overlap between $P1$ and $P2$. The number of edges that cross by the shortest path connecting $P1$ and $P2$ represents the semantic distance between $P1$ and $P2$, which is denoted $Dis(P1, P2)$. When the semantic distance is the same, the conceptual semantic similarity of two words increases with the summation of the two levels they belong to, and decrease with the difference of the two levels. Therefore, in the calculation of semantic similarity between words, the depth of concept level is taken into account and the regulatory factor l set to adjust it.

1) SEMANTIC SIMILARITY OF TOPICS

The semantic similarity between the two noun sets η_1 and η_2 for example, is calculated, and the calculation between the verb sets is similar.

To obtain the similarity between the two subjects, the similarity between the words of the same part of speech is first calculated based on OWL, and then the similarity between the sets is calculated. Finally, the semantic similarity between subjects is calculated according to the matching logarithm of words between sets. To make the result more intuitive, the final similarity result is normalized. (1) Based on OWL, the semantic similarity between the two nouns n_1^1 and n_1^2 is calculated as

$$Sim'(n_1^1, n_1^2) = \frac{C(n_1^1) \cap C(n_1^2)}{Dis(n_1^1, n_1^2) \times (l \times |h_{n1}^1 - h_{n1}^2| + 1)}, \tag{4}$$

where $C(n_1^1) \cap C(n_1^2)$ is the semantic overlap between n_1^1 and n_1^2, $Dis(n_1^1, n_1^2)$ is the semantic distance between n_1^1 and n_1^2, l is a regulatory factor, and h_{n1}^1 and h_{n1}^2 are the conceptual depths corresponding to n_1^1 and n_1^2,respectively.

(2) The formula to adjust the semantic similarity according to the level of the keywords is:

$$l = \left(\frac{level(n_1^1)}{level_1} + \frac{level(n_1^2)}{level_2} \right) /2 \tag{5}$$

$$Sim''(n_1^1, n_1^2) = l \times Sim'(n_1^1, n_1^2), \tag{6}$$

where $level_1$ and $level_2$ are the depths of η_1 and η_2, respectively; and $level(n_1^1)$ and $level(n_1^2)$ are the depths of n_1^1 and n_1^2, respectively, in sets η_1 and η_2.

The sum of the maximum semantic similarity between the nouns in sets η_1 and η_2 can then be calculated.

$$\sum_{n_i^1 \in \eta_1, n_j^2 \in \eta_2} Max^{Sim(n_i^1, n_j^2)}, \quad i, j = 1, 2 \cdots n. \tag{7}$$

After extracting the maximum semantic similarity in the current set, the corresponding two words are removed from the set.

The maximum semantic similarity between ζ_1 and ζ_2 can be similarly calculated as

$$\sum_{v_i^1 \in \zeta_1, v_j^2 \in \zeta_2} Max^{Sim(v_i^1, v_j^2)}, i, j = 1, 2 \cdots n. \tag{8}$$

The semantic similarity between subjects can be obtained by multiplying the matching pairs of words between different sets by the maximum semantic similarity set of the set, and then adding and dividing by the total matching pairs:

$$Sim'(T_1, T_2) = \frac{C_1 \times \sum\limits_{n_i^1 \in \eta_1, n_j^2 \in \eta_2} Max^{Sim(n_i^1, n_j^2)} + C_2 \times \sum Max^{Sim(v_i^1, v_j^2)}_{v_i^1 \in \zeta_1, v_j^2 \in \zeta_2}}{C_1 + C_2}, \tag{9}$$

where C_1 and C_2 are the matching pairs of nouns and verbs, respectively.

(3) Result normalization:

$$Sim(T_1, T_2) = 1 - \mu_0^{Sim'(T_1,T_2)}. \quad (10)$$

where μ_0 is a normalized factor that is a positive real number with a value greater than 1.

2) SIMILAR TOPICS CLUSTERING

The semantic similarity calculated in the preceding section is stored in the similarity matrix $SimArr \in R^{n\times n}$. n is the number of topics to be matched, $SimArr[i][j]$ is the semantic similarity between T_i and T_j, and it can be seen that $SimArr$ is a symmetric matrix, and the main diagonal element of the matrix is set to 0.

Based on the similarity matrix, a bottom-up aggregation method is used to cluster similar topics.

(1) Treat each topic to be assigned as a class cluster in advance, and set the semantic similarity threshold α for clustering.

(2) The semantic similarity of two class clusters is the maximum semantic similarity between the current class clusters, and if its value is greater than α, then the two class clusters are merged.

(3) The value of semantic similarity between the merged clusters is defined as the mean value of semantic similarity between the pre-merged clusters.

(4) Repeat (2) and (3) until all topics with semantic similarity greater than a are clustered and the remaining topics are grouped into separate topic domains.

C. MONITOR TOPIC GROWTH

After clustering all topics, the growth rate of all topics in the topic domain within the current timestamp is monitored. If the monitoring topic is $TD[i]$, statistics of the number of tweets in each time stamp that can be classified as $TD[i]$, i.e., N_k, are generated in the sliding window of the current time. The growth rate of $TD[i]$ per time stamp, e.g., from the jth to the $j+1$th time stamp, is

$$G_j^{j+1}(TD[i]) = \frac{N_{TD[i]}^{j+1} - N_{TD[i]}^{j}}{N_{TD[i]}^{j}}, \quad (11)$$

where $N_{TD[i]}^{j}$ and $N_{TD[i]}^{j+1}$ are the numbers of tweets that can be classified as $TD[i]$ in the jth time stamp and the $j+1$th time stamp.

After a large number of statistics, it can be concluded that the growth rate of similar topics in two consecutive timestamps is maintained at $[\beta_1, \beta_2]$, if the current timestamp within $G_j^{j+1}(TD[i]) > \rho \times \beta_2$, ($\rho > 10$), labels the event corresponding to $TD[i]$ as an SE and it can then be inserted into the SL. In the next section the propagation path of SE is further predicted.

D. ALGORITHM IMPLEMENTATION

See Algorithm 1.

Algorithm 1 OWL-Based Method to Identify and Monitor Suspected Bursty Hot Events (IMSB).

input: Dataset of tweets W
output: list of suspected bursty hot events SL
1: Add all tweets within current time stamp to dataset of tweets W
2: for (each tweet $w_i \in W$)
3: Extract $Top-k$ keywords based on TF-IDF algorithm and put nouns in η_i, put verbs in ζ_i.
4: end for
5: for (η_i, $i \leftarrow 1$ to n)
6: for (η_j, $j \leftarrow 1$ to n)
7: if (i!=j) then
8: Compute semantic similarity between all nouns in two sets $Max^{Sim(n_s^i, n_t^j)}$
9: end if
10: end for
11: end for
12: Sum maximum semantic similarity in noun set $\sum_{v_s^i \in \zeta_i, v_t^j \in \zeta_j} Max^{Sim(v_s^i, v_t^j)}$
13: for (ζ_i, $i \leftarrow 1$ to n)
14: for (ζ_j, $j \leftarrow 1$to n)
15: if (i!=j) then
16: Compute semantic similarity between all verbs in two sets $Max^{Sim(v_s^i, v_t^j)}$
17: end if
18: end for
19: end for
20: Sum maximum semantic similarity in verb set $\sum_{v_s^i \in \zeta_i, v_t^j \in \zeta_j} Max^{Sim(v_s^i, v_t^j)}$
21: for (any two tweets w_i, $w_j \in W$)
22: Calculate semantic similarity of topics in tweet $Sim(T_i, T_j)$
23: end for
24: Save topic semantic similarity in similarity matrix, and set main diagonal element to 0, to obtain topic domain $TD[k]$
25: while (in sliding timestamp with time span of p)
26: Monitor growth rate of topics in $TD[k]$, mark an event as SE for a topic whose growth rate exceeds the threshold $\rho \times \beta_2$, and put it into SL.
27: end while
28: return SL

IV. METHOD TO PREDICT BURSTY HOT EVENTS BASED ON USER RELATIONSHIP NETWORK (PBUR)

To predict PE, the user relationship network according to data of user behavior and the social relationships among users in Twitter are first constructed. The propagation path of the SE captured in the preceding section is predicted, its heat calculated, and L in descending order of heat finally obtained.

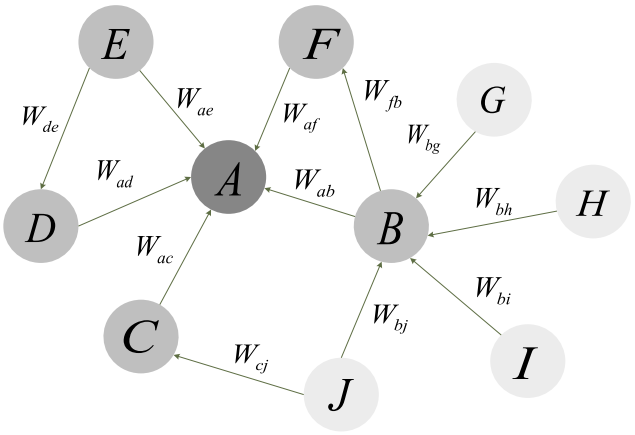


FIGURE 3. Twitter user relationship graph for bursty hot event prediction.

A. BUILDING USER RELATIONSHIP NETWORK

Based on the complex network [20], a Twitter user relationship network is constructed as shown in Figure 3. Node A is the original user that posted the tweet; that is, the central node. If user B pays attention to user A, then there is a one-way edge from B to A marked $e_{B\to A}$ between the two nodes, and the weight of $e_{B\to A}$, W_{ab}, is the intimacy between users A and B. The degree of exit and degree of entry of a node are the amount of the users' followees and followers.

When calculating the activation probability of an active node to an inactive node, it is necessary to consider the influence of the neighbor node of the inactive node, whether @ is included, the intimacy of the two nodes to the users, and the interest of the inactive node to the information.

The influence of a node is calculated as follows.

After much research, several main factors that impact the influence of nodes are selected: degree of entry for a node Deg, degree of node activity Act, and degree of a node to the user's past popularity of tweets Hot.Principal component analysis is used to determine the weights of the factors p_1, p_2, and p_3. The influence of node A can be obtained by multiplying each influence factor by its corresponding weight and then adding them together.

$$I(A) = Deg(A) \times p_1 + Act(A) \times p_2 + Hot(A) \times p_3. \quad (12)$$

Those nodes that have the most influence are those having the most social impact on society. These nodes are called Hub nodes. In the complex network, they play a key role in the dissemination of news and opinions [21]–[24].

B. PROPAGATION PATH PREDICTION

The user information network constructed in the preceding section is used to predict the path and trend of suspected hot spots in Twitter.

Twitter users are online or offline. In propagation path prediction, only the retweeting behavior of a current online user is predicted, and the state of the user remains unchanged in the prediction. Nodes with topics that are in the same domain $TD[k]$ are put into the same domain $ND[k]$. In the Twitter user relationship network, the propagation paths of events corresponding to different themes are predicted, and the propagation paths between different themes do not affect each other.

The activation probability of a node is calculated based on a binary logistic regression model, and then the re-tweeting behavior of users is predicted. The first forwarder-tweet case and the iterative forwarder-tweet case are described below.

1) FIRST RE-TWEETING

From Figure 3, node A is the original publishing user, i.e., the central node, and the topic $T_1 = \{\eta_1, \zeta_1\} \in TD[k]$ to be predicted is taken as an example. The prediction of an event's propagation path in the user relationship network must predict the re-tweeting behavior of non-active nodes, i.e., the re-tweeting behavior of nodes B, C, D, E and F. Taking node B as an example, the remaining nodes are the same. Node B must satisfy the following conditions.

There is a directed edge $e_{B\to A}$ from B to A between node B and node A, $B \notin ND[1]$.

When considering the influence factors of a node activation probability, the intimacy degree between users and the interest degree of users are introduced as the new influence factors based on the traditional method. The factors are as follows. Influence of node A: whether the tweet posted by node A contains an @ to the node; that is whether node A uses @ to call nodes B; the intimacy between nodes A and B; and how interested node B is in the topic T_1.

The probability that A will activate B is

$$P_{AB} = \frac{1}{1 + e^{-\lambda^T x}}, \quad 0 \le P_{AB} \le 1. \quad (13)$$

where $x^T = \{x_1, x_2, x_3, x_4\}$, x_1 is the influence of node A, x_2 is the intimacy between nodes A and B, x_3 is the node A whether containing @ or not, and x_4 is the interest of node B to the topic T_1; $\lambda^T = \{\lambda_1, \lambda_2, \lambda_3, \lambda_4\}$, λ_1, λ_2, λ_3, $\lambda_4$are the weights of each component.

We treat the activation behaviors between nodes as independent events that do not affect each other. In the case of node B, the re-tweeting probability is affected by nodes A and F, which can be expressed as

$$P_{retweet}(B) = 1 - (1 - P_{AB})(1 - P_{FB}). \quad (14)$$

When $P_{retweet}(B)$ is greater than or equal to 0.5, node B is considered to re-tweet, and vice versa.

2) ITERATIVE RETWEETING

From (1), it is predicted that if a node re-transmits an original tweet, then nodes G, H, I, and J all have the possibility of re-tweeting behavior. Taking node G as an example below, the remaining nodes are the same. Node G should satisfy the following conditions.

There is a directed edge $e_{G\to B}$ from G to B, $G \in ND[1]$.

Because a tweet is retransmitted, users may be affected by the influence of central nodes (such as celebrities and other influential nodes). Therefore, when considering the situation

of iterative re-tweeting, based on the four factors in the first re-tweeting situation, the influence of the central node is introduced as one of the factors that affects the activation probability, and the overall factors are the following: the central node A influence, the influence of node B, the degree of intimacy between users B and G, whether or not node B uses @ to connect node G; and node G's interest in topic T_1.

Furthermore, the re-tweeting behavior of node G is predicted using the binary logistic regression model.

Step (2) is repeated to predict all nodes in the network.

3) TRAINING

Since there is an error between the predicted and true values, a difference function (cost function) is defined as

$$Cost(\lambda) = -\frac{1}{n}\left[\sum_{i=1}^{n} y_i \log(P_{retweet}) + (1 - y_i)\log(1 - P_{retweet})\right], \tag{15}$$

where $P_{retweet}$, y_i is the training set, and n is the attention of the predicted nodes.

To ensure the accuracy of the prediction is the minimum of the difference function, the gradient descent parameter is used to obtain the optimal parameter.

C. SPECULATING ON TOPIC POPULARITY

For theme $T_k = \{\eta_k, \zeta_k\}$, if the number of nodes covered in the current timestamp i is $|V|^i$, then the heat of theme T_k is

$$HEAT(T_k)^i = \frac{|V|^i}{V_{sum}}, HEAT(T_k)^i \in (0, 1), \tag{16}$$

where V_{sum} is the number of all Twitter users in the current time stamp.

The events are sorted in descending order of heat, and the list of bursty hot events L is obtained.

D. ALGORITHM IMPLEMENTATION

See Algorithm 2.

Algorithm 2 Method to Predict Bursty Hot Events Based on User Relationship Network (PBUR).

input: List of suspected bursty hot events SL, dataset of Twitter user U
output: List of quasi-burst hot events L
1: Construct Twitter user relationship network based on U
2: for (every suspected bursty hot events $SE \in SL$)
3: Select central node
4: while (select followers of central node)
5: Predict the probability that central node activates its followers
6: while (use breadth-first search algorithm to traverse entire network)
7: if (current node has not been predicted)
8: Calculate activation probability of all its neighbor nodes, and judge their re-tweeting behavior accordingly
9: if (re-tweet)
10: Continue to predict their followers
11: end if
12: if (not re-tweet)
13: break
14: end if
15: end if
16: end while
17: end while
18: Extrapolate heat of event corresponding to topic
19: end for
20: Sort events in descending order of heat and generate quasi-burst hot event list L
21: return L

V. EXPERIMENT

A. ACQUISITION OF DATASETS

To verify the validity and rationality of the algorithm proposed in this paper, an experiment based on web crawler technology and the Twitter API interface was conducted. Recent tweets were collected as the training dataset to predict bursty hot events. The dataset contained 90,753 tweets from 1.98 million Twitter users with 25 million followers. Each tweet had approximately 70 re-tweets on average.

After obtaining the dataset, the data was pre-processed, zombie users whose followers were below a certain threshold filtered, and characters that interfered with the extraction of the keywords, such as emojis, removed, and short tweets filtered. The symbols used in the experiment are shown in Table 1.

TABLE 1. Identifier symbols.

Symbol	Explanation
TP	number of correctly predicted bursty hot events
FP	number of incorrectly predicted normal events
FN	number of incorrectly predicted bursty hot events
TN	number of correct predicted normal events

B. LABORATORY TOOLS AND ENVIRONMENT

The experimental environment comprised the Windows 10 operating system running on an Intel I5-7200u CPU with 16 Gb of memory. The experiment was realized on PyCharm in the Python language.

C. INDICATORS OF EVALUATION

To solve the problem of data imbalance [25], the experimental results were analyzed by establishing a confusion matrix [26]. To facilitate the description of the problem, the confusion matrix is shown in Table 2.

The evaluation index of the algorithm that was selected includes recall rate (RR), precision rate (PR) and $F-measure$ [21] as follows.

TABLE 2. Confusion matrix.

Confusion matrix		Actual sample	
		1	0
predicted sample	1	*TP*	*FP*
	0	*FN*	*TN*

TABLE 3. Experimental parameters.

Parameter name	Value
Depth of traversal	1, 2, 3, 4, 5
Semantic similarity threshold α	0.5, 0.6, 0.7, 0.8, 0.9
Number of tweets	1000, 3000, 5000, 7000, 9000

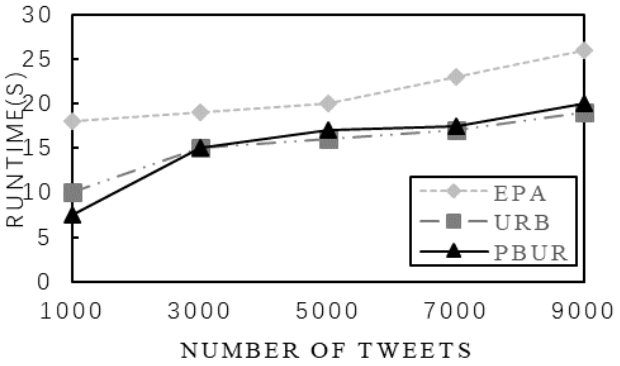


FIGURE 4. Comparison of runtime of the three methods.

(1) The recall rate is

$$RR = \frac{TP}{(TP + FN)}. \tag{17}$$

(2) The precision rate is

$$PR = \frac{TP}{(TP + FP)}. \tag{18}$$

(3) The F-measure is

$$F = \frac{2 \times PR \times RR}{(PR + RR)}. \tag{19}$$

D. EXPERIMENTAL DESIGN

The EPA [4] and URB algorithms [27] were selected for comparison with the proposed algorithm. The EPA, URB, and PBUR algorithms were compared and analyzed in three aspects: precision rate, recall rate, and F-measure. The experimental parameters are shown in Table 3.

Analysis of time complexity:

The results of the runtime experiment are shown in Figure 4. Generally speaking, the runtime of the PBUR algorithm is almost the same as that of the URB algorithm, but much less than that of the EPA algorithm. In particular, when the number of tweets changed from 1000 to 9000, the three algorithms' runtimes increased slowly within a reasonable range but the EPA algorithm took the longest.

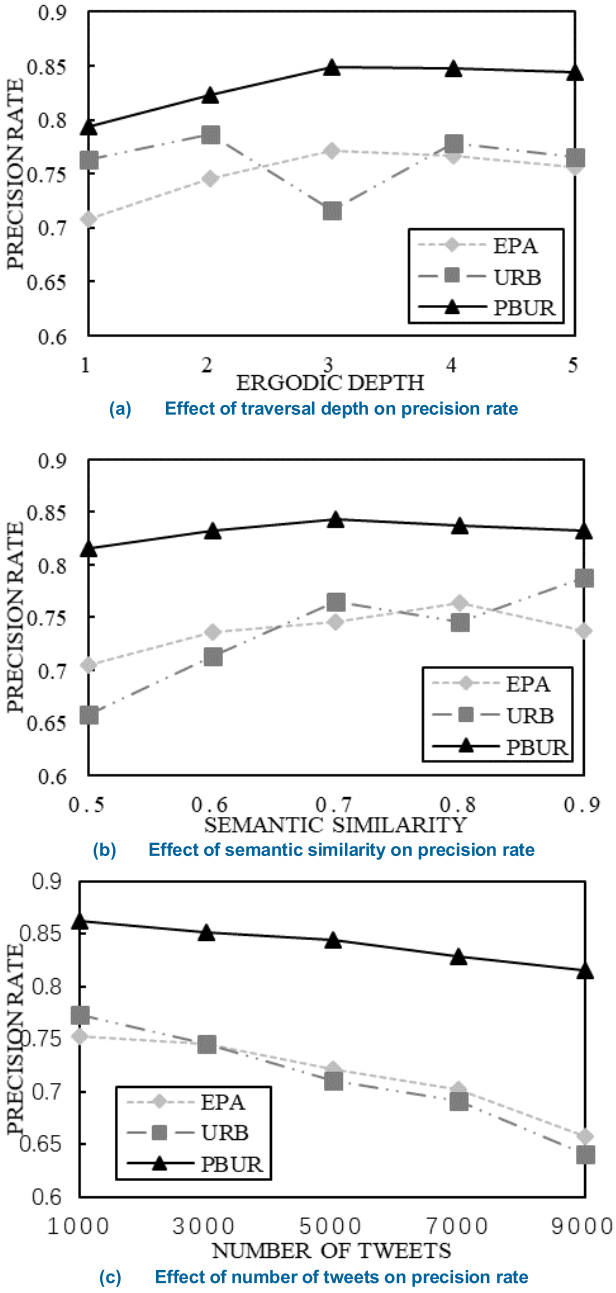
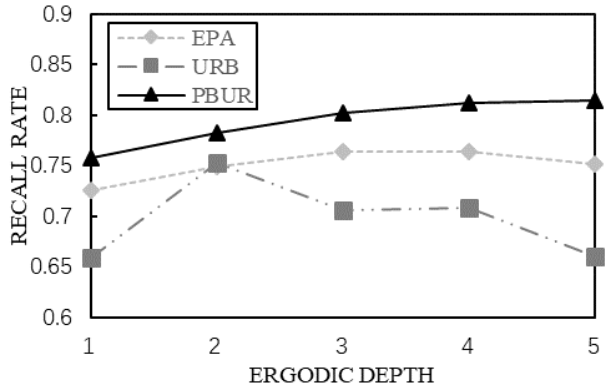
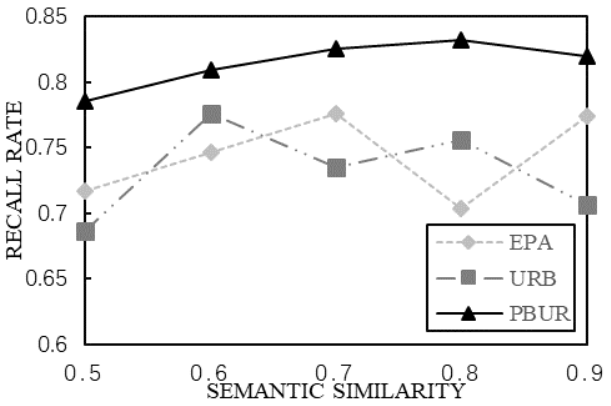


FIGURE 5. Comparison of accuracy of the three methods studied.

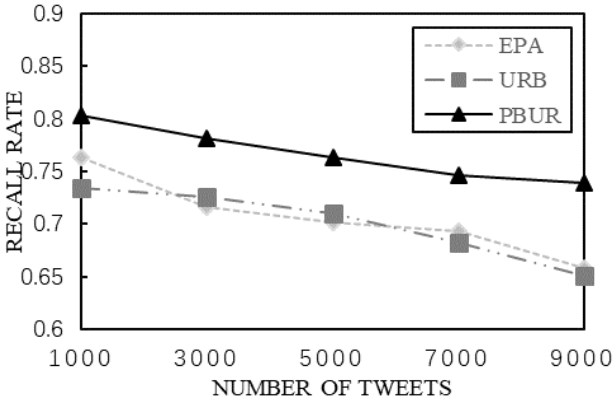
Regarding the precision rate of prediction, the specific experimental results are shown in Figure 5. Compared with the EPA and URB algorithms, the precision rate of the PBUR algorithm is greatly improved. When the ergodic depth increases, the precision rate of the PBUR and EPA algorithms increases, but the precision rate of the URB algorithm fluctuates greatly because its re-tweeting probability greatly depends on users' historical behavior; when the semantic similarity threshold α increases, the precision rates of both the EPA and URB algorithms fluctuate greatly, while that of the



(a) Effect of traversal depth on recall rate

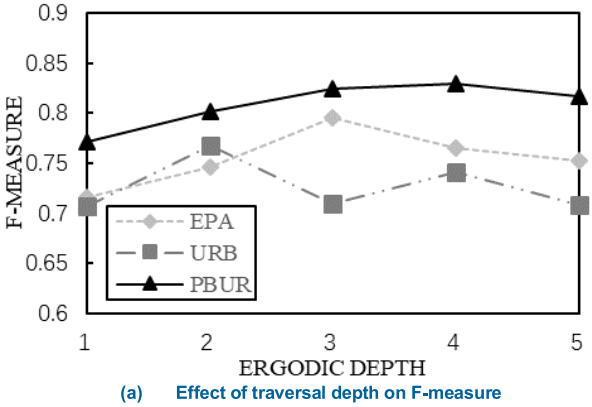


(b) Effect of semantic similarity on recall rate

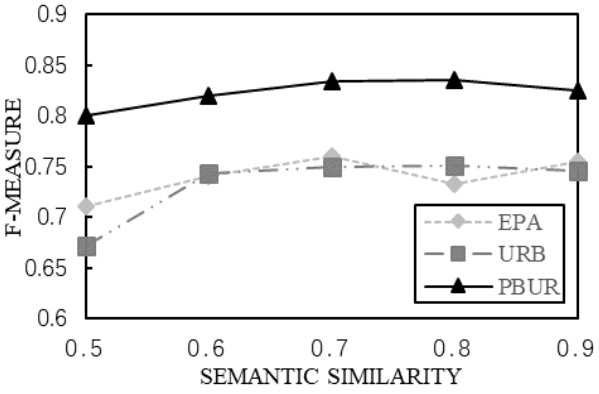


(c) Effect of number of tweets on recall rate

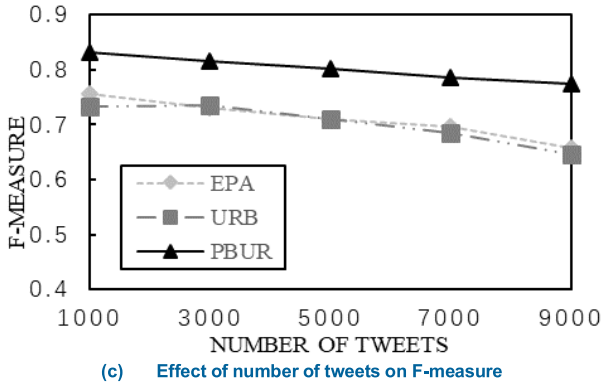
FIGURE 6. Comparison of recall of the three methods studied.



(a) Effect of traversal depth on F-measure



(b) Effect of semantic similarity on F-measure



(c) Effect of number of tweets on F-measure

FIGURE 7. Comparison of F-measures of the three methods studied.

PBUR algorithm remains relatively stable and its precision rate is always higher than that of the other two algorithms. When the number of tweets increases, the precision rate of all three algorithms decreases to some extent, but that of the PBUR algorithm remains relatively stable.

Regarding the recall rate of prediction, the specific experimental results are shown in Figure 6. The recall rate of the PBUR algorithm is always superior to those of the EPA and URB algorithms. The ergodic depth has little influence on the PBUR and EPA algorithms, and the recall rate of the URB algorithm varies greatly. When the semantic similarity threshold α is increased, the recall rates of the PBUR and EPA algorithms both increase slightly.

Regarding the F-measure of prediction, the experimental results are shown in Figure 7. When the ergodic depth increases, the F-measure of the PBUR algorithm is always superior to that of the EPA and URB algorithms; when the semantic similarity threshold α increases, the F-measure of the PBUR algorithm is stable but the F-measures of both the EPA and URB algorithms are in an unstable state; when the number of tweets increases, the F-measure of the PBUR algorithm changes slightly, but it is always better than that of the EPA and URB algorithms.

It is not difficult to acknowledge the following from the above experimental results.

(1) In three aspects, the PBUR algorithm is always better than the EPA and URB algorithms.

(2) When the number of tweets increases, the precision rate, recall rate, and F-measure all remain relatively stable.

(3) When the ergodic depth increases, the EPA algorithm fluctuates greatly because the re-tweeting probability of it greatly depends on users' historical behavior.

(4) The fluctuation of the semantic similarity threshold α has a great impact on both the EPA and URB algorithms.

In summary, it can be seen that the proposed method has the value of practical application and can effectively solve the problem of inaccurate prediction caused by lack of considering user information.

VI. CONCLUSION

In this paper, prediction of bursty hot events in the Twitter environment were studied and various factors considered, such as the degree of intimacy between users, the degree of users' interest in information, and the influence of the central node on information dissemination.

The method proposed solves the problem of low forecasting efficiency caused by the lack of consideration of factors in previous research. The experimental results show that the proposed method is effective and feasible, and it provides a new idea for the prediction of bursty hot events in the Twitter environment.

Planned future research will focus on the following two areas.

(1) Combining the neural network in deep learning to extract the sequence information and block information in tweets, and then constructing the model for each word in order to obtain the keywords and suspected bursty hot events.

(2) Consideration of other influencing factors of information transmission and train the model to improve its efficiency.

REFERENCES

[1] Z. M. Zhong, Y. Guan, and C. H. Li, "Localized top-K bursty event detection in microblog," *Chin. J. Comput.*, vol. 41, no. 427, pp. 76–88, 2018.

[2] C. J. Hu, W. W. Xu, and Y. Hu, "Review of information diffusion in online social networks," *J. Electron. Inf. Technol.*, vol. 39, no. 4, pp. 794–804, 2017.

[3] D. Helbing, D. Brockmann, and T. Chadefaux, "What complexity science and information systems can contribute," *J. Stat. Phys.*, vol. 158, no. 3, pp. 735–781,2015.

[4] X. Zhu, Y. Jia, and Y. P. Nie, "Event propagation analysis on microblog," *J. Comput. Res. Develop.*, vol. 52, no. 2, pp. 437–444, 2015.

[5] C. Wang, W. Chen, and Y. Wang, "Scalable influence maximization for independent cascade model in large-scale social networks," *Data Mining Knowl. Discovery*, vol. 25, no. 3, pp. 545–576, Apr. 2012.

[6] W. Galuba, K. Aberer, and D. Chakraborty, "Outtweeting the Twitterers-predicting information cascades in microblogs," in *Proc. WOSN*, Jun. 2010, pp. 1–9.

[7] G. Amit, W. Lu, and V. S. Lakshmanan, "SIMPATH: An efficient algorithm for influence maximization under the linear threshold model," in *Proc. IEEE 11th Int. Conf. Data Mining*, Vancouver, BC, Canada, Dec. 2011, pp. 211–220.

[8] L. Dickens, I. Molloy, J. Lobo, P.-C. Cheng, and A. Russo, "Learning stochastic models of information flow," in *Proc. IEEE 28th Int. Conf. Data Eng.*, Apr. 2012, pp. 570–581.

[9] D. Kempe, J. Kleinberg, and É. Tardos, "Maximizing the spread of influence through a social network," in *Proc. 9th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2003, pp. 137–146.

[10] K. Saito, M. Kimura, and K. Ohara, "Behavioral analyses of information diffusion models by observed data of social network," in *Proc. 3rd Int. Conf. Social Comput., Behav. Modeling, Predict.*, Bethesda, MD, USA, Mar. 2010, pp. 149–158.

[11] K. Saito, M. Kimura, and K. Ohara, "Selecting information diffusion models over social networks for behavioral analysis," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, 2010, pp. 180–195.

[12] Y. Li, Y.-H. Chen, and T. Liu, "Survey on predicting information propagation in microblogs," *J. Softw.*, vol. 27, no. 2, pp. 247–263, 2016.

[13] S. A. Myers and J. Leskovec, "Clash of the contagions: Cooperation and competition in information diffusion," in *Proc. IEEE 12th Int. Conf. Data Mining*, Dec. 2012, pp. 539–548.

[14] M. Gomez-Rodriguez, J. Leskovec, and B. Schölkopf, "Modeling information propagation with survival theory," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 666–674.

[15] L. Weng, A. Flammini, A. Vespignani, and F. Menczer, "Competition among memes in a world with limited attention," *Sci. Rep.*, vol. 2, no. 1, p. 335, Mar. 2012.

[16] Y. Su, X. Zhang, L. Liu, S. Song, and B. Fang, "Understanding information interactions in diffusion: An evolutionary game-theoretic perspective," *Frontiers Comput. Sci.*, vol. 10, no. 3, pp. 518–531, Jan. 2016.

[17] S. Martinčić-Ipšić, E. Moćibob, and M. Perc, "Link prediction on Twitter," *PLoS ONE*, vol. 12, no. 7, Jul. 2017, Art. no. e0181079.

[18] M. Jalili, Y. Orouskhani, M. Asgari, N. Alipourfard, and M. Perc, "Link prediction in multiplex online social networks," *Roy. Soc. Open Sci.*, vol. 4, no. 2, Feb. 2017, Art. no. 160863.

[19] W. S. Zhang, Y. Xiao, and J. J. Liang, "Personalized recommendation of Web resources based on topic clustering," *Microelectron. Comput.*, vol. 4, pp. 35–39, Aug. 2015.

[20] T. Zhou, W. J. Bai, and B. H. Wang, "A brief review of complex networks," *Physics*, vol. 34, no. 1, pp. 31–36, 2016.

[21] J. Yang and J. Leskovec, "Modeling information diffusion in implicit networks," in *Proc. IEEE Int. Conf. Data Mining*, Dec. 2010, pp. 599–608.

[22] E. Bakshy, J. M. Hofman, and W. A. Mason, "Everyone's an influencer: Quantifying influence on Twitter," in *Proc. WSDM*, Hong Kong, 2011, pp. 65–74.

[23] E. Bakshy, B. Karrer, and L. A. Adamic, "Social influence and the diffusion of user-created content," in *Proc. 10th ACM Conf. Electron. Commerce*, 2009, pp. 325–334.

[24] N. Barbieri, F. Bonchi, and G. Manco, "Topic-aware social influence propagation models," *Knowl. Inf. Syst.*, vol. 37, no. 3, pp. 555–584, Apr. 2013.

[25] M. Yang, J. M. Yin, and G. L. Ji, "Classification methods on imbalanced data: A survey," *J. Nanjing Normal Univ.*, vol. 8, no. 4, pp. 7–12, 2008.

[26] Y. Hu, "Research on the topics discovery and hot preciction of weibo public opinion in emergencies," M.S. thesis, Jun, Xi'an Univ., Xi'an, China, Jun. 2018.

[27] Z. Yang, J. Guo, K. Cai, J. Tang, J. Li, L. Zhang, and Z. Su, "Understanding retweeting behaviors in social networks," in *Proc. 19th ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, 2010, pp. 1633–1636.

XICHAN NIE was born in Wuhan, Hubei, China, in 1999. She is currently pursuing the degree with Hubei University, China.

Her research interests include knowledge domains map and spatial crowdsourcing. She received the 2019 National Scholarship.

WANSHAN ZHANG was born in Wuhan, Hubei, China, in 1973. He received the M.S. degree from Hubei University, China.

He is currently a Teacher with the Computer Science Department, School of Computer and Information Engineering, Hubei University, and a Researcher with the Hubei Education Information Engineering Technology Center. He has published more than ten scientific articles in core journals and conferences. In the past five years, he has hosted and participated in several projects, including the Natural Science Foundation of Hubei Province and the Education Department of Hubei Province.

YANG ZHANG was born in Enshi, Hubei, China, in 1999. He is currently pursuing the degree with Hubei University, China. His research interest includes data mining.

DUNHUI YU was born in Wuhan, Hubei, China, in 1974. He received the M.S. degree from Hubei University, China, and the Ph.D. degree from Wuhan University, China.

He is currently the Director of the Software Engineering Department, School of Computer and Information Engineering, Hubei University, and the Deputy Director of the Hubei Education Information Technology Center. His research interests include service computing and big data. In the past five years, he has hosted several projects, including the Natural Science Foundation of Hubei Province and the Education Department of Hubei Province. He has published more than 20 scientific articles in core journals and conferences. He participated in the preparation of an academic monograph and applied for invention patents.

Dr. Yu is a member of China Computer Application Committee and Chinese Computer Society. He received the Wuhan Excellent Academic Paper Award. He has participated in many projects such as 973, 863, and the National Natural Science Foundation. He presided over a number of horizontal issues such as the ''four-in-one'' performance appraisal system for Guizhou local taxes, with a total investment of more than 3.5 million yuan.