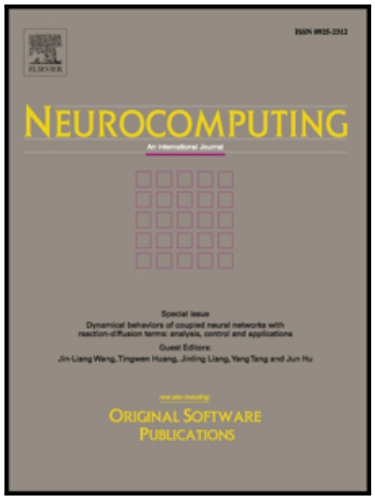
Communicated by E.P.J van den Heuvel

# Accepted Manuscript




A multi-feature probabilistic graphical model for social network semantic search

Feifei Kou, Junping Du, Congxian Yang, Yansong Shi, Meiyu Liang, Zhe Xue, Haisheng Li

PII: S0925-2312(18)31274-8
DOI: https://doi.org/10.1016/j.neucom.2018.03.086
Reference: NEUCOM 20098

To appear in: *Neurocomputing*

Received date: 31 July 2017
Revised date: 21 February 2018
Accepted date: 21 March 2018

Please cite this article as: Feifei Kou, Junping Du, Congxian Yang, Yansong Shi, Meiyu Liang, Zhe Xue, Haisheng Li, A multi-feature probabilistic graphical model for social network semantic search, *Neurocomputing* (2018), doi: https://doi.org/10.1016/j.neucom.2018.03.086

# A multi-feature probabilistic graphical model for social network semantic search

Feifei Kou[a], Junping Du[a,*], Congxian Yang[a], Yansong Shi[a], Meiyu Liang[a], Zhe Xue[a], Haisheng Li[b]

[a]*Beijing Key Laboratory of Intelligent Telecommunication Software and Multimedia, School of Computer Science, Beijing University of Posts and Telecommunications, Beijing 100876, China*

[b]*School of Computer and Information Engineering,Beijing Technology and Business University, Beijing 100048, China*

**Abstract**

With the rapid development of social network platforms, more and more people are using them to search for material related to their interests. As the texts of social media messages are usually so short, when traditional existing document modeling methods are used in social network search tasks, the problem of semantic sparsity arises, leading to low-quality semantic representation and low-precision social network search results. Fortunately, besides of short text, social media data also has other features, such as timestamps, locations, and its user information. In light of this, to realize precise social network search, we propose a multi-feature probabilistic graphical model (MFPGM), which can generate high-quality semantic representation. To deal with the problem of semantic sparsity, we exploit two strategies in MFPGM. First, we propose a concept named special region and utilize location information to aggregate short text into long text. Second, we introduce the biterm pattern that can generate dense semantic space by supposing that a biterm occurring in the same context has the same topic. In order to generate high-quality semantic representations, we simultaneously model multiple features (i.e. biterm, user, location and timestamp) of social network data to enhance the semantic learning process

*Corresponding author : Junping Du
*Email address:* junpingdu@126.com (Junping Du)

of MFPGM. We conduct a lot of experiments on real-word datasets, and the comparisons with several state-of-art baseline methods have demonstrated the superiority of our MFPGM on topic quality and search performance. Additionally, with the help of the generated semantic representations, MFPGM allows people to analyze the relationships between time and the popularities of topics.

*Keywords:* social network platform; short text; multi-feature; semantic search

## 1. Introduction

Social network platforms are developing rapidly [1], allowing users to conveniently publish messages [2][3]. With the spread of information on social networks, many real-life events and emerging topics have attracted a great deal of attention through social network platforms [4][5]. These platforms have already become the main sources of information for many people [6]. However, the text of social media data is very concise. For example, the length of messages in Sina Weibo (one of the most popular social platforms in China) is usually shorter than 140 characters [7][8]. It is hard to conduct accurate search on social network platforms only through short text [9].

In the textual search field, typical methods mainly include two categories: vector space models (VSM) and document modeling methods [10]. VSM such as TF-IDF and BM25 are difficult to deal with polysemy and synonymy, as they ignore the meaning of terms in corpus. Document modeling methods can explore interweaving relationship among terms by modeling data into latent semantic space [11][12]. The most popular document modeling method, the LDA and its variants are all probabilistic graphical models [13], which have also been widely applied in various fields, such as action recognition [14], point of interest recommendation [15], cross-media retrieval [16], 3D object retrieval [17], etc. However, although probabilistic graphical models perform well in various fields, especially in long document modeling [18][19], they do not resolve the problem of semantic sparsity [20][21] when they are used for the social network search, because social network data texts are so short.

Many researchers have attempted to improve the semantic representation ability of probabilistic graphical models. An appealing way of doing this is to overcome semantic sparsity by turning short texts into normal long texts. Typical methods such as the (Word Network Topic Model) WNTM [22] and the (Distributed Representation-based Expansion) DREx [23] exploit term frequency and word similarities, respectively, to expand the original short texts. Other researchers use location information [24], user information [25] and hashtag information [26] to aggregate short texts. By turning short texts into long texts, the corpus is enlarged and the semantic sparsity is reduced. However, this method may introduce irrelevant terms, which interfere with the semantic modeling of these graphical models. Hence, the qualities of the semantic representations generated are improved to a certain degree but are still limited. Another recent attempt is to generate dense semantic space by modeling biterm pattern [27][28]. These methods represented by (Biterm Topic model) BTM [29] assume that two words appearing in the same sliding window share the same topic. By modeling the biterm pattern, the generated semantic spaces of these methods become denser and the qualities of their semantic representations improve. However, with supposing two words in a biterm share the same topic, biterm pattern brings too strong constraints, which will limit the effectiveness of these methods.

In addition to short texts, social networks also include many other features, such as time, geographic location, user, etc. [30][31], which can all play positive roles in improving the semantic representation capabilities of probabilistic graphical models. Therefore, some researchers utilized multiple features of social media data to constrain how topics are generated. Examples of this include the TOT [32], which models time information in its generative process, LTT [33], which simultaneously models time information, location information and words, and the STM-TwitterLDA [24], which models a topic as a mixture of words, timestamps, pictures and hashtags. Modeling features in the generative process of probabilistic graphical models can certainly improve the quality of generated semantic representations. However, as texts in social networks are

so short, it is not enough to model multiple features without dealing with the problem of semantic sparsity. In this paper, we tackle the above challenges and propose a multi-feature probabilistic graphical model (MFPGM) for social network search tasks that can generate high-quality semantic representations by effectively combining multiple features of social media data and overcome the problem of semantic sparsity. This model solves the semantic sparsity problem in two ways. On one hand, in the MFPGM, we put forward a concept named "special region" and assume that data posted in the same region has the same topic proportion in this paper. With this strategy, we are able to aggregate short texts into long texts. On the other hand, we introduce the biterm pattern proposed by Yan et al. [29] to generate dense semantic space. In addition to solving the semantic sparsity problem, we add the timestamp feature and user feature during the generative process of our MFPGM to improve the quality of the semantic representation. By modeling these attribute features, we not only acquire higher quality semantic representations of text, but also map the text feature, timestamp feature and user feature into the same semantic space. The semantic mapping process of multiple features is shown in Figure 1. Because these features are represented by a unified form, we are able to measure them at the same scale and fulfill multiple search interest. The main contributions of this paper are summarized as follows:

- We combine multiple features of social media data and propose a novel probabilistic graphical model MFPGM that can generate high-quality semantic representations and realize precise social network search.
- The semantic modeling ability of the proposed MFPGM is improved because we simultaneously solve the problems of semantic sparsity and modeling multi-features of social network data.
- The effectiveness of the MFPGM in solving semantic sparsity is also twofold. On one hand, it can turn short texts into long texts by utilizing the proposed concept of the special region. On the other hand, it can generate dense semantic space by introducing the biterm pattern.

- The MFPGM can satisfy a variety of search interests, as multiple features of social media data are represented with a unified form. In particular, it can show the topic distribution profile over time.

The remainder of this paper is organized as follows. In section 2, we introduce related work on probabilistic graphical models for social network search. In section 3, we present the proposed multi-feature probabilistic graphical model MFPGM. In section 4, we present the details of our experiments. Finally, we conclude this paper in section 5.

## 2. Related work

Multiple features of social network data, such as timestamps, locations, users, and short texts, entail both opportunities and challenges to precise social network search. Modeling these features simultaneously can improve the semantic representations of probabilistic graphical models, however, short texts cause semantic sparsity, which presents obstacles in realizing precise search performance for probabilistic graphical models. Many researchers have attempted to deal with the problem of semantic sparsity and make the best use of social network features, so in the following part, we review relevant works dealing with these issues.

### *2.1. Methods for solving semantic sparsity of short text*

Short texts have been extensively studied by researchers. These methods can be broadly divided into two categories. One is to convert short texts into

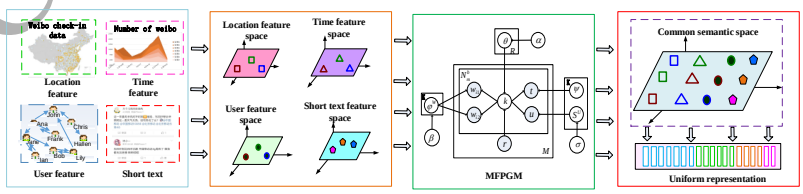



Figure 1: The semantic mapping process of multiple features

long texts by extension or aggregation. The other is to fully exploit relationships between words by proposing new probabilistic graphical models for short texts. Typical methods of expanding short texts include the WNTM [22], the DREx [23] and the CoFE [34]. The WNTM constructs a word co-occurrence network that treats words as nodes, and if two words co-exist in the same document, they will be connected by an edge. This word co-occurrence network allows one word to be represented by a group of words with co-occurrence relationships, thereby, modeling words groups instead of the original short document. The DREx relies on distributed word vector representations, and uses typical word embedding methods such as word2vec and GloVe to designate a vector to represent a word. By defining a distance measurement to find similar words, the DREx finally uses these similar words as candidate words to expand original short texts. The CoFE, similar to the WNTM, also uses words co-occurrence frequencies to expand short texts. However, instead of modeling words groups, it uses words co-occurrence frequencies to generate candidate words, and uses these words to expand the original short document. Typical short text aggregating methods are based on author information [35][25]or location information [24]. These methods assume that a message has only one topic, and messages belonging to the same author or the same location have a common topic proportion, which means that all the messages belonging to the same author or the same location are aggregated into a single long document.

To deal with short text semantic sparsity, other researchers have attempted to fully exploit relationships among words by proposing new probabilistic graphical models. Of the different examples of this kind of methods, we have highlighted the BTM method proposed by Yan et al. [29] and the PTM method proposed by Zuo et al. [36]. The BTM enhances topic learning by modeling word pairs and supposing word pairs co-occurring in a document have the same topic. Through this assumption, it can generate dense semantic space, which alleviates semantic sparsity to a certain degree. The PTM introduces the concept of a pseudo document to aggregate short text against data sparsity. However, the performance of the PTM is influenced by the number of pseudo documents;

if the number is not suitable for the dataset, the performance will be poor. Furthermore, it does not offer a means of identifying the most appropriate number of pseudo documents.

### *2.2. Applications of social network data features in topic learning*

Social network data has multiple features such as user, timestamp, location information, etc, which can enhance the topic-learning process of probabilistic graphical models. Some graphical models are proposed by utilizing the above features. For example, TOT [32] models timestamps in the process of generating topics. Compared to typical graphical models that do not utilize timestamps, the quality of its generated semantic representations is higher. Additionally, with the acquired relationships between timestamps and topics, people can observe the evolution of topics and predict the release time of a text. The LTT [33] uses time and location information to constrain its generative process. It generates topics that are mixtures of text, location, and time, and can perform well in the task of social event monitoring. The STM-TwitterLDA [24] models five features: (time feature, location feature, text feature, hashtag feature, and image feature) simultaneously to enhance the topic-learning process. Its generated topics perform well in detecting and tracking events. Besides enhancing the generative process of graphical models, multiple features of social network data can also be used as auxiliary information to aggregate short texts into long documents [26][37]. Typically, Mehrotra et al. [26] design a variety of pooling schemes based on social network features such as author-wise pooling, hashtag-based pooling, temporal pooling, and so on. Experiments verify all of these pooling schemes can improve the topic modeling performance to some extent. Therefore, in this paper, we choose three features in addition to text (location, user and time) that are helpful in improving the semantic modeling ability of graphical models to generate high-quality semantic representations and realize precise social network search.

## 3. Multi-feature probabilistic graphical model

In this section, we first introduce the definitions and notations that will be used throughout this paper, and then present the MFPGM we proposed for social network semantic search.

### *3.1. Definitions*

To describe this process formally, we give some definitions as follow.

**DEFINITION1. A social media message** can be represented as a 4-tuple $(d, l, t, u)$, where, $d$ denotes the short text, $l$ denotes the location feature, $t$ denotes the time feature, and $u$ denotes the user feature. We choose these four features to represent a social message for the following reasons. 1) Short texts are the most important feature of a social media message, because we can extract rich semantics from words or biterms (i.e. two words co-occurring in the same context); 2) location, time and user are also important features of a message, as different users usually have different interests and a user's interest is always associated with location and change over time.

**DEFINITION2. Special region** $R$ denotes the common character that all messages share. It can be the same author, the same location, or the time slice. When messages have the same character, we have designated them as belonging to the same special region, and sharing a common topic distribution. By introducing special region $R$, we are able to aggregate short text together to alleviate the problem of semantic sparsity. In this paper, for simplicity, we directly used location information for special region $R$.

**DEFINITION3. Topics** in this paper are influenced by many features. A special region has a multinomial topic distribution. A topic is a mixture of a word multinomial distribution, a time beta distribution and a user multinomial distribution.

**DEFINITION4. Uniform representation** means the uniform representation acquired from the common semantic space conducted using the MFPGM. In this paper, each special region, word, user, and timestamp can be represented by a vector, and each element in a vector represents a topic.

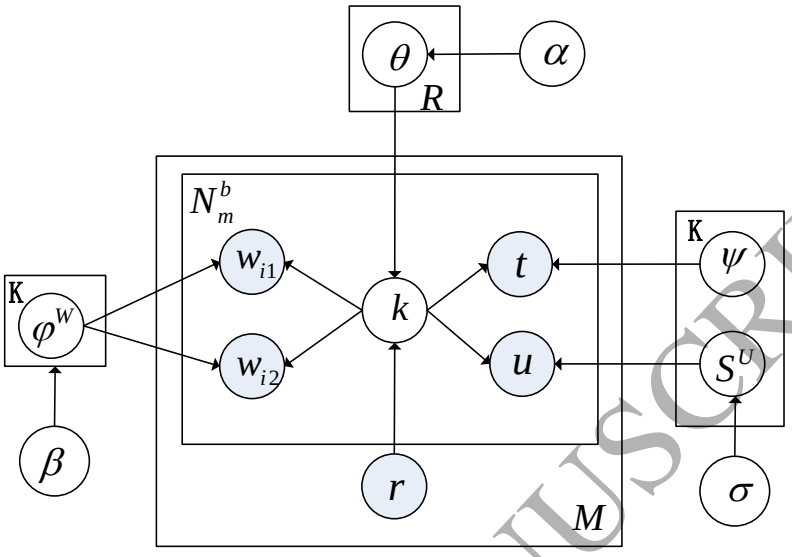



Figure 2: Graphical model of MFPGM

### 3.2. Model description

For realizing precise semantic search, we propose MFPGM to generate high-quality semantic representations of social media messages. Figure 2 shows the graphical model of the MFPGM.

As described in definition 1, social media messages consist of four features: short texts, location, time, and user information. Our MFPGM is designed to take full advantage of these features to generate high-quality semantic representations. Because short texts lead to the problem of semantic sparsity, in the MFPGM, we exploit two strategies to overcome it. On one hand, as described in definition 2 above, we propose the concept of special region $R$, based on which, we can aggregate short texts that have common characteristics into a long text. This strategy is inspired by method proposed in the Ref. [35], which assumes data posted by the same author share the same topic proportions. As social media data is location-specific, for this study we simply choose location information as the common character. On the other hand, to generate dense semantic space, we introduce the biterm pattern proposed in Ref. [29], which

assumes that biterm (two words) co-occurring in the same context have the same topic. The biterm pattern allows us to mine richer relationships of words and generate denser semantic spaces.

To further improve the quality of generated semantic representations, we also introduce the time and user features. Each topic generated by the MFPGM is a mixture of topic-word multinomial distribution, topic-time beta distribution and topic-user multinomial distribution. By mapping multiple features into a topical space, finally, we can finally obtain the uniform representation and measure multiple features with the same form.

Formally, different from typical LDA, which assigns each message to one topic proportion, the MFPGM assumes that all social messages within the same special region $r$ have the same topic proportion. A topic proportion is modeled as a Dirichlet distribution with parameter $\alpha$. Topic-word distributions are modeled as multinomial distributions $\varphi$. As topics change over time, we model topic-time distributions as beta distributions [32][24]. In addition, as a user may focus on many topics and a topic is usually of interest to many different users, we model topic-user distributions as multinomial distributions $S$. As a result, in the proposed MFPGM, topic proportions are mixtures of words, users and time.

Table 1 summarizes the notations and corresponding descriptions of our proposed MFPGM.

### *3.3. Generative process of MFPGM*

This generative process of MFPGM can be described more formally by defining some variables in this paper. Assuming we have $K$ topics and $R$ spatiotemporal regions, we can exploit a matrix $\Theta$ of size $K \times R$ to parameterize the multinomial distribution over topics for each special region. Every element $\theta_{kr}$ stands for the probability of topic $k$ and it will be assigned to biterms in the special region $r$, thus $\sum_{k=1}^{K} \theta_{kr} = 1$. For convenient, we use $\theta_r$ denotes the $r-th$ column of the matrix. Similarly, a matrix $\Phi$ is used to denote word-topic multinomial distributions. Each element $\varphi_{wk}$ denotes the probabilities of word

Table 1: Notations and corresponding descriptions

| Notations | Description |
|---|---|
| $r$ | Special region |
| $k, w, b$ | Topic, word and biterm |
| $b_i, w_{i1}, w_{i2}$ | the i-th biterm and its two words |
| $t_i, u_i$ | Timestamp and user of the i-th biterm |
| $N_m^b$ | Numbers of biterms in m-th message |
| $K, W$ | Number of topics and words |
| $R, M$ | Number of special regions and messages |
| $\alpha, \beta, \sigma$ | Dirichlet priors for $\theta, \varphi, S$ |
| $\theta_r$ | Multinomial distributions for topics in the r-th special region |
| $\varphi^W, S^U$ | Multinomial distributions for words and users |
| $\psi$ | Beta distribution for topic-time |

$w$ generated from topic $k$. A matrix $S$ is used to denote user-topic multinomial distribution. Each element $s_{uk}$ denotes the probabilities of user $u$ generated from topic $k$. we also parameterize timestamp-topic beta distributions using function $\Psi$, the value of $\psi_{tk}$ denotes the probabilities that timestamps $t$ generated from topic $k$, and the beta function $\psi_k$ denotes topic $k$ changing over time. As a generative model, the generative process of MFPGM can be formally described as follows:

1. For the messages belonging to the same special region $r$, we draw their topics proportion as Dirichlet distribution with parameter $\alpha$: $\theta_r \sim Dirichlet(\alpha)$.
2. For each topic $k$, we respectively draw a Dirichlet distribution with parameter $\beta$: $\varphi_k \sim Dirichlet(\beta)$ denoting word distribution, a Dirichlet distribution with parameter $\sigma$ : $s_k \sim Dirichlet(\sigma)$ denoting user distribution, and a Beta distribution $\psi_k$.
3. For each message $m$, we can know its special region $r_m$, so we get it topic distribution $\theta_{r_m}$.

   For its each biterm (word pairs) $b_i$:

   a) Choose a topic from the region-topic multinomial distribution: $k \sim Multi(\theta_{r_m})$;

b) Choose two words from the topic-word multinomial distribution: $\mathrm{w}_{i1}, w_{i2} \sim \varphi_k$;

c) Choose a user from the topic-user multinomial distribution: $u_i \sim S_k$;

d) Choose a timestamp from the topic-time beta distribution: $t_i \sim \psi_k$.

### *3.4. Model inference*

There are one latent variable topic $k$ and four parameters $\{\theta, \varphi, \psi, S\}$ in MFPGM. Like other topic model, we use a collapsed Gibbs sampling algorithm to infer parameters. Sampling formula for the topic of each biterm is as follow:

$$p\left(k_i|K_{\neg i}, B, U, T, R\right) \propto \frac{p\left(K, B, U, T, R|\Theta\right)}{p\left(K_{\neg i}, B_{\neg i}, U_{\neg i}, T_{\neg i}, R_{\neg i}|\Theta\right)} \tag{1}$$

In formula (1), $\neg i$ denotes the exception of the element $i$ , and $\Theta$ represents all the parameters. We first calculate their joint probability to derive the sampling formula, and their joint probability is shown as below:

$$\begin{aligned} p\left(K, B, U, T, R|\Theta\right) &= p\left(K|\alpha, R\right) p\left(B|K, \beta\right) p\left(U|K, \sigma\right) p\left(T|K, \psi\right) p\left(R\right) \\ &= \prod_{m=1}^{M} \frac{\Delta(\overrightarrow{n_{r_m}^{K}}+\alpha)}{\Delta(\alpha)} \prod_{k=1}^{K} \frac{\Delta(n_k^{N^b}+\beta)}{\Delta(\beta)} \prod_{k=1}^{K} \frac{\Delta(\overrightarrow{n_k^{U}}+\sigma)}{\Delta(\sigma)} \prod_{i=1}^{N^b} p(t_i|\psi_{k_i}) \end{aligned} \tag{2}$$

In Eq.(2), $\overrightarrow{n_{r_m}^{K}}, \overrightarrow{n_k^{N^b}}, \overrightarrow{n_k^{U}}$ respectively represents a $K$-dimensional, $N^b$-dimensional and $U$-dimensional vector. Each value of the vectors respectively denotes the number of topic $k$ occurring in document $m$, the number of times that is assigned to topic $k$ for each biterm, and the number of times that is assigned to topic $k$ for each user.

Combining formula (1) and formula (2), sampling formula is derived in formula (3).

$$\begin{aligned} p\left(k_i = k|K_{\neg i}, B, U, T, R\right) \propto & \frac{n_{k,\neg i}+\alpha}{\sum_{k=1}^{K} n_{k,\neg i}+K\alpha} \times \frac{(n_{k,\neg i}^{w_{i1}}+\beta)(n_{k,\neg i}^{w_{i2}}+\beta)}{(\sum_{w=1}^{W} n_{k,\neg i}^{w}+W\beta)\cdot(\sum_{w=1}^{W} n_{k,\neg i}^{w}+1+W\beta)} \\ & \times \frac{(n_{k,\neg i}^{u_i}+\sigma)}{(\sum_{u=1}^{U} n_{k,\neg i}^{u}+U\sigma)} \times \frac{(1-t_i)^{\psi_{k1}-1} t_i^{\psi_{k2}-1}}{B(\psi_{k1},\psi_{k2})} \end{aligned} \tag{3}$$

We summarize the Gibbs sampling procedure of MFPGM in Algorithm 1. Iterative executions of above sampling rules until a steady state results allow us to estimate parameters by the following equations:

$$\theta_{k,r} = \frac{n_{k,m} + \alpha}{\sum_{k=1}^{K} n_{k,m} + K\alpha}, \varphi_{k,w} = \frac{n_k^w + \beta}{\sum_{w=1}^{W} n_k^w + W\beta} \tag{4}$$

$$\psi_{k1} = \overline{t_k}\left(\frac{\overline{t_k}(1-\overline{t_k})}{r_k^2} - 1\right), \psi_{k2} = \left(1-\overline{t_k}\right)\left(\frac{\overline{t_k}(1-\overline{t_k})}{r_k^2} - 1\right) \tag{5}$$

$$s_{k,u} = \frac{n_k^u + \sigma}{\sum_{u=1}^{U} n_k^u + U\sigma} \tag{6}$$

We utilize the method of moments to estimate $\psi_{k1}, \psi_{k2}$ [24] [32]. In formula (5), $\overline{t_k}, r_k^2$ are respectively used to denote sample mean and sample bias variance of timestamps that belong to topic k.

### *3.5. Complexity analysis*

We show computational complexity of MFPGM by analyzing its time complexity and stored variables. We also compare them with LDA. In LDA and the proposed MFPGM, the most time-consuming procedure is sampling operations. We use $K$ denotes the topic number, $l$ denotes the average length of messages and $M$ represents number of messages. Through $M \times l$, we can calculate number of total words $N^w$ in corpus. In MFPGM, we use a sliding window to generate biterm, the length of sliding window is recorded as $l_w$, which can be assigned as the average length of social network messages. The number of total biterms $N^{\mathrm{b}}$ is equal to $M\frac{l_w(l_w-1)}{2}$. For LDA, we need to sample a topic for each word, then its time complexity is $O(KN^w)$. For MFPGM we need to sample a topic for each biterm, then its time complexity is $O(KN^b)$. As number of total biterms is $(l_w-1)/2$ times larger than the number of total words, the time complexity of MFPGM is $(l_w-1)/2$ times larger than time complexity of LDA. Fortunately, the length of a social network message is usually short, so the time complexity of MFPGM is still comparable with that of LDA.

**Algorithm 1** Gibbs sampling algorithm for MFPGM

**Input:**

Topic number $K$, parameters $\alpha, \beta, \sigma$, short text, region No., normalized timestamp and user ID of messages

**Output:**

$\Phi, \Psi, \theta, S$

Initialize topic assignment randomly for short text;
**for** $iter = 1$ to $N$ **do**
  **for** $m = 1$ to $M$ **do**
    **for** each $bitermb_i = (w_{i,1}, w_{i,2}) \in B_m$ **do**
      Draw topic $k$ from Eq.(3)
      Update $n_{r,k}, n_k^u, n_k^{w_{i1}}, n_k^{w_{i2}}$
    **end for**
  **end for**
  **for** $k = 1$ to $K$ **do**
    Update $\psi_k$ in Eq.(5)
  **end for**
**end for**
Compute the posterior estimates of $\theta$ and $\varphi$ in Eq.(4), $\psi$ in Eq.(5)and $s$ in Eq.(6)

Next, we illustrate space complexity by analyzing stored variables. As for LDA, we need to store document-topic matrix, topic-word matrix, and topic assignments of total words in the corpus, so space complexity of LDA is $MK + KW + Ml$. As for MFPGM, we need to store region-topic matrix, topic-word matrix, topic-user matrix, and parameters for topic-time distribution. Besides, we need to store total biterms in the corpus, and each biterm includes all its relevant information (topic assignment, user, region and timestamp). So the space complexity is $RK + KW + KU + K + N^b$ and it can be simplified to $RK + KW + KU + K + \frac{(l-1)}{2}Ml$. The number of special region is very small, so $RK << MK$ and space complexity of MFPGM is similar with that of LDA.

We list time complexity and stored variables of the two methods in Table 2.

Table 2: Time complexity and variables need to be maintained of LDA and MFPGM

| Method | Time complexity | Number of stored variables |
|---|---|---|
| LDA | $O(KN^w)$ | $MK + KW + Ml$ |
| MFPGM | $O(KN^b)$ | $RK + KW + KU + K + \frac{(l-1)}{2}Ml$ |

## 4. Experiments

### *4.1. Experiment setting*

#### *4.1.1. Description of datasets*

In this paper, we crawled data from Sina Weibo, a very popular social network platform in China. We obtained our raw data set mainly through two steps. In the first step, we utilize keywords such as "traffic accident", "domestic violence" and "rainstorm" to collect related microblogs. Each acquired message contained the following information: text, user, timestamp and location (i.e. the province where the user was located). In the second step, we applied the user and timestamp information of each message to enlarge the dataset. Specifically, for each message, we first delineated a time range based on its timestamp, and then filtered out all the microblogs posted by the user within this time range. To verify the data sensitivity of the MFPGM, we also constructed three sub-datasets from the entire dataset. Collection 1 was concerned with the traffic accidents, and its data is closely related to regional information. Collection 2 was about domestic violence, and its data is closely related to different users. Collection 3 was devoted to rainstorms and its data was closely to time. By conducting experiments on these datasets, we wanted to verify that in which cases these features play their parts as much as possible. As most of the raw data is in Chinese, we first preprocessed the crawled data through the following operations: 1) segmenting sentence into words; 2) removing stop words and low-frequency words; 3) removing duplicate micro-blogs and short micro-blogs.

Information of the whole dataset and three data subsets after preprocessing is presented in Table 3.

Table 3: Datasets

| Data sets | Number of Messages | Vocabulary Size | Time Period | Number of Regions | Number of Users |
|---|---|---|---|---|---|
| The Whole Dataset | 112053 | 77418 | 2011.05.20-2013.10.23 | 27 | 20734 |
| Collection 1 | 20186 | 8026 | 2012.01.01-2012.12.31 | 18 | 4015 |
| Collection 2 | 18152 | 7624 | 2011.08.01-2013.04.31 | 12 | 1963 |
| Collection 3 | 23551 | 8723 | 2012.05.01-2012.10.31 | 7 | 6897 |

*4.1.2. Evaluation metrics*

We evaluated the proposed MFPGM by two measures. On one hand, as the MFPGM is a topic model, we evaluated the quality of its generated topics. On the other hand, as uniform representation acquired through MFPGM can be used to calculate similarities between the query and the searched documents we measured the MFPGM by its performance on search task. In this paper, we used two most commonly used evaluation indicators: NPMI [38] and UMass coherence [39] to assess the quality of generated topics. The higher the values of these two evaluations, the better the performance of methods. In addition, we crawled other data from Sina Weibo as external evaluation dataset, and the size of this dataset is 5M. PMI is widely used to evaluate topic quality of topic models, and NPMI is the normalized version of PMI. When given a topic and its top N most probable words, NPMI-score can be calculated by the following equation:

$$NPMI_{score}(t, N) = mean\{NPMI(w_i, w_j), i, j \in 1, ..., N, i \neq j\} \quad (7)$$

$$NPMI(w_i, w_j) = (ln\frac{P(w_i, w_j) + \zeta}{P(w_i)P(w_j)})/(-ln(P(w_i, w_j) + \zeta) \quad (8)$$

Where, $\zeta$ is used to avoid logarithm of zero, $P(w_i)$ and $P(w_i, w_j)$ respectively represent the probability of word $w_i$ and word pair $w_i, w_j$ occurring in the external datasets.The score of UMass coherence can be calculated by the following equation:

$$C(k, N^{(k)}) = \sum_{n=2}^{N} \sum_{L=1}^{n} \log \frac{D(w_n^{(k)}, w_l^k) + \tau}{D(w_l^k)} \quad (9)$$

In Eq.(9), $W^{(K)} = (w_1^{(k)}, ..., w_N^{(k)})$ represents the list of top $N$ most probable words of topic $k$. $D(w_l^k)$ and $D(w_n^{(k)}, w_l^k)$ denote the document frequency that a single word occurring and two words co-occurring respectively.$\tau$ is used to avoid zero value of log.

We used the uniform representation on search tasks, and adopted two widely used standard metrics, NDCG (Normalized Discounted Cumulative Gain) [40]and MAP (Mean Average Precision) [41] to evaluate the search performance. MAP is the mean value of AP and the average precision(AP) can be calculated as follows.

$$AP = \frac{1}{R'} \sum_{r=1}^{R} prec(r)\delta(r) \quad (10)$$

In Eq.(10), $R$ represents the number of all the returned search results, $R^{'}$ represents the number of the returned search results related to the query, $prec(r)$ represents the precision of the $r - th$ returned search results, and $\delta(r)$ is an indicator function, where 1 indicates the result has correlation and 0 indicates the related has no correlation.

The NDCG is defined as follows:

$$\mathrm{NDCG\,(n)} = Z_n \sum_{j=1}^{n} \left(2^{r(j)} - 1\right) \Big/ \log\left(1 + j\right) \quad (11)$$

where, $Z(n)$ is a normalization factor, and $r(j)$ is the graded relevance of the $j - th$ search result.

#### *4.1.3. Baseline methods*

The MFPGM is a probabilistic graphical model that models multiple features of social network data and can deal with the problem of semantic sparsity caused by short text. Therefore, in this paper, to verify the superiority of our MFPGM, we compared it with other state-of art topic models that are also designed to generate high-quality topics by alleviating semantic sparsity or modeling attribute features of social network data. Baseline methods used in this paper are listed as follows:

**LDA** [42]: A typical topic model that can achieve good performance in long text. We implemented it by using collapsed Gibbs sampling. [1]

**BTM** [27]: A topic model that proposes the biterm pattern, which assumes that two words in the same context have the same topic. Its generated topic quality is improved and semantic space becomes denser by adopting biterm pattern. We implemented it using the code provided by its authors. [2]

**WNTM** [22]: A topic model in which a word co-occurrence graph is constructed to expand one word into a pseudo document. Although it utilizes only the relationships between words without exploiting additional features, as the original corpus is expanded, it can alleviate the problem of semantic sparsity to a certain degree.The code used to implement this method was provided by its authors. [3]

**PTM** [36]: In this model, pseudo-documents are generated to solve the semantic sparsity problem of short texts. It is specifically designed to deal with short texts without using any additional attribute features. We used the code provided by its authors to implement it. [4]

**UCT** [37]: This topic model introduced both the user information and biterm pattern to obtain the semantics of short text.

[1] http://jgibblda.sourceforge.net/
[2] https://github.com/xiaohuiyan/BTM
[3] http://ipv6.nlsde.buaa.edu.cn/zuoyuan/
[4] http://ipv6.nlsde.buaa.edu.cn/zuoyuan/

**TOT** [32]: This topic model utilizes the time information to enhance the leaning process of topics by modeling the topic-time distribution as beta distributions.

*4.1.4. Parameters tuning*

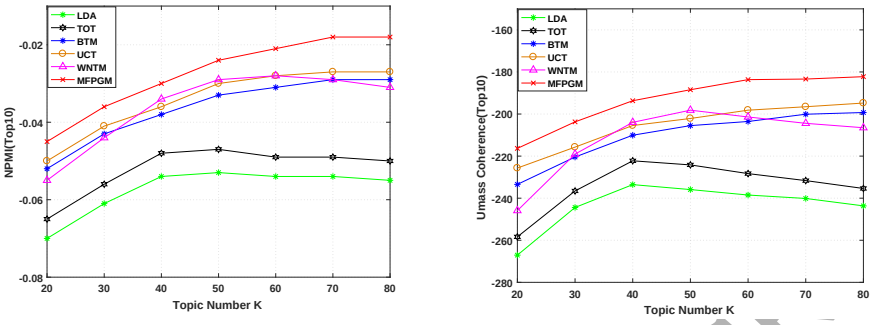
There are some common parameters shared by the MFPGM and all baseline methods: Dirichlet priori parameters $\alpha$ and $\beta$, respectively for topic proportions and topic-word distributions, the number of sampling iteration and the number of topics $K$. These common parameters are all empirically set to achieve their best performance, and their values are listed in Table 4.

Table 4: The main common parameters setting for MFPGM and the baseline methods

| | MFPGM | WNTM | BTM | PTM | LDA | TOT | UCT |
|---|---|---|---|---|---|---|---|
| $\alpha$ | $50/K$ | $50/K$ | $50/K$ | 0.1 | 0.1 | $50/K$ | $50/K$ |
| $\beta$ | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 | 0.01 |
| $K$ | 20 to 80 | 20 to 80 | 20 to 80 | 20 | 20 to 80 | 20 to 80 | 20 to 80 |
| $N_{iter}$ | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 |

The number of topics $K$ is the most important parameter that we need to mention. To verify the superiority of our MFPGM, we ranged the number of topics from 20 to 80. However, because the PTM is too time-consuming, for convenience, we set its number of topics as 20. We extracted the top 10 probable words of each topic and used NPMI and Umass scores to measure the qualities of the topics generated by each method. The performances of algorithms changed as the topic number grew, and the results are shown in Figure 3.

Figure 3 illustrates that when a topic number is too large, the performance of the LDA and the TOT will become worse, which occurs because too large a topic number exacerbates the problem of semantic sparsity. As the BTM, UCT and WNTM all alleviate semantic sparsity to some extent, they performed better than the LDA and TOT. The superiority and stability of our MFPGM




(a) NPMI score (b) UMass values

Figure 3: Topic quality vs topic number K

are also reflected in Figure 3, as it achieves the best performance compared to the baseline methods. Another important observation revealed by Figure 3 is that when the number of topics $K$ is set to 50, it is fairly to compare the performances of algorithms. Therefore, in the following experiments, $K$ is set to 50.

Aside from these common parameters, the PTM and the MFPGM both have their special parameters. For MFPGM, it has a special Dirichlet priori parameter for topic-user distribution, and in this paper, we set it as 0.01. For the PTM, we set the priori parameters to the same valuses as those set by its authors [22]. Because the performance of the PTM improves as the number of pseudo-documents increased, considering the dataset size that we used in this paper, we set the number of pseudo documents at 1,000 to ensure the best performance, which was very time-consuming and far beyond the special region number of our dataset.

### *4.2. Topic quality*

We measured the proposed MFPGM topic model by evaluating its topic quality in this paper. First, we verified the effectiveness of modeling multiple features and then compared the MFPGM with other baseline methods.

*4.2.1. Influence of multi-features*

In this section, we will explain the influences of modeling multiple features on the topic quality. The basic assumption of the proposed model is that two words in a biterm have the same topic. We used MFPGM(B) to represent the simplified version of our MFGPM, which just modeled biterms. MFPGM(U) was another simplified version of our MFGPM, which just modeled user feature besides of biterms. Similarly, MFPGM(R) and MFPGM(T) respectively modeled only region feature and time feature besides of biterm. The number of topics for all the methods was set as 50, and the iteration number was set as 2000. We conducted experiments on the three collections and the scores of NPMI and UMass coherence are listed respectively in Table 5 and Table 6.

From Table 5 and Table 6, we can find that the following observations: (1) MFPGM(T), MFPGM(U) and MFPGM(R) all performed better than MFPGM(B), verifying that modeling each feature can effectively improve topic quality. (2) Among MFPGM(T), MFPGM(U) and MFPGM(R), MFPGM(R) achieved the best in collection 1, MFPGM(U) achieved the best in collection 2 and MFPGM(T) achieved the best in the collection 3. These phenomena are caused by the following reasons. First, the data drawn from collection 1, which was mainly about traffic accidents, was closely correlated to the special region. In this case, regional information played a positive role, so MFPGM(R) performs better than the other two methods. Second, collection 2 was mainly about domestic violence. As only a small percentage of users were concerned about this kind of messages, collection 2 was closely correlated to users. So MFPGM(U) performed better than the other two methods in this collection. Third, collection 3 mainly include data about rainstorms, which usually occurs over the summer, so in this collection, the time feature played a very important role and MFPGM(T) performed better than the other two methods. (3) The proposed MFPGM performed best in all three of the collections, because the multiple features were all helpful in generating high-quality topics, and when we modeled all these features, they demonstrated the best performance. All

Table 5: NPMI scores of top N probable words in each data subset (scores in bold=best results while scores in bold and italics=second-best results)

| Datasets | Methods | Top5 | Top10 | Top15 | Top20 |
|---|---|---|---|---|---|
| Collection 1 | MFPGM(B) | -0.042 | -0.045 | -0.048 | -0.051 |
| | MFPGM(T) | -0.039 | -0.043 | -0.047 | -0.049 |
| | MFPGM(U) | -0.039 | -0.042 | -0.045 | -0.046 |
| | MFPGM(R) | ***-0.037*** | ***-0.039*** | ***-0.043*** | ***-0.044*** |
| | MFPGM | **-0.034** | **-0.036** | **-0.04** | **-0.042** |
| Collection 2 | MFPGM(B) | -0.038 | -0.044 | -0.045 | -0.049 |
| | MFPGM(T) | -0.039 | -0.043 | -0.047 | -0.047 |
| | MFPGM(U) | ***-0.034*** | ***-0.038*** | ***-0.039*** | ***-0.042*** |
| | MFPGM(R) | -0.037 | -0.045 | -0.047 | -0.048 |
| | MFPGM | **-0.032** | **-0.033** | **-0.039** | **-0.042** |
| Collection 3 | MFPGM(B) | -0.029 | -0.033 | -0.036 | -0.039 |
| | MFPGM(T) | ***-0.026*** | ***-0.029*** | ***-0.031*** | ***-0.034*** |
| | MFPGM(U) | -0.030 | -0.032 | -0.037 | -0.041 |
| | MFPGM(R) | -0.027 | -0.030 | -0.032 | -0.033 |
| | MFPGM | **-0.023** | **-0.024** | **-0.039** | **-0.031** |

Table 6: UMass coherence scores of top N probable words in each data subset (scores in bold=best results while scores in bold and italics=second-best results

| Datasets | Methods | Top5 | Top10 | Top15 | Top20 |
|---|---|---|---|---|---|
| Collection 1 | MFPGM(B) | -37.27 | -240.38 | -696.91 | -1383.62 |
| | MFPGM(T) | -34.21 | -227.35 | -667.49 | -1347.74 |
| | MFPGM(U) | -36.15 | -235.52 | -641.57 | -1323.42 |
| | MFPGM(R) | ***-32.45*** | ***-201.42*** | ***-627.02*** | ***-1308.33*** |
| | MFPGM | **-29.05** | **-189.54** | **-601.83** | **-1286.62** |
| Collection 2 | MFPGM(B) | -36.31 | -231.43 | -692.48 | -1369.47 |
| | MFPGM(T) | -37.16 | -236.87 | -695.53 | -1354.65 |
| | MFPGM(U) | ***-28.26*** | ***-227.64*** | ***-634.42*** | ***-1296.45*** |
| | MFPGM(R) | -38.45 | -229.35 | -676.36 | -1343.72 |
| | MFPGM | **-26.22** | **-211.94** | **-594.81** | **-1273.43** |
| Collection 3 | MFPGM(B) | -30.13 | -210.46 | -630.93 | -1298.77 |
| | MFPGM(T) | ***-27.62*** | ***-200.13*** | ***-606.74*** | ***-1208.66*** |
| | MFPGM(U) | -29.81 | -204.31 | -616.36 | -1223.57 |
| | MFPGM(R) | -28.25 | -201.46 | -611.82 | -1215.35 |
| | MFPGM | **-25.72** | **-192.16** | **-591.73** | **-1203.66** |

of these experimental results reflect that, although the influence of each feature is different according to the characteristics of the dataset used, they are all positive, and modeling multiple features is a very helpful strategy in topic learning.

Table 7: NPMI scores of top N probable words on whole corpus (scores in bold=best results while scores in bold and italics=second-best results)

| Methods | Top 5 | Top 10 | Top 15 | Top 20 |
|---|---|---|---|---|
| MFPGM(B) | -0.036 | -0.043 | -0.046 | -0.048 |
| MFPGM(T) | -0.037 | -0.041 | -0.044 | -0.045 |
| MFPGM(U) | -0.035 | -0.042 | -0.045 | -0.049 |
| MFPGM(R) | ***-0.033*** | ***-0.039*** | ***-0.041*** | ***-0.042*** |
| MFPGM | **-0.029** | **-0.03** | **-0.036** | **-0.039** |

We also conducted experiments on the whole dataset, the results of which are presented in Table 7 and 8. We found that, although simplified MFPGM methods such as MFPGM(T), MFPGM(U) and MFPGM(R) are all data-sensitiv, the NPMI and UMass coherence scores of these three methods were all higher than that of BTM. All the experimental results showed that modeling multiple features was meaningful and effective.

Table 8: UMass coherence scores on whole corpus (scores in bold=best results while scores in bold and italics=second-best results)

| Methods | Top 5 | Top 10 | Top 15 | Top 20 |
|---|---|---|---|---|
| MFPGM(B) | -32.31 | -220.43 | -682.93 | -1341.06 |
| MFPGM(T) | -30.84 | -191.91 | -657.48 | -1337.64 |
| MFPGM(U) | -30.11 | -213.56 | -663.48 | -1352.31 |
| MFPGM(R) | ***-29.98*** | ***-208.11*** | ***-624.34*** | ***-1324.94*** |
| MFPGM | **-29.36** | **-203.72** | **-613.02** | **-1297.48** |

*4.2.2. Comparisons with baseline methods*

In this section we will compare the proposed MFPGM method with baseline methods on the whole corpus and evaluate the generated topic quality in two ways. First, to verify the superiority of the MFPGM and make the experiment results more interpretable, we identify some common topics and their Top-N words generated by the MFPGM and the BTM. The experiment results are presented in Table 9.

Table 9: The top 12 Words in some common topics of BTM and MFPGM

| Topics ID | Method | The top 12 words in a topic | | | |
|---|---|---|---|---|---|
| Topic 1 | BTM | /accident<br>/month<br>/day | / scene<br>/ expressway<br>/ vehicle | /road<br>/dead<br>/car | / a car<br>/woman<br>/ experience |
| | MFPGM | /accident<br>/ traffic police<br>/ scene | /driver<br>/drive<br>/bus | / bump<br>/dead<br>/car | / vehicle<br>/ damage<br>/ expressway |
| Topic 2 | BTM | /plant man<br>/doctor<br>/lie | /awaken<br>/hope<br>/ uncertain | / accident<br>/ miracle<br>/read | /become<br>/ say<br>/speak |
| | MFPGM | /doctor<br>/hospital<br>/ kinsfolk | /accident<br>/ operation<br>/ miracle | /plant man<br>/hope<br>/lie | /awaken<br>/ give up<br>/uncertain |
| Topic 3 | BTM | / the left roll<br>/dad<br>/ Li family | / the right roll<br>a surname<br>/dead | /the top scroll<br>Li gang<br>/ Domestic violence | /shuangjiang<br>/ hate<br>/ Submachine gun |
| | MFPGM | / the left roll<br>/father<br>/ Li family | / the right roll<br>/Mr li<br>/hate | /the top scroll<br>/Li gang<br>/Domestic violence | /shuangjiang<br>/Niubi<br>/ Li yang |

In Table 9, we list three common topics for MFPGM and BTM. Topic 1 is the description of a traffic accident, topic 2 describes the treatment of those injured in accidents when they are taken to the hospital, and topic 3 relates to some events involving key persons with the surname Li. However, the BTM included a few less relevant words, such as "woman," "month," and "day," in topic 1, "become" and "read" in topic 2, and "submachine gun" in topic 3.

We also compared the MFPGM with other state-of-art methods, and conducted experiments on the whole corpus. Their NPMI scores are presented in Table 10 and their UMass scores in Table 11 below.

These two tables reveal the following observations. First, the TOT per-

Table 10: NPMI scores of top N probable words on whole corpus

| Methods | Top5 | Top10 | Top15 | Top20 |
|---|---|---|---|---|
| LDA | -0.041 | -0.061 | -0.062 | -0.064 |
| TOT | -0.038 | -0.056 | -0.058 | -0.059 |
| BTM | -0.036 | -0.043 | -0.046 | -0.048 |
| UCT | -0.034 | -0.037 | -0.041 | -0.043 |
| WNTM | -0.031 | -0.034 | -0.048 | -0.049 |
| PTM | -0.032 | -0.036 | -0.039 | -0.043 |
| MFPGM | -0.029 | -0.03 | -0.036 | -0.039 |

Table 11: UMass coherence scores of top N probable words on whole corpus

| Methods | Top 5 | Top 10 | Top 15 | Top 20 |
|---|---|---|---|---|
| LDA | -39.18 | -244.41 | -733.47 | -1369.66 |
| TOT | -36.19 | -235.76 | -713.38 | -1356.41 |
| BTM | -32.31 | -220.43 | -682.93 | -1341.06 |
| UCT | -31.92 | -219.56 | -679.65 | -1338.93 |
| WNTM | -31.23 | -219.04 | -685.76 | -1359.64 |
| PTM | -32.45 | -221.82 | -659.63 | -1321.09 |
| MFPGM | -29.36 | -203.72 | -613.02 | -1297.48 |

formed better than the LDA using both NPMI and UMass scores, because the TOT modeled time in its topic generative process, which improved the quality of the generated topics. Second, all the methods designed for short texts (the BTM, the UCT, the WNTM, and the PTM) performed better than the LDA and the TOT, because they all alleviated the problem of semantic sparsity to a certain degree. Third, of the BTM, the WNTM, and the PTM, only the BTM performed stably, verifying the effectiveness of modeling biterm patterns. Fourth, because the UCT exploits biterm patterns and user information at the same time, it performed better than the BTM using both NPMI and UMass scores, verifying that user information is useful in mining the semantics of social media data. Fifth, of all these methods, the proposed MFPGM performed the best because it solved the problem of semantic sparsity by aggregating short texts in the same special regions and introducing a biterm pattern; modeling multiple features also enhanced the topic learning process.

### *4.3. Search task*

Through the proposed MFPGM, we were able to obtain uniform representations for multiple features. A text, either a word or a sentence, can be represented as a vector, allowing us to observe how a topic evolves over time, and to identify all the hot topics relating to a specific timestamp. Each element in the uniform vector representation denoted a topic, and all of these features were mapped into a topic semantic space. In this section we discuss the many search experiments we conducted on the whole dataset to verify the quality of the uniform vector representation. We explain the similarity calculation method in section 4.3.1, the text search results in section 4.3.2, and the topic-time distribution in section 4.3.3.

#### *4.3.1. Similarity calculation method*

In this paper, multiple features are not directly used to sort search results but they are conducive to increasing the quality of uniform representation. To verify the semantic analysis and modeling ability of the MFPGM, the uniform

representation is used for search task.

The entered query in this paper is a short sentence composed by a couple of word ,where,$w$ represents word in a sentences, and $n$ represents the total number of words. The messages to be searched has the following information $(d, r, t, u)$, where $d$ denotes the short text, $r$ denotes the region information, $t$ denotes the timestamp, and $u$ denotes the user information. Through the parameters obtained by the MFPGM, we can represent each message as a K-dimensional vector. The vector denotes the topic proportion of a message, and its $k-th$ element represents the probability that the message belongs to the $k-th$ topic. The value of each element is affected by a variety of parameters. In this paper, we assume that the effects of each factor are independent of each other. So the probability that a message m belongs to the $k-th$ topic can be calculated by Eq.(12).

$$\begin{aligned} & P(z=k|m) \\ & = \sum_{i=1}^{B_m} P(z=k, b_{m,i}|d_m) P(z=k|u_m) P(z=k|t_m) \\ & = \sum_{i=1}^{B_m} P(z=k|b_{m,i}, d_m) P(b_{m,i}|d_m) S_{k,u_m} \psi_{k,t_m} \\ & = \sum_{i=1}^{B_m} \frac{\theta_{rk} \varphi_{k,w_{m,i,1}} \varphi_{k,w_{m,i,2}}}{\sum_{k=1}^{K} \theta_{rk} \varphi_{k,w_{m,i,1}} \varphi_{k,w_{m,i,2}}} \frac{n^{b_{m,i}}}{N_m^b} S_{k,u_m} \psi_{k,t_m} \end{aligned} \tag{12}$$

After each message is represented as a vector, similarity between it and the query text can be measured by the probability that query text will be generated at the topic proportion of this message. This probability can be treated as the similarity score, and it can be calculated by Eq. (13).

$$Score_i = p\left(Q/\theta_i\right) = \prod_{n=1}^{N} p\left(w_n/\theta_i\right) = \prod_{n=1}^{N} \sum_{k=1}^{K} \left(\varphi_{k,w_n} \theta_{i,k}\right) \tag{13}$$

*4.3.2. Search performance*

To fairly evaluate the search performance, we set the size of the top returned search results $T$ to 5, 10, 20, and 30, and computed the respective MAP and NDCG scores. We invited several members of other projects in our laboratory

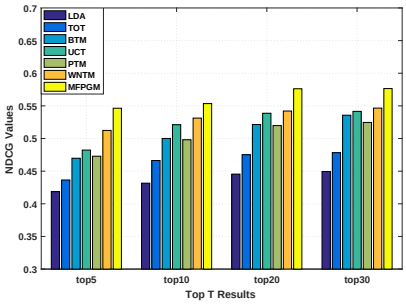



Figure 4: NDCG Values

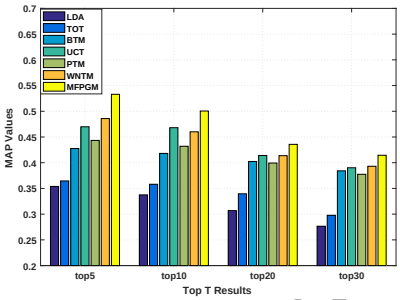



Figure 5: MAP Values

to volunteer in evaluating the relevance between each query and the top 30 returned results in the search list, and used the resulting average evaluation values as the relevance scores. The MAP scores and the NDCG scores of all the methods in the whole dataset are shown in Figures 4 and 5 respectively.

These experiments yielded the following observations. First, the NDCG values and MAP values of the TOT were both better than that of the LDA, which verifies that the modeling time feature is useful in constraining the topic learning process of topic models. Second, of all the methods designed for short texts, the WNTM, the BTM, the UCT, the PTM, and our proposed MFPGM performed better than the LDA or the TOT. Although the LDA and the TOT perform very well using long texts, when they are used on social network short texts they are affected by semantic sparsity. Third, the performances of the WNTM and the PTM both verified that combining short texts into a long text is helpful in resolving the semantic sparsity problem: the WNTM extends short texts into a long text by constructing a word co-occurring graph, and the PTM uses a pseudo-document to aggregate short texts together. Fourth, the performance of the BTM verified that supposing a biterm shares a common topic can enhance the effectiveness of topic learning. Fifth, we observed that the UCT performed better than the BTM with regard to both NDCG and MAP values, verifying that user information is conducive to mining the semantics of social media data. Sixth, it is not surprising that the proposed MFPGM demonstrated the

best performance. This algorithm works so well for the following reasons: it aggregates short texts into long texts using region information; it enhances the effectiveness of topic learning by assuming that a biterm shares a common topic; and modeling multiple features is helpful in generating high-quality topics.

#### *4.3.3. Displaying of topic-time distributions*

Because the MFPGM models the topic-time distribution as a beta distribution, after the training process we were able to obtain each topic-time distribution. These distributions allowed us to identify the temporal variation rule of topics that we were interested in.

We present the topic-time beta distributions of the top five topics in Figure 6 below. The X axis represents the normalized times, and Y axis represents the density of beta distribution. If a topic has a high value on the Y axis at a certain moment, the heat of that topic was high. Figure 6 also illustrates that topics 1-4 were fairly evenly distributed over time, while topic 5 was prevalent during the normalized time period 0.4-0.6 and reached its highest value at normalized time 0.5 (corresponding to a real date in August 2012). This is due to the fact that topics 1-4 were roughly uniformly distributed over time, while topic 5 related to an emergency (" / the director smiled in the tragic accident scene") that occurred on August 26, 2012. These beta distributions effectively show the distribution of topics over time, which is helpful in analyzing hot events.

## 5. Conclusions

In this paper, we have designed multi-feature probability graphical model(MFPGM) for the social network semantic search task. The MFPGM exploits features of social media data to overcome the semantic sparsity problem and generate high-quality semantic representations. First, by utilizing location information, we proposed a special region concept to aggregate short texts and overcome the semantic sparsity problem. Second, we introduced a biterm pattern to generate dense semantic space. Finally, we simultaneously modeled time features, user features, and biterms in the generative process of the MFPGM to enhance the

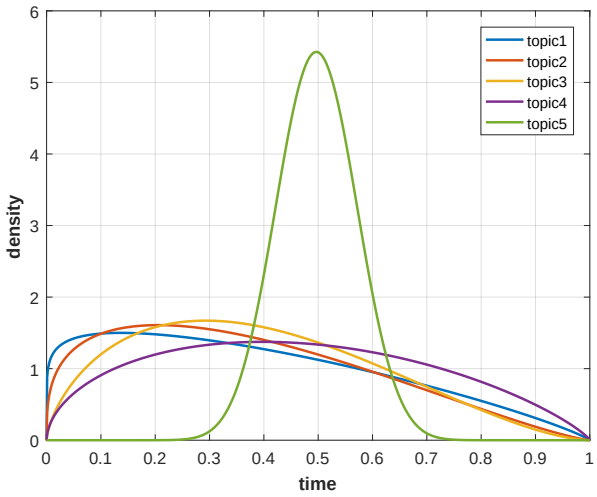



Figure 6: beta distribution of top 5 topics

learning process of topics. Modeling topic-time distribution contributes not only to improving topic quality, but also to displaying how the popularity of topics changes over time.

We conducted many experiments to verify the effectiveness of the proposed MFPGM and compared it with other state-of-th-art baseline methods. Because the MFPGM is a topic model, we measured it according to topic quality and search task. By filtering certain data subsets from the whole dataset, we tested its data-sensitivity and validated its effectiveness in modeling multiple features. We also demonstrated the capability of our method to display time-varying patterns of topics by showing the topic-time distribution. Our experiments have confirmed that the proposed MFPGM can generate high-quality topics and perform precise social network semantic searches. In the future, as images and videos continue to proliferate exponentially over social networks, we must take full advantage of visual feature as we continue to improve search capabilities.

**Acknowledgments**

This work is supported by the National Natural Science Foundation of China (No. 61320106006, No.61532006, No. 61772083).

**References**

[1] F. Huang, S. Zhang, J. Zhang, G. Yu, Multimodal learning for topic sentiment analysis in microblogging, Neurocomputing 253 (2017) 144–153. doi:10.1016/j.neucom.2016.10.086.

[2] L. Deng, Y. Jia, B. Zhou, J. Huang, Y. Han, User interest mining via tags and bidirectional interactions on sina weibo, World Wide Web (1) (2017) 1–22. doi:10.1007/s11280-017-0469-6.

[3] L. Gao, Y. Wang, D. Li, J. Shao, J. Song, Real-time social media retrieval with spatial, temporal and social constraints, Neurocomputing 253 (2017) 77–88. doi:10.1016/j.neucom.2016.11.078.

[4] W. Cui, P. Wang, Y. Du, X. Chen, D. Guo, J. Li, Y. Zhou, An algorithm for event detection based on social media data, Neurocomputing 254 (2017) 53–58. doi:10.1016/j.neucom.2016.09.127.

[5] J. Huang, M. Peng, H. Wang, J. Cao, W. Gao, X. Zhang, A probabilistic method for emerging topic tracking in microblog stream, World Wide Web 20 (2) (2017) 325–350. doi:10.1007/s11280-016-0390-4.

[6] C. Xing, Y. Wang, J. Liu, Y. Huang, W. Y. Ma, Hashtag-based sub-event discovery using mutually generative lda in twitter, in: Thirtieth AAAI Conference on Artificial Intelligence, 2016, pp. 2666–2672.

[7] Y. Wang, J. Liu, J. Qu, Y. Huang, J. Chen, X. Feng, Hashtag graph based topic model for tweet mining, in: Data Mining (ICDM), 2014 IEEE International Conference on, IEEE, 2014, pp. 1025–1030. doi:10.1109/TKDE.2016.2531661.

[8] Y. Wang, H. Huang, C. Feng, Query expansion based on a feedback concept model for microblog retrieval, in: International Conference on World Wide Web, 2017, pp. 559–568. doi:10.1145/3038912.3052710.

[9] Y. Wang, J. Liu, Y. Huang, X. Feng, Using hashtag graph-based topic model to connect semantically-related words without co-occurrence in microblogs, IEEE Transactions on Knowledge and Data Engineering 28 (7) (2016) 1919–1933. doi:10.1109/TKDE.2016.2531661.

[10] Y. Wang, I. C. Choi, H. Liu, Generalized ensemble model for document ranking in information retrieval, IEEE Transactions on Knowledge and Data Engineering 41 (2) (2017) 367–395.

[11] M. H. Alam, W.-J. Ryu, S. Lee, Hashtag-based topic evolution in social media, World Wide Web (3) (2017) 1–23. doi:10.1007/s11280-017-0451-3.

[12] Y. Wang, J.-S. Lee, I.-C. Choi, Indexing by latent dirichlet allocation and an ensemble model, Journal of the Association for Information Science and Technology 67 (7) (2016) 1736–1750. doi:10.1002/asi.23444.

[13] X. Fu, K. Yang, J. Z. Huang, L. Cui, Dynamic non-parametric joint sentiment topic mixture model, Knowledge-Based Systems 82 (2015) 102–114. doi:10.1016/j.knosys.2015.02.021.

[14] S. Yang, C. Yuan, B. Wu, W. Hu, F. Wang, Multi-feature max-margin hierarchical bayesian model for action recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1610–1618. doi:10.1109/CVPR.2015.7298769.

[15] X. Ren, M. Song, E. Haihong, J. Song, Context-aware probabilistic matrix factorization modeling for point-of-interest recommendation, Neurocomputing 241 (2017) 38–55. doi:10.1016/j.neucom.2017.02.005.

[16] T. Chen, H. M. SalahEldeen, X. He, M.-Y. Kan, D. Lu, Velda: Relating an image tweet's text and images., in: AAAI, 2015, pp. 30–36.

[17] W. Nie, X. Li, A. Liu, Y. Su, 3d object retrieval based on spatial+lda model, Multimedia Tools and Applications 76 (3) (2017) 4091–4104. doi:10.1007/s11042-015-2840-x.

[18] S. Yang, W. Lu, D. Yang, L. Yao, B. Wei, Short text understanding by leveraging knowledge into topic model, in: HLT-NAACL, 2015, pp. 1232–1237.

[19] C. Li, H. Wang, Z. Zhang, A. Sun, Z. Ma, Topic modeling for short texts with auxiliary word embeddings, in: Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, ACM, 2016, pp. 165–174. doi:10.1145/2911451.2911499.

[20] Y. Yang, F. Wang, J. Zhang, J. Xu, S. Y. Philip, A topic model for co-occurring normal documents and short texts, World Wide Web (2017) 1–27doi:10.1007/s11280-017-0467-8.

[21] F. Zhao, Y. Zhu, H. Jin, L. T. Yang, A personalized hashtag recommendation approach using lda-based topic model in microblog environment, Future Generation Computer Systems 65 (C) (2016) 196–206. doi:10.1016/j.future.2015.10.012.

[22] Y. Zuo, J. Zhao, K. Xu, Word network topic model: a simple but general solution for short and imbalanced texts, Knowledge and Information Systems 48 (2) (2016) 379–398. doi:10.1007/s10115-015-0882-z.

[23] P. Bicalho, M. Pita, G. Pedrosa, A. Lacerda, G. L. Pappa, A general framework to expand short text for topic modeling, Information Sciences 393 (2017) 66–81. doi:10.1016/j.ins.2017.02.007.

[24] H. Cai, Y. Yang, X. Li, Z. Huang, What are popular: exploring twitter features for event detection, tracking and visualization, in: Proceedings of the 23rd ACM international conference on Multimedia, ACM, 2015, pp. 89–98. doi:10.1145/2733373.2806236.

[25] W. X. Zhao, J. Jiang, J. Weng, J. He, E.-P. Lim, H. Yan, X. Li, Comparing twitter and traditional media using topic models, in: European Conference on Information Retrieval, Springer, 2011, pp. 338–349.

[26] R. Mehrotra, S. Sanner, W. Buntine, L. Xie, Improving lda topic models for microblogs via tweet pooling and automatic labeling, in: Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval, ACM, 2013, pp. 889–892. doi:10.1145/2484028.2484166.

[27] X. Cheng, X. Yan, Y. Lan, J. Guo, Btm: Topic modeling over short texts, IEEE Transactions on Knowledge and Data Engineering 26 (12) (2014) 2928–2941. doi:10.1109/TKDE.2014.2313872.

[28] L. Jiang, H. Lu, M. Xu, C. Wang, Biterm pseudo document topic model for short text, in: IEEE International Conference on TOOLS with Artificial Intelligence, 2016, pp. 865–872.

[29] X. Yan, J. Guo, Y. Lan, X. Cheng, A biterm topic model for short texts, in: Proceedings of the 22nd international conference on World Wide Web, ACM, 2013, pp. 1445–1456. doi:10.1145/2488388.2488514.

[30] Y. Hu, C. Hu, S. Fu, P. Shi, B. Ning, Predicting the popularity of viral topics based on time series forecasting, Neurocomputing 210 (2016) 55–65. doi:10.1016/j.neucom.2015.10.143.

[31] D. Guo, J. Xu, J. Zhang, M. Xu, Y. Cui, X. He, User relationship strength modeling for friend recommendation on instagram, Neurocomputing 239 (2017) 9–18. doi:10.1016/j.neucom.2017.01.068.

[32] X. Wang, A. McCallum, Topics over time: a non-markov continuous-time model of topical trends, in: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2006, pp. 424–433. doi:10.1145/1150402.1150450.

[33] X. Zhou, L. Chen, Event detection over twitter social media streams, Vldb Journal 23 (3) (2014) 381–400.

[34] G. Pedrosa, M. Pita, P. Bicalho, A. Lacerda, G. L. Pappa, Topic modeling for short texts with co-occurrence frequency-based expansion, in: Intelligent Systems (BRACIS), 2016 5th Brazilian Conference on, IEEE, 2016, pp. 277–282. doi:10.1109/BRACIS.2016.058.

[35] M. Rosen-Zvi, C. Chemudugunta, T. Griffiths, P. Smyth, M. Steyvers, Learning author-topic models from text corpora, ACM Transactions on Information Systems (TOIS) 28 (1) (2010) 4. doi:10.1145/1658377.1658381.

[36] Y. Zuo, J. Wu, H. Zhang, H. Lin, F. Wang, K. Xu, H. Xiong, Topic modeling of short texts: A pseudo-document view, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2016, pp. 2105–2114. doi:10.1145/2939672.2939880.

[37] Y. Zhao, S. Liang, Z. Ren, J. Ma, E. Yilmaz, M. de Rijke, Explainable user clustering in short text streams, in: Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, ACM, 2016, pp. 155–164. doi:10.1145/2911451.2911522.

[38] G. Bouma, Normalized (pointwise) mutual information in collocation extraction, Proceedings of GSCL (2009) 31–40.

[39] D. Mimno, H. M. Wallach, E. Talley, M. Leenders, A. McCallum, Optimizing semantic coherence in topic models, in: Proceedings of the conference on empirical methods in natural language processing, Association for Computational Linguistics, 2011, pp. 262–272.

[40] Y. Shen, X. He, J. Gao, L. Deng, G. Mesnil, Learning semantic representations using convolutional neural networks for web search, in: Proceedings of the 23rd International Conference on World Wide Web, ACM, 2014, pp. 373–374. doi:10.1145/2567948.2577348.

[41] H.-F. Yang, K. Lin, C.-S. Chen, Supervised learning of semantics-preserving hash via deep convolutional neural networks, IEEE Transactions on Pattern Analysis and Machine Intelligencedoi:10.1109/TPAMI.2017.2666812.

[42] D. Blei, A. Ng, M. Jordan, Latent dirichlet allocation, The Journal of Machine Learning Research 3 (2003) 993–1022.

**Feifei Kou** was born in 1989. She received her M.S. degree in Computer technology from Beijing Technology and Business University. She is now a Ph.D. candidate in Computer Science and Technology of Beijing University of Posts and Telecommunications. Her research interests include social network search, semantic analysis and semantic learning.

**Junping Du** was born in 1963. She is now a professor and Ph.D. tutor at the School of Computer Science and Technology, Beijing University of Posts

and Telecommunications. Her research interests include artificial intelligence, image processing and pattern recognition.




755

**Congxian Yang** was born in 1994. He received the B.S. degree in Computer Science and Technology from Shandong University in 2016,. He is currently pursuing the Master's degree in Computer Science and Technology from Beijing University of Posts and Telecommunications. His current research interest is social network search and machine learning.

**Yansong Shi** was born in 1993. He received his Bachelor Degree in Software Engineering from Sichuan University. He is now a M.S. candidate in Computer Technology of Beijing University of Posts and Telecommunications. His research interests include social network search and data mining.

**Meiyu Liang** was born in 1985. She received her Ph.D. degree in School of Computer Science from Beijing University of Posts and Telecommunications, Beijing, China, in 2014. She ever did postdoctoral research in School of Computer Science from Beijing University of Posts and Telecommunications from 2014 to 2016. She is currently an Associate Professor in School of Computer Science, Beijing University of Posts and Telecommunications, Beijing, China. Her research interests include image and video processing, data mining and computer vision.

**Zhe Xue** received the Ph.D. degree in computer science from University of Chinese Academy of Sciences, Beijing, China in 2017, and the B.S. degree in Electronic Engineering from Civil Aviation University of China, Tianjin, China, in 2010. He is currently an assistant professor with the school of computer science, Beijing University of Posts and Telecommunications, Beijing, China. His research interests include machine learning, computer vision and multimedia

data mining.

**Haisheng Li** was born in 1974, He received his Ph.D. degree in computer graphics from Beihang University, China in 2002. He is a professor in school of Computer and Information Engineering, Beijing Technology and Business University, China. He is the director of network center in BTBU. He is also the discipline leader in Computer science and technology in BTBU. His current research interests include computer graphics, scientific visualization, 3D model retrieval etc.