Article

JIS

Journal of Information Science
1–16

DOI: 10.1177/0165551514541614
jis.sagepub.com
SAGE

Detecting hot topics from Twitter: A multiview approach

Yixiang Fang
Shenzhen Graduate School, Harbin Institute of Technology

Haijun Zhang
Shenzhen Graduate School, Harbin Institute of Technology

Yunming Ye
Shenzhen Graduate School, Harbin Institute of Technology

Xutao Li
Shenzhen Graduate School, Harbin Institute of Technology

Abstract
Twitter is widely used all over the world, and a huge number of hot topics are generated by Twitter users in real time. These topics are able to reflect almost every aspect of people's daily lives. Therefore, the detection of topics in Twitter can be used in many real applications, such as monitoring public opinion, hot product recommendation and incidence detection. However, the performance of traditional topic detection methods is still far from perfect largely owing to the tweets' features, such as their limited length and arbitrary abbreviations. To address these problems, we propose a novel framework (MVTD) for Twitter topic detection using multiview clustering, which can integrate multirelations among tweets, such as semantic relations, social tag relations and temporal relations. We also propose some methods for measuring relations among tweets. In particular, to better measure the semantic similarity of tweets, we propose a new document similarity measure based on a suffix tree (STVSM). In addition, a new keyword extraction method based on a suffix tree is proposed. Experiments on real datasets show that the performance of MVTD is much better than that of a single view, and it is useful for detecting topics from Twitter.

Keywords
Multirelation; multiview clustering; suffix tree; Twitter topic detection

1. Introduction

The growth of the Internet in recent years is making social media (such as blogger and Twitter) part of daily life. Twitter, a micro-blogging platform, fills the gap between blogging and instant messaging service. Unlike other social media, it has some special features [1, 2], such as allowing users to post short texts (140 characters or less) as well as images, videos and urls, and the relationship of following and being followed by users requiring no reciprocation. Twitter has attracted thousands of millions of participants. Since users can post anything that pops into their mind at any time, the information in Twitter can reflect almost every aspect of people's daily lives. Thus, the detection of topics, especially hot topics, from Twitter can be used in many applications, such as monitoring public opinion, hot product recommendation and incidence detection. Among these topics, this paper mainly focuses on topics of broad attention, that is, hot topics, since they can reflect the mainstream opinions of the society.

Corresponding author:
Yixiang Fang, Shenzhen Graduate School, Harbin Institute of Technology, Rm 202, C# Building, HIT Campus at Xili University Town, Shenzhen 518055, People's Republic of China.
Email: yxfang@cs.hku.hk

Topic detection, also known as event detection, has been studied for decades. There are many topic detection methods such as single-pass based topic detection [3], a Latent Dirichlet Allocation (LDA) model [4] and a graph-based method [5]. All of these traditional methods mainly work on lengthy texts, and they often assume that the feature spaces extracted from the available texts are effective enough to detect topics. However, the text information in Twitter is extremely limited owing to the limitation of the number of words in tweets. As a result, it is unable to provide sufficient statistical information for measuring the similarity between two tweets. Therefore, even though the text information is important for topic detection, the traditional topic detection methods may not perform well for Twitter datasets. In addition, arbitrary abbreviations, meaningless symbols and newly created words are widely used in Twitter. All of these aspects of Twitter make it difficult to detect topics using traditional methods, and thus new approaches for detecting topics from Twitter should be developed with careful consideration of these aspects.

In order to better understand the more detailed features of hot topics, we conducted a careful observation of hot topics in Twitter and found that: (a) the hot topics are usually represented by consecutive words, that is, phrases, which can convey more detailed meaning; (b) activities such as annotation, retreet and reply for hot topics will be much greater than those for topics receiving less attention, and hot topics usually appear with many hashtags, which implies that Twitter users have treated them as topics by themselves; and (c) in most cases, the related tweets of a particular topic appear in a particular range of time and regions, with few topics receiving attention for a long time from people all over the world.

With these observations of hot topics in Twitter, the authors believe that, instead of only focusing on text semantic information, some new features can also be useful for detecting topics. For example, the user-annotated hashtags, generated by Twitter users when posting their tweets, can be easily identified since they are often in the form of '#tag', that is, a combination of a character '#' and some other characters. They often indicate particular hot topics in Twitter, and thus they can be used for measuring the relations of tweets. In addition, if one tweet is a retweet or a reply to another, this suggests that they may belong to the same topic with a high probability. Furthermore, the temporal relation, spatial relation and user profile relation of tweets can also be used for detecting topics from Twitter.

Therefore, tweets can be regarded as entities with multirelations, some of which are useful for detecting topics from Twitter. Figure 1 describes some possible relations among tweets. Fusing these relations may compensate for the weakness of text semantic relations among tweets, and thus may achieve better topic detection results. Relying on the above assumption, this paper introduces a new framework (MVTD) for detecting topics in Twitter using multiview clustering, which is able to integrate multirelations among tweets. Even though there have been a few studies trying to fuse social tags and temporal information for topic detection, MVTD can incorporate various relations among tweets. It is scalable for many relations and thus may achieve higher performance.

The main contributions of this paper are summarized as follows: (a) a new topic detection framework (MVTD) using multiview clustering is proposed, which can fuse multirelation among tweets; (b) to measure semantic relations between tweets, this paper proposes a new document similarity measure based on suffix tree (STVSM) as well as methods for measuring the social tag relation and temporal relation among tweets; and (c) a novel method for extracting keywords from tweets based on suffix tree is proposed. In addition, a comparative study of the methods for fusing multirelation on real Twitter datasets is given.

The organization of the rest of this paper is as follows. In Section 2, the related work is discussed. In Section 3, an overview of MVTD is provided. In Section 4, the new methods of measuring the semantic relation, social tag relation and temporal relation are introduced. In Section 5, three methods for multirelation fusion are discussed. Section 6 introduces a new keyword extraction method. Section 7 describes the experiments results. Finally, Section 8 ends this paper with conclusions.

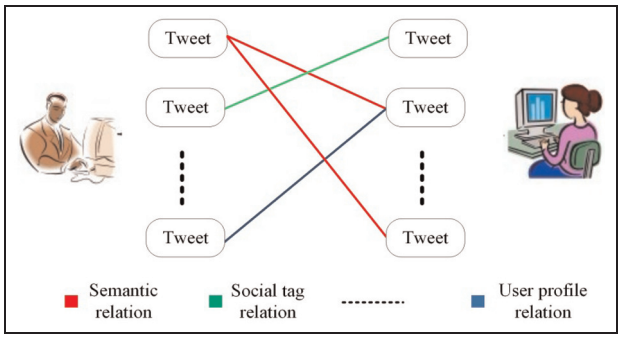

Figure 1. Multirelations in Twitter.

2. Related work

The previous studies on topic detection can be divided into two categories: traditional topic detection methods and new methods for detecting topics from Twitter.

2.1. Traditional topic detection methods

Papka and Allan proposed an on-line new event detection method using single pass clustering and a novel thresholding model that incorporates the properties of events as a major component [3]. Blei et al. [4] proposed an LDA model, which is a three-level hierarchical Bayesian model. It extends the generative model to achieve the capacity of generalizing the topic distributions such that the model can be used to generate unseen documents. Some work [6] extends LDA with consideration of temporal information using tensor factorization. Recently, Zhou et al. [5] proposed a method for detecting topics from professional blogs by constructing a network of co-occurrences of the keywords. By analyzing the structure of the word network, they first extracted candidate topics and then identified the final hot topics from the candidate topics.

These traditional methods, however, are mainly focused on traditional long texts such as news articles and blogs. However, the content of a tweet, that is, text, is usually limited owing to the limited number of characters. In addition, some information, such as images and meaningless symbols, may be useless or even noisy for topic detection. Therefore, the traditional methods may not perform well for tweets.

2.2. New methods for detecting topics from Twitter

The new proposed topic detection methods for Twitter often extend traditional algorithms, such as graph mining, classification and community discovery. Hurst et al. [7] proposed a new event detection algorithm by creating a keyword graph and using community detection methods (e.g. clustering) to discover and describe events. Sakaki et al. [8] devised a classifier of tweets based on features of Twitters, and then they produced a probabilistic spatiotemporal model for the target event. Popescu et al. [9] used the Gradient Boosted Decision Trees framework to detect topics. They used ML algorithms (e.g. regression and two-step decomposition) to address the task of identifying controversial events from Twitter [10]. Huang et al. [11] proposed detecting topics from Twitter by combining single-pass clustering and LDA. Ishikawa et al. [12–14] proposed a method for detecting hot topics in a local area during a particular period. In addition, many studies detect topics in Twitter from the perspective of stream data [15–18].

All these new methods for detecting topics from Twitter mainly focus on the text semantic information, despite some of them having considered other additional information (i.e. the spatiotemporal information). Therefore, the main difference between this work and previous work is that this paper treats tweets as entities with multirelations, and instead of only considering text semantic relations, it has considered all of the possible relations among tweets, such as semantic relations, social tag relations and temporal relations, to detect topics from Twitter. These relations are used to compensate for the weakness of the single relation, that is, the text semantic relation. This paper proposes a new framework (MVTD) to detect topics using multiview clustering, which can fuse multirelations.

3. Overview of MVTD framework

We first give the definition of topic detection from Twitter in Section 3.1. Then we discuss the framework of our approach in Section 3.2. We discuss the multirelations of Twitter in Section 3.3.

3.1. Problem definition

Given that a dataset D consists of n tweets, d_1, d_2, …, d_n. Each tweet is an original tweet posted by a Twitter user, and the words, tags, urls, user identifiers, time and location of posting are kept. The task of Twitter topic detection is to detect the top k topics from D which can closely reflect the real-life events, where k, the number of expected topics, is defined by the user. Each topic is represented by p keywords or phrases, where p is also predefined by the user.

3.2. Framework of MVTD

This paper regards tweets as entities that contain multirelations and proposes to detect topics by fusing multirelations. Specifically, we propose detecting hot topics from Twitter using multiview clustering (MVTD). The general steps of

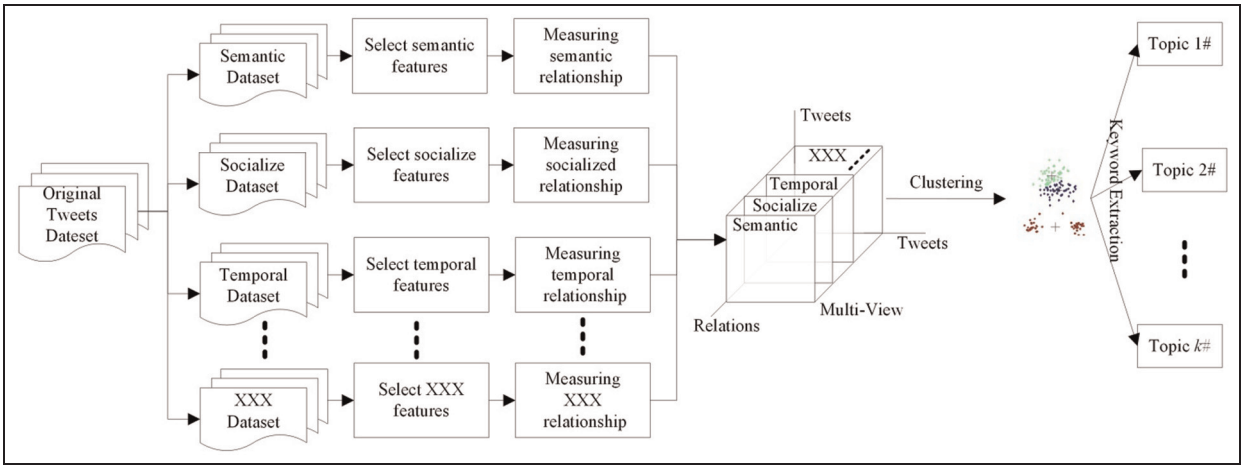

Figure 2. Framework of multiview topic detection method (MVTD).

MVTD are as follows: (a) measure multirelations among tweets; (b) cluster on multiview for tweets; and (c) extract representative keywords from clusters. The overall framework of MVTD is illustrated in Figure 2. Given a Twitter dataset, it is pre-processed into different feature spaces first according to the features selection criteria relying on different relations. Then the relations among tweets are measured into matrices. After measuring the relations among tweets, a series of matrices that represent the relations can be obtained. Each matrix can be treated as a view. Then a multiview over the dataset is formed by putting them together. All the clusters that represent different topics can be achieved using multiview clustering methods. Then the keyword extraction methods can be applied to extract the representative keywords as topics from clusters, and thus MVTD obtains all the detected topics.

3.3. Multirelations among tweets

For the task of topic detection, we summarize all the possible useful relations and give their definitions in this paper.

(1) *Semantic relation* – the 'semantic relation' is defined as the relation measured by text information, that is, meaningful words, in tweets. The text information is the most commonly used feature for conventional topic detection. We believe that it is also very useful for Twitter topic detection, although the maximum number of characters in a tweet is 140.

(2) *Social tag relation* – the 'social tag relation' is defined as the relation measured by the hashtags. A hashtag is usually in the form of '#tag', where 'tag' can be a combination of any characters, not necessarily a meaningful word. Social tags in tweets are often highly generalized descriptions of topics contained in that tweet. Let us take two tweets as examples, that is, 'I will receive lots of gifts during this festival #Christmas'and '#Christmas is the most welcomed festival for kids in U.S.A.' Even though there is no common meaningful word between them from the perspective of a semantic relation, these two tweets share a common tag, '#Christmas'. Therefore, they may have a high probability of belonging to the same topic, '#Christmas'. Thus the social tag relation is very useful for detecting topics from Twitter, and may compensate for the weakness of the semantic relation.

(3) *Temporal relation* – this is defined as the relation measured by the posting time of tweets. Most of the natural hot topics are generated with a particular lifecycle. Specially, after some event, such as a hurricane, related tweets are usually posted in a short span of time.

(4) *Other relations* – besides the above relations, there exist many other relations: (a) *retweet–reply relation* – if one tweet is the re-posting or reply of the other, they should belong into the same topic with a high probability; (b) *geographical relation* – most events appear in a particular region; and (c) *user profile relation* – if two users share a highly similar background, they may post tweets with a high similarity. Thus, the user profile relation is also useful for topic detection.

In our topic detection framework, we are able to fuse different relations in a collective manner. In comparison to traditional text-based topic detection methods, our method utilizes different types of relations and thus may achieve better topic detection results. For simplicity, in this paper, we only consider the first three relations for topic detection.

4. Measuring multirelations

4.1. Semantic relations

A suffix tree of string S is a compact trie tree containing all the suffixes of S. The suffix tree has been widely used for document clustering [19–21], web mining [22], bioinformatics [23], etc. Zamir et al. [19] proposed clustering documents in the same cluster if they share many phrases, which can be detected by a suffix tree. Chim and Deng [20, 21] propose a new document similarity computation method using a suffix tree. However, all these algorithms are aimed at traditional texts such as news documents. For texts in tweets, the number of words in each tweet is limited. If a suffix tree is built for a tweet dataset, the height of the tree must be very limited. As a result, the above algorithms may not be effective.

A phrase is often more meaningful than random combinations of single words. Given an example of the phrase 'apple company', this means an IT company established by Steve Jobs. However, if the phrase is split into two single words 'company' and 'apple', it is not easy to determine whether it means an IT company or not exactly. In addition, it is intuitive to observe that long phrases can convey more meaningful things than short ones. This is because, for long phrases, meanings are often more concrete and unambiguous than those of short phrases, so a long shared phrase between two tweets indicates their high semantic similarity. Thus, the similarity value between two tweets should be 'made' higher if they share long common phrases.

For example, let us consider three tweets: 'apple sale company', 'iphone apple company' and 'apple company excellent'. We assume that their ids are 1, 2 and 3. The suffix tree built by them is shown in Figure 3. All the nodes denoted by *a–i* in the suffix tree are drawn as circles. Each internal node is attached by a rectangle, which contains the identifiers of tweets that share common phrases. For instance, the node labelled *g* has a box that contains the ids of tweets 2 and 3, as they share a common phase, that is, 'apple company'.

Since words in tweets vary greatly, the number of nodes is very large. However, the height of the suffix tree is limited, because the numbers of characters in tweets are limited. After building the suffix tree using all the suffix strings, all the common phrases of each node can be obtained by traversing from the root node to all the leaf nodes in the tree. Therefore, all the common phrases between any pair of tweets can be obtained. The results are shown in Table 1.

Based on this observation, this paper extends the traditional vector space model (VSM) to a novel document similarity calculation method (STVSM) by assigning extra weights for words in phrases detected by suffix tree. The steps of STVSM are as follows: (a) delete special symbols such as @, RT, #tag, short URL etc. and the stopwords, and then extract the stems of words; (b) build a suffix tree using an $O(n)$ algorithm [24]; and (c) compute the pairwise tweet similarity by adding extra weights for words in phrases detected by suffix tree.

Suppose the size of the lexicon in D is m, which contains all the possible words. For two specific tweets d_1 and d_2, using VSM, two vectors, x_1 and x_2, can be formed. Each vector is used to represent the tweet in the m-dimensional space. The weight of the ith term is calculated by the TFIDF scheme.

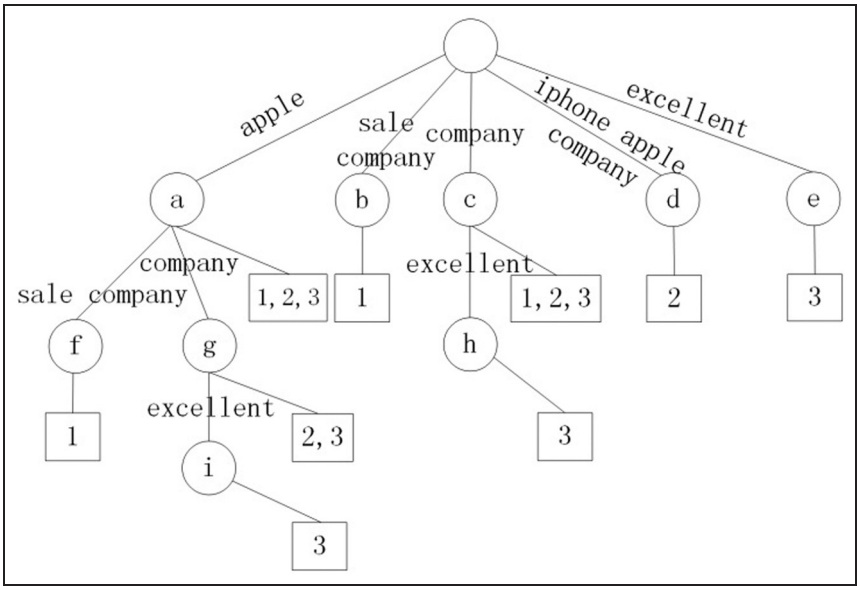

Figure 3. An example of a suffix tree.

Table 1. Common phrases of each pair of tweets.

Document id	1	2	3
1	—	apple; company	apple; company
2	apple; company	—	apple; company; apple company
3	apple; company	apple; company; apple company	—

For the similarity computation between two tweets, STVSM mainly focuses on the common words shared by them. So a set of words can be formed by collecting all the common words shared by the two tweets, and this set of common words can be denoted as $C = \{p_1, p_2, \cdots, p_s\}(1 \leq s \ll m)$. After forming the common words set, two new vectors x'_1 and x'_2 can be constructed using C. The sizes of them are represented by s. The ith value of each vector is assigned by the TFIDF scheme. Thus, the semantic similarity between d_1 and d_2 can be calculated by

$$semanticSim(d_1, d_2) = \cos(x_1, x_2) = \frac{x_1 \cdot x_2}{|x_1| \cdot |x_2|} = \frac{x'_1 \cdot x'_2}{|x_1| \cdot |x_2|} = \frac{\Delta}{\sqrt{\Delta_1 + M} \cdot \sqrt{\Delta_2 + N}} \tag{1}$$

$$\Delta = \sum_{i=1}^{s} x'_{1,i} \cdot x'_{2,i},\ \Delta_1 = \sum_{i=1}^{s} x^2_{1,i},\ \Delta_2 = \sum_{i=1}^{s} x^2_{2,i} \tag{2}$$

$$\begin{aligned} M &= \sum_{i=1}^{l_1} x^2_{1,i} (i - th\ word \notin C, i - th\ word \in C_1) \\ N &= \sum_{i=1}^{l_2} x^2_{2,i} (i - th\ word \notin C, i - th\ word \in C_2) \end{aligned} \tag{3}$$

where C_1 and C_2 represent the word sets of d_1 and d_2, respectively; l_1 and l_2 represent the sizes of C_1 and C_2, respectively; $x'_{1,i}$ and $x'_{2,i}$ represent the weighting values of the ith word of C in x'_1 and x'_2, respectively; and $x_{1,i}$ and $x_{2,i}$ represent the weighting values of ith word, which is not included in C but in C_1 and C_2, respectively. By using this formula, the speed of computing similarity is faster, because STVSM mainly focuses on the limited words appearing in two tweets, instead of two vectors whose dimensions are m.

In VSM, the order of all the words is overlooked after forming the vectors. This causes the loss of certain useful information. Since the suffix tree can detect the common phrases between any pair of tweets, STVSM measures their similarity by adding an extra weight to a word which belongs to an accumulating common phrase. Generally speaking, the more the number of words in the common phrase, the larger the weighting value that should be added to the word in it. The final tweet text similarity computation formula of STVSM is

$$sim(d_1, d_2) = \frac{\Delta^*}{\sqrt{\Delta_1^* + M} \cdot \sqrt{\Delta_2^* + N}} \tag{4}$$

$$\Delta^* = \sum_{i=1}^{s} (1 + w_i)^2 x_{1,i} \cdot x_{2,i},\ \Delta_1^* = \sum_{i=1}^{s} (1 + w_i)^2 x^2_{1,i},\ \Delta_2^* = \sum_{i=1}^{s} (1 + w_i)^2 x^2_{2,i} \tag{5}$$

where w_i denotes added-weight for the ith word in C and it can be calculated by using the weighting function $w(l)$; l denotes the maximum length of the accumulating common phrase shared by two tweets which contain the word. Equation (6) gives the detailed weighting function.

If $l = 1$, it means that no extra weight will be added for phrases that are single words. The empirical experiment in Section 7.2 demonstrates that the lengths of most of the common phrases are 2 or 3, so STVSM gives them weights α and $\alpha + \beta$, respectively.

$$w(l) = \begin{cases} 0 & l = 1 \\ \alpha & l = 2\ (\alpha > 0) \\ w(l-1) + \beta & l \geq 3\ (\beta > 0) \end{cases} \tag{6}$$

Table 2. Methods of computing term weight.

Document id	Description
TF	Use the term frequencies of tags to form vectors.
TFIDF	Use the term frequencies and inverse document frequencies of tags to form vectors.
HTF	Regard words which appear in tags as tags, and use the term frequencies of tags to form vectors.
HTFIDF	Regard words which appear in tags as tags, and use the term frequencies and inverse document frequencies of tags to form vectors.

4.2. Social tag relation

Social tagging has emerged as one of the best ways to associate metadata with web objects and it has been studied in a large body of literature [25]. Gupta et al. [25] made a survey on tag-related topics such as the motivation, modelling, application and recommendation of tags. Chua et al. [26] used social tags as detectors for detecting news events.

Social tags in a tweet are highly generalized description of topics contained in that tweet. Thus, we may obtain topical groups by clustering tweets based on their social tags. This paper also measures the social tag relation based on the VSM (Vector Space Model), in which tags are regarded as terms and each tweet is represented by a vector. Similarly, the values of terms in vectors can be computed by TF or TFIDF. Moreover, since the number of tags is often smaller than the number of words in a tweet, the feature space of tags is even sparser. With a careful observation of a large number of tweets, we believe that the tag feature space may be enhanced by regarding words appearing in tags, as tags. For example, there are two tweets which are 'I will receive lots of gifts during #Christmas' and 'Christmas is the most welcomed festival for kids'. The word 'Christmas' appears in the second tweet, and also appears in the tag '#Christmas' in the first tweet, so the word 'Christmas' can be regarded as a tag because they reflect the same topic. Therefore, this paper proposes to extend the traditional methods, that is, TF and TFIDF, for computing weights of terms, as HTF and HTFIDF, respectively. Table 2 summarizes four different methods to compute the weights of terms when forming vectors. Even though this may cause some noise by regarding words appearing in tags as tags, our further experiment results on a real dataset in Section 7.3 demonstrate that it is still very effective.

4.3. Temporal relation

In Twitter, the temporal information, that is, the timestamp for posting a tweet, is an essential feature of tweets. Plenty of studies have tried to exploit the temporal information for data mining tasks [7, 27, 28]. This paper also uses the temporal relations among tweets for topic detection. Instead of treating Twitter data as a data stream, it measures the temporal relation of each pair of tweets. A Gaussian kernel function (GKF) based similarity method is employed. For two specific tweets d_1 and d_2 their temporal similarity is defined as

$$temporalSim(d_1, d_2) = e^{-\frac{\|d_{1,t} - d_{2,t}\|^2}{\sigma^2}} \tag{7}$$

where $d_{1,t}$ and $d_{2,t}$ represent the time when these two tweets are posted; $d_{1,t} - d_{2,t}$ represents the number of hours between their posting time; σ is a predefined parameter.

4.4. Discussion of interactions of relations

Even though the semantic relation is the most commonly used feature for traditional topic detection, it may inefficient for Twitter owing to its sparse feature space. On the other hand, the social tags can be used to describe the topics contained in tweets, but users' annotations may be arbitrary. This implies that some tags may be meaningless, useless or even noisy. For the temporal relation, it is also useful because most of the natural topics appear in a specific range of time, but it is also inefficient for topic detection because many topics may appear in the same range of time.

From the above discussions, we can easily conclude that: (a) each of these relations is useful for topic detection; and (b) only the use of one of these relations, that is, only using the semantic relation, social tag relation or temporal relation, is still inefficient to detect topics from Twitter. This may also be true for other relations such as the retweet–reply relation and geographical relation. Therefore, instead of only considering single relations, we propose integrating different kinds

of relations to detect topics. By fusing these relations, we may compensate for the incomplete information in any single relation and reduce the noise caused by single relations, and thus we may achieve a more reliable topic detection result.

5. Fusion of multirelation

Given a set of tweets D, for each relation R_i, we can obtain a pairwise similarity matrix M_i according to its similarity measurement. We can fuse all the relations into a multiview $M = [M_1, M_2, \cdots, M_m]$ by putting these matrices together, where m is the number of relations. Then, we can cluster tweets based on the multiview M [29].

In this paper, we not only attempt to fuse multirelations which may compensate for the weakness of single views, but also attempt to exploit the correlation of multirelations for improving the clustering performance. To give a systematic comparison, we first fuse the multiview without considering the correlation of views, and then we fuse the multiview considering the correlation among views. For the former, we employ word-tag fusion [30] and SMC [31]; for the latter, we employ CMC [32]. The word-tag fusion can only fuse the semantic relation and social tag relation, while SMC and CMC are scalable for multiviews. The difference between SMC and CMC is that the former does not consider the correlation of multiviews, while the latter does.

5.1. Word-tag fusion

Based on the collections of both words and tags, it is straightforward to concatenate them together [30]. Suppose that the bag of words and the bag of tags of D are C_W and C_T respectively. Each tweet is split into two separate sets, which are the word set and tag set. Then two vectors, which are word vector, $V_W = (w_1, w_2, \cdots, w_u)$, and tag vector, $V_T = (t_1, t_2, \cdots, t_v)$, can be formed, where u and v denote the sizes of C_W and C_T, respectively. The weights of terms in vector can be calculated by TFIDF. By normalizing and giving them equal weights after concatenating them, the final vector can be obtained:

$$V = \left(0.5^{0.5} V_W, 0.5^{0.5} V_T\right) \tag{8}$$

After representing tweets as vectors, their similarities can be computed using cosine similarity and then a spectral clustering algorithm [33] can be applied to cluster the similarity matrix.

5.2. Stage-based multiview clustering (SMC)

SMC is a stage-based multiview clustering [31], which is based on a spectral clustering algorithm [33]. There are four stages in a typical spectral clustering algorithm [33] and these stages correspond to four matrices: the similarity matrix, the utility matrix, the feature matrix and the partition matrix. For multiview clustering, it is straightforward to integrate them directly during each stage. Once they have been integrated into one matrix in a particular stage, it can be regarded as a single view and then the remaining steps of spectral clustering will be performed on it. So there are four methods for integrating them: similarity integration, utility integration, feature integration and partition integration.

5.3. Co-training-based multiview clustering (CMC)

CMC was studied in the literature [32, 34]. Unlike integrating the matrices in different stages in SMC, this method is similar to the co-training process, which assumes that, if two tweets belong to the same topic in one view, they should belong to this topic in all the views. On the other hand, if two tweets belong to different topics, they should be so in all the views. Since the top l eigenvectors contain the discriminative information of tweets in one view, we can project the similarity matrix in one view along these directions in another view to retain the expected information for clustering and eliminate the confused information.

5.4. Discussion of different fusion methods

Here, we give a brief discussion of these three fusion methods. The word–tag fusion is the most commonly used fusion method for fusing words and tags, but it cannot be applied to multiview cases when there are some other relations which cannot be represented as term vectors.

SMC and CMC are based on the spectral clustering [33], and they are scalable for multiviews. SMC fuses matrices of different relations at the same stages directly, in a similar manner to word–tag fusion. However, when some relations are too sparse or the level of correlation between views is too low, its performance may degrade since SMC adopts

linear operation on views, while CMC may perform more robustly because it projects and back-projects from one view to another iteratively, which overcomes the weakness of SMC. Compared with SMC, CMC exploits the correlation of views and thus may achieve better clustering performance. In the following section, we describe comparative studies using the above fusion algorithms.

6. Keyword extraction

After clustering tweets on multiview, groups of tweets that represent different topics can be obtained. The next step of topic detection is to extract representative keywords from each cluster. The traditional keyword extraction methods [35–41] focus on extracting single words. Therefore, a topic is usually represented as a set of words.

However, owing to the existence of many new words, abbreviations and symbols in Twitter, a set of single words may not be easy for people to understand. So a better way to represent a topic is to use a set of phrases. Just as described in Section 4.1, suffix trees can be used to detect common phrases among tweets, which contain more semantic information than the random combinations of words in them. Therefore, this study proposes using phrases to represent topics, instead of only words.

The detailed steps of our keyword extraction method can be summarized as follows:

- Step 1 – input a group of tweets and conduct preprocessing.
- Step 2 – build a suffix tree for those inputted tweets.
- Step 3 – collect the common phrases by traversing the tree.
- Step 4 – compute the TFIDF value of each word, tag and phrase. For phrases, extra weights are added for them according to equation (6).
- Step 5 – reverse the order of words and phrases into a ranking list according to their TFIDF weights, and choose the top p terms as representative keywords of this topic (note that if a phrase has been ranked in the front, the suffix strings of this phrase are removed from the ranking list).

7. Experiments

7.1. Dataset

This study used the Twitter API (https://dev.twitter.com/) to crawl hot topics. Each topic contains around 1500 tweets. The topics and their corresponding tweets, which were distributed during 1–9 January 2012, were collected. These topics include festivals, movies and sports, which were the hot topics during that period.

First, all the data were preprocessed. Then, we randomly chose 12,000 tweets, consisting of 60 topics. Each topic contained 200 tweets, and each of them contained at least one social tag. Three sub-datasets were created: datasets 1–3. They contained 20, 40 and 60 topics, respectively. To measure the clustering performance, three measures were used: F-measure, NMI and Entropy. Higher F-measure and NMI values indicate better clustering performance, while higher Entropy values indicate lower clustering performance.

In the following experiments, we evaluated the performance of measurements on different relations, fusion methods and topic detection methods. Since the experimental results were very similar on all the datasets, to save the space and avoid redundancy, we use dataset 2 as the default dataset to show the various experimental results in details.

7.2. Semantic relation

According to the weighting function, equation (6), when computing the semantic similarity between two tweets, the weight of a word is defined according to the maximum length of all the common phrases that contain it. In order to assign the proper weight, the distribution of a number of common phrases with different lengths for Twitter datasets should be studied. Therefore, we investigated the lengths of common phrases.

First, a suffix tree was built using dataset 2. Then, by traversing the suffix tree, all the common phrases among tweets were collected. Finally, the numbers of phrases with different lengths were counted. Here only phrases whose lengths were bigger than 1 were considered. Figure 4 shows the number of phrases with different lengths. It can be observed that the lengths of most of the common phrases (70%) are 2 and 3. This suggests that most of the phrases contain only 2 or 3 words. Therefore, in the weighting function $w(l)$, this study mainly focused on weights for phrases whose lengths were 2 and 3, namely α and $\alpha + \beta$.

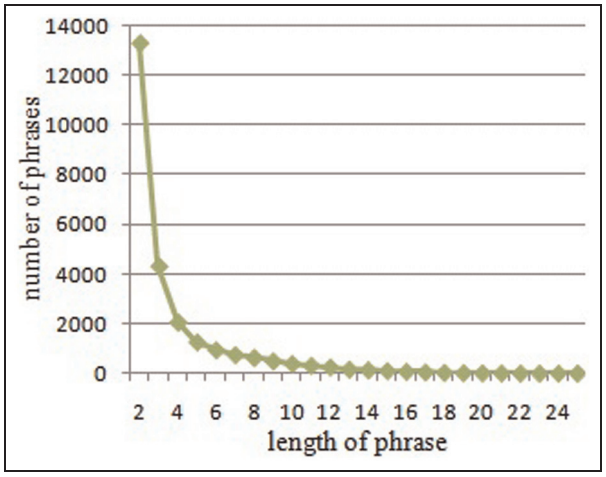

Figure 4. Number of phrases with different lengths in dataset 2.

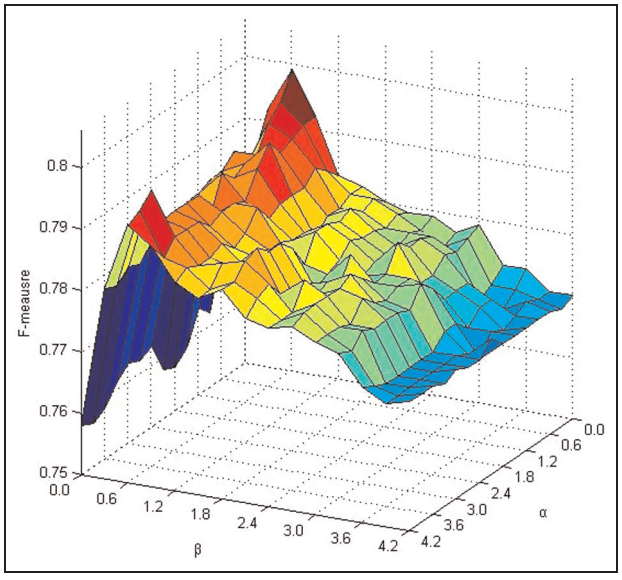

Figure 5. Parameter adjustment on dataset 2.

In order to achieve a promising clustering performance, an experiment was conducted for parameter adjustment on dataset 2. Let $\alpha \in [0.0, 4.2]$, $\beta \in [0.0, 4.2]$. The corresponding similarity matrix of semantic relation was computed by using STVSM. After clustering on the semantic matrix using a spectral clustering algorithm [33], the F-measure was used to evaluate its performance. Note that, to reduce the random noise, each experiment was repeated 30 times. The average F-measure value was used as the final result (note that the following clustering experiments also include this step).

Figure 5 illustrates the parameter adjustment experimental results. When α and β are 0, the F-measure value is 0.774. Note that, in this case, STVSM performs the same as VSM. It is clear that the leading parameter is α although there are two parameters. This implies that a large proportion of phrases have a length of 2. When α is in (0.0, 0.6], STVSM is able to consistently achieve better performance than VSM. This shows that the extra weights added to words in phrases can indeed improve performance. However, since not every common phrase is a real phrase in practice, for example, random combinations of words appearing in several tweets, the weighting function may magnify the effect of that noise, especially when the weight is large. When α is bigger than 3.0, the performance is slightly worse than that of the VSM model. The F-measure value of the best performance is 0.8 when $\alpha = 0.6$ and $\beta = 0.0$, and we use this set of parameters in the following experiments.

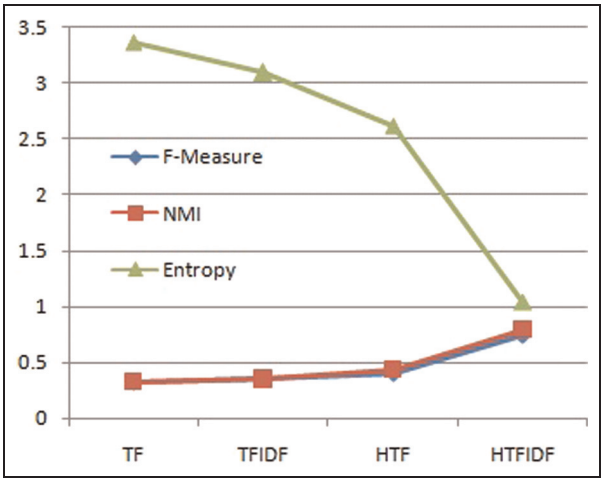

Figure 6. Comparison of methods to compute weighting on dataset 2.

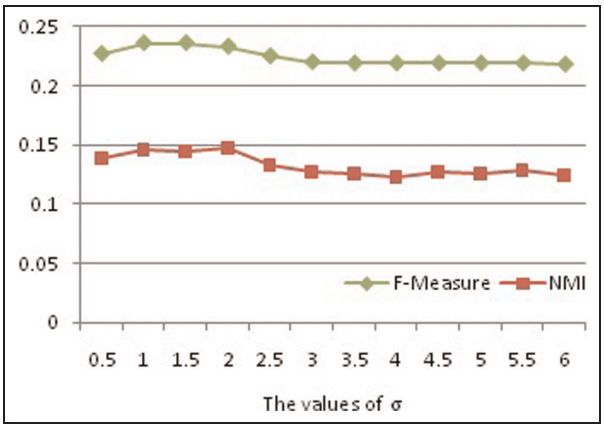

Figure 7. Clustering tweets in dataset 2 only using temporal relation.

7.3. Social tag relation

Section 4.2 proposes regarding words that appear in tags as tags. In this experiment, we compared the performance of the four methods in Table 2 on dataset 2. For each tweet, four vectors could be formed according to the methods shown in Table 2. After obtaining the similarity matrices of social relation, a spectral clustering algorithm [33] was applied to cluster them.

Figure 6 shows the experimental results. The performance of TFIDF is better than that of TF. The performance of the proposed HTF and HTFIDF is better than that of TF and TFIDF under all the three evaluation measures. The values of *F*-measure, NMI and Entropy of HTFIDF are 0.702, 0.759 and 1.308, respectively, while the values of *F*-measure, NMI and Entropy of TFIDF are 0.346, 0.346 and 3.089, respectively. Therefore, the performance of tweet clustering can be improved significantly by treating words that appear in tags as tags. This also indicates that words appearing in tags can deliver similar performance compared with using tags. Therefore, MVTD uses HTFIDF to measure the social tag relation in the following experiments.

7.4. Temporal relation

In this experiment, tweets were clustered using a spectral clustering algorithm [33] by only considering the temporal relation. The similarities between tweets were computed using equation (7) with a parameter σ. The experimental result for dataset 2 is shown in Figure 7. The overall values of *F*-measure and NMI (the average values of *F*-measure and NMI are

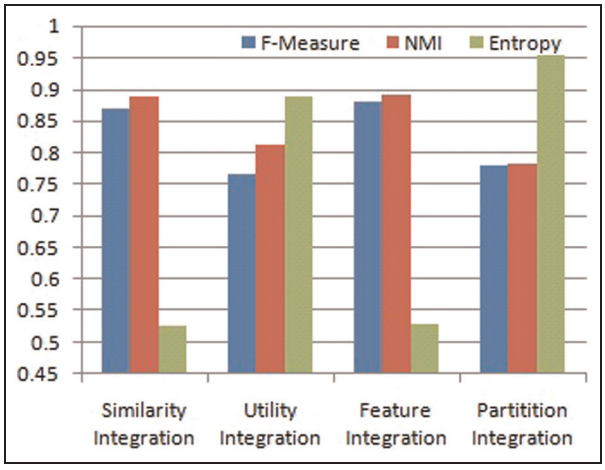

Figure 8. Performance comparison of four integration methods on dataset 2.

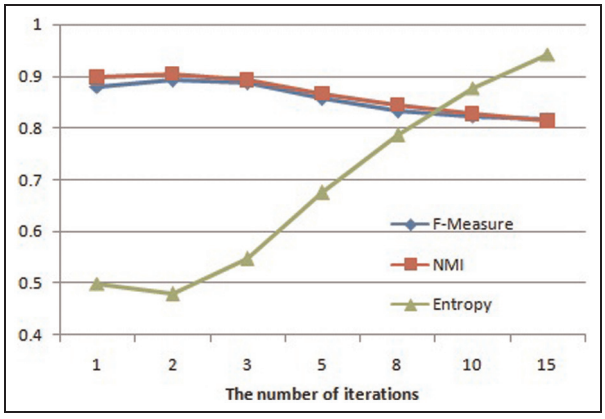

Figure 9. Co-training based multiview clustering performance with the variation of the number of iterations on dataset 2.

0.224 and 0.133) are very low compared with those of semantic and social tag relations. Compared with the original dataset, the time ranges of many topics appear overlapped, so compared with semantic relation and social tag relation, the performance is very low when clustering using only the temporal relation, but it may still provide some information for distinguishing topics. Also, when $\sigma = 1.0$, the highest values of F-measure and NMI are obtained, so σ is set as 1.0 in the following experiments.

7.5. Multirelation fusion

7.5.1. SMC fusion. To compare the performances of the four integration methods of SMC on the Twitter dataset, an overall view of the performance comparison of four integration methods on dataset 2 is given in Figure 8. The performances of similarity integration and feature integration are better than those of utility integration and partition integration. In practice, if some views are too sparse or contain too much noise, feature integration may be more robust than similarity integration, because the feature integration has higher de-noising capability. Therefore, feature integration is used in the following experiments.

7.5.2. CMC fusion. The CMC method has to project and back-project from one view to another view iteratively. Here an empirical study of the number of iterations required for multiview in Twitter dataset is given. For simplicity, a multiview using the semantic relation and social tag relation is formed first, and then it is clustered using CMC fusion. Figure 9 depicts the co-training based multiview clustering performance with the variation of the number of iterations under three evaluation methods on dataset 2. As can be seen, the best performance (F-measure, 0.892; NMI, 0.904; Entropy, 0.479)

Table 3. Clustering performance comparison on dataset 1.

Number of views	View	Algorithm	*F*-Measure	NMI	Entropy
1	SM	VSM	0.783	0.766	0.953
1	SM	STVSM	0.793	0.781	0.824
1	SO	HTFIDF	0.808	0.797	0.819
1	TM	GKF	0.222	0.269	3.113
2	SM + SO	Word-tag	0.832	0.826	0.704
2	SM + SO	SMC	0.899	0.869	0.540
2	SM + SO	CMC	0.903	0.886	0.451
2	SM + TM	SMC	0.813	0.759	1.025
2	SM + TM	CMC	0.767	0.769	0.962
2	SO + TM	SMC	0.776	0.721	1.171
2	SO + TM	CMC	0.811	0.782	0.923
3	SM + SO + TM	SMC	0.917	0.907	0.385
3	SM + SO + TM	CMC	0.946	0.953	0.267

Table 4. Clustering performance comparison on dataset 2.

Number of views	View	Algorithm	*F*-Measure	NMI	Entropy
1	SM	VSM	0.774	0.792	1.052
1	SM	STVSM	0.799	0.812	0.926
1	SO	HTFIDF	0.743	0.790	1.041
1	TM	GKF	0.236	0.146	3.984
2	SM + SO	Word-tag	0.842	0.859	0.716
2	SM + SO	SMC	0.868	0.889	0.539
2	SM + SO	CMC	0.881	0.891	0.479
2	SM + TM	SMC	0.848	0.865	0.668
2	SM + TM	CMC	0.853	0.872	0.652
2	SO + TM	SMC	0.716	0.761	1.196
2	SO + TM	CMC	0.796	0.802	1.032
3	SM + SO + TM	SMC	0.889	0.915	0.412
3	SM + SO + TM	CMC	0.926	0.935	0.316

is achieved when the number of iterations is set at 2. Since the optimal parameter with respect to the number of iterations is very small, the speed of CMC is fast.

7.6. Fusion comparison

The clustering performances of two-view clustering case and three-view clustering case are compared in this section. Tables 3–5 depict the clustering performance on datasets 1–3. The performance of single-view clustering is also listed in the tables. SM, SO and TM represent semantic relation, social tag relation and temporal relation, respectively. VSM, STVSM, HTFIDF and GKF are the algorithms for single view clustering, while Word-tag, SMC and CMC are the algorithms for fusing multirelation.

In the two-view clustering case, the performance of SM + SO is the best among all the combinations of two views, and SM + TM is always better than SO + TM. This indicates that the semantic relation is still the most important relation among those three relations for clustering tweets. The performance of word-tag is also better than that of any single-view clustering. In addition, it is observed that, by fusing SO or TM with SM, its performance is always better than that of only using SM. This is also the same with SO. The performance of the fusion of SO and TM is better than that of TM, but less well than SO. This is mainly because the temporal relation is a very weak relation for clustering tweets, as demonstrated in Section 7.4, but it is still a useful relation for clustering tweets, which can be seen from the fusion of SM + TM.

In the three-view clustering case, the clustering performance of SM + SO + TM is consistently better than that of any single or two views in the three datasets. Given an example of the clustering performance of SM + SO + TM on dataset 1, the best clustering performance can be achieved using CMC, that is, the values of *F*-measure, NMI and Entropy are 0.946, 0.953 and 0.267, while the values of *F*-measure, NMI and Entropy of the best performance in the

Table 5. Clustering performance comparison on dataset 3.

Number of views	View	Algorithm	*F*-Measure	NMI	Entropy
1	SM	VSM	0.788	0.818	1.018
1	SM	STVSM	0.800	0.832	0.827
1	SO	HTFIDF	0.702	0.759	1.308
1	TM	GKF	0.096	0.201	4.452
2	SM + SO	Word-tag	0.816	0.847	0.856
2	SM + SO	SMC	0.875	0.891	0.617
2	SM + SO	CMC	0.892	0.904	0.479
2	SM + TM	SMC	0.851	0.871	0.694
2	SM + TM	CMC	0.860	0.879	0.666
2	SO + TM	SMC	0.766	0.777	1.236
2	SO + TM	CMC	0.784	0.798	1.130
3	SM + SO + TM	SMC	0.896	0.920	0.416
3	SM + SO + TM	CMC	0.928	0.936	0.342

Table 6. Topic detection comparison on dataset 2.

Topic	Algorithm	Representration
Topic 1: 'Christian Watford'	Single-pass	watford; christian; dai; indiana; michigan; zeller; point; 25;
Topic 1: 'Christian Watford'	LDA	watford; christian; point; michigan; hand; half; ap; 25;
Topic 1: 'Christian Watford'	MVTD	christian watford; ichigan;indiana; 25 point; kentucki;christian watford plai; christian watford buzzer; lucki shot;
Topic 2: 'Beauty and the beast'	Single-pass	beauti; beast; 3d; theater; funtim; check; wathc; lol;
Topic 2: 'Beauty and the beast'	LDA	beast; beauti; 3d; watch; feel;bad; movi; hernandez;
Topic 2: 'Beauty and the beast'	MVTD	beauti beast; beauti beast 3d; watch beauti beast; beauti beast check; watch beauti beast check; theater; beauti beast come; beast theater;

single-clustering view case are 0.808, 0.797 and 0.819, and the values of *F*-measure, NMI and Entropy of the best performance in the two-view clustering case are 0.903, 0.886 and 0.451. Thus, the *F*-measure value of three-view clustering is 13.8% better than that of the single-view clustering and 4.3% better than that of any two-view clustering. Therefore, these experiments demonstrate that fusing multiviews by clustering on multiviews indeed can compensate for the weakness of single views, and thus it achieves better clustering performance than any single view.

7.7. Topic detection comparison

To demonstrate the topic detection performance of MVTD, we compared it with other commonly used topic detection methods, that is single-pass and LDA-based methods. Here, the number of representative keywords is set as p=8. The single-pass based method clusters the input dataset using single-pass clustering. The threshold is set as $\sigma = 0.01$ for the partition of clusters. The LDA-based method generates the LDA model using the input dataset with parameters $k = 40$, $\alpha = 0.2$, $\beta = 0.1$ and number of iterations = 1000. MVTD uses the CMC algorithm to fuse three views on dataset 2.

Here, for simplicity, we only show the topic detection results of two topics, 'Christian Watford' and 'Beauty and the Beast', using three different methods (Table 6). MVTD- and LDA-based methods can outperform single-pass based method, because their extracted keywords, such as 'chirstian' and 'watford', are highly related to the topics, while some keywords extracted by the single-pass-based method are unrelated to the topic. For example, the word 'dai' extracted by single-pass based method has little relationship to topic 1.

In addition, since this paper proposes to use a suffix tree to extract keywords from a given cluster of tweets, the extracted keywords are a set of phrases, in which the order of words is well kept, so the extracted keywords of MVTD are more meaningful than the sets of single words extracted by single-pass and LDA.

8. Conclusion

In this paper, we present a new framework (MVTD) for detecting hot topics from Twitter using multiview clustering. We first conducted a careful observation of hot topics and found that hot topics are usually represented by phrases, tags, etc.,

and each topic often appears in a particular range of time and region. Based on these features of hot topics, we believe that tweets can be regarded as entities with multirelations, such as semantic relation, social tag relation and temporal relation. Hence, we can detect hot topics based on these relations.

In this study, we mainly used the above three relations for topic detection, but our framework can be scalable for many other relations. To measure the relations of each pair of tweets, we propose methods to compute their similarity values. In particular, to better measure the semantic similarity of tweets, we propose a new document similarity measure based on a suffix tree (STVSM), which exploits the extract meaning carried by phrases. We also propose regarding words appearing in tags as tags too and this can enhance the tag feature space. To cluster tweets, we employ three multiview-based methods, which are word-tag, SMC and CMC. In addition, we propose a new keyword extraction method based on a suffix tree. The extensive experimental results on three real Twitter datasets also demonstrate that the performance of multiview clustering is better than any single-view clustering, and thus MVTD achieves better performance in detecting topics from Twitter.

In the near future, we plan to extend our work in two directions. First, it will be interesting to study how to measure other relations such as the retweet–reply and geographical relations, which may improve multiview clustering performance and results in better topic detection results. Second, we will perform our approach on some more real datasets such as posts from Facebook, Sina Weibo, etc.

Funding

This work was supported in part by the National Nature Science Foundation of China under grant no. 61300209, the Shenzhen Foundation Research Fund under grant no. JCY20120613115205826 and the Shenzhen Technology Innovation Program under grant no. CXZZ20130319100919673.

References

[1] Kwak H, Lee C, Park H and Moon S. What is Twitter, a social network or a news media? In *Proceedings of the 19th international World Wide Web conference*, 2010.
[2] Wu S and Homan JM. Who says what to whom on Twitter. In: *Proceedings of the 20th international World Wide Web conference*, 2011.
[3] Papka R and Allan J. On-Line new event detection using Single Pass Clustering. Technical report no. TR98-21, UMass Computer Science, 1998.
[4] Blei DM, Ng AY and Jordan MI. Latent Dirichlet allocation. *Journal of Machine Learning Research* 2003; 3: 993–1022.
[5] Zhou E, Zhong N and Li Y. Hot Topic detection in professional blogs. In: *Proceedings of the 7th international active media technology conference*, 2011, pp. 141–152.
[6] Guo X, Xiang Y, Chen Q, Huang Z and Hao Y. LDA-based online topic detection using tensor factorization. *Journal of Information Science* 2013; 39(4): 459–469.
[7] Sayyadi H, Hurst M and Manykov A. Event detection and tracking in social streams. In: *Proceedings of the international conference on weblogs and social media*. AAAI, 2009.
[8] Sakaki T, Okazaki M and Matsuo Y. Earthquake shake Twitter users: Real-time event detection by social sensors. In: *Proceedings of the 19th international World Wide Web conference*, 2010.
[9] Popescu A and Pennacchiotti M. Detecting controversial events from Twitter. In: *Proceedings of conference on information and knowledge management*, 2010.
[10] Popescu A and Pennacchiotti M. Extracting events and event descriptions from Twitter. In: *Proceedings of the 20th international World Wide Web conference* (Companion Volume), 2011, pp. 105–106.
[11] Huang B, Yang Y, Mahmood A and Wang HJ. Microblog topic detection based on LDA model and single-pass clustering. In: *Proceedings of international conference on rough sets and current trends in computing*, 2012, pp. 166–171.
[12] Ishikawa S, Arakawa Y, Tagashira S and Fukuda A. Hot topic detection in local areas using Twitter and Wikipedia. In: *International conference on architecture of computing systems workshops*, 2012, pp. 165–174.
[13] Cataldi M, Caro LD and Schifanella C. Emerging topic detection on Twitter based on temporal and social terms evaluation. In: *Proceedings of the 10th international workshop on multimedia data mining*, 2010.
[14] Yuan Q, Cong C, Ma Z, Sun A and Magnenat-Thalmann N. Who, where, when and what: Discover spatio-temporal topics for Twitter users. In: *Proceedings of international conference on knowledge discovery and data mining*, 2013.
[15] Guo J, Zhang P, Tan J and Guo L. Mining hot topics from Twitter streams. In: *Proceedings of international conference on computational science*, 2012, pp. 2008–2011.
[16] Mathioudakis M and Koudas N. TwitterMonitor: Trend detection over the twitter stream. In: *Proceedings of international conference on management of data*, 2010, pp. 1155–1158.
[17] Peterovic S, Osborne M and Lavrenko V. Streaming first story detection with application to Twitter. In: *Proceedings of 2010 annual conference of the North American Chapter of the Association for Computational Linguistics*, 2010, pp. 181–189.

[18] Benhardus J and Kalita J. Stream trend detection in Twitter. *International Journal of Web Based Communities* 2013; 9(1): 122–139.
[19] Zamir O and Etzinoi O. Web document clustering: A feasibility demonstration. In: *Proceedings of international conference on information retrieval*, University of Washington, Seattle, WA, 1998.
[20] Chim H and Deng X. A new suffix tree similarity measure for document clustering. In: *Proceedings of the 16th international World Wide Web conference*, 2007.
[21] Chim H and Deng X. Efficient Phrase-based document similarity for clustering. *IEEE Transactions on Knowledge and Data Engineering* 2008; 20: 1217–1229.
[22] Xie X, Fang Y, Zhang Z and Li L. Extracting data records from Web using suffix tree. In: *Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics*, 2012.
[23] Nicolas J, Durand P, Ranchy G, Tempel S and Valin A. Suffix-tree analyser (STAN): Looking for nucleotidic and peptidic patterns in chromosomes. *Bioinformatics* 2005; 21(24): 4408–4410.
[24] Ukkonen E. On-line construction of suffix trees. *Algorithmica* 1995; 14: 249–260.
[25] Gupta M, Li R, Yin Z and Han J. Survey on social tagging techniques. In: *Proceedings of international conference on knowledge discovery and data mining explorations*, 2010, Vol. 12(1), pp. 58–72.
[26] Chua A, Razikin K and Goh D. Social tags as news event detectors. *Journal of Information Science* 2010; 37(1): 3–18.
[27] Wang X, Zhu F and Jiang J. Real time event detection in Twitter. In: *Proceedings of international conference on Web-age information management*, 2013, pp. 502–513.
[28] Milajevs D and Bouma G. Real time discussion retrieval from twitter. In: *Proceedings of the 22th international World Wide Web conference*, 2013, pp. 795–800.
[29] Bickel S and Scheffer T. Multi-view clustering. In: *Proceedings of international conference on data mining*, 2004, pp. 19–26.
[30] Ramage D, Heymann P, Manning DC and Molina HG. Clustering the tagged web. In: *Proceedings of International ACM conference on web search and data mining*, 2009, pp. 54–63.
[31] Tang L, Wang X and Liu H. Community detection in multi-dimensional networks, Technical report, *School of Computing, Informatics, and Decision System Engineering, Arizona State University*, 2010.
[32] Kumar A and Daum III. A co-training approach for multi-view spectral clustering. In: *Proceedings of international conference on machine learning*, 2011, pp. 393–400.
[33] Ng A, Jordan M and Weiss Y. On spectral clustering: Analysis and an algorithm. In: *Proceedings of advances in neural information processing systems*, 2001, pp. 849–856.
[34] Kumar A, Rai P and Daum III. Co-regularized multi-view spectral clustering. In: *Proceedings of advances in neural information processing systems*, 2011, pp. 1413–1421.
[35] Frank E, Paynter GW, Witten IH, Gutwin C and Nevill-Manning CG. Domain-specific keyphrase extraction. In: *Proceedings of the 16th international joint conference on artificial intelligence*, 1999, pp. 668–673.
[36] Turney PD. Learning algorithms for keyphrase extraction. *Information Retrieval* 2000; 2(4): 303–336.
[37] Mihalcea R and Tarau P. TextRank: Bringing order into texts. In: *Conference on empirical methods in natural language processing*, Vol. 4. Barcelona: ACL, 2004, pp. 404–411.
[38] Wartena C, Brussee R and Slakhorst W. Keyword extraction using word co-occurrence. In: *Proceedings of international conference on database and expert systems applications workshops*, 2010, pp. 54–58.
[39] Timonen M, Toivanen T, Teng Y, Chen C and He L. Informativeness based keyword extraction from short documents. In: *Proceedings of international joint conference on knowledge discovery, knowledge engineering and knowledge management*, 2012, pp. 411–421.
[40] Li Z, Zhou D, Juan YF and Han J. Keyword extraction for social snippets. In: *Proceedings of the 19th international World Wide Web conference*, 2010, pp. 1143–1144.
[41] Park J, Kim J and Lee J. Keyword extraction for blogs based on content richness. *Journal of Information Science* 2013; 40(1): 38–49.