Expert Systems With Applications 192 (2022) 116405

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

TT-graph: A new model for building social network graphs from texts with time series

Wei Jia [a], Ruizhe Ma [b,*,1], Li Yan [a,2], Weinan Niu [a], Zongmin Ma [a,c,3]

[a] College of Computer Science & Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China
[b] Department of Computer Science, University of Massachusetts Lowell, Lowell, MA 01854, USA
[c] Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing 210000, China

ARTICLE INFO

Keywords:
Social network graph
Text semantics
Time series
User similarity

ABSTRACT

Social network analysis is a fundamental problem inherent in various applications, which can be handled mainly based on a graph model given in advance. However, it is generally ignored in most existing studies that the social networks may only contain users and no users' connections are explicitly available. Then the connections between users play a considerable role in building the graph, and it is necessary to calculate users' similarities. Traditional methods of user similarity calculation are extensively based on the topics of text content because they can effectively reflect users' interests. Nevertheless, these methods mostly ignore the importance of time series in the texts, where the texts with time series can reveal the activity trends of users in social networks. In this work, we explore a new problem of building a social network graph over text with time series. Our basic idea is that social media users are more similar if they have similar text semantics in similar time sequences. To obtain the semantics of text with time series, we extract topic words of each user from the corresponding text with our proposed Time-Biterm Topic Model (*T*-BTM), which improves the BTM model by taking the time-topic distribution into account. On this basis, we further propose a novel time series-based graph model with text, called Text with Time series for Graph (TT-Graph) model, which explicitly considers the user similarity and time series similarity. With the TT-Graph model, we propose novel methods for topic detection, community detection, and link prediction in social network analysis. Extensive experiments demonstrate that topic detection, community detection and link prediction can be effectively conducted on the TT-Graph model, and the credibility of our model can be proved.

1. Introduction

With the evolution of social networks platforms, social network analysis is receiving increasing attention. Many efforts of social network analysis have been carried out, such as *influence maximization* (Tang, Tang, & Yuan, 2017; Jung, Heo, & Chen, 2013; Chu, Zhao, & Liu, 2014; Goyal, Lu, & Lakshmanan, 2011; Bozorgi, Haghighi, & Zahedi, 2016; Shen, Mao, & Chung, 2020), *information diffusion model* (Krishnamurthy & Hoiles, 2016; Wang, Rho, & Chen, 2017; Li, Zhang, & Xu, 2016), *community detection* (Yan, Fanrong, & Yong, 2014; Zhang, Ren, Song, Jia, & Zhang, 2017; Han, Chen, & Yang, 2019; Cheng, Ma, & Chen, 2020; Zhu, Pan, & Wang, 2020) and so on. Social network analysis can reveal the internal structure of a network as well as the process of information propagation. It is essential to many social activities such as viral market, recommendation systems, and so on. We take community detection as an example, which is a traditional problem in social network analysis. Generally, the users with the same community frequently contact each other, and the users who belong to different communities are less connected and have fewer common languages or topics. This phenomenon has significant implications for social marketing. People with common interests are more likely to belong to the same community (e.g., football community, music community, and scientific community). Suppose that a music company would hold a concert. The organizer should primarily promote the concerts in the

* Corresponding author.
E-mail addresses: ruizhe_ma@uml.edu (R. Ma), yanli@nuaa.edu.cn (L. Yan).
[1] ORCID: 0000-0003-2749-3063.
[2] ORCID: 0000-0002-1881-3128.
[3] ORCID: 0000-0001-7780-6473.

https://doi.org/10.1016/j.eswa.2021.116405
Received 12 March 2021; Received in revised form 4 August 2021; Accepted 11 December 2021
Available online 17 December 2021

music community rather than the football community or the science community.

According to the relationships in social networks, we view social network analysis from two scenarios. The first one is that the relationships between users are known in advance, such as author cooperation network, paper citation network, circle of friends, and so on. The second one is that users are known in advance, but relationships between users may be unknown. The users in Sina Weibo, for example, may not have following relationships but only have similar topics or interests. Currently, most efforts dealing with social network analysis mainly rely on the first scenario of social networks. That is, the relationships between users are given in advance. In the context of influence maximization, for example, a social network graph is widely applied to find the influential users (Tang et al., 2017; Jung et al., 2013; Chu et al., 2014; Goyal et al., 2011; Bozorgi et al., 2016; Shen et al., 2020).

Social networks can also be categorized into social networks with text information and social networks without text information (Cheng et al., 2020). In the early stage of social network analysis, research works mainly focus on social networks without text information, which chiefly relies on network topological structure. Due to the enormous text information produced by social network platforms, increasing efforts are devoted to social network analysis by explicitly considering the text information (Sun, Jia, & Huang, 2020; Aslay, Barbieri, & Bonchi, 2014; Škrlj, Kralj, & Lavrač, 2019; Liu & Guo, 2020; Cheng, Yan, & Lan, 2014; Jing & Liu, 2019). These methods can effectively improve the accuracy and reliability of social network analysis.

It is shown from the above discussion that, for the second scenario of social network analysis, it is significant to propose a model to build a social network graph by text information. To deal with this above problem, some methods have been proposed from different perspectives, such as *position-based method* (Zhong et al., 2020; Zheng, Xie, & Ma, 2010; Yin, Cui, & Zhou, 2016; Wang, 2015), *co-occurrence-based method* (Adamic & Adar 2003; Arif, Asger, & Malik, 2015; Smith, Shneiderman, & Milic-Frayling, 2009), *text-based method* (Eichinger et al., 2019; Han, Wang, & Farahbakhsh, 2016; Makrehchi, 2011; Dusan & Mária, 2011; Wu, Zhang, Shen, Huang, & Gu, 2020; Spaeth & Desmarais, 2013; Han, Li, & Li, 2018) and so on. Note that, in social networks, hot events and opinions are not always static and actually change over time. Such changes reflect the trend of events or the variation of users, which fits the dynamic of a real social network (Zhang, Liu, & Du, 2011). At this point, time series can play a significant role in social network analysis. So, combining text information and time series can better explain the relationships between users, which is very useful in building a social network. The traditional social network analysis often ignores the context semantics information. Some efforts in the semantics network are based on the Latent Dirichlet Allocation (LDA) model (Blei et al., 2003), which can effectively intensify the accuracy. Nevertheless, text information from some social platforms such as Sina Weibo is generally short and sparse, and such texts cannot be handled by the LDA model because its sterling performance only exits for long text. To make up for the weakness of LDA, Pan, Yin, and Liu (2014) proposed the BTM model to effectively mine the topics, which relies on the biterm rather than the document.

Nowadays, a large volume of information has been produced on social network platforms. Because users would like to communicate or present their opinions through social network media, it is essential to discover the relationships between users. In this paper, we consider two categories of information in social networks: text semantic information and time series information in the text. The semantic information can exhibit the latent significance or the latent interest of users. The time series information narrates the external dynamics of users, which can represent users' activities. We aim to propose a new model to build a social network without any prior knowledge. For this purpose, we first extract topic words from texts. It is difficult to mine user's interests or labels of users from long text. So, we apply the topic words to characterize the users. We then calculate user similarity. We argue that the time series in texts can reveal a user's activity and is very significant in calculating user similarity. In this paper, we combine the text semantic similarity and time series similarity to calculate user similarity to improve the accuracy. Finally, based on user similarity (including the text semantic similarity and the time series similarity), we build the social network. In the paper, we evaluate the credibility of the TT-Graph model under three different scenarios: *topic detection*, *community detection,* and *link prediction*. The main contributions of this paper are summarized as follows.

- The traditional topic models neglect the situation of data with time series. Based on the BTM model, we propose a *T*-BTM model by considering time-topic distribution, which can effectively discover the topic words from the text with time series.
- Based on the text with time series, we propose a novel model of building a social network graph TT-Graph. To the best of our knowledge, this is the first work that combines text semantics and time series to build a social network without any prior knowledge.
- Based on the TT-Graph, we propose a novel community detection algorithm, TTSLPA. Not only does it consider the text semantics and time series of each node, but it also takes the propagation order of nodes into account.
- The experimental results show that our model can achieve excellent performance in topic detection, community detection, and link prediction.

The remainder of this paper is organized as follows. In Section 2, we introduce the related work about user similarity calculation and applications of time series in social network analysis. In Section 3, we present the preliminaries. We describe our proposed method in detail in Section 4. In Section 5, we evaluate the performance of our proposed TT-Graph model in the aspect of topic detection, community detection, and link prediction. We conclude this paper in Section 6.

2. Related work

In this section, we provide a concise retrospect of the related works in two main areas: user similarity calculation and applications of time series in social network analysis.

2.1. User similarity calculation

We identify users' similarity calculation from three aspects: position-based, co-occurrence items-based, and text-based.

2.1.1. Position-based user similarity

The geographical position information regularly indicates individual preferences and can be used to extract the similarity of users. Numerous researchers focus on location-related information such as shopping, traveling, location type, and so on. Zhong, Lyu, and Zhang (2020) proposed that people who share more similar common locations are more similar. To investigate the mobile patterns of users, they put forward a multi-centra-based similarity method and preference spot approach to calculate the final user similarity. Generally, historical traveling tracks of users can excavate user's similarities, which plays a significant role in the travel location recommendation. Zheng et al. (2010) build three different graphs to infer user similarity: a location-location graph, which shows the relationships for the locations that are frequently traversed in a trip; a user-location graph, which indicates the traversed location with travel times as weight; the hierarchical graphs, which are applied to infer the user-user relationships based on the above two graphs.

Data sparsity is an extreme challenge in the recommendation problem. To deal with this, Yin et al. (2016) propose a probabilistic generative model, called Topic-Region Model (TRM). This model combines semantic, temporal, and spatial information to calculate user similarity. Considering that the interest of users may influence the locations, they

present a topic-region factor to indicate similar semantic locations.

2.1.2. Co-occurrence items-based user similarity

It is commonly believed that the users with more co-occurrence items (i.e., browse webs, purchase, and so on) should be more similar. Based on this idea, Adamic and Adar (2003) calculated user similarity from four aspects: In-links, Out-links, Mailing lists, and Text. They primarily concentrated on the user's homepage. For the users who link to each other, the co-occurrence of text in their homepages typically declares a common interest, such as history classes.

Based on the co-authorship, Arif et al. (2015) build an academic collaborations network. To obtain the co-authorship relationships, they collect and investigate academic social networks among participants such as conferences, technical reports, and so on). Moreover, since it is possible that some author names look very similar but are essentially different, they propose an author name disambiguation method based on the work in (Arif, Ali, & Asger, 2014). They utilize NodeXL (Smith et al., 2009) to obtain the co-authorship network in mining social relationships of conference participants.

2.1.3. Text-based user similarity

In social networks, users generally produce enormous text information. Many efforts are devoted to extracting user interest from the text information produced by the user. The user similarity is generally calculated with the topic model. In Makrehchi (2011), users' topics are first extracted with the LDA (Latent Dirichlet Allocation) model, then a semi-bipartite graph is built by combining the node-topic and topic-topic, and finally, the Katz and short path scores are utilized to build a new network. Similar to the work in Makrehchi (2011), Dusan and Mária (2011) calculate user similarity with three stages: text selection, latent feature pre-computation, and final similarity calculation. They utilize a similarity tree to represent the similarity of articles, which can effectively discover the user's interest.

Wu et al. (2020) combine the Glove and BTM to discover the topics on Microblog. They use the JS divergence to evaluate the text similarity. Integrating text similarity and tags similarity, Spaeth and Desmarais (2013) utilize the LSA (Latent Semantic Analysis) model to analyze use similarity. Since most of the existing works ignore the APP usage periodicity, Han et al. (2018) propose a dynamic fusion framework. Based on user similarity calculated by collecting the user's App installation lists, their method can provide users with appropriate recommendations for the App.

2.2. Time series and applications in social network analysis

We present a brief overview of the related work from the following two aspects: time series similarity method and time series applications in social network analysis.

2.2.1. Time series similarity

Triangle similarity is a simple and effective method in processing time series similarity (Faloutsos, Ranganathan, & Manolopoulos 1994). This method combines the triangle similarity and machine learning method to cluster time series. In the triangle similarity, each time series is treated as t-dimensional space. And the similarity of the two time series is from -1 to 1. Moreover, the triangle similarity method can effectively address noise, amplitude scaling, offset translation, and linear drift.

Note that the triangle similarity measure faces a problem: it cannot deal with two time series with different lengths. In order to make up for the shortcoming of triangle similarity, dynamic time warping (DTW) distance is proposed in (Berndt & Clifford, 1994). Aiming to solve the difficulty of scaling search to large datasets, Rakthanmanon, Campana, and Mueen (2012) propose four original optimization methods under DTW. Recently, Aggarwal and Aggarwal (2017) apply random forest to measure the similarity between data objects, which has some significant advantages over the traditional methods.

2.2.2. Applications of time series in social network analysis

Based on the similarity method proposed in (Aggarwal & Aggarwal, 2017), time series are utilized in social network analysis (Óskarsdóttir et al., 2018) to represent the dynamics of customer behavior so that customer churn in telco can be predicted. Their approach mainly includes two steps: extracting multivariate time series data and applying a time series classification technique.

In addition, a wide range of studies focuses on the use of time series in dealing with the problem of rumor detection (Kotteti et al., 2019; Ma, Gao, & Wei, 2015; Lan, Li, & Li, 2018). In Kotteti, Dong, and Qian (2019), the propagation pattern is applied to rumor detection, which divides time series into different time stamps. Ma et al. (2015) focus on the importance of changing these social context features over time and propose a model named Dynamic Series-Time Structure (DSTS). In this model, they consider both features in the interval and the slopes between two consecutive intervals. Being different from the above directions, Lan et al. (2018) investigated the variation of semantics over time series to rumor detection. They divided time series by the semantics between two consecutive intervals.

Similarly, many efforts are investigated in event detection based on time series (Nguyen & Jung, 2017; Nguyen, Anh, & Yang, 2019; Saeed, Abbasi, & Razzak, 2019; Seifikar & Farzi, 2019; Wang & Guo, 2020). Based on deep learning, Nguyen et al. (2019) proposed a new method for temporal event detection. Firstly, the input textual data is pre-processed by combining a convolutional neural network with multi-embedding. Then, an event detection model is proposed to learn time series data features by utilizing a recurrent neural network. In addition, Saeed et al. (2019) analyzed the event detection on Twitter by time series. In their work, they combined each pair of the adjacent graphs in a resultant graph series to generate enhanced heartbeat graph series. And the weights between nodes are gradually updated in subsequent graphs to characterize temporal relationships. Wang and Guo (2020) debunked the rumored event in three steps. First, they developed a Sentiment Dictionary (SD) to capture the fine-grained human emotional reactions to different events by an automatic construction method. Then, a new Two-steps Dynamic Time Series (TsDTS) algorithm is proposed, which involves the sentimental information in the division process. Finally, based on the above two steps, they proposed a novel model for rumor event detection. Moreover, the work in Seifikar and Farzi (2019) aims to complete surveying the recent text-based trend detection. They investigated the following three aspects: algorithms, dimension, and diversity of events.

From a new perspective, a recent work (Ozer, Sapienza, & Abeliuk, 2020) designs a multifaceted temporal analysis to study the timeline of popular online content, which provides insight on how popularity is gained over time. Primarily, they proposed a multidimensional shape-based time series clustering algorithm. With this algorithm, the temporal multidimensional behavior of online content is analyzed in two scalable real-world datasets.

3. Preliminaries

In this section, we first provide a brief introduction to several types of social network graphs. Then, we present the issues of building a social network with time series.

3.1. Social network graph

In our daily lives, we have diverse social networks, such as scientist cooperation networks, mail delivery networks, Microblog networks, and so on. We take the Microblog network as an example. For a microblog posted by one person, others may take different actions: ignoring, commenting, or forwarding. Suppose that two users have a relationship of commenting or forwarding, and then there is an edge between these

two users. Taking users and their relationships as nodes and edges, respectively, we can have a social network graph. To have a better explanation for the social network graph, some formalized definitions are given as follows.

Definition 1. (*(Social network graph)*) *A social network can be modeled as a graph G = (V, E, P), in which V is the set of nodes (users), E denotes the set of edges between users, and P is the set of weights for edges. A weight associated with an edge is applied to describe how one user influences another user.*

Depending on whether edges are directed or undirected, we can identify-two types of social network graphs: *directed graphs* and *undirected graphs*.

Definition 2. (*(Directed/undirected social network graph)*) *In a directed social network graph G, where node v follows node u means a directed edge (u → v) from u to v, this represents that u has an influence on v. If G is an undirected graph, there is an edge (u − v) between u and v, which can be regarded as two edges (u → v) and (v → u).*

In addition, we can identify-two types of social network graphs according to the weight set *P* in *G*, which are *weighted graphs* and *unweighted graphs*.

Definition 3. (*(Weighted/unweighted social network graph)*) *In the unweighted graph, the edges connected to different users are weighted to be* 1, *and other edges are weighted to be* 0. *As to the weighted graph, each edge is associated with a weight, which can be calculated mainly based on users' links.*

Let us look at an example. In Fig. 1, we present two major types of social network graphs. In Fig. 1 (a), the social network graph contains 5 nodes as well as 7 directed edges with weights (say, the directed edge ($u1 \rightarrow u2$) has a weight of 0.8). The social network graph in Fig. 1 (a) is, therefore, a weighted directed graph. The social network graph in Fig. 1 (b) is an unweighted undirected graph. It contains 5 nodes and 5 undirected edges without weights. When the undirected edge (u1–u2) is without weight, for example, this means the same influence is observed for u1 to u2 as well as u2 to u1, signifying a mutual influence.

3.2. Problem statement

In this paper, we concentrate on a new scenario of social networks: Building Social Network with Time series (TBSN). We aim to build a new network graph by calculating the users' similarity over text with time series. In this section, we primarily formulate the TBSN problem.

TBSN Problem. The basic graph building problem relies on *f: (U, T) → G*, where *U* is the user, and *T* denotes the text information of the user. Note that the user's text is time-sensitive. For the TBSN problem, we can construct a social network graph *f*: $(U, T)^t \rightarrow G$, where $(U, T)^t$ represents the user *U* with text information *T* over time series *t*. Here *G* is the final social network graph. The problem of TBSN is described in Fig. 2, which includes users with text over time series as the input and a graph as the output.

The TBSN problem is mainly based on the following definitions and assumptions.

Definition 4. (*(User similarity)*) *User similarity is used to measure how users are similar. The user similarity is relevant to semantic information similarity and time series similarity. Here the semantic information similarity depends on the topic of text, while the time series similarity depends on the shape of the time series.*

Definition 5. (*(Text)*) *Texts associated with users are used to represent the microblog message, co-items, co-positions, self-description, and so on. The texts in this paper represent microblogs that users posted previously.*

Definition 6. (*(Time series)*) *A time series t is denoted as* $t = t_1, t_2, \ldots, t_q$, *where* t_i $(1 \leq i \leq q)$ *is the ith time point of t and q represents the time series length.*

Definition 7. (*(Text over time series)*) *For user u, the text over time series can be denoted as* $T = \{Tt1, Tt2, \ldots, Ttq\}$ *with time series* $t=\{t_1, t_2, \ldots, t_q\}$. *Here Tt*1 *represents the text information in* t_1. *The number of text information in each time point* t_i $(1 \leq i \leq q)$ *can be regarded as the activity frequency of users at* t_i.

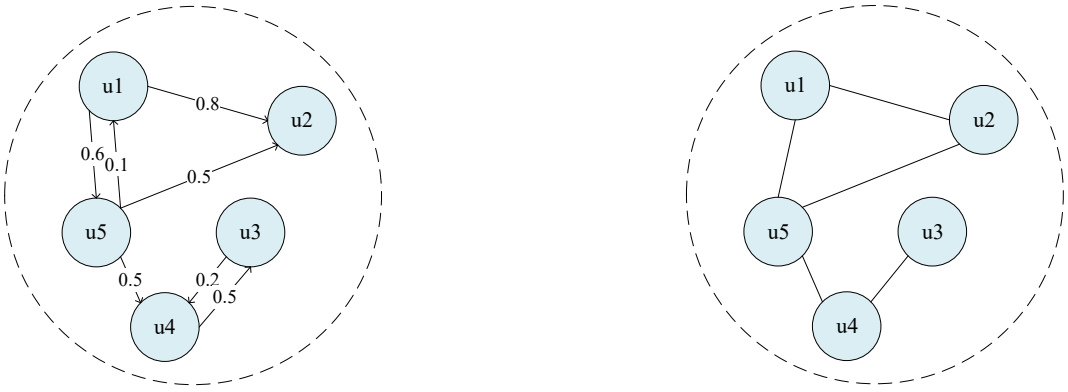
Definition 8. (*(Social network graph with text over time series)*) *These kinds of social networks can be described as G = (V, E, S, T, t), where V is the set of users, E denotes the edge between users, and S is the similarity between two users. In addition, T is the text information with users over time series t.*

Assumption 1. *Users' texts can reflect their interests. By collecting users' texts and then analyzing the topic words of these texts, we can discover users' interests.*

Assumption 2. *The variation of text over time series can reflect the change of users' interests. In addition, the value of each time point in time series can show the users' activities.*

4. Our proposed model TT-Graph

This paper devotes to building a new social network graph with our proposed TT-Graph model. We present this processing framework in Fig. 3, which is roughly divided into three portions: input, user similarity, and output. Here the similarity network is a major portion. We first concentrate on the calculation of the text semantic similarity in Section 4.1, which can be obtained by the Bert model and our proposed



(a) Weighted directed graph (b) Unweighted undirected graph

Fig. 1. Two major types of social network graphs: weighted directed graph and unweighted undirected graph.

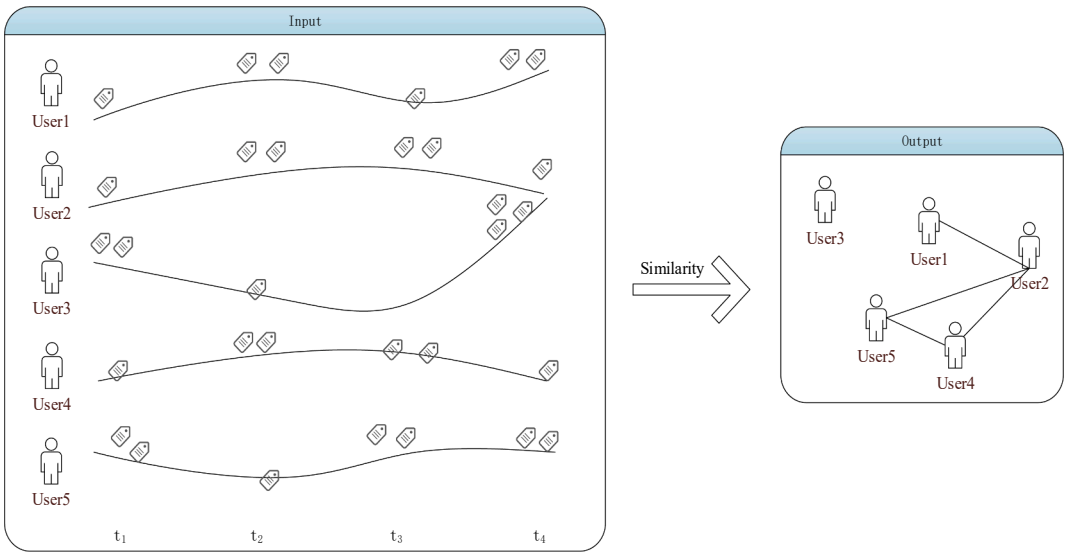


Fig. 2. The TBSN problem where input is users with text over time series and output is a graph.

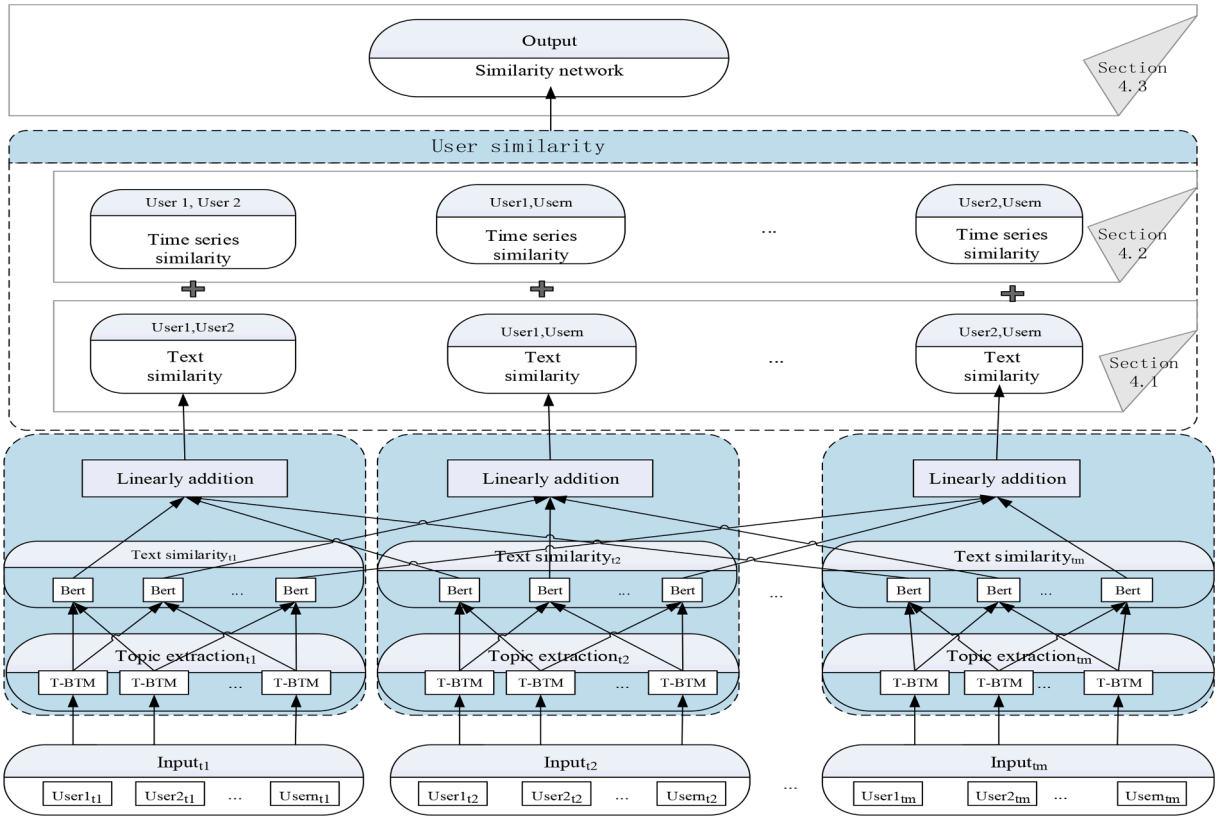


Fig. 3. The framework of the TT-Graph model.

T-BTM model. Then, in Section 4.2, we calculate the user similarity from the aspect of time series similarity. Finally, based on the text semantic similarity and the time series similarity, we build a similarity social network graph in Section 4.3. The built graph model is the output. The input is users' texts over each time point in time series. The users' topics are extracted to calculate text similarities among users.

4.1. User similarity text with time series

With the evolution of social network platforms, abundant information is produced in social network media. It is essential for us to search the latent semantic information to understand users from this massive data. Let us look at an example from Sina Weibo. In real life, some users of Sina Weibo may post blogs frequently to give their opinions, express their emotions, or record their daily routines. It is arduous to detect topics for the large amount of text information published by users. In this paper, we consider time-topic distribution and propose a novel model to obtain the semantic information of users by extracting topic words over text with time series. On this basis, we can easily calculate users' similarities.

The BTM model supposes that when words co-occur together frequently in the corpus, these words may belong to the same topic (Pan et al., 2014). Compared with the LDA model (Han et al., 2018), the BTM model can efficaciously solve the sparsity of short text problem because it does not rely on the document level but rather on the biterm level. When a biterm is assigned to a topic, it is more likely for words to belong to the same topic than a single word. It should be noted that, however, the BTM model loses sight of the time series information in text, which is crucial to social networks. A typical example is that a user may belong to different communities at different times. In this situation, it is essential to consider the time factor since users may belong to different clusters at different times. To meet our need, we incorporate a time-topic distribution in the BTM model and propose the Time-Biterm Topic Model (*T*-BTM) in Fig. 4, which is different from the traditional BTM model. The *T*-BTM model can be divided into the following four processes:

- for each topic z, we exact topic-word distribution: $\varphi z\widetilde{Dir}(\varsigma)$
- for a topic distribution from the corpus with a given time from Dirichlet $\eta\widetilde{Dir}(\kappa)$
- for each biterm b, we assign two words $wi, wj\widetilde{Multi}(\phi z)$
- for each topic z, we assign a topic with a given time $z\widetilde{Multi}(\eta)$

For the Gibbs sampling, we can easily get the global topic distribution $\eta z|t$, the topic-biterm distribution $\phi wi|z$, and $\phi wj|z$. Therefore, from the above analysis, the topic-document distribution with time can be concluded as follows.

Here $P(z|b)$ is calculated as below.

$$P(z|d) = \sum_{b} P(z|b)P(b|d) \tag{1}$$

$$P(z|b) = \frac{P(z|t)P(wi|z)P(wj|z)}{\sum_t\sum_z P(z|t)P(wi|z)P(wj|z)} = \frac{\eta z|t\phi wi|z\phi wj|z}{\sum_t\sum_z \eta z|t\phi wi|z\phi wj|z} \tag{2}$$

In addition, we can conclude that the generation probability of a biterm b=(wi, wj) in document d with a given time:

$$P(b|d) = \frac{nd(b)}{\sum_b nd(b)} \tag{3}$$

We need to obtain the topic words of each user with time series. In this paper, we need to get the similarities among users. Here we use the pre-trained BERT to acquire the similarity between users by their own topic words. The BERT model proposed in 2018 is popular in natural language processing because, compared to other methods, it has better performance in accuracy and effectiveness (Devlin, Chang, & Lee, 2019). The framework of BERT is shown in Fig. 5.

We argue that text in closer time point should have a greater influence on the corresponding user. So, we assign different weights for different time points. Having the similarity between users in each time point, we can calculate semantic similarity between users u and v on the time series as follows.

$$ss(u, v) = \sum_{i=1}^{|t|} \alpha i si(u, v) \tag{4}$$

$$\alpha i = \frac{1}{|t|} + \beta(ti - tmid) \tag{5}$$

Here $|t|$ is the length of the time series and α_i is the semantic similarity on time point i between user u and v. And α_i is a weight, denoting the proportion of influence on time i. In addition, β is a tuning parameter to deal with data dispersion.

4.2. User similarity on time series

Time series data occurs in wide applications of diverse fields, such as financial, gene expression, medical care, and so on. Many efforts, for example, investigate cluster time series from the commercial consumption. In addition to the text semantic similarity between users in Section 4.1, we also need to calculate the time series similarity by the time series shape of users. Let us again take Sina Weibo as an example. In Sina Weibo, each time series represents the number of blogs posted by a user. It can be thought that the users with similar time series shapes should have a similar activity or interest in similar topics or things at a given time. So, we can use the time series information to cluster users. The triangle similarity can effectively address noise, offset translation, and amplitude scaling. In this paper, we utilize the triangle similarity to cluster similar users with similar time series.

Let $oi = \{oi1, oi2, oi3, \ldots oit\}$ be a time series of a user where its length is t. Let O_i and O_j be the time series of two users u and v, respectively. Then the time series similarity of two users u and v can be calculated by triangle similarity as follows:

$$dT(oi, oj) = \frac{\sum_{l=1}^{t} oilojl}{\left(\sum_{l=1}^{t} o^2 il\right)^{1/2}\left(\sum_{l=1}^{t} o^2 jl\right)^{1/2}} \tag{6}$$

4.3. Build a similarity social network

After acquiring the users' similarities from the text semantics and time series, we can build a social network graph. The social network is modeled as a graph $G = (V, E)$, where V represents the set of users and E is the set of edges between users. In addition, each edge $(u, v) \in E$ with a weight w is called propagation probability. As shown in Fig. 6, we build a similarity social network graph following two steps:

(a) regarding each user as a node.
(b) calculating the similarity between a pair of users. If the similarity of two users is greater than or equal to the given threshold θ, there should be an edge between these two users. Otherwise, there is no edge between these two nodes.

Let us look at an example. Suppose that there are five users $U = \{u_1, u_2, u_3, u_4, u_5\}$ in a social network. The similarities among all users are shown in Table 1. As shown in Table 1, there may be small weights between users, which implies less contact between users. Therefore, we adopt a parameter θ to control whether there is a link between each user pair. Let $\theta = 0.5$. For u_1, there should be an edge between u_1 and u_3, which weight is 0.8. Similarly, there is an edge between u_1 and u_5 with a weight of 0.6. This process is repeatedly performed on u_2, u_3, u_4, and u_5. Finally, we give the graphic process of u_1 and the final social network graph in Fig. 7. Note that the value of θ will be tested in our experiment section to find an optimal result for our model.

As shown in Fig. 6, the final similarity of each pair of users is calculated by a linear combination of the semantic text similarity and time series similarity. The similarity between users u and v can be calculated with Formula (7).

$$fs(u, v) = \alpha ss(u, v) + (1 - \alpha)ts(u, v) \tag{7}$$

Here fs, ss, and ts represent the final similarity, semantic similarity, and time series similarity, respectively. In addition, α is a parameter used to adjust the weight of semantic similarity and time series

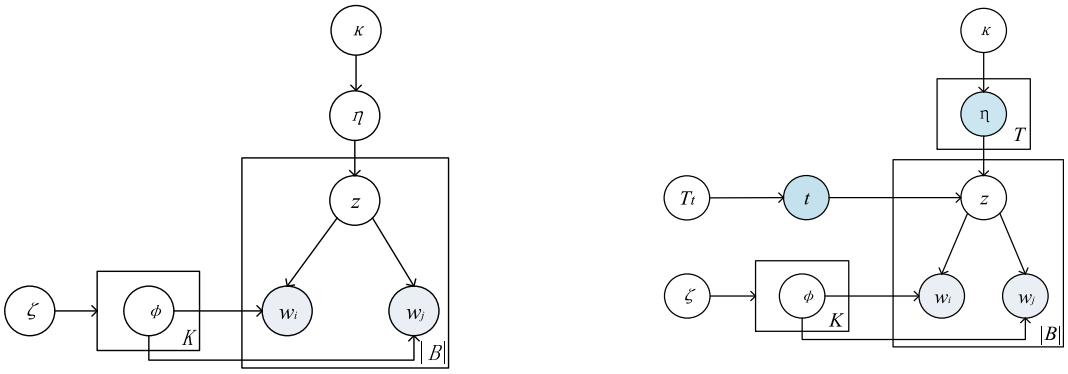


Fig. 4. Graphical structures of BTM and *T*-BTM models.

similarity.

To build a similarity network graph that combines semantic information and time series information, we give an algorithm description in Algorithm 1.

Algorithm1: Building a similarity network graph (TBSN problem)

Input: nodes V with topic words, time series, and similarity parameter θ
Output: similarity network
//Initialize
1: For v in V:
2: $u[v]$ = list($topic1$,$topic2$) // Top 2 topic words by *T*-BTM model
3: $t[v]$ = list(t_1,t_2,t_3,t_4,t_5) //The number of microblogs of each user
// Semantic similarity
4: For u in V:
5: For v in $V\backslash u$:
6: $ss[u,v]$ = cosine-similarity($n[u]$,$n[v]$) //Bert Model
// Time series similarity
7: For u in V:
8: For v in $V\backslash u$:
9: $ts[u,v]$ = triangle-similarity($t[u]$,$t[v]$) //Triangle-similarity
// Final similarity
10:$fs[u,v] = \alpha ss[u,v] + (1-\alpha)ts[u,v]\backslash$* *MERGEFORMAT*
//Building a similarity network
11: For u in V:
12: For v in $V\backslash u$:
13: if $fs[u,v]>=\theta\backslash$* *MERGEFORMAT*:
14: G.add_edge(u,v,weight = $fs[u,v]$)
15: Return G

This process can be concluded into five parts: *initialize*, *text semantic similarity*, *time series similarity*, *final similarity*, and *building a similarity social network*. As to the initialization, we aim to extract the top 2 topic words of each user each year and get the time series of each user, respectively (lines 1–3). The time complexity of this algorithm is $O(N_{iter}KN_B)$, in which N_{iter}, K, N_B, and l respectively represent the iteration times, topic number and biterms number, and $N_B = l(l\text{-}1)/2$. Note that the $l(l\text{-}1)/2$ biterms is generated by a document with l distinct words. In the phase of text semantics, we utilize the pre-trained Bert model to calculate the similarity between each pair of nodes according to their topic words each year. The final semantic similarity is calculated through total five years (lines 4–6), and its time complexity is $N_B(N_B - 1)$. In addition, lines 7–9 show the calculation of time series similarity of users, which utilizes the triangle-similarity in Formula (6), and its time complexity is $N_B(N_B - 1)$. After obtaining the semantic similarity and time series similarity, the final similarity can be computed with Formula (7) (line 10). In lines 11–15, we build a social network graph in which if the similarity between two nodes satisfies the condition, we add an edge between these two nodes. Otherwise, there will not be any edges between these two nodes. Finally, according to the above steps, we obtain a novel social network graph by inputting the nodes with topic words only. We know the time complexity of TT-Graph is $O(N_{iter}KN_B) + 2N_B(N_B - 1)$ from the analysis above.

5. Experimental evaluations of TT-Graph model

We carry out our experiment from two perspectives: 1) building a social network without any prior knowledge with the TT-Graph model; and 2) evaluating the performances of the TT-Graph on topic detection, community detection, and link prediction. We present the experimental datasets in Section 5.1; then, in Section 5.2, we utilize the TT-Graph model to build a social network. Section 5.3 evaluates the performance of the *T*-BTM model with some baselines on perplexity. Subsequently, with the built graph, we evaluate the proposed model on community detection and link prediction in Section 5.4 and Section 5.5, respectively.

All the experiments were achieved under the given environments: the processor is Intel(R) Core(TM) i7-9700 CPU@ 3.00 GHz, and the graphics card is NVIDIA 2080Ti. In addition, we applied the Anaconda 4.6.11 with Python 3.7, and the version of TensorFlow is 1.15 in our experiments.

5.1. Dataset

In our experiment, we use three datasets: *Microblog*, *Zhihu*, and *HepTh*. The Microblog dataset is randomly crawled from Sina Weibo.[4] Zhihu (Tu et al., 2017) and HepTh (High Energy Physics Theory)[5] are two publicly available datasets, which contain a graph file and a content file. The graph file shows the nodes and edges, while the content file consists of users with text information. For different experimental tasks, we need different datasets. We use these three datasets because they have different compositions and are suitable for different tasks. Microblog is utilized for building the network with the proposed TT-Graph model and detecting topics and communities. Zhihu and HepTh are applied for link prediction.

(1) Microblog consists of 257,465 microblogs posted by 1000 unique users from June 2011 to December 2015. Here the people crawled from Sina Weibo are the users. The users have followers, ranging from 1 to 49301. Note that, among these users, only a few of them are verified users and can be regarded as popular people. In addition, each text of users may contain one or multiple posts with a timestamp.
(2) Zhihu is an online community website of Q&A, where people can search for and answer questions. The dataset is formed by randomly crawling the users' following lists and following question lists. In our experiment, we regard the descriptions of their concerned topics as text information.

[4] https://weibo.com.
[5] https://snap.stanford.edu/data/cit-HepTh.html.

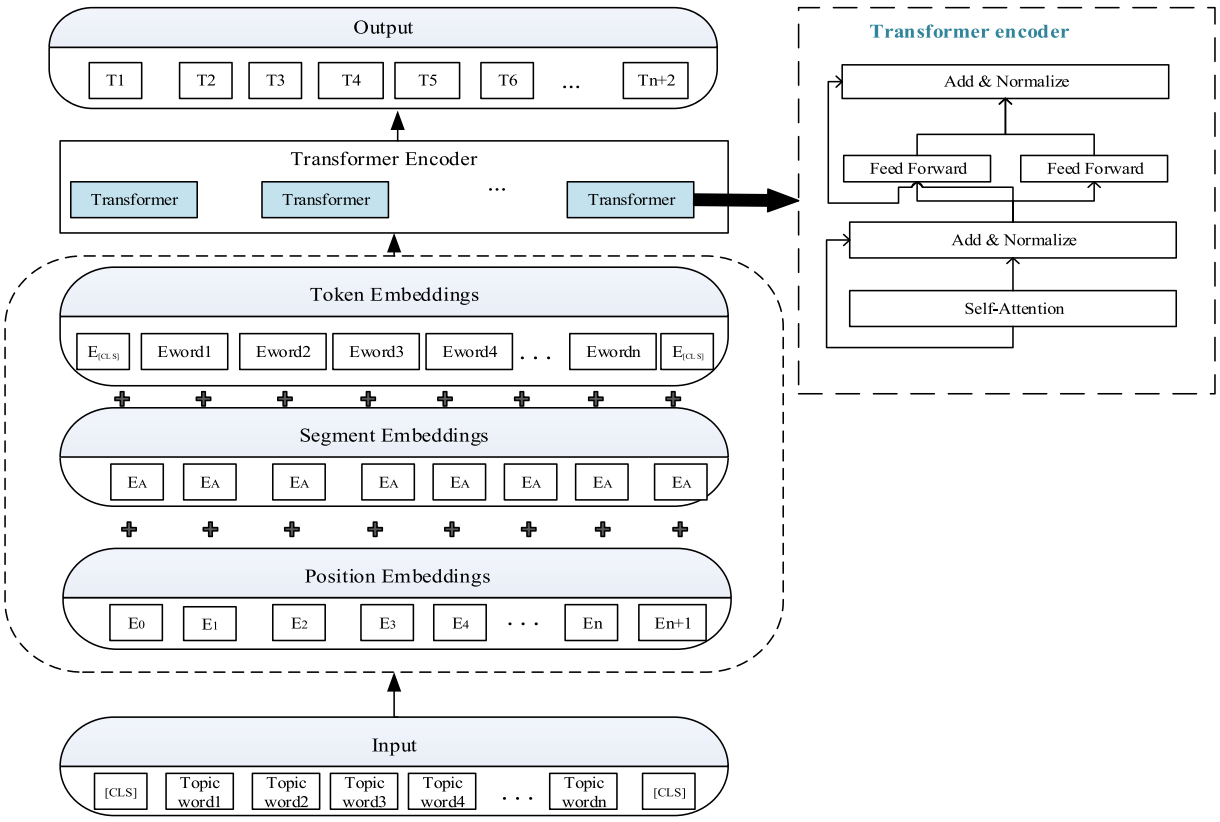


Fig. 5. The framework of the BERT model.

(3) HepTh is a citation network from arXiv3. It contains the papers published from January 1993 to April 2003. In our experiment, we filtered out the papers without any abstract information and retained 1038 papers.

Three datasets used in the experiments are summarized in Table 2.

5.2. Social network building

We first investigate latent semantic information and external time series information from users' text. Based on these two aspects, the final similarities among users are calculated by linearly adding the semantic similarity and time series similarity. This paves the way for building a similarity social network. Finally, according to the user similarity, we set a similarity parameter to filter the useless relationships between users and build a social network graph.

5.2.1. Text similarity between users

Text similarity between users is calculated according to the main interest topics of users in the texts; our proposed *T*-BTM model is used to mine topic words in texts. For our experiment with the Microblog dataset, we select the top 2 topic words for each user for each year. Then, we utilize the Bert model to calculate the similarity of the topics between users in each year. Finally, we obtain the final semantic similarity by the similarity of the topics between users over five years. As a show of the text similarity result, it is not necessary to present the result of all the 1000 users in the dataset. In our experiment, for the purpose of illustration, we randomly select five users from the 1000 users according to their id. Tables 3 and 4 present the top 2 topic words of the selected five users and their semantic similarity each year, respectively. In the experiment, the length of Microblog data is 5, and the adjustment parameter β is set to be 0.05 according to Formulas (4) and (5). The final semantic similarities are calculated as follows.

$$ss(u,v) = \sum_{i=1}^{5}\left[\frac{1}{5} + 0.05^{*}(i - imid)\right] * si(u,v)$$

5.2.2. Time series similarity between users

For the five selected users above, in Fig. 8, we list the time series information of these five users from 2011 to 2015. We note that each user has a different activity level for each year, which can reveal the external information of the users. The more similar shapes of users on the aspect of time series, the more similar activities they have.

It is reasonable to suppose that, compared to the users with different activities, the users with the same activity are more similar. Therefore, we utilize triangle similarity to calculate the time series similarity according to Formula (4) to (6), and the results of time series similarity are shown in Table 5.

5.2.3. Social network building

After obtaining the semantic similarity and time series similarity, we can calculate the final similarities among users according to Formula

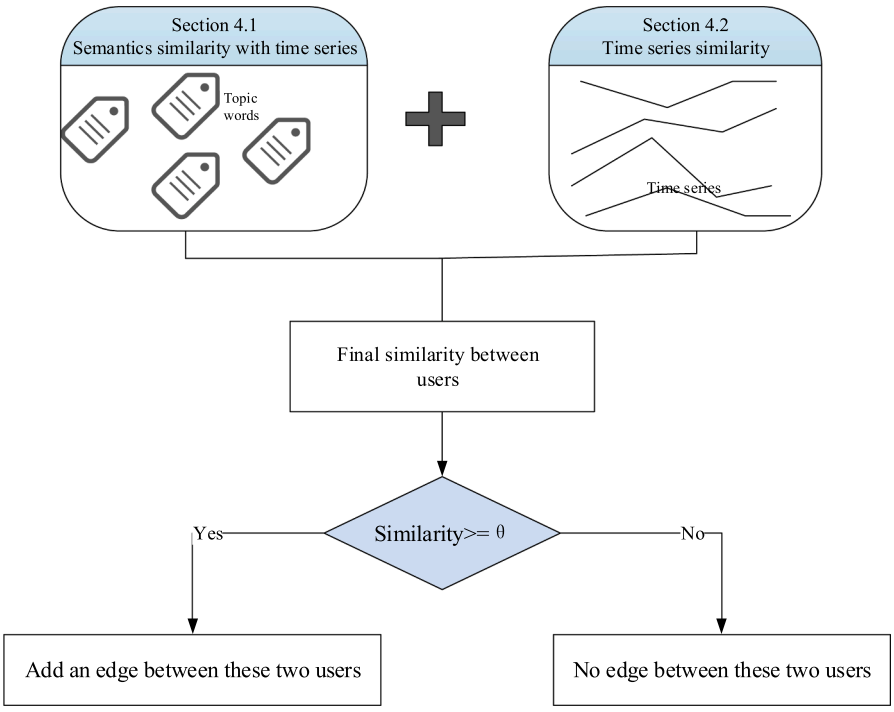


Fig. 6. Process of building a similarity network.

Table 1
The similarity between each pair of users.

User	u_1	u_2	u_3	u_4	u_5
u_1	\	0.2	**0.8**	0.3	**0.6**
u_2	0.2	\	0.1	0.4	**0.5**
u_3	**0.8**	0.1	\	**0.7**	0.3
u_4	0.3	0.4	**0.7**		**0.5**
u_5	**0.6**	**0.5**	0.3	**0.5**	\

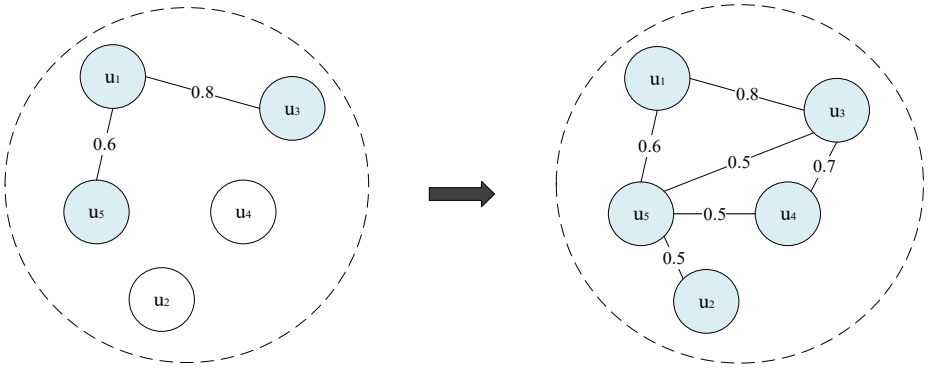
(7). The parameter α is set to be 0.5. The final similarities among the five users above are illustrated in Table 6.

According to the final similarities, we can build a social network graph by adjusting the parameter θ, which is applied to determine if the edges are filtered or not while the graph is built. In other words, θ influences the size of the built graph. Generally, a smaller θ can result in a larger scale of the social network graph. To inspect the influence of different θ, we select 100 users from the 1000 users in the Microblog dataset to building the corresponding graphs when θ is set to be 0.2, 0.4, 0.6, and 0.8, respectively. We selected 100 users rather than directly using 1000 users in the dataset to build the graph so that the built graph does not contain too many nodes and can be easily identified (for the same reason, we select 100 users from the 1000 users in the Microblog dataset to show the detected communities in Section 5.4 as well). Since the similarities of all nodes are less than 0.8, we only make an evaluation when θ is respectively 0.2, 0.4, and 0.6 in our experiment.

The built network graphs with 100 users under different θ are shown in Fig. 9. It is clearly shown in Fig. 9 that, with the increase of θ, the network graph becomes sparse because some edges whose weights are less than the given θ are filtered. Furthermore, the network graph with $\theta = 0.6$ is consistent with the real social network, here few people have a large degree, and the majority of the people have a small degree.

5.3. Topic detection

In Section 4.1, we propose a novel *T*-BTM model that is applied to



(a) The network graph built over u_1 (b) The network graph built over all five users

Fig. 7. Process of u_1 and the final social network with $\theta = 0.5$.

extract topic words from the text information. Here the topic words are very useful in calculating user's similarities which further pave the way for building the social network. In this section, we mainly evaluate the performance of the *T*-BTM model with baselines on the Microblog dataset described in Table 2.

5.3.1. Baseline methods

We roughly identify-two categories of methods for topic detection: static topic model and dynamic topic model. The former contains the BTM model (Pan et al., 2014) and the LDA model (Blei et al., 2003). The latter contains the mDTM model (Li, Kveton, Wu, & Kashyap, 2012) and the OSDATM model (Yang, Qu, & Chen, 2019). The details of these models are described as follows.

- BTM model (Pan et al., 2014): models the generation of word co-occurrence patterns (i.e., biterms) in the whole corpus to learn the topics and work well on short texts.
- LDA model (Blei et al., 2003): is a widely-used baseline method for topic modeling on long documents, implemented by the Gensim toolkit.[6]
- mDTM model (Li et al., 2012): is a flexible topic evolution model, which can effectively incorporate metadata effects.
- OSDATM model (Yang et al., 2019): combines the advantages of both the topic model and the neural language model. The OSDATM model can extract more information in time stamped documents.

5.3.2. Results

To verify the performance of the *T*-BTM model, we conduct a set of experiments on perplexity. Here perplexity is a widely used measure that can estimate the predictive power of a generative model. For a model, its lower perplexity means that it has a better performance. In the context of a *T*-BTM model, its perplexity is formally described as follows.

$$Perplexity = exp\left\{ -\frac{\sum_z lnp(b)}{B} \right\} \tag{8}$$

$$p(b) = \sum_z p(z|t)p(wi|z)p(wj|z) = \sum_z \eta z|t\phi wi|z\phi wj|z \tag{9}$$

Here B represents all biterms in the corpus, and $p(b)$ represents the joint probability of a biterm b.

For the selected five users, we sketch their perplexity scores with BTM and *T*-BTM from 2011 to 2015 in Table 7. It is shown in Table 7 that, generally, the *T*-BTM model outperforms 14 times, and the BTM outperforms 4 times. In addition, for User 3, User 4, and User 5, their perplexities of *T*-BTM perform better than the BTM in all five years. In particular, the *T*-BTM of User 5 performs 1.97%, 0.31%, 0.30% and 57.06% better than its BTM in 2012, 2013, 2014 and 2015, respectively. We can conclude that considering the time-topic distribution, the *T*-BTM model is able to get a better performance than the traditional BTM model.

In addition, for all the 1000 users, we present in Table 8 their average perplexities of five different models from 2011 to 2015. It is shown in Table 8 that the LDA model is worse than other models. The reason may be that the LDA model only focuses on the words level and ignores the timestamps information. The *T*-BTM model and the OSDATM model perform better than other models because of their relatively low perplexity scores. Note that the OSDATM is better than the *T*-BTM in 2011 because the OSDATM considers both the ordering of words as well as their timestamps. In general, the *T*-BTM model can obtain a better performance since it considers both biterms and timestamps. Let us look at the line of 2015 in Table 8. The perplexity of *T*-BTM is 2.12%, 3.86%, 1.51% and 0.42% better than the perplexities of BTM, LDA, mDTM and OSDATM models, respectively. This verifies the strong predictive power of our proposed *T*-BTM model.

[6] https://radimrehurek.com/gensim/

5.4. Community detection

With the TT-Graph model, we apply our proposed TTSLPA algorithm for community detection with experiments. To evaluate the performance of the TTSLPA algorithm, we compare it with four traditional algorithms (i.e., SLPA, Louvain, LILPA, and SLPA + word2vec) from six aspects: *precision*, *recall*, *F1-measure*, *NMI*, *ARI*, and *accuracy*. Note that the original dataset does not contain the categories of each user, and we label a subset of users according to their topics. With the topics, we manually select 1000 users that are labeled as 10 classes.

5.4.1. TTSLPA algorithm

Community detection, which can reveal the network topological structure, mainly relies on a social network graph. In this section, with the built social network graph, we propose a new community detection algorithm, which can detect community in the context of text with time series. We note that the Speaker-Listener Label Propagation Algorithm (SLPA) algorithm can effectively and rapidly detect the overlapping community (Xie, Szymanski & Liu, 2011). But, the traditional SLPA algorithm mainly has two shortcomings. The first one is that randomly selecting labels and nodes may result in instability. Subsequently, it may lead to the community detection inaccurate because it ignores important semantic information and time series information.

To solve the above problems, based on our network graph, we propose a new algorithm, called Text and Time series based on Speaker-Listener Label Propagation Algorithm (TTSLPA). Our algorithm not only considers the text semantics and time series of each node, but also takes the propagation order of nodes into consideration, and this determines the influence of nodes. The process of this algorithm mainly includes the following six steps:

(a) each node regards its topic words as its labels,
(b) label propagation starts from the node with the largest influence,
(c) node propagates its labels to its neighbors with a specified propagation rate,
(d) collecting the time which each node label is received,
(e) the labels that occur more than λ times are regarded as its final labels, and
(f) the nodes with the same labels are divided into the same communities.

In step (b), the influence of node u is calculated as follows:

$$Infu = D(u) + \sum_{v \in N(u)} puv \cdot D(v) \tag{10}$$

Here *D(u)* and *D(v)* are respectively the degree of node u and v, and p_{uv} denotes the propagation probability between u and v. The *N(u)* represents the neighbors of u.

We present the TTSLPA algorithm in Algorithm 2.

Algorithm 2: TTSLPA algorithm for community detection

Input: the network graph $G = (V, E)$ with topic words and the filter parameter λ
Output: clusters
// Initialize
1: For v in V:
2: *Label(v)=* MERGEFORMAvtopic* MERGEFORMAT* // *T*-BTM model
3: $u \leftarrow maxinfv$ // formula (10)
// Evolution
4: lo = sorted($N[u].items()$, $key = lamada\ item{:}item[1], reverse = True$) // listening order
5: For l in lo:
6: $labels = Label(l) + Label(u)$
7: $Label(l) \leftarrow max(Labels, key = labels.get)$
// Filter
8: For i in V:
9: filter $Label(i) < \lambda$
10: Cluster nodes with a same label

Table 2
Descriptions of the three datasets.

Dataset	# users	#edges	# posts	From (CST)	To (CST)	Max degree	Average clustering coefficient
Microblog	1,000	/	257,465	26 June 2011	31 December 2015	/	/
Zhihu	10,000	43,894	/	/	/	2191	0.15
HepTh	1,038	1, 974	/	January 1993	April 2003	24	0.43

Table 3
Top 2 topic words of five users over five years.

	Year				
User	2011	2012	2013	2014	2015
User 1	Pain Experience	European Cup Boring	Move Sina	Through Fast	Red Army Our army
User 2	Male Mysterious	Just Thinking	Consultation Request	Found New book	Telephone Speak
User 3	None None	None None	None None	None None	New year's Time
User 4	Morning Later	Get up confirm	None None	None None	Millet Scraping card
User 5	None None	Year Video	Crime Word	Success Membership	Lijiang Plan

Note that users may repost many useless blog posts, and we do not extract topics from such microblogs. We regard these actions as normal activity in this paper.

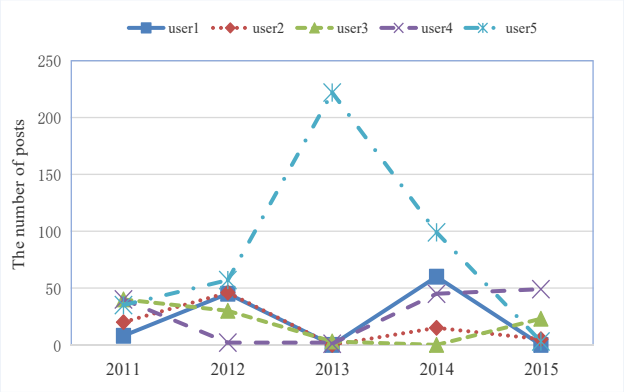


Fig. 8. Time series of five users.

Table 4
Semantic similarities among five users.

User	User	2011	2012	2013	2014	2015	Semantic similarity
1	***2***	0.44	0.38	0.40	0.41	0.43	***0.41***
1	***3***	0.00	0.00	0.00	0.00	0.40	***0.12***
1	***4***	0.40	0.40	0.00	0.00	0.43	***0.23***
1	***5***	0.00	0.42	0.40	0.41	0.42	***0.37***
2	***3***	0.00	0.00	0.00	0.00	0.42	***0.13***
2	***4***	0.39	0.36	0.00	0.00	0.43	***0.22***
2	***5***	0.00	0.37	0.43	0.42	0.43	***0.38***
3	***4***	0.00	0.00	0.00	0.00	0.41	***0.12***
3	***5***	0.00	0.00	0.00	0.00	0.40	***0.12***
4	***5***	0.00	0.40	0.00	0.00	0.43	***0.19***

In the first stage, the running time of initializing the maximum influence nodes is $O(ND_{max})$, where N is the number of nodes, and D_{max} denotes the maximum degree. Subsequently, the time complexity of evolution is $O(Tnlogn)$, where T is the iteration times. In the filter stage, the time complexity is $O(n)$. Consequently, the total time complexity of TTSLPA is $O(ND_{max} + Tnlogn + n) \approx O(ND_{max} + Tnlogn)$.

5.4.2. Baseline methods

To evaluate the effectiveness of the proposed community detection algorithm, we compare the TTSLPA algorithm with several traditional community detection approaches, including TTSLPA + word2vec, SLPA, LILPA, and Louvain algorithm. These baselines are described as follows.

- TTSLPA + Bert: Based on the semantic social network constructed by the Bert model, the TTSLPA algorithm considers the semantics of each node and selects the node with the largest influence as the initial node.
- TTSLPA + word2vec: Based on the similarity social network constructed by the Word2vec model, the TTSLPA algorithm considers the semantics of each node and selects the node with the largest influence as the initial node.
- SLPA (Xie, Szymanski, & Liu, 2011): the SLPA algorithm detects overlapping communities by updating the category labels of their communities, which are primarily used to mine overlapping communities.
- LILPA (Zhang et al., 2017): the LILPA combines the label importance, nodes importance, and node attraction to identify the label update order, which can effectively improve the stability and enhance its accuracy.
- Louvain (Blondel et al., 2008): the Louvain algorithm is a representative algorithm for its effectiveness in clustering.

Table 5
Time series similarity of five users.

	user1	user 2	user 3	user 4	user 5
user1	/	0.72	0.51	0.69	0.81
user2	0.72	/	0.63	0.80	0.24
user3	0.51	0.63	/	0.79	0.06
user4	0.69	0.80	0.79	/	0.38
user5	0.81	0.24	0.06	0.38	/

5.4.3. Evaluation criterion

We first utilize *Precision*, *Recall*, and *F1-measure* to evaluate the performance of our TTSLPA algorithm. Precision P denotes the rate of correctly clustered Microblogs to the total number of Microblogs in a cluster. Recall R denotes that the rate of Microblogs with the same topic is finally divided into the same cluster. And the F1-measure $F1$ depends on Precision and Recall.

$$P = \frac{A}{A+B} \tag{11}$$

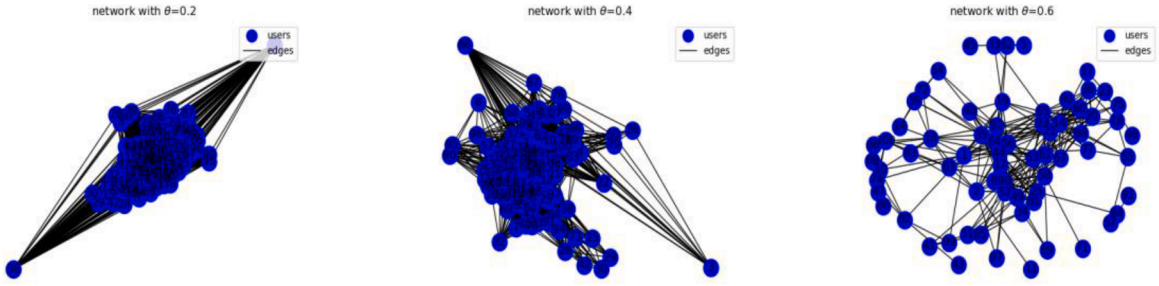
$$R = \frac{A}{A+C} \tag{12}$$

$$F1 = \frac{2 \times P \times R}{P+R} \tag{13}$$

Here A and B represent the number of Microblogs clustered to the correct cluster, and the number of Microblogs does not cluster to this cluster, respectively. And C is the number of the Microblogs that should be clustered into this cluster but finally clustered into other clusters.

Table 6
Final similarities among five users.

User	User	2011	2012	2013	2014	2015	Semantic similarity	Time series similarity	Final similarity
1	2	0.44	0.38	0.40	0.41	0.43	**0.41**	**0.72**	**0.56**
1	3	0.00	0.00	0.00	0.00	0.40	**0.12**	**0.51**	**0.32**
1	4	0.40	0.40	0.00	0.00	0.43	**0.23**	**0.69**	**0.46**
1	5	0.00	0.42	0.40	0.41	0.42	**0.37**	**0.81**	**0.59**
2	3	0.00	0.00	0.00	0.00	0.42	**0.13**	**0.63**	**0.38**
2	4	0.39	0.36	0.00	0.00	0.43	**0.22**	**0.80**	**0.51**
2	5	0.00	0.37	0.43	0.42	0.43	**0.38**	**0.24**	**0.31**
3	4	0.00	0.00	0.00	0.00	0.41	**0.12**	**0.79**	**0.46**
3	5	0.00	0.00	0.00	0.00	0.40	**0.12**	**0.06**	**0.09**
4	5	0.00	0.40	0.00	0.00	0.43	**0.19**	**0.38**	**0.28**



(a) Network graph with $\theta = 0.2$ (b) Network graph with $\theta = 0.4$ (c) Network graph with $\theta = 0.6$

Fig. 9. Three different network graphs with 100 users under different θ.

Table 7
Perplexities of five users in the BTM and *T*-BTM models over of five years.

Perplexity / Model in Year / User	2011		2012		2013		2014		2015	
	BTM	T-BTM	BTM	T-BTM	BTM	T-BTM	BTM	T-BTM	BTM	T-BTM
User 1	22.03	**21.79**	**62.52**	63.51	**89.34**	90.26	65.96	**64.25**	**57.61**	58.35
User 2	40.94	**40.57**	53.03	**52.73**	48.47	**48.28**	2.66	**0.00**	**112.60**	115.30
User 3									4.06	**4.03**
User 4	87.19	**85.08**	23.51	**22.85**					105.95	**104.92**
User 5			6.61	**6.48**	16.16	**16.11**	36.60	**35.52**	1.70	**0.73**

Table 8
Average perplexities of 1000 users in five models over five years.

Perplexity / Model / Year	T-BTM	BTM	LDA	mDTM	OSDATM
2011	55.09	57.84	62.66	56.18	54.75
2012	80.60	84.31	88.94	84.45	81.32
2013	74.56	75.01	77.73	75.63	74.89
2014	46.66	48.22	48.37	47.62	46.71
2015	40.57	41.45	42.20	41.19	40.74

In addition to *P*, *R* and *F1*, we also utilize *Accuracy*, *NMI*, and *ARI* to compare our approach with the traditional algorithms. Among them, the *NMI* can be calculated as follows.

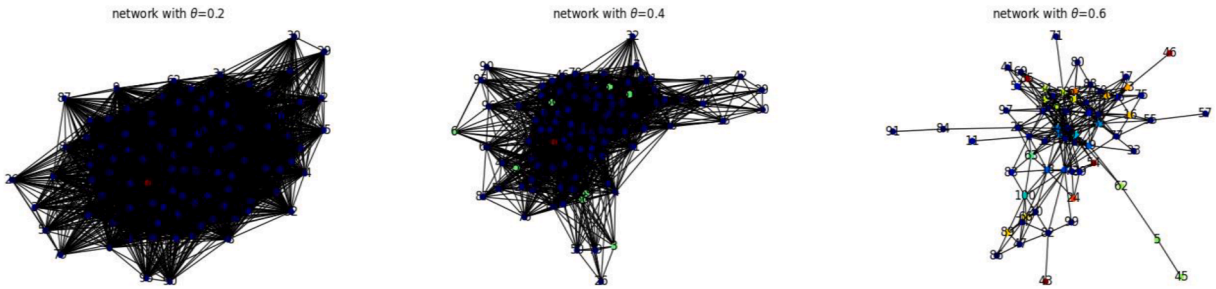
$$NMI(\Omega, C) = \frac{I(\Omega; C)}{(H(\Omega) + H(C))/2} \tag{14}$$

Here I denotes the mutual information, and H is an entropy. In addition, N is the number of samples, $\Omega = \{w_1, w_2, \ldots, w_k\}$ is s set of clusters, and $C = \{c_1, c_2, \ldots, c_k\}$ represents the real class.

5.4.4. Parameter setting

Two parameters need to be tested: similarity θ and filter λ, which are applied to build the similarity networks and filter useless labels, respectively. In Section 5.2.3, we evaluate the influence of θ on building a social network. In this section, we first investigate the influence of θ to community detection on the Microblog dataset.

We first look at the parameter of similarity θ. As described in Section 3.2, θ is a parameter that is used to decide whether edges are filtered or not during building a similarity social network. It can influences the



(a) Community with $\theta = 0.2$ (b) Community with $\theta = 0.4$ (c) Community with $\theta = 0.6$

Fig. 10. Different communities of networks with 100 users under different θ.

social network size. Considering that it is difficult to clearly identify the nodes of the detected communities if they contain too many nodes, here we only show 100 users to obtain the communities in Fig. 10. In Fig. 10, we present different communities of networks with 100 users when θ is set to be 0.2, 0.4, and 0.6, respectively. It is shown that corresponding to these three different values of θ, we respectively get 2, 5, and 9 detected communities, where the same colors represent the same communities. When θ is 0.2, the relationships in the social network are relatively close, and most users are classified into two communities. When θ is 0.4, the community structure is relatively clear, and there may be a large gap between nodes. When θ is 0.6, many communities are clearly detected, and the users in the same community are very similar.

We further evaluate the precision, recall, F1-measure, and the size of the similarity network of a total of 1000 users in the dataset with different θ. As shown in Fig. 11, with the increase of θ, the performance of our algorithm TTSLPA gets better in precision, recall, and F1-measure. The reason may be that, with the increase of θ, the network structure becomes clear, and there are no useless edges.

We also evaluate the performance of our algorithm with different θ in NMI, ACC, and ARI. We easily find from Fig. 12 that when $\theta = 0.6$, the TTSLPA obtain 0.67, 0.87, and 0.64 in NMI, ACC, and ARI, respectively. It is shown that $\theta = 0.6$ has excellent advantages over $\theta = 0.2$ and $\theta = 0.4$. Note that, in the performance verification of θ, the parameter λ is set to be 0.4. The reason why λ is set to be 0.4 is demonstrated in the following evaluation.

We can finally draw a conclusion that when $\theta = 0.6$, the TTSLPA outperforms *NMI*, *ACC*, and *ARI* in *precision*, *recall*, and *F1-measure*.

Now we look at the parameter of filter λ. To make the clustering result more accurate, the parameter λ is introduced to filter the scale of labels. For clear node identification, Fig. 13 presents the detected communities of 100 users with different λ. Here we set λ to be ranging from 0.1 to 0.6, and we get 15, 24, 24, 24, 24, and 23 communities, respectively.

Now we evaluate the precision, recall, F1-measure, and the size of

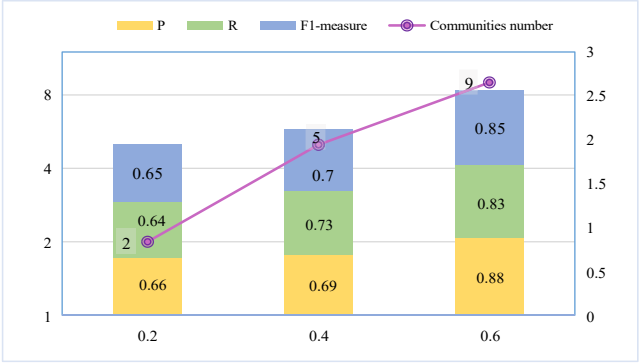


Fig. 11. Results of 1000 users in P, R and F1-measure under different θ.

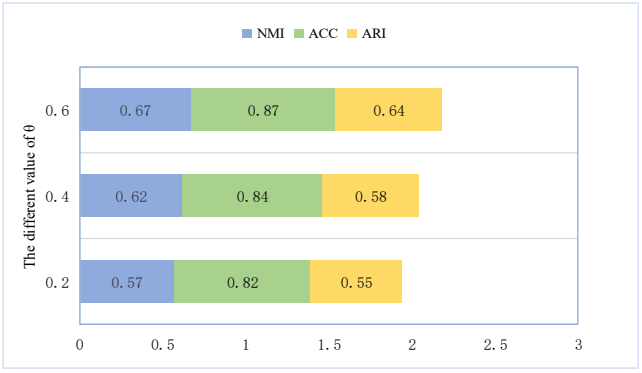


Fig. 12. Results of 1000 users in NMI, ACC and ARI under different θ.

similarity network of the entire 1000 users in the dataset with different λ. As shown in Fig. 14, with the increase of λ, the number of communities our algorithm detects decreases. The reason is that when λ is small, a node may have many labels, and this can result in many communities. Conversely, when λ is large, the algorithm can filter many node labels, resulting in fewer communities. Moreover, when $\lambda = 0.4$, our approach achieves the best results of 0.85, 0.83, and 0.88 in precision, recall, and F1-measure. We also evaluate the performance of our algorithm with different λ in NMI, ACC, ARI and the number of clusters. It is shown in Fig. 15 that the NMI, ACC, and ARI can get excellent manifestation with $\lambda = 0.4$.

5.4.5. Results

After verifying the optimal parameters θ and λ, in this section, we compare our TTSLPA with four traditional approaches. We set $\theta = 0.6$ and $\lambda = 0.4$ to evaluate their performances in detecting communities from seven aspects: *Running time*, *P*, *R*, *F1-measure*, *NMI*, *ACC*, and *ARI*. The experimental results are presented in Fig. 16 and Table 9.

It is shown that, compared with 4.16 s (SLPA), 1.47 s (Louvain), and 4.69 s (LILPA), the TTSLPA cannot obtain the best performance with its running time being 5.34/5.35 s. The reason is that the TTSLPA algorithm propagates several labels to its neighbors while others have only one label, and this needs additional running time. As to the TTSLPA + Bert and the TTSLPA + Word2vec, the TTSLPA + Bert can perform 12.82%, 0.00%, and 44.07% better than the TTSLPA + Word2vec in *P*, *R*, and *F1-measure*, respectively. The reason is that the Bert model considers the context information and polysemy situation, and this results in its better performance than the Word2vec model. In general, the TTSLPA + Bert and the TTSLPA + Word2vec greatly outperform the methods of SLPA, Louvain, and LILPA in terms of precision, recall, and F1-measure. Based only on the topological structure, the SLPA and Louvain ignore the related semantic information, resulting in inaccurate results. In addition, the LILPA considers the semantic information but ignores the time

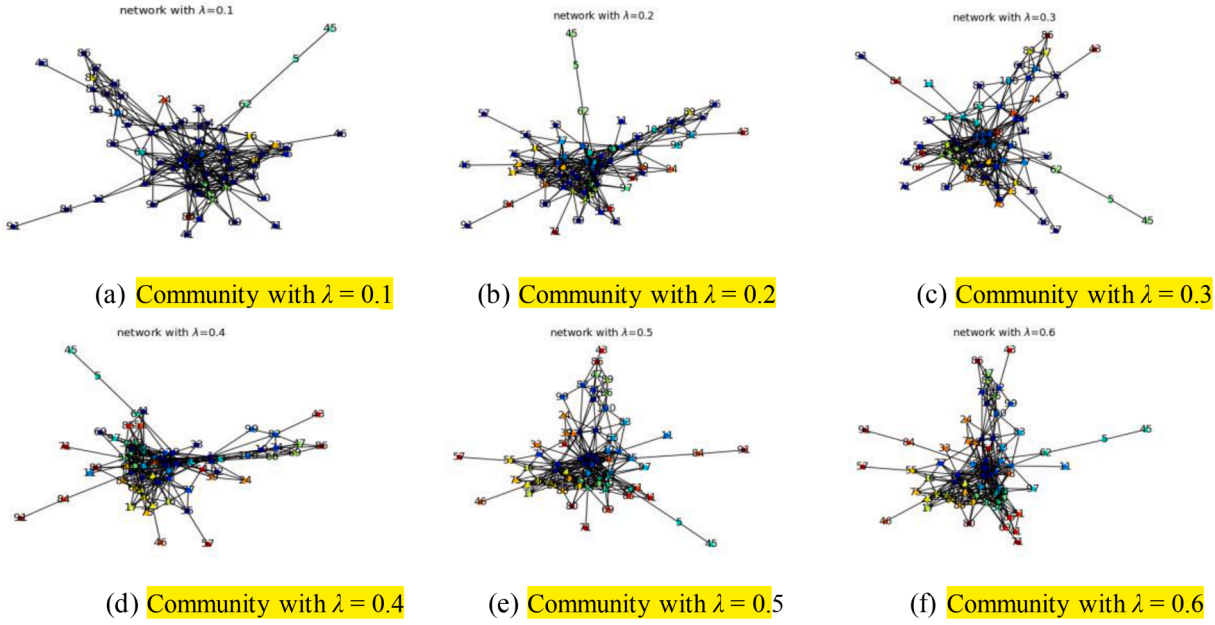


Fig. 13. Different communities of networks with 100 users under different λ.

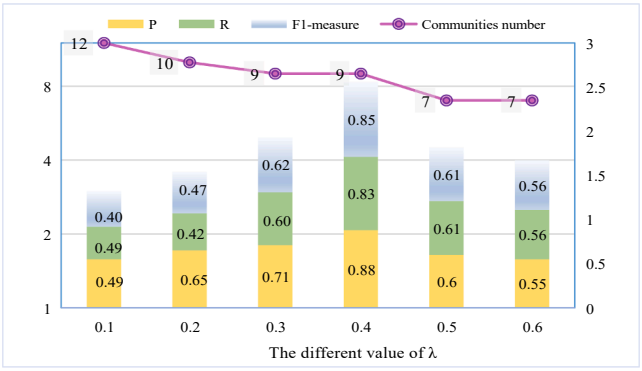


Fig. 14. Results of 1000 users in P, R, and F1-measure under different λ.

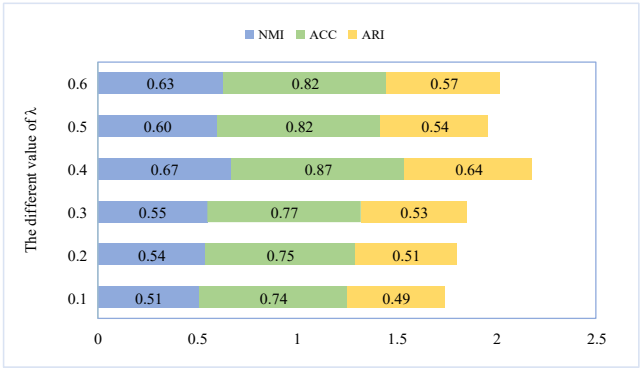


Fig. 15. Results of 1000 users in NMI, ACC, and ARI under different λ.

series information, which makes its performance better than the Louvain but worse than the TTSLPA. It is also shown in Table 9 that, in terms of *NMI*, *ACC*, and *ARI*, the TTSLPA + Bert basically performs better than other approaches. This is because these traditional approaches only consider the topological structure of the network but ignore the semantic information and time series information.

Overall, we conclude that the TTSLPA can obtain an ideal performance, which shows the effectiveness of our proposed algorithm.

5.5. Link prediction

In this section, we evaluate the link prediction task, which assesses the ability of TT-Graph in building the network. The original Microblog dataset contains only nodes with no edges. So it cannot be utilized for the link prediction task. In our evaluation, the datasets for link prediction should be the social network graph with text over time series. Although some public datasets contain graph and text information, there are few public datasets containing time series information. Here we use two datasets Zhihu, and HepTh, described in Table 2, containing two files of the graph information and text information of users. Moreover, in our experiment, we utilize AUC (Areas Under ROC), which is widely adopted in link prediction, to evaluate the performance of our proposed method. AUC can be interpreted as the probability that a randomly chosen missing link has higher score than a randomly chosen nonexistent link (Wang, Xu, Wu, & Zhou, 2015).

5.5.1. Baseline methods

In the experiment, we use the following five methods as our baselines, including three structure-based methods and two text-based methods:

- LINE (Jian, Meng, & Wang, 2015): this method learns the embedding vectors for each node by concatenating the first-order and the second-order proximity of the network, respectively.
- Jaccard Coefficient: it normalizes the size of common neighbors.
- Comment Neighbor (Newman, 2001): the CN metric is one of the most popular measurements in link prediction for its simplicity.
- CENE (Sun, Guo & Ding, 2016): leverages both structure modeling and text modeling by regarding text content as a special kind of vertices.
- CANE (Tu et al., 2017): this method aims to learn various context-aware embeddings for a node when interacting with different neighbor nodes.

6. Results

The goal of our experiment is to predict the links between users. We

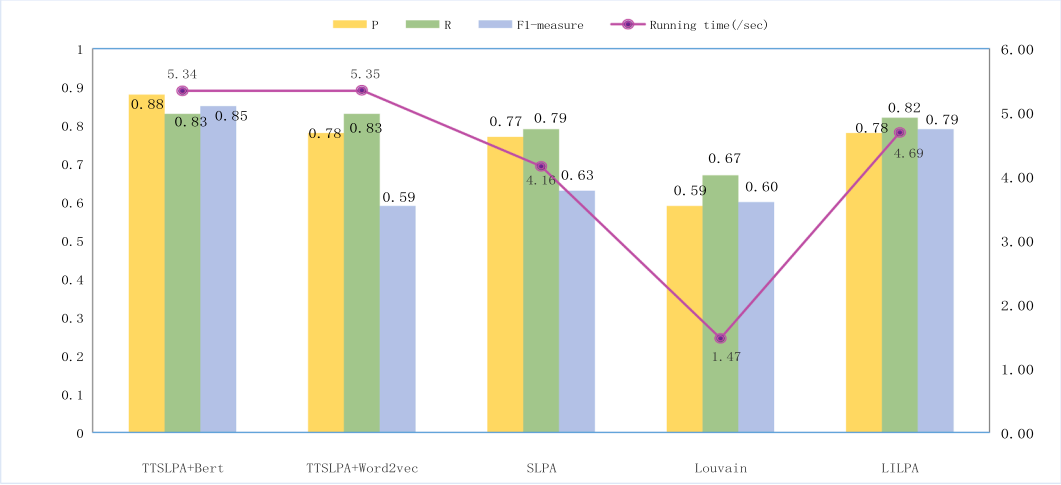


Fig. 16. Results of five different algorithms with 1000 users in P, R, F1-measure and running time.

Table 9
Results of five evaluation methods with 1000 users in seven different aspects.

Evaluation	TTSLPA + Bert	TTSLPA + Word2vec	SLPA	Louvain	LILPA
Running time	5.34	5.35	4.16	**1.47**	4.69
P	**0.88**	0.78	0.77	0.59	0.78
R	**0.83**	**0.83**	0.79	0.67	0.82
F1-measure	**0.85**	0.59	0.63	0.60	0.79
NMI	0.67	**0.68**	0.63	0.59	0.65
ACC	**0.87**	0.84	0.80	0.73	0.82
ARI	**0.64**	0.59	0.50	0.43	0.54

Table 10
AUC values of 10,000 users in Zhihu.

%Training edges	Jaccard Coefficient	Common Neighbor	LINE	CENE	CANE	TT-Graph
0%	/	/	/	/	/	0.21
5%	0.47	0.43	0.51	**0.53**	0.52	/
15%	0.46	0.46	0.52	**0.56**	**0.56**	/
25%	0.50	0.49	0.56	**0.57**	**0.57**	/
35%	0.51	0.51	**0.60**	**0.60**	0.57	/

Table 11
AUC values of 1038 users in HepTh.

%Training edges	Jaccard Coefficient	Common Neighbor	LINE	CENE	CANE	TT-Graph
0%	/	/	/	/	/	**0.13**
5%	0.29	0.40	0.43	**0.79**	0.75	/
15%	0.53	0.56	0.54	**0.86**	0.84	/
25%	0.59	0.58	0.60	**0.85**	**0.85**	/
35%	0.59	0.59	0.67	**0.90**	0.87	/

evaluate the AUC values while removing different ratios of edges on Zhihu and HepTh datasets, respectively. Here we adopt $\theta = 0.9$ according to their similarities in the experiment. The experimental results are shown in Tables 10 and 11. According to the results, we have the following observations.

(1) Without training edges. Our proposed TT-Graph can achieve 0.21 and 0.13 AUC values on Zhihu and HepTh datasets, respectively. It indicates the effectiveness of TT-Graph when applied to link prediction tasks without training edges. Also, it verifies that the TT-Graph has the capability of modeling relationships between nodes. The other five baselines do not apply to the scenes without training edges.

(2) With training edges. Jaccard Coefficient, Common Neighbor, LINE, CENE, and CANE exhibit unstable performance under various training ratios. With the increase of training edges, their performance becomes better. Especially, text-based methods such as CENE and CANE can obtain better performance than others. And this proves that the context information benefits the link prediction task. Note that our proposed TT-Graph is not sensitive to the radio of training edges, and this results in bad performance with various training edges.

In summary, all the above observations demonstrate that the text-based method can improve the performance of the link prediction task. Considering the text information, our TT-Graph model can get better performance without training edges, indicating the effectiveness of TT-Graph in building a social network without prior knowledge. However, compared with other methods, the TT-Graph model cannot deal with the link prediction task well when we face various training ratios.

7. Conclusions

In this paper, we proposed a novel model to build a social network without any prior acknowledge from a new perspective by considering semantic information and time series information. First, we utilize a new *T*-BTM model to extract the topics from text information with time series. Then, we combine the *T*-BTM and the Bert model together to obtain semantic similarity. Subsequently, we build a similarity network by considering the above semantic similarity and time series similarity. Based on our similarity network, by specifying the initial labels and sorting the propagation order of the nodes, a new method, TTSLPA, is proposed for community detection, which addresses problems that the SLPA algorithm suffers. We conducted extensive experimental results on several datasets, and the results clearly showed the credibility of our proposed model on topic detection, community detection, and link prediction.

In the future, we will focus on two research directions. First, we will try to apply our model to other domains with text information with time series. Furthermore, we will explore the impact of user intrinsic characteristics in complex networks and build a powerful social network graph model with more semantics such as temporal and/or spatial information (Fani, et al., 2020; Huang, Chen, Ren, & Wang, 2021).

CRediT authorship contribution statement

Wei Jia: Methodology, Writing – original draft. **Ruizhe Ma:** Methodology, Writing – review & editing. **Li Yan:** Formal analysis. **Weinan Niu:** Data curation, Software. **Zongmin Ma:** Investigation, Project administration.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Arif, T., Asger, M., Malik, M. B., et al. (2015). Extracting academic social networks among conference participants. In *Proceedings of the 8th International Conference on Contemporary Computing* (pp. 42–47). IEEE.

Arif, T., Ali, R., & Asger, M. (2014). Author name disambiguation using vector space model and hybrid similarity measures. In *Proceedings of the 7th International Conference on Contemporary Computing* (pp. 135–140). IEEE.

Aslay, C., Barbieri, N., Bonchi, F., et al. (2014). Online topic-aware influence maximization queries. In *Proceedings of the 2014 International Conference on Extending Database Technology* (pp. 295–306).

Adamic, L. A., & Adar, E. (2003). Friends and neighbors on the Web. *Social Networks, 25*(3), 211–230.

Aggarwal, C. C., & Aggarwal, C. C. (2017). Similarity forests. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 395–403). ACM.

Bozorgi, A., Haghighi, H., Zahedi, M. S., et al. (2016). INCIM: A community-based algorithm for influence maximization problem under the linear threshold model. *Information Processing & Management, 52*(6), 1188–1199.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet Allocation. *Journal of Machine Learning Research, 3*, 993–1022.

Berndt, D. J., & Clifford, J. (1994). Using dynamic time warping to find patterns in time series. *KDD Workshop*, 359–370.

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., & Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Stat. Mech. Theory Exp., 10*, 155–168.

Chu, Y., Zhao, X., Liu, S., et al. (2014). An efficient method for topic-aware influence maximization. In *Proceedings of the 16th Asia-Pacific Web Conference on Web Technologies and Applications* (pp. 584–592).

Cheng, J., Ma, T., Chen, X., et al. (2020). A seed-expanding method based on TOPSIS for community detection in complex networks. *Complexity, 2020*, 1–14.

Cheng, X., Yan, X., Lan, Y., et al. (2014). BTM: Topic modeling over short texts. *IEEE Transactions on Knowledge & Data Engineering, 26*(12), 2928–2941.

Dusan, Zeleník, & Mária, Bieliková (2011). New recommending based on text similarity and user behavior. In *Proceedings of the 7th International Conference on Web Information Systems and Technologies* (pp. 302–307).

Devlin, J., Chang, M. W., Lee, K., et al. (2019). BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 4171–4186).

Eichinger, T., Beierle, F., Khan, S. U., et al. (2019). Affinity: A system for latent user similarity comparison on texting data. In *Proceedings of the 2019 IEEE International Conference on Communications* (pp. 1–7). IEEE.

Faloutsos, C., Ranganathan, M., & Manolopoulos, Y. (1994). Fast subsequence matching in time-series databases. In *Proceedings of the 1994 ACM SIGMOD International Conference on Management of Data* (pp. 419–429). ACM.

Fani, H., et al. (2020). User community detection via embedding of social network structure and temporal content. *Information Processing & Management, 57*(2), Article 102056.

Goyal, A., Lu, W., & Lakshmanan, L. V. S. (2011). SIMPATH: an efficient algorithm for influence maximization under the linear threshold model. In *Proceedings of the 2012 IEEE International Conference on Data Mining* (pp. 211–220). IEEE.

Huang, X., Chen, D., Ren, T., & Wang, D. (2021). A survey of community detection methods in multilayer networks. *Data Mining and Knowledge Discovery, 35*(1), 1–45.

Jung, K., Heo, W., & Chen, W. (2013). IRIE: Scalable and robust influence maximization in social networks. *Proceedings of the 2012 IEEE 12th International Conference on Data Mining*. IEEE.

Jing, D., & Liu, T. (2019). Context-based influence maximization with privacy protection in social networks. *EURASIP Journal on Wireless Communications and Networking, 142* (2019).

Han, X., Chen, D., & Yang, H. (2019). A semantic community detection algorithm based on quantizing progress. *Complexity, 2019*, 1–13.

Han, X., Wang, L., Farahbakhsh, R., et al. (2016). CSD: A multi-user similarity metric for community recommendation in online social networks. *Expert Systems with Applications, 53*, 14–26.

Han, Di, Li, Jianqing, & Li, Wenting (2018). Predictor: An app cold start recommender system based on user similarity and app periodicity. *Proceedings of the 10th International Conference on Internet Multimedia Computing and Service*. ACM, 1-42:4.

Jian T, Meng Q, Wang M, et al. (2015). LINE: Large-scale information network embedding. In *Proceedings of the 24th International World Wide Web Conferences* (pp. 1067–1077).

Krishnamurthy, V., & Hoiles, W. (2016). Information diffusion in social sensing. *Numerical Algebra Control and Optimization, 6*(3), 365–411.

Kotteti, C. M. M., Dong, X., & Qian, L. (2019). Rumor detection on time-series of tweets via deep learning. In *Proceedings of the 2019 IEEE Military Communications Conference* (pp. 1–7). IEEE.

Li T, Kveton B, Wu Y, & Kashyap A. (2012). *Incorporating metadata into dynamic topic analysis*. abnms org, 2014.

Li, D., Zhang, Y., Xu, Z., et al. (2016). Exploiting information diffusion feature for link prediction in Sina Weibo. *Scientific Reports, 6*, 0058.

Liu, C., & Guo, C. (2020). STCCD: Semantic trajectory clustering based on community detection in networks. *Expert Systems with Applications, 162*, Article 113689.

Lan, T., Li, C., & Li, J. (2018). Mining semantic variation in time series for rumor detection via recurrent neural networks. In *Proceedings of the 2018 IEEE 20th International Conference on High Performance Computing and Communications* (pp. 282–289). IEEE.

Makrehchi, M. (2011). Social link recommendation by learning hidden topics. In *Proceedings of the 2011 ACM Conference on Recommender Systems* (pp. 189–196). ACM.

Ma, J., Gao, W., Wei, Z., et al. (2015). Detect rumors using time series of social context information on microblogging websites. In *Proceedings of the 24th ACM International Conference on Information and Knowledge Management* (pp. 1751–1754). ACM.

Óskarsdóttir, M., Calster, T. V., Baesens, B., et al. (2018). Time series for early churn detection: Using similarity based classification for dynamic networks. *Expert Systems with Applications, 106*, 55–65.

Ozer, M., Sapienza, A., Abeliuk, A., et al. (2020). Discovering patterns of online popularity from time series. *Expert Systems with Applications, 151*, Article 113337.

Newman, M. (2001). Clustering and preferential attachment in growing networks. *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics, 64*(2), Article 025102.

Nguyen, D. T., & Jung, J. E. (2017). Real-time event detection for online behavioral analysis of big social data. *Future Generation Computer Systems, 66*, 137–145.

Nguyen, V. Q., Anh, T. N., & Yang, H. J. (2019). Real-time event detection using recurrent neural network in social sensors. *International Journal of Distributed Sensor Networks, 15*(6), 155014771985649.

Pan, Yali, Yin, Jian, Liu, Shaopeng, et al. (2014). A biterm-based dirichlet process topic model for short texts. *Proceedings of the 3rd International Conference on Computer Science and Service System*.

Rakthanmanon, T., Campana, B., Mueen, A., et al. (2012). Searching and mining trillions of time series subsequences under dynamic time warping. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 262–270). ACM.

Saeed, Z., Abbasi, R. A., Razzak, I., et al. (2019). Enhanced heartbeat graph for emerging event detection on twitter using time series networks. *Expert Systems with Applications, 136*, 115–132.

Seifikar, M., & Farzi, S. (2019). A comprehensive study of online event tracking algorithms in social networks. *Journal of Information Science, 45*(2), 156–168.

Shen, X., Mao, S., & Chung, F. L. (2020). Cross-network learning with fuzzy labels for seed selection and graph sparsification in influence maximization. *IEEE Transactions on Fuzzy Systems, 28*(9), 2195–2208.

Sun, H., Jia, X., Huang, R., et al. (2020). Distance dynamics based overlapping semantic community detection for node-attributed networks. *Computational Intelligence*. https://doi.org/10.1111/coin.12324

Škrlj, B., Kralj, J., & Lavrač, N.d. (2019). CBSSD: Community-based semantic subgroup discovery. *Journal of Intelligent Information Systems, 53*, 265–304.

Smith, M. A., Shneiderman, B., Milic-Frayling, N., et al. (2009). Analyzing social media networks with NodeXL. *Proceedings of the Fourth International Conference on Communities and Technologies*. PA, USA: University Park.

Spaeth, A., & Desmarais, M. C. (2013). In *Combining Collaborative Filtering and Text Similarity for Expert Profile Recommendations in Social Websites and Personalization* (pp. 178–189). Springer.

Tang, J., Tang, X., & Yuan, J. (2017). Influence maximization meets efficiency and effectiveness: A hop-based approach. In *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, Beijing* (pp. 64–71).

Tu C, Han L, Liu Z, et al. (2017). CANE: Context-aware network embedding for relation modeling. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics* (pp. 1722-1731).

Wang, P., Xu, B., Wu, Y., & Zhou, X. (2015). Link prediction in social networks: The state-of-the-art. *Science China Information Sciences, 58*(1), 1–38.

Wang, R., Rho, S., Chen, B. W., et al. (2017). Modeling of large-scale social network services based on mechanisms of information diffusion: Sina Weibo as a case study. *Future Generation Computer Systems, 74*, 291–301.

Wang, Weiqing, et al. (2015). Geo-SAGE: A Geographical Sparse Additive Generative Model for Spatial Item Recommendation. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 1255–1264). ACM.

Wang, Z., & Guo, Y. (2020). Rumor events detection enhanced by encoding sentimental information into time series division and word representations. *Neurocomputing, 397*(2), 224–243.

Wu, D., Zhang, M., Shen, C., Huang, Z., & Gu, M. (2020). BTM and GloVe similarity linear fusion-based short text clustering algorithm for microblog hot topic discovery. *IEEE Access, 8*, 32215–32225.

Xie, J. R., Szymanski, B. K., & Liu, X. M. (2011). SLPA Uncovering overlapping communities in social networks via a speaker-listener interaction dynamic process. In *Proceedings of the 11th IEEE International Conference on Data Mining Workshops* (pp. 344–349). IEEE.

Yin, H., Cui, B., Zhou, X., et al. (2016). Joint modeling of user check-in behaviors for real-time point-of-interest recommendation. *ACM Transactions on Information Systems, 35*(2), 1631–1640.

Yan, X., Fanrong, M., Yong, Z., et al. (2014). A node influence based label propagation algorithm for community detection in networks. *The Scientific World Journal, 2014*, Article 627581.

Yang, M., Qu, Q., Chen, X., et al. (2019). Discovering author interest evolution in order-sensitive and Semantic-aware topic modeling. *Information Sciences, 486*, 271–286.

Zhang, Xiaohang, Liu, Jiaqi, Du, Yu, et al. (2011). A novel clustering method on time series data. *Expert Systems with Applications, 38*(9), 11891–11900.

Zhang, X. K., Ren, J., Song, C., Jia, J., & Zhang, Q. (2017). Label propagation algorithm for community detection based on node importance and label influence. *Physics Letters, A, 381*(33), 2691–2698.

Zhu, G., Pan, Z., Wang, Q., et al. (2020). Building multi-subtopic Bi-level network for micro-blog hot topic based on feature Co-Occurrence and semantic community division. *Journal of Network and Computer Applications, 170*, Article 102815.

Zhong, H., Lyu, H., Zhang, S., et al. (2020). Measuring user similarity using check-ins from LBSN: A mobile recommendation approach for e-commerce and security services. *Enterprise Information Systems, 14*(3), 368–387.

Zheng, Y., Xie, X., & Ma, W. Y. (2010). GeoLife: A Collaborative Social Networking Service among user, location and trajectory. *IEEE Data Engineering Bulletin, 33*(2), 32–40.