IEEE Transactions on Knowledge and Data Engineering,Volume:28,Issue:7,Issue Date :July.1.2016

Using Hashtag Graph-based Topic Model to Connect Semantically-related Words without Co-occurrence in Microblogs

Yuan Wang, Jie Liu, Yalou Huang and Xia Feng

Abstract—In this paper, we introduce a new topic model to understand the chaotic microblogging environment by using hashtag graphs. Inferring topics on Twitter becomes a vital but challenging task in many important applications. The shortness and informality of tweets leads to extreme sparse vector representations with a large vocabulary. This makes the conventional topic models (e.g., Latent Dirichlet Allocation [1] and Latent Semantic Analysis [2]) fail to learn high quality topic structures. Tweets are always showing up with rich user-generated hashtags. The hashtags make tweets semi-structured inside and semantically related to each other. Since hashtags are utilized as keywords in tweets to mark messages or to form conversations, they provide an additional path to connect semantically related words. In this paper, treating tweets as semi-structured texts, we propose a novel topic model, denoted as *Hashtag Graph-based Topic Model* (*HGTM*) to discover topics of tweets. By utilizing hashtag relation information in hashtag graphs, HGTM is able to discover word semantic relations even if words are not co-occurred within a specific tweet. With this method, HGTM successfully alleviates the sparsity problem. Our investigation illustrates that the user-contributed hashtags could serve as weakly-supervised information for topic modeling, and the relation between hashtags could reveal latent semantic relation between words. We evaluate the effectiveness of HGTM on tweet (hashtag) clustering and hashtag classification problems. Experiments on two real-world tweet data sets show that HGTM has strong capability to handle sparseness and noise problem in tweets. Furthermore, HGTM can discover more distinct and coherent topics than the state-of-the-art baselines.

Index Terms—Hashtag graph, topic modeling, sparseness of short text, weakly-supervised learning

1 Introduction

Microblogging platforms such as Twitter have gone global. With billions of active users, Twitter is popular because of its massive spreading of instant messages (i.e. tweets), bursts of world news, entertainment gossip about celebrities, and discussions over recently released products are all spreading on Twitter vividly. Text content is one of the most important elements of social networks. It has been well recognized that uncovering topics of these user-generated contents is crucial for a wide range of content analysis tasks, such as natural disaster awareness [3], emerging topic detecting [4], interesting content identification [5], user interest profiling [6], realtime web search [7], et al.

Characterizing contents of documents is a standard problem addressed in information retrieval and statistical natural language processing. Achieving good representations of documents could benefit tasks of organizing, classifying and searching a collection of documents. In recent years, topic models such as Probabilistic Latent Semantic Analysis (PLSA) [8] and Latent Dirichlet Allocation (LDA) [1], have been recognized as powerful methods of learning semantic representations for a corpus. According to the assumption that each document has a multinomial distribution over topics and each topic is a mixture distribution over words.

Although traditional methods have achieved success in uncovering topics for normal documents (e.g., news articles, technical papers), the characteristics of tweets bring new challenges and opportunities to them. There are three key reasons. First, the severe sparsity problem of tweet corpora invalidates traditional topic modeling techniques. Typically, LDA and PLSA both reveal the latent topics by capturing the document-level word co-occurrence patterns. Compared with normal texts, tweets usually contain only a few words. Furthermore, the usage of informal language enlarges the size of the dictionary. Second, conventional topic models are designed for flat texts without structure. On Twitter, hashtags, prefixing one or more characters with a hash symbol as "#hashtag", are a community-driven convention for adding both additional context and metadata to tweets, making tweets semi-structured texts. Hashtags are created or selected by users to categorize messages and highlight topics. They provide a crowdsourcing way for tagging short texts, which is usually ignored by Bayesian statistics and machine learning methods. Last but not least, such crowd wisdom information clashes with the assumption of Independent Identical Distribution (i.i.d) of documents. The weakly-supervised information provided by hashtags can build direct semantic relations between tweets so that the words in tweets have more complex topical relationships than in normal texts. Typically, it is reasonable to assume that the tweets containing the same hashtags have similar underlying topics [9] [10] [11]. Hence, the i.i.d assumption does not hold anymore.

Therefore, in addition to the bag-of-words within a tweet, it is crucial to consider semantic information in semi-structured contexts conveyed by hashtags. We find that there are two kinds

- *Y. Wang, J.Liu and Y. Huang are with College of Computer and Control Engineering and College of Software, Nankai University, Tianjin, China, 300071.*
  E-mail: {yayaniuzi23@mail., jliu@, huangyl@}nankai.edu.cn
- *X. Feng is with Information Technology Research Base of Civil Aviation Administration of China, Civil Aviation University of China, Tianjin, China, 300071.*
  E-mail: xfeng@cauc.edu.cn
- *Jie Liu is the corresponding author of the paper*

Manuscript received XXX XX, XXXX; revised XXX XX, XXXX.

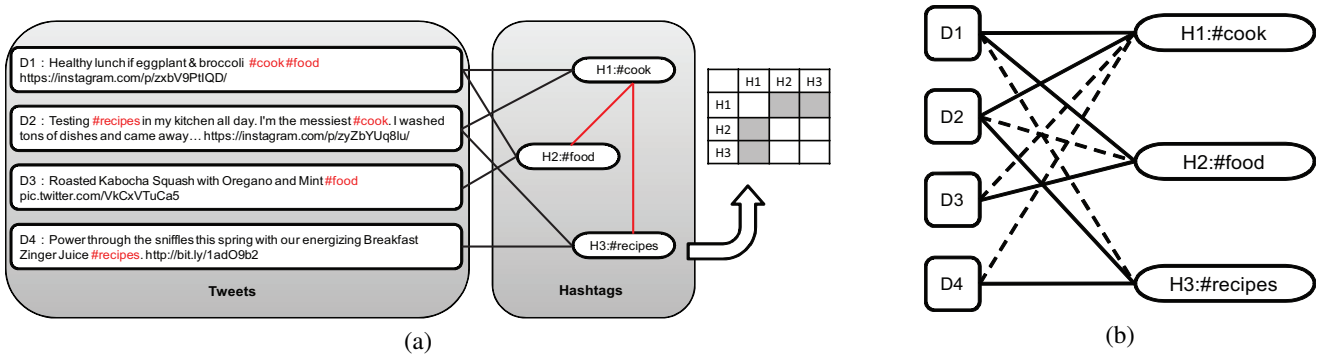


Fig. 1. An illustration of semantic relationships in tweets. (a) Explicit Relationship. One is the inclusion relation between tweets and hashtags marked with black lines, the other one is the co-occurrence relation between hashtags marked with red links. The hashtag relationship can be formulated as a relation graph represented by a matrix. (b) Potential Relationship. The potential inclusion relation between tweets and hashtags are marked with dotted lines. It means tweets probably connect with hashtags that are not included.

of relationships in tweets that lead to semantic connections. One is *explicit relationship* that contains inclusion relations between tweets and hashtags and co-occurrence relations between hashtags, as Figure 1(a) shows. Due to the explicit relationship, tweets sharing the same hashtags have highly overlapping correlated topics. The other one is *potential relationship* shown as dotted lines in Figure 1(b). A tweet should have a possibility to connect or contain those hashtags that have no explicit relationship with, but have a lot of co-occurrences with hashtags the tweet has already contained. Hence, hashtag co-occurrences in tweets indirectly contribute wider semantic relationship between tweets. It is easy to figure out, as shown in Figure 1(a), users anticipate the topic of "Cook" by adding the hashtags "#cook", "#food", "#cook" in tweet D1, D2, D3 and D4. The same hashtag bridges tweets with explicit relationship (i.e., hashtag inclusion relation) as an aggregation solution. Furthermore, hashtag co-occurrences in a whole corpus indirectly give a chance to connect tweets with no hashtag sharing. For example, word "Breakfast" in tweet D4 and word "lunch" in tweet D1 are obviously semantically related. Unfortunately, one tweet or the aggregation solution couldn't handle or find out such a semantic relationship. Whereas, we can connect these two words through the path "D4"-"#recipes"-"#cook"-"D1" based on the hashtag co-occurrences in the whole dataset shown in Figure 1(a). That means D4 should have a potential relationship with "#cook" (in a dotted link as Figure 1(b) shows), and D1 can be connected to "#recipes" as well. These connections tackle the problem of sparseness in tweets as a weakly-supervised information and build a meaningful semantic relation between words.

Inspired by the observations mentioned above, we construct different kinds of hashtag graphs based on statistical information of hashtag occurrence in a crowdsourcing manner that can be acquired without human efforts such as labeling. Based on these hashtag graphs, we propose a novel framework of Hashtag Graph-based Topic Model (HGTM). The basic idea of HGTM [12] is to project tweets into a coherent semantic space by using latent variables via user-contributed hashtags. HGTM provides a robust way for noisy and sparse tweets, which is different from traditional topic models since they normally consider only content information and ignore explicit and potential semantic connection via noisy hashtags. HGTM is a probability generative model that incorporates such weakly-supervised information based on a weighted hashtag graph. The model links tweets via both explicit and potential tweet-hashtag relationship, so that hashtag relationship can connect semantically-related words with or without co-occurrences, which alleviates severe sparse and noise problem in short texts. In our previous work [12], we have verified the effectiveness that HGTM can bridge semantic-related words when they share no co-occurrences. In this paper, we extend the previous work and further explore the influence of different hashtag graph construction methods and discuss more details about HGTM, including time complexity analysis and the key process of hashtag assignment analysis. We evaluate HGTM on two real-world Twitter data sets to understand different kinds of hashtag graphs and the working of HGTM on extensive tweet mining tasks such as clustering, classification, and topic quality evaluation. Compared to the state-of-the-art methods, HGTM shows the ability of handling the sparseness and noise problem in mining tweets by exploiting both explicit and potential relations between hashtags and tweets.

The remainder of the paper is organized as follows. Section 2 gives a brief summary of related work and draws a comparison to our approach. In Section 3, we describe Hashtag Graph-Based Topic Model and discuss its time complexity. Experimental results and analysis are given in Section 4. Section 5 concludes our new findings.

2 Related Works

In this section, we briefly summarize related works of topic models on flat text and semi-structured text.

2.1 Topic Models on Flat Text

Topic models have been widely used to discover latent semantic structures in a corpus. The topic structures in corpora have certain theoretical and practical value. Researchers have already proposed many powerful topic models for document analysis, such as Latent Semantic Analysis (LSA) [2], Probabilistic Latent Semantic Analysis (PLSA) [8], Latent Dirichlet Allocation (LDA) [1] and Correlated Topic Model (CTM) [13]. They use different techniques and assumptions to analyze a corpus. LSA applies singular value decomposition to reduce dimensions of documents; PLSA is an extension of LSA from the perspective of probability. LDA introduces Dirichlet priors for generating a document's distribution over topics, and gives a way to model new documents. CTM models topic correlation between documents by replacing

Dirichlet priors with Logistic Normal priors. They have achieved success in traditional tasks of long document understanding, such as text classification and clustering [14], information retrieval [15], semantic analysis [16]. However, traditional topic models fail in modeling tweets due to the severe sparseness and noise in short tweets [9] [10]. Hong, et al. [9] made a comprehensive study of topic modeling on Twitter and suggested that specific topic models for tweets were in demand.

Several methods have been proposed to tackle the serious noise and lack of context problems in tweets. One intuitive method is to aggregate tweets as a long document. Typically, Hong, et al. [9] aggregated tweets by the same user, the same word or the same hashtag. Mehrotra, et al. [10] investigated different pooling schemes with hashtags for the later LDA process. Weng, et al. [17] introduced "a pseudo document" by collecting tweets under the same author. Yan, et al. [18] clustered tweets by a non-negative matrix factorization. They all achieved a better performance than original LDA on tweet mining via tweet relevancy assumption. Furthermore, Yan, et al. [19] extracted biterms from tweets to enlarge word co-occurrence patterns in LDA. The other alternative is to inflate or link short texts with additional information. Some works focus on introducing external knowledge from auxiliary long texts to enrich short texts, such as using Wikipedia and WordNet [20]. Additionally, some researchers directly model sparseness in tweets. Zhao, et al. [21] assumed that one tweet contained only a single topic. However, this assumption is too strong to control the multi-semantic tweet modeling. To relax the constraint, Lin, et al. [22] achieved focused topics and focused terms for short texts in the way of adding a dual-sparse constraint on topic mixtures of documents and words by applying a "Spike and Slab" prior. The above models all tackle the problem from the point of content, but they consider tweets as flat texts and ignore tag-related information contained in twitter data.

2.2 Topic Models on Semi-structured Text

Several works have been carried out to utilize semi-structured information (tags or labels) for content modeling, which can model semantic relevancy better.

In the study of tweets, Labeled LDA [23] takes manually selected labels as supervision information. Ramage, et al. [24] applied Labeled LDA on tweet topic modeling, drawing the topic distribution by picking out hyperparameter components related to a tweet's labels. Lim, et al. [11] made use of hashtags for tweet aggregation to improve performance on aspect clustering.

Besides tweets, many approaches take advantage of tags or labels for normal text mining, such as Tag-LDA Model [25], Partially Labeled Topic Model (PLDA) [26], Dirichlet-multinomial Regression (DMR) topic model [27], Tag-Weighted Topic Model (TWTM) [28] and Tag-Weighted Dirichlet Allocation (TWDA) [29]. Tag-LDA Model treats tags as extended words and then learns topics by LDA. PLDA constricts each topic to a specific label which is associated with a topic class. TWTM infers a topic distribution for each individual document with a function of tag-weighted topic assignment. DMR and TWDA both include label priors on the topic distribution of each document. In DMR, the prior distribution over topics is a log-linear function of metadata features in the document while TWDA considers the weight of metadata features and adds a Dirichlet prior when generating document's topic distribution. The idea of tag weighting in TWTM and TWDA is related to ours to some extent, but our hashtag weighting information is based on the wisdom of crowds rather than a prior determined by academic experience or data validation. Therefore, our method allows to utilize user-generated information to build word relevancy and further to handle short text. The Author-Topic Model (ATM) [30] [31] also can be seen as a way of modeling text via tags, by treating tags as authors. In this regard, Tsai [32] showed a reliable result of applying ATM on blog mining. Hence, we compare with ATM as a strong baseline of our model. Note that ATM just leverages tag information by a uniform distribution of tags, but ignores the potential tag relation that is vitally helpful to build the latent semantic relationship between words. So, ATM still suffers from the lack of word co-occurrences. Topic Model with Biased Propagation (TMBP) [33] and contextual Focused Topic Model (cFTM) [34] take text-rich heterogeneous information networks to model the topic assignment. They both leverage contextual information, the authors and venues for documents' topic distribution generation. However, they capture the relation between documents in a similar way as ATM does.

In our previous work [12], we have found the hashtag graph is helpful for tweet clustering and hashtag clustering. This paper extends that work with the following significant improvements. 1) We introduce two more strategies of constructing hashtag graphs. 2) More comprehensive experiments are conducted on more datasets and new findings are reported. 3) The capability of HGTM with various hashtag graphs is verified.

3 Hashtag Graph-based Topic Model

In this section, we first introduce notations and hashtag graphs. Secondly, we mathematically investigate and explore Hashtag Graph-based Topic Model (HGTM). Thirdly, we discuss the parameter inference method. In the end, we analyze the complexity of HGTM.

3.1 Notations and Definitions

In HGTM, words are discrete random variables coming from a fixed dictionary. We define the tweet corpus as $D = \{d\}_{d=1:M}$, with a word dictionary $\{w\}_{w=1:W}$ and a hashtag dictionary $\{h\}_{h=1:H}$. Suppose that document d has a word sequence $\mathbf{w}_d = \{w_{d1}, \ldots, w_{dj}, \ldots, w_{dN_d}\}$ and a hashtag sequence $\mathbf{h}_d = \{h_{d1}, \ldots, h_{dj}, \ldots, h_{dH_d}\}$, where w_{dj} is the j^{th} word in document d and h_{dj} is the j^{th} hashtag in document d.

A hashtag graph is an undirected graph, denoted as $\mathcal{G} = (V, E)$, where nodes V are hashtags from the hashtag dictionary $\{h\}_{h=1:H}$ and edges $E = \{(h, h')\}$ are obtained from co-occurrence relations between hashtags in the explicit relationship. The edge $e_{hh'}$ is weighted based on the association weight between hashtag h and hashtag h'. There are various hashtag relations in the corpus, such as appearing in the same tweets, used by the same users and added with the same URLs, all of which reflect semantic relevancy between hashtags. Such information can be stored as a hashtag relation matrix G, in which the entry g_h at the h^{th} row represents hashtag h's incident vector and $g_{hh'}$ is the association weight obtained by measuring the number of one kind of co-occurrences mentioned above. We use $g_{\mathbf{h}_d}$ to denote the multiple rows in G, where hashtag indexes are in $\mathbf{h}_d$.

Traditional probabilistic generative topic models (e.g. LDA [1]) suppose that there are T topics in the whole corpus. Each document is typically characterized by a distribution over topics as θ, and each topic is represented by a distribution over words as ϕ. Taking LDA for example, each word w_{dj} in the document

d is assigned with a latent variable z_{dj} (topic assignment). The generative process of a document d via latent topic variables is given as follows:

$$\theta_i|\alpha \sim Dirichlet(\alpha)$$
$$\phi_i|\beta \sim Dirichlet(\beta)$$
$$z_{dj}|\theta_d \sim Multinomial(\theta_d)$$
$$w_{dj}|\phi_{z_{dj}} \sim Multinomial(\phi_{z_{dj}})$$

where α and β are hyperparameters of Dirichlet priors. Obviously, LDA treats tweets (including hashtags) as a flat structure without considering the semantic relevance between tweets.

Different from LDA, we characterize each hashtag as a distribution over topics as θ. A topic assignment z_{dj} and a hashtag assignment y_{dj} are allocated for each word w_{dj} in document d. Here, we use bold letters $\mathbf{z}$ and $\mathbf{y}$ which are both N-dimensional vectors, to denote the topic and hashtag assignments for all words respectively. The related assignment vectors of document d in $\mathbf{z}$ and $\mathbf{y}$ are $\mathbf{z}_d$ and $\mathbf{y}_d$.

According to explicit and potential relationships mentioned in the introduction, there are two types of corresponding hashtags for a tweet, *explicit hashtags* and *potential hashtags*.

Definition 3.1 (Explicit Hashtags). Explicit hashtags of tweet d refer to the hashtags that are contained in tweet d, i.e., $\mathbf{h}_d$ of tweet d.

For example, in Figure 1, the explicit hashtags of tweet D1 are "#cook" and "#food". The behavior of explicit hashtags shows a directly semantic relationship between hashtags and words in tweets. In tweet D1, it means word "lunch" and word "eggplant" both share related semantic meanings with "#cook" and "#food".

The advantage of explicit hashtags is obvious: the semantic associations between words that have co-occurrences with the same hashtags are established. So many related works [9] [10] [11] used the aggregating or pooling techniques via hashtags to learn topic structure of a tweet corpus. However, due to the sparseness of tweets, some of semantically-related words probably have a low or even zero co-occurrence probability. In this case, explicit hashtags are not enough to detect the relationship between words.

Definition 3.2 (Potential Hashtags). Potential hashtags of tweet d refer to the hashtags that do not appear in tweet d, but have co-occurrence with hashtags in $\mathbf{h}_d$, i.e., the ones with non-zero association weights with explicit hashtags in a hashtag graph.

For example, in Figure 1, the potential hashtag of tweet D1 is "#recipes". In spite of no appearance in tweet D1, "#recipes" has a chance to connect with words in tweet D1 because of co-occurrence with "#cook" in tweet D2. So, word "lunch" can share related semantic with "#recipes". In this way, we can generate an effective connection between semantically-related words via hashtags and hashtag graphs.

Using these definitions, HGTM tries to handle the relationship of semantically-related words with co-occurrences via explicit hashtags and to connect semantically related words without co-occurrence via potential hashtags.

3.2 The Generative Process of Tweets in HGTM

HGTM is a probabilistic generative model that describes the process of generating a semi-structured tweet collection with weakly-supervised information from hashtag graphs. The model associates each word position with a "hashtag-topic" assignment pair. We generate a hashtag assignment y first, then allocate a topic z from the topic distribution of hashtag y for the current word position, and finally generate the specific word from topic z's distribution over words. Drawn from Dirichlet hyperparameter α, each hashtag is represented as a multinomial distribution over topics. By assigning the latent hashtag assignment to each word, each hashtag has its own contribution to the topic distribution of tweets. The word distribution specific to each topic is drawn from Dirichlet hyperparameter β. HGTM parameterization is given as follows:

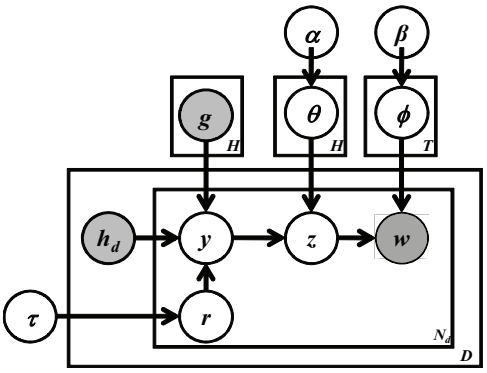


Fig. 2. The graphical model representation for HGTM, where θ is topic distribution matrix of hashtags, ϕ is word distribution matrix of the topics, y indicates the tag assignment for current word.

$$\theta_h|\alpha \sim Dirichlet(\alpha)$$
$$\phi_t|\beta \sim Dirichlet(\beta)$$
$$y_{di}|\gamma_d \sim Multinomial(\gamma_d; \mathbf{h}_d, g_{\mathbf{h}_d}, \tau)$$
$$z_{di}|\theta_{y_{di}} \sim Multinomial(\theta_{y_{di}})$$
$$w_{di}|\phi_{z_{di}} \sim Multinomial(\phi_{z_{di}}),$$

where γ_d (the detail is shown in 3.3) is a distribution conditioned on $\mathbf{h}_d$, $g_{\mathbf{h}_d}$ and τ. Here we introduce a Bernoulli variable τ to decide whether to assign explicit hashtags or potential hashtags for the current word position. Thus, we can connect semantically-related words by applying hashtag graphs.

The generative process for HGTM is given by the following steps (as shown in Figure 2) :

1 T, α, β, τ are predefined
2 For each of the hashtags $h = 1 : H$, draw $\theta_h \sim Dir(\alpha)$
3 For each of the topics $t = 1 : T$, draw $\phi_t \sim Dir(\beta)$
4 For each of the documents $d = 1 : D$, draw its length N_d, given a hashtag set $\mathbf{h_d}$ referred to the document d
  For each word w_{di}, $\quad i = 1 : N_d$
    1) draw an initial hashtag assignment $y^1_{di} \sim Uni(\mathbf{h}_d)$
    2) draw $r \sim Bern(\tau)$
    3) if $r = 1$, draw a hashtag assignment $y_{di} = y^1_{di}$,
      if $r = 0$,
    draw a hashtag assignment $y_{di} \sim Multi(norm(g_{y^1_{di}}))$
    4) draw a topic assignment $z_{di} \sim Multi\left(\theta_{y_{di}}\right)$
    5) draw a word assignment $w_{di} \sim Multi\left(\phi_{z_{di}}\right)$

In Step 3), $norm(g_{y^1_{di}})$ is an H-dimension association probability vector by normalizing row values of the hashtag graph, where the j^{th} element is

$$p(y_j|y^1_{di}) = \frac{g_{y^1_{di},y_j}}{\sum_{j'} g_{y^1_{di},y_{j'}}}. \quad (1)$$

The Equation (1) reflects the compactness of the semantic relationship between hashtags. It indirectly tells the semantic relationship

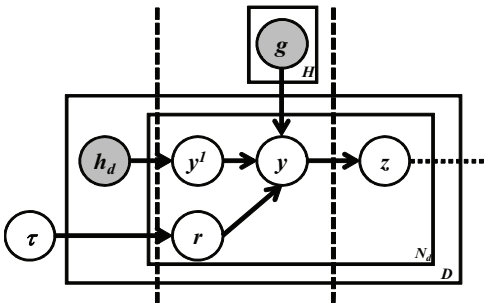


Fig. 3. The graphical model representation of two-step hashtag sampling in HGTM. τ indicates the probability of hashtag assignment within hashtags in the current tweet, y^1 is sampled from hashtags in current tweet, hashtag assignment y is codetermined by r, y^1 and hashtag graph G.

of words from different tweets that contain related hashtags separately. In HGTM, the association weight shows the similarity between topic distributions of different hashtags. As shown in Figure 1, the toy example reflects the equal similarity between "#cook" and "#food" (or "#recipes"). From the whole dataset, hashtag co-occurrence frequency indicates the similarity between two hashtags' topic distributions, and further tells the similarity of topic distributions of two tweets related to these hashtags. The statistics of hashtag relation avoid the necessity of word co-occurrence, transmit semantic information and then enhance topic modeling. We assume that such a phenomenon is in accord with the way that users exchange information and communicate in the microblogging platforms.

3.3 Key Process of Hashtag Assignment

The key step of the generative processes is to sample a correlated hashtag for the current word position. Under the observation of the hashtags in tweets, we model the process of assigning hashtags by using a two-step procedure of hashtag selection. In the first step, we sample a hashtag y^1_{di} uniformly from $\mathbf{h}_d$. In the second step, we sample an r from a Bernoulli distribution with parameter τ to decide whether the current word position is related to explicit hashtags.

Note that we model semantic arbitrary relation between hashtags and words via a sticky factor $\tau \sim [0,1]$. The hyperparameter τ defines the possibility that the assigned hashtag is from the hashtags in this tweet, i.e. $\tau \sim p(y \in \mathbf{h}_d)$. Meanwhile, there is $(1-\tau)$ chance for the current word position to be assigned to highly semantically-related potential hashtags. Through the sampling process, on one hand, we simulate randomness during hashtag selection, on the other hand, we seamlessly integrate correlated hashtags to enhance the semantic relationship between short texts that are semantically similar but contain literally different hashtags. When τ is less than 1, we introduce meaningful word co-occurrence via latent hashtag assignment even if they have less or no co-occurrence. The process of two-step sampling is showed in Figure 3.

Specifically, we find out that original hashtag relationships are of the following aspects: 1) two hashtags co-occur in the same tweets, 2) two hashtags are added by the same group of users, 3) two hashtags are inserted with a number of the same URLs, et al. We can directly apply these frequencies as weight schemas in hashtag relation matrix G to construct hashtag graphs.

During hashtag assigning process, let vector γ_d represent the probability of hashtag sampling, where the h^{th} element is the probability of hashtag h being sampled. Let vector $\mathbf{s}_d$ represent the original sampling probability, where $s_{dh} = 1$ only when $h \in \mathbf{h}_d$. So, the hashtag sampling probability distribution is

$$\gamma_d = \tau s_d + (1-\tau) \sum_{t \in \mathbf{h}_d} norm(g_t). \tag{2}$$

As shown in Equation (2), only those hashtags that occur in the current tweet, or share a large number of co-occurrences with $\mathbf{h}_d$ in a whole tweet corpus, can achieve the highest probability to be assigned. It shows how semantically-related words are connected by hashtags. Meanwhile, the hyperparameter τ controls the contribution of potential hashtags in HGTM. The less τ is, the higher randomness is, and vice versa.

3.4 Parameter Estimation

HGTM is a probabilistic model that describes the text generation process. As shown in Figure 3, words, hashtags and hashtag graphs are observed while the topic structure (i.e. the hashtag assignment $\mathbf{y}$ and the topic assignment $\mathbf{z}$) is hidden. The hidden variables are guided by latent distribution parameters, i.e., the H hashtag-topic distribution θ and the T topic-word distribution ϕ. Thus, the central problem for our model is to infer the hidden variables via the observed ones. Therefore, we compute the posterior distribution of the hidden variables given the observed variables. We infer θ and ϕ via the sample assignment z and y. Assuming that each document is independent, generating probability of the whole corpus is

$$p(\mathbf{w}|\theta,\phi,r,\mathbf{h},G) = \prod_{d=1}^{D} p(\mathbf{w}_d|\theta,\phi,r,\mathbf{h}_d,\mathbf{g}_{h_d}). \tag{3}$$

For each document, the probability of word vector $\mathbf{w}_d$ conditioned on the model parameters is

$$\begin{aligned} p(\mathbf{w}_d|\theta,\phi,r,\mathbf{h}_d,\mathbf{g}_{h_d}) &= \prod_{i=1}^{N_d} p(w_{di}|\theta,\phi,r,\mathbf{h}_d,\mathbf{g}_{\mathbf{h}_d}) \\ &= \prod_{i=1}^{N_d}\sum_{s=1}^{H}\sum_{t=1}^{T} p(w_{di},z_{di}=t,y_{di}=s|\theta,\phi,r,\mathbf{h}_d,\mathbf{g}_{\mathbf{h}_d}) \\ &= \prod_{i=1}^{N_d}\sum_{s=1}^{H}\sum_{t=1}^{T} p(w_{di}|z_{di}=t,\phi)p(z_{di}=t|y_{di}=s,\theta)p_{sy_{di}} \\ &= \prod_{i=1}^{N_d}\sum_{s=1}^{H}\sum_{t=1}^{T} \phi_{w_{di}t}\theta_{ts}p_{sy_{di}}, \end{aligned} \tag{4}$$

where topic distribution over words ϕ and hashtag distribution over topics θ are conditional independent, $p_{sy_{di}}$ indicates the probability of hashtag assignment s within the known explicit hashtags $\mathbf{h}_d$ and potential hashtags. From the procedure of two-step hashtag sampling discussed earlier, we choose the related hashtag according to the co-occurrence and statistic correlation between hashtags:

$$\begin{aligned} p_{sy_{di}} &= p(y_{di}=s|r,\mathbf{h}_d,g_{\mathbf{h}_d}) \\ &= [p(y^1_{di}=s|\mathbf{h}_d)p(y_{di}=s|y^1_{di}=s)]^r \\ &\quad [\sum_{j=1}^{H_d} p(y^1_{di}=h_{dj}|\mathbf{h}_d)p(y_{di}=s|y^1_{di}=h_{dj},g_{h_{dj}})]^{1-r}. \end{aligned} \tag{5}$$

When $r=1$, there is no chance to assign potential hashtags. Then, the formula degenerates to

$$\begin{aligned} &p(y_{di}=s|r,\mathbf{h}_d,g_{\mathbf{h}_d}) \\ &= p(y^1_{di}=s|\mathbf{h}_d) = \frac{1}{H_d}. \end{aligned} \tag{6}$$

When $r = 0$, the model assigns only potential hashtags that have high semantic relationship with $\mathbf{h}_d$ in the hashtag graph. The formula degenerates to

$$\begin{aligned} & p(y_{di} = s | r, \mathbf{h}_d, g_{\mathbf{h}_d}) \\ &= \sum_{j=1}^{H_d} p(y_{di}^1 = h_{dj} | h_d) p(y_{di} = s | y_{di}^1 = h_{dj}, g_{h_{dj}}) \\ &= \frac{1}{H_d} \sum_{j=1}^{H_d} \frac{g_{h_{dj},s}}{\sum_{j'} g_{h_{dj},j'}}. \end{aligned} \tag{7}$$

When applied to probabilistic topic models [8], this method is susceptible to local maxima and computationally inefficient [1]. Hence, we employ an alternative parameter estimation strategy, the Gibbs sampling procedure [35] – a fast and efficient Markov Chain Monte Carlo (MCMC) algorithm to carry out approximated parameters instead of estimating the model parameters directly. It infers complex probability distributions by iterative sampling. By applying a Euler integration for Equation (4), we can obtain the sample posterior distribution:

$$\begin{aligned} & p(z_{di} = t, y_{di} = s, r_{di} = u | w_{di} = w, \mathbf{z}_{-di}, \mathbf{y}_{-di}, \mathbf{w}_{-di}, H, G, \alpha, \beta, \tau) \\ & \propto \frac{C_{wt,-di}^{WT} + \beta}{\sum_{w'} C_{w't,-di}^{WT} + W\beta} \cdot \frac{C_{ts,-di}^{TH} + \alpha}{\sum_{t'} C_{t's,-di}^{TH} + T\alpha} \cdot p_s^{y_{di}}, \end{aligned} \tag{8}$$

where C^{WT} is the count matrix of the times that a specific word is assigned to a specific topic, C^{TH} is the count matrix of the times that a specific topic is assigned to a specific hashtag, and $-di$ means hashtag assignments and topic assignments except that for the current word.

After iterative sampling, it reaches convergence. The final results of θ and ϕ are

$$\begin{aligned} \theta_s &\propto \frac{C_{ts}^{TH} + \alpha}{\sum_{t'} C_{t's}^{TH} + T\alpha} \\ \phi_t &\propto \frac{C_{wt}^{WT} + \beta}{\sum_{w'} C_{w't}^{WT} + W\beta} \end{aligned} \tag{9}$$

Thus, according to the detected topic structure, HGTM can conclude distinguishable topics in tweets and find out clear representative words for each topic. Besides, HGTM finds out hashtags' semantic meaning and key hashtags under each topic as well. The Gibbs sampler process is summarized in Algorithm 1.

3.5 New Tweet's Topic Distribution Inference

After parameter estimation, we can get hashtags' probability distributions over topics and topics' probability distribution over words. For a new tweet with known hashtags, we infer its topic assignments by the same sampling process as parameters inference, but the latent variable probability distributions are static by using the parameters learned above.

The algorithm is summarized in Algorithm 2. On each iteration, we fix topic distributions over hashtags θ. Firstly, we run two-step hashtag sampling with the same τ used in the training process for each word position in the new tweet. Secondly, we draw a topic assignment z_{di} according to the topic distribution of hashtag $\theta_{y_{di}}$. Finally, we normalize all topic assignment $\mathbf{z}_d$ in topic dimensions to get the topic distribution of tweet d. So the topic distribution of tweet d is:

$$p(z|d) = \frac{\sum_{i=1}^{N_d} \delta(z_{di} == z)}{N_d}, \tag{10}$$

where $\delta(.)$ the indicator function.

Algorithm 1 Gibbs sampling algorithm for HGTM.

Input: topic number T, hashtag graph G, iteration times NN, α, β, τ, word sequence $\mathbf{w}$, hashtag sequence $\mathbf{h}$;
Output: Θ, ϕ;
Initialization: randomly initialize the hashtag assignments $\mathbf{y}$ and topic assignments $\mathbf{z}$ for all words;

```
for ii = 1 : NN do
    for d = 1 : D do
        for i = 1 : N_d do
            Draw y_di^1 ~ Uni(h_d)
            Draw r ~ Bern(τ)
            if r = 1 then
                y_di = y_di^1
            else
                Draw y_di ~ Multi(norm(g_{y_di^1}))
            end if
            Draw a topic z_di ~ Multi(θ_{y_di})
            Update C^WT_{w_di,z_di} and C^TH_{z_di,y_di}
        end for
    end for
    Calculate Θ, φ as as Equation 9
end for
return Θ, φ;
```

Algorithm 2 HGTM Inference for A New Tweet.

Input: iteration times NN, $\theta, \tau, G, \mathbf{w}_d, \mathbf{h}_d$;
Output: tweet d's hashtag assignments $\mathbf{y}_d$ and topic assignments $\mathbf{z}_d$;
Initialization: randomly initialize the hashtag assignments $\mathbf{y}_d$ and topic assignments $\mathbf{z}_d$;

```
for ii = 1 : NN do
    for i = 1 : N_d do
        Draw y_di^1 ~ Uni(h_d)
        Draw r ~ Bern(τ)
        if r = 1 then
            y_di = y_di^1
        else
            Draw y_di ~ Multi(norm(g_{y_di^1}))
        end if
        Draw a topic z_di ~ Multi(θ_{y_di})
        Update y_di and z_di in y_d and z_d
    end for
end for
return y_d and z_d;
```

3.6 Complexity Analysis

The major time-consuming part of parameter estimation is to calculate the conditional probability of hashtags and topics in Equation (8), with time complexity $O(NTH_d^{max}G_{\mathbf{h}_d}^{max})$, where H_d^{max} is the maximum number of hashtags in a tweet, and $G_{\mathbf{h}_d}^{max}$ is the maximum number of hashtags that a hashtag can be connected to in the hashtag graph. However, in most cases, users would not add too many hashtags in one short tweet, that is to say, H_d^{max} and $G_{\mathbf{h}_d}^{max}$ always show up with a relatively small number, which can be replaced by a fixed integer.

According to our observation, there are 106,682 hashtags in one public twitter dataset[1] used in our experiment. H_d^{max} is 17, which means $H_d^{max} \ll H$. Meanwhile, hashtags are always associated with an actual event or a hot topic in real life, which leads to $G_{\mathbf{h}_d}^{max} \ll H$. In our dataset, the maximum numbers of hashtag co-occurrences in the same tweets, related to the same URLs, added by the same users are 1198, 329, and 3088 respectively. We can see the time complexity of HGTM is $H_d^{max} \cdot G_{\mathbf{h}_d}^{max}$ times of LDA. Therefore, when topic number T is fixed, HGTM has the linear

.

3.7 Connection with Author Topic Model

Here we make a clear comparison between HGTM and ATM [30]. When τ equals to 1, HGTM degenerates into ATM: the latent hashtag assignment can only come from the tweet's hashtag set.

In a real-life scenario, users' arbitrary activity of adding hashtags often results in various hashtags related to the same event or topic. For example, "#jan25", "#Cairo" and "#25jan" are all hashtags used to discuss "the Egypt Revolution". However, there are two differences between these hashtags: 1) "#jan25" and "#Cairo" have many occurrences in tweets. 2) Users rarely use "#jan25" and "#25jan" together, but they have a chance of being added with the same URLs. Traditional ATM only can detect the equal semantic relations under the hashtag co-occurrence in one tweet. However, when two hashtags have another semantic relationship, such as co-occurring with the same URLs or used by many of the same authors frequently, ATM will fail to model such flexible and complex information. What's more, different kinds of co-occurrence frequencies reflect the degree of semantic relations between hashtags and further tell the degree of semantic relations between words that these hashtags are related to. ATM couldn't capture this vital information as well. Even though users add only some of all the correlated hashtags in one tweet, people have common sense with all of these correlated hashtags.

Our two-step hashtag sampling process (the first stage is decided by explicit hashtags, the second stage is decided by both explicit hashtags and potential hashtags) can model randomness during hashtag selection. Handling various hashtag relationships provides us a way of connecting semantically-related words with or without co-occurrence. This is HGTM's main improvement during the generative process for a corpus.

4 Experimental Analysis

In this section, we first empirically evaluate the effectiveness and efficiency of HGTM and fulfill the task of text clustering, hashtag clustering and hashtag classification. Then we analyze the topics we learned.

4.1 Experimental Settings

4.1.1 Datasets

In order to verify the effectiveness of our model, we use two tweet collections observed in different years for evaluation.

- **Tweet2011** collection[1] is a standard tweet collection published in TREC 2011 microblog track. It contains nearly 16 million tweets sampled from January 23rd to February 8th, 2011. Users discussed much about "Super bowl 2011"[2] and "Egyptian Revolution of 2011"[3] during that period.
- **Tweet2015** collection is a subset collection that we crawled Twitter.com by selected hot keywords[4] from June 17th to June 23rd, 2015. Users discussed much about "Charleston church shooting"[5] and some related gunshots (Newton Tragedy[6] and Aurora Shooting[7]), "Father's Day"[8] and "Uber"[9] during that period.

TABLE 1
Summary of the two tweet collections.

Dataset	#tweet	#word	#hashtag	avgDocLen	avgHashtag
Tweet2011	333,491	12,420	106,682	5.22	1.42
Tweet2015	250,306	8,300	66,384	7.22	1.76

The raw data of these collections is very noisy. To reduce low-quality tweets, we process the raw dataset via similar normalization steps as Biterm Topic Model [19] does. We also do stemming by using Stanford NLP tools[10]. Finally, we conduct our experiments on tweets with hashtags and remove retweets. Table 1 shows the number of tweets, distinct words and distinct hashtags, the average length (i.e., number of words) of tweets and the average hashtag numbers of tweets in the two collections after preprocessing.

We divide each dataset into a training and a testing set. Due to the fact that HGTM does not deal with time dimension, we split the two datasets with different strategies to see whether our model is time-sensitive. For Tweet2011, the training set contains tweets from January 23rd to February 6th, used for inferring the model's parameters, and the remaining tweets from February 7th to February 8th are in the testing set. For Tweet2015, we randomly split the dataset into training and testing subsets with the same ratio 7 : 1 as we do with Tweet2011. So experiment results for Tweet2011 are time-sensitive, those for Tweet2015 are non time-sensitive.

4.1.2 Baseline Methods

We compared HGTM with ten typical models for tweet mining.

- **VSM**, the traditional **V**ector **S**pace **M**odel, which represents a tweet as word frequencies in a vector space.
- **LSA** [2], the **L**atent **S**emantic **A**nalysis model, which decomposes the "document-word" matrix by Singular Value Decomposition.
- **LSAH**, which aggregates the tweets containing the same hashtag to a pseudo document before training.
- **LDA** [1], the standard **L**atent **D**irichlet **A**llocation, which takes each tweet as a document.
- **LDAH** [10] [11], which has the same aggregation strategy with LSAH and learns LDA parameters with pseudo documents.
- **LDAHW**, the variant **L**atent **D**irichlet **A**llocation including **H**ashtags as **W**ords, which does the same as Tag-LDA [25] does and treats each hashtag as a word in tweets.
- **LDAHGW**, the variant **L**atent **D**irichlet **A**llocation including **H**ashtags from **G**raphs as **W**ords, which extends a tweet with other hashtags of high co-occurrence frequency with hashtags in that tweet. In this method, we extend words of each tweet using hashtags from hashtag graphs.

1. http://trec.nist.gov/data/tweets/
2. http://en.wikipedia.org/wiki/Super_bowl
3. http://en.wikipedia.org/wiki/Egyptian_uprising
4. http://kdd.nankai.edu.cn/sourcecode/HGTM.html
5. https://en.wikipedia.org/wiki/Charleston_church_shooting
6. https://en.wikipedia.org/wiki/Sandy_Hook_Elementary_School_shooting
7. https://en.wikipedia.org/wiki/2012_Aurora_shooting
8. https://en.wikipedia.org/wiki/Father's_Day
9. https://en.wikipedia.org/wiki/Uber_(company)
10. http://nlp.stanford.edu/software/corenlp.shtml

- **TWTM** [28], the **T**ag-**W**eighted **T**opic **M**odel, which infers a topic distribution for each individual document with a function of tag-weighted topic assignment.
- **TWDA** [29], the **T**ag-**W**eighted **D**irichlet **A**llocation, which is based on TWTM, adds a Dirichlet prior to assume that each document exists a latent tag that can impact the topic distribution of the document as well.
- **ATM** [30], the **A**uthor **T**opic **M**odel, where we treat hashtags as authors of a tweet, as Tsai [32] did for blog mining.

In our experiments, we explore our hashtag graphs via different graph construction schemes. Firstly, we evaluate them on text clustering and hashtag clustering tasks. Secondly, we evaluate our HGTM on hashtag classification. Thirdly, we show details of topics learned by models. For different methods of graph construction, we use HGTM-L, HGTM-R, HGTM-T to denote our models when the hashtag graph is constructed by one hashtag relationship which is "inserted with the same URLs", "added by the same group of users" and "showing up in the same tweets" respectively. Similarly, we use LDAHGW-L, LDAHGW-R, LDAHGW-T to differentiate the hashtag graphs used for tweet extension in LDAHGW. HGTM (or LDAHGW) is still a general name for them all. Note that these three kinds of graphs have overlaps in nodes and edges, and convey different semantic information as well. Such complexity makes it hard to combine them by naive methods like linear combination, so we leave it as future work.

The number of topics T is fixed at 60 which is obtained through a cross validation set. The hyperparameters in generative models (LDA, LDAH, LDAHW, LDAHGW, ATM, HGTM) are set at $50/T$ for α and 0.01 for β respectively. We ran 5 independent Gibbs sampling chains for 2000 iterations on two datasets. For exclusive parameters of TWTM and TWDA, we set them as default in their released code[11]. For LDAHGW, we extend a tweet using top-10 hashtags from hashtag graphs with the highest degree of co-occurrence with hashtags in this tweet. In HGTM, we set τ as 0.7 for a little preference to select explicit hashtags in current tweets.

4.2 Clustering

This part discusses the effectiveness of different methods of graph construction by clustering performance of HGTM.

4.2.1 Evaluation Metrics

We aim to evaluate the effectiveness of HGTM algorithms on different hashtag graphs for tweets. In recent years, many works [19] [36] [37] [38] show that topic modeling identifies topic distributions in a document collection, which can effectively identify clusters in a collection. Topic modeling is a viable way to quantify document similarity, so it helps to cluster documents. After reducing representation dimension of a document by topic models, we can calculate similarity between documents in a semantic (topic) space. So our evaluation is based on quantified similarity measures and clustering requests. The good clusters should have lower intra-cluster distances and higher inter-cluster distances.

We denote tweet representation as $\mathbf{d}$. In VSM, we use word frequencies in a vector space to represent a tweet. In LDA, LDAHW, LDAHGW, TWTM and TWDA, $\mathbf{d}$ is the topic distribution of a tweet inferred by the model. LDAH, ATM and HGTM do not explicitly model topic distributions of tweets, but they can infer topic assignments for words in a tweet. Denote tweet d's topic assignment count vector as C^d, where C^d_t is the number of words that have been assigned to a specific topic t in tweet d. To get $\mathbf{d}$, we normalize a vector to sum to 1. By LSA and LSAH, each tweet is mapped to a low-dimensional vector.

We use cosine similarity to measure the degree of similarity between two tweets. So the distance between two tweets is

$$dis(d_1,d_2) = 1 - \frac{\mathbf{d_1}\cdot\mathbf{d_2}}{\|\mathbf{d_1}\|\|\mathbf{d_2}\|}. \tag{11}$$

The average intra-cluster distance is

$$IntraDis(C) = \frac{1}{K}\sum_{k=1}^{K}[\sum_{d_i,d_j\in C_k, i\neq j}\frac{2dis(d_i,d_j)}{\|C_k\|\|C_k-1\|}]. \tag{12}$$

The average inter-cluster distance is

$$InterDis(C) = \frac{1}{K(K-1)}\sum_{C_k,C_{k'}\in C,k\neq k'}[\sum_{d_i\in C_k}\sum_{d_j\in C_{k'}}\frac{2dis(d_i,d_j)}{\|C_k\|\|C_{k'}\|}]. \tag{13}$$

If average intra-cluster distance is much less than the average inter-cluster distance, it means that model achieves a clearer topic description. So we calculate the ratio H-score between them. Smaller H-score represents better performance.

$$H = \frac{IntraDis(C)}{InterDis(C)} \tag{14}$$

We use H-score [19] to measure the performance of models on both tweet clustering and hashtag clustering.

4.2.2 Tweet Clustering

For text clustering, there is no obvious category information in microblogging data sets. Thus we take hashtags as cluster labels. Thus tweets with the same hashtags are automatically assigned to the same cluster. We manually take 50 frequent hashtags that mark events or topics as our cluster labels (shown in Table 2). Note that it is possible for a tweet to belong to more than one clusters when the tweet contains two or more selected hashtags for tweet clustering experiments on Tweet2011. It indicates the semantic overlap relationship between topics of two clusters labeled by hashtags, such as topics about hashtag "#weather" and hashtag "#snow". Nevertheless, we added constraint to the testing data on Tweet2015, where we limited only one cluster for each testing tweet on Tweet2015 to see the difference.

The results are shown in Table 3. From the table, we have the following conclusions. (1) HGTM achieves the best performance with each kind of hashtag graphs. HGTM-T achieves the best performance with improvement of 0.024 to HGTM-L and 0.016 to HGTM-R on Tweet2011. It proves that for a mixed tweet data, hashtag co-occurrences in tweets show stronger and closer semantic relationship of words than URL-based relationship and user-based relationship. Based on the performance of HGTM-R, it is conclusive that the degree of users' similar interests is worse for defining semantic relationship than word co-occurrence in short tweets (used in HGTM-T), but it is still better than URL-based hashtag relationships (used in HGTM-L). However, for Tweet2015, HGTM-L achieves the best performance among different hashtag graphs. Here is a probable reason: the content

11. https://github.com/Shuangyinli

TABLE 2
Hashtags selected for evaluation.

Tweet2011	**Tweet2015**
#jan25 #superbowl #sotu #tcyasi	#espn #charleston #gunsense
#wheniwaslittle #mobsterworld #jobs	#sports #racism #bitchimmadonna
#agoodboyfriend #bieberfact #glee	#pjnet #confederate #android #ff
#lfc #rhoa #itunes #thegame	#origin #nra #football #mufc
#celebrity #americanidol #cancer	#2a #soccer #quote #tbt #nfl #cnn
#socialmedia #jerseyshore #kindle	#technology #tech #blacklivesmatter
#jp6foot7remix #meatschool	#lgbt #batman #nowplaying #ytff
#factsaboutboys #libra #android	#nbafinals2015 #mlb #afc #uber
#sagittarius #thissummer #tnfisherman	#life #tcot #welcomebackway5
#sagawards #ausopen #bears #weather	#iran #ebay #thesuperhuman
#jaejoongday #skins #bfgw #fashion	#card #obama #dylannroof
#pandora #realestate #teamautism	#endausteritynow #confederateflag
#travel #nba #football #marketing	#gameinsight #madonna #jon_stewart
#design #oscars #food #dating	#music #ameshooting #closeupatytff
#snow #obama #photography	#drswamy4bapujibail #nba

of each URL can be related to one or multiple aspects, and tweets that have different hashtags including the same URL maybe talk about the same or different aspects about the same topic in different cases. (2) The performance of LDAHGW shows that hashtags from our hashtag graphs can benefit traditional topic model LDA with an average improvement around 0.27, but still can not beat HGTM. (3) The traditional models improve in text clustering after the aggregation strategy is implied. LDAH and LSAH outperform LDA by around 0.1 and LSA by around 0.04 respectively on both of two datasets. The results verify that word co-occurrence frequency has a great impact on LSA and LDA. Due to data sparsity and noise, LSA and LDA could not get distinguished results even when aggregating tweets by user-contributed hashtags. (4) VSM is the worst for semantic similarity measure on Tweet2011. It shows that the arbitrary nature of languages has severely impacted VSM, while it has affected ATM and HGTM less. However, VSM performs better than LSA and LSAH on the keyword-specific dataset Tweet2015. (5) TWTM and ATM consider the hashtag co-occurrence in a single tweet, and exceed most of other models except HGTM. We can infer that modeling the strength of semantic relationships of words by considering both explicit hashtags and potential hashtags via two-step sampling is quite helpful. Potential hashtags do bridge words and beat the sparse problem. (6) Even though TWDA adds a Dirichlet prior to TWTM, it fails to model semantic distance of short texts like tweets.

Furthermore, we know that LDAH explicitly collects words co-occurring with a hashtag and solves the sparse problem to some extent. However, it still can not beat HGTM. That is because LDAH links words to every co-occurred hashtag, but ATM and HGTM link words with one of the co-occurred or related hashtags in each tweet during hashtag assignment process. Note that the relationship between words and hashtags in a tweet has been modeled as "AND" relationship in LDAH and "OR" relationship in ATM and HGTM. The "AND" relationship assigns every hashtag to each word in one tweet, while the "OR" relationship assigns one of these hashtags to each word in one tweet. When dealing with a polysemic hashtag, LDAH may mix irrelevant words within aggregational documents, thus it provides some unreasonable co-occurrences. Unreasonable word co-occurrences would result in fuzzy word distributions over topics. Meanwhile, ATM and HGTM enhance meaningful word co-occurrences via hashtags and reduce ambiguity. So ATM and HGTM can learn a better topic structure for a tweet corpus.

TABLE 3
H-score for text clustering. A smaller H-Score indicates better clustering performance.

	Dataset	
Method	**Tweet2011**	**Tweet2015**
VSM	0.961	0.575
LSA	0.877 ± 0.001	0.645 ± 0.001
LSAH	0.838 ± 0.002	0.602 ± 0.001
LDA	0.817 ± 0.001	0.574 ± 0.004
LDAH	0.718 ± 0.002	0.482 ± 0.002
LDAHW	0.562 ± 0.001	0.395 ± 0.001
LDAHGW-L	0.514 ± 0.005	0.300 ± 0.003
LDAHGW-R	0.562 ± 0.005	0.344 ± 0.006
LDAHGW-T	0.570 ± 0.006	0.325 ± 0.001
TWTM	0.469 ± 0.003	0.297 ± 0.002
TWDA	0.564 ± 0.002	0.336 ± 0.006
ATM	0.477 ± 0.003	0.286 ± 0.002
HGTM-L	0.467 ± 0.002	**0.272 ± 0.002**
HGTM-R	0.459 ± 0.001	0.295 ± 0.003
HGTM-T	**0.443 ± 0.003**	0.286 ± 0.001

TABLE 4
Label information of hashtags.

Label (#hashtag)	Examples
IDIOMS (126)	#ihate, #cantcandidateyou, #followback
POLITICAL (39)	#Jan25, #tcot, #glennbeck, #obama, #hcr
TECHNOLOGY (57)	#nikeplus, #teamautism, #amwriting
SPORTS (42)	#golf, #yankees, #nhl, #cricket, #lakers
MOVIES (32)	#lost, #glennbeck, #bones, #newmoon
CELEBRITY (4)	#mj, #brazilwantsjb, #regis, #iwantpeterfacinelli
GAMES (13)	#mafiawars, #spymaster, #mw2, #zyngapirates
MUSIC (23)	#lastfm, #thisiswar, #musicmonday, #pandora

4.2.3 Hashtag Clustering

Hashtag topic distribution is an important by-product of HGTM. In order to illustrate that our method is susceptible to different categories of hot topics and events, we evaluate the model by a hashtag clustering task. The aim is to see the capacity of HGTM to distinguish hashtags with different semantic domains. We aggregate the tweets containing the same hashtag to construct a pseudo document for the hashtag to calculate its word vector representation in SVM and inferring its topic distribution in LSA, LDA, LDAHW and LDAHGW. For LSAH, LDAH, TWTM, TWDA and ATM, we directly obtain hashtag topic distribution from model parameters. We use manual hashtag label information [39] as cluster labels. The ambiguous hashtags in the "NONE" cluster have been removed, and finally 336 hashtags in 8 clusters are used in our experiment. The details are shown in Table 4. We take the same evaluation metric (H-score) as tweet clustering in Section 4.2.2 for evaluation. Table 5 presents the results.

From results in Table 5, we observe that HGTM performs significantly better than other baseline models on hashtag clustering. With different schemes of hashtag graph construction, performance increases from the URL-based HGTM-L, the user-based HGTM-R to the tweet-based HGTM-T on Tweet2011, while performance increases in reverse order on Tweet2015. Our model takes advantage of the semantic bridge built by hashtag graphs and achieves H-score of 0.552 on Tweet2011 and 0.610 on Tweet2015. Even though the tweet-based HGTM-T achieves the best performance on Tweet2011, HGTM-R and HGTM-L still

have their own advantages. As mentioned in Section 3.6, when the size of the corpus and the number of topics are fixed, the time complexity of models is affected by the maximum number of hashtags in a tweet H_d^{max} and the maximum number of hashtags that one hashtag can connect to in the hashtag graph $G_{\mathbf{h}_d}^{max}$. According to the corpus, H_d^{max} is constant (17 on Tweet2011, 9 on Tweet2015), while $G_{\mathbf{h}_d}^{max}$ changes according to different kinds of hashtag graphs, from 329 (URL-based), 3088 (user-based) to 1198 (tweet-based) on Tweet2011 and from 99 (URL-based), 998 (user-based) to 199 (tweet-based). The HGTM-L takes the least amount of time for inference and achieves the best performance on the keyword-specific Tweet2015.

As with the results of tweet clustering, the performance of unsupervised semantic methods can be improved by an aggregation strategy in the case of noisy text data. LSAH and LDAH achieve a lower H-score than LSA and LDA respectively. In particular, LDAH outperforms LDA by 30.1% $((0.991-0.693)/0.991)$ improvement on Tweet2011 and 29.7% $((1.006-0.707)/1.006)$ improvement on Tweet2015. Comparing LSA-type methods to LDA-type methods, the latent topic structure discovered by LDAH is more suitable for hashtag semantic understanding and has more consistent results with human labeled clusters.

Additionally, we know from the results in Table 5 that LDA, LDAHW and LDAHGW are even worse than VSM. In particular, LDA and LDAHW perform much worse than others with a H-score larger than 1 on Tweet2015. We learn that due to short and casual tweets, modeling topic relevance information from a single post probably captures only weak and indirect semantic description around topics for each word (or hashtag). However, aggregated messages give us a more accurate and comprehensive topic illustration [9] [10] [11]. SVM can take advantage of integration in the testing phase, because the model is only a counting process. As to LDA, LDAHW and LDAHGW, ambiguous and noisy topic information learned in the training process is accumulated and amplified in the aggregated pseudo documents. LSA extract topic concepts via a low-dimensional approximation process and discards the noise component. It explains why LSA-style methods are much better than LDA-style methods. Even though LDAH would mix irrelevant words due to polysemic hashtags as Section 4.2.2 mentions, LDAH achieves a better performance than ATM. The reason is that LDAH has much more impact on polysemic hashtags than hashtags that have a single topic. Due to sparseness in tweets, TWTM and TWDA both fail to represent hashtags in a distinguishable semantic space although they perform good on text clustering.

TABLE 5
H-score for hashtag clustering. A smaller H-Score indicates better clustering performance.

	Dataset	
Method	**Tweet2011**	**Tweet2015**
VSM	0.946	0.906
LSA	0.823 ± 0.001	0.768 ± 0.002
LSAH	0.751 ± 0.001	0.718 ± 0.001
LDA	0.991 ± 0.002	1.006 ± 0.001
LDAH	0.639 ± 0.001	0.707 ± 0.001
LDAHW	0.994 ± 0.002	1.009 ± 0.001
LDAHGW-L	0.980 ± 0.004	0.995 ± 0.003
LDAHGW-R	0.977 ± 0.005	0.993 ± 0.001
LDAHGW-T	0.974 ± 0.005	0.996 ± 0.003
TWTM	0.978 ± 0.002	0.935 ± 0.002
TWDA	0.995 ± 0.002	0.890 ± 0.001
ATM	0.659 ± 0.003	0.635 ± 0.002
HGTM-L	0.589 ± 0.002	**0.610 ± 0.001**
HGTM-R	0.573 ± 0.003	0.611 ± 0.002
HGTM-T	**0.552 ± 0.002**	0.622 ± 0.002

4.3 Hashtag Classification

In addition to hashtag clustering, we conduct a classification experiment on the same hashtags used in Section 4.2.3 to explore the linear separability of semantic represented hashtags by different methods (hashtag representations are the same as Section 4.2.3). We use a linear Support Vector Machine (SVM) open source tool, libSVM[12], to train a linear classifier and set SVM regularization parameters by grid search. Figure 4 shows performance on 5-fold cross-validation of models on Tweet2011 and Tweet2015.

From the results shown in Figure 4, we see that HGTM dominates the ten baselines, increased by 34.61% (compared with LDAHGW-L) at most and 7.37% (compared with SVDH) at least on Tweet2011. LSA and LSAH are ranked in the second place on both Tweet2011 and Tweet2015. On the Tweet2011 collection, the performance of two methods both are over 60%, while all of the other topic models, such as ATM, LDA and LDAH are all far below. We deduce that LSA can discard noise in tweet data, and then improve generalization of data representation. The results of LSA make a good explanation for sparsity and noise of tweets. LSA and LSAH map hashtags represented by the related tweets into a linearly separable space. Traditional LDA performs even 3.51% worse than VSM, while aggregational LDA improves by 6.34% and achieves a little better performance than VSM. ATM, which only considers hashtag appearance in one tweet, is far below LSA and LSAH as well. It indicates that it's vital to model correlation carefully for freestyle documents, especially when they are so short. Meanwhile, LDAHW and LDAHGW fall to the bottom of performance. Thus, it's inappropriate to treat hashtags just as words for hashtag semantic learning. The co-occurrences of related hashtags and words in tweets are far from enough to make the meanings of hashtags clear. Even though TWTM and TWDA achieve good performance on tweet clustering, they can not understand hashtags well. Further investigation finds that TWTM and TWDA share the similar idea with ATM, and they all take advantage of internal tags. However, the tag weighted schema in both TWTM and TWDA, and even the Dirichlet prior in TWDA are not suitable for hashtag modeling in short texts. Besides, the statistical information in hashtag graphs is a better choice on this problem. We get similar results on Tweet2015.

Graph associated information in data pushes tweets from flat data to semi-structured data, where connection is vital to model semantics out of noise and sparseness. Topic information of hashtags is shown via discussion among people and language exchange, which is the key point of HGTM. Thus, HGTM's performance on hashtag classification is striking, because it models both the gap between co-occurrence of hashtags in a tweet and the statistical relationship of hashtags in the whole dataset.

4.4 Quality of Topics

In this section, we show more details of topics learned by models. We first evaluate topic coherence, and then show the distribution of topics learned by different models, and finally observe most probable words and hashtags of topics learned by HGTM.

12. http://www.csie.ntu.edu.tw/~cjlin/

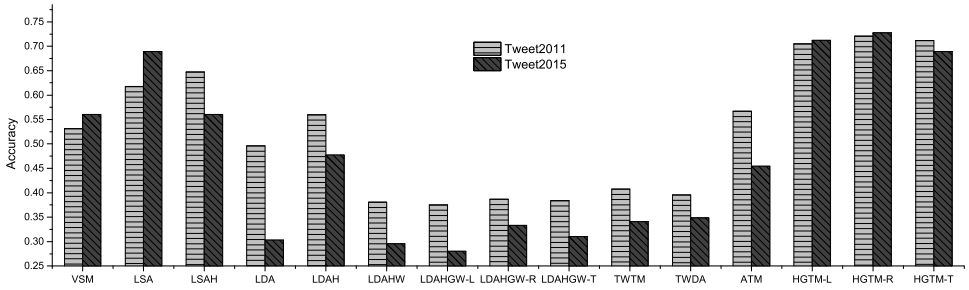


Fig. 4. Accuracy on hashtag classification.

4.4.1 Topic Coherence

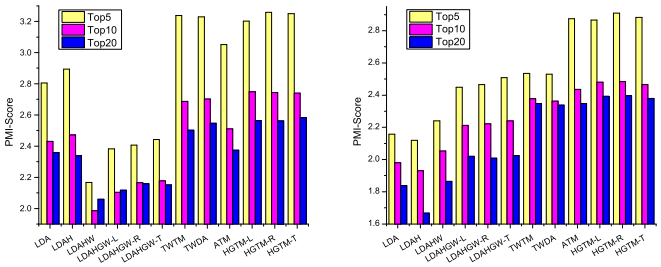
How to evaluate topic models is an open question. From the view of models, it's typical to use perplexity or marginal likelihood evaluation metrics [1] [40]. However, for text understanding and organization, we expect that topic models have the ability to learn interpretable topics, organize and summarize documents effectively. There is no technical reason to show that topic models with a higher held-out likelihood can give a credible result that could be easily understood and explained [41]. In recent years, some automatic evaluation methods are proposed to measure the quality of the topics discovered. One is the PMI-Score [42], which broadly agrees with human-judged topic coherence. PMI-Score measures the coherence of a topic based on pointwise mutual information using external text data sources, e.g., Wikipedia. Due to model-independence of the external data set, PMI-Score is fair for all the topic models. Therefore, we exploit PMI-Score to verify topic quality.

Given the M most probable words of topic k, $(w_1, \cdots, w_M)$, PMI-Score is motivated by measuring word associations between them:

$$PMI\text{-}Score(k,M) = \frac{1}{M(M-1)} \sum_{1 \le i < j \le M} PMI(w_i, w_j), \qquad (15)$$

where $PMI(w_i,w_j) = log\frac{P(w_i,w_j)}{P(w_i)P(w_j)}$, $P(w_i,w_j)$ and $P(w_i)$ are the probabilities of co-occurring word pair (w_i,w_j) and word w_i estimated empirically from the external data sets, respectively. For a topic model of T topics, PMI-Score will refer to the average of T PMI-Scores. We compute the PMI-Score using 4M English Wikipedia articles to evaluate topic model coherence on Tweet2011 and Tweet2015.

Figure 5 shows the results on the Tweet2011 and Tweet2015 collections with the number of most probable words M ranging from 5 to 20. We see that HGTM with different hashtag graphs outperforms LDA and all LDA based models on both the two datasets. Aggregational strategy, like LDAH and word extension methods, like LDAH or LDAHW fall far behind HGTM. On the mixed dataset Tweet2011, LDAHW and LDAHGW perform even worse than LDA and LDAH, but the contrary is the case on keyword specific dataset Tweet2015. LDAH does not always perform well on datasets with different sampling settings. It performs much better on mixed Tweet2011 dataset than on keyword-based Tweet2015. Thus aggregating documents cannot fully resolve the sparsity problem. The performance of TWTM, TWDA and ATM is rather changeable on different datasets as well. TWTM and TWDA can discover more coherent topics on miscellaneous data (Tweet2011), but surprisingly fall behind ATM on Tweet2015. The results show that HGTM performs more stable than other models



(a) Tweet2011 (b) Tweet2015

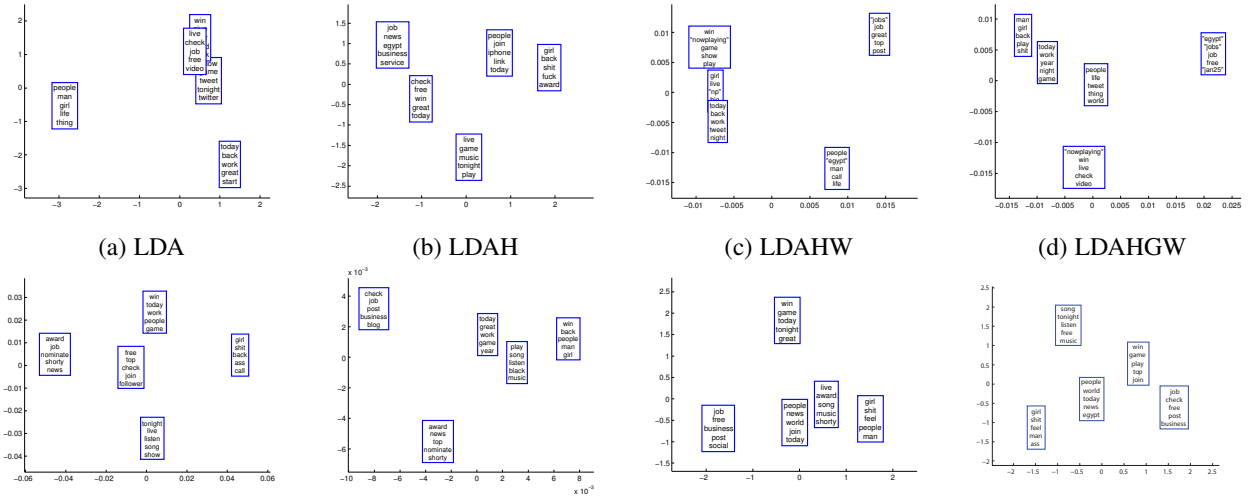
Fig. 5. PMI-Scores on (a) Tweet2011 and (b) Tweet2015 data sets. A larger PMI-score indicates more coherent topics.

on different datasets and beats several previously published topic models designed for specific data. As to different hashtag graphs, HGTM-R outperforms both HGTM-L and HGTM-T. Thus, user-based hashtag relationship releases more consistent semantic information than the other kinds of hashtag relationships.

4.4.2 Space Mapping Analysis

This section focuses on comparing topic distribution learned by each topic model (LDA, LDAH, ATM and HGTM). We use a classical multidimensional scaling technique to visualize all pairwise topic distances on a 2D map. For each topic pair, the symmetrized Kullback Leibler distance between topic distributions is calculated. We collect the top-5 probable words of each topic to represent topic points in a 2D map. Figure 6 shows the results with top-5 probable words (or hashtags enclosed in quotes in LDAHW or LDAHGW) of each topic. The size of the rectangle is defined by the length of words and irrelevant to the probability of topics. Due to space limitation, we show the results on Tweet2011. Here, we choose the fastest URL-based LDAHGW-L and HGTM-L to show the results of LDAHGW and HGTM.

As shown in Figure 6, topics learned by LDA have a serious confusion. Except topic "people-man-girl-life-thing" and topic "today-back-work-great-start", other three topics are very close in the mapping space. LDAH scatters topics, but it could not get clear topics, such as topic "job-news-egypt-business-service" at $(-1.6, 1)$. We can see both political event "Egyptian Revolution of 2011" and business affairs in it. Topic "check-free-win-great-today" and topic "people-join-iphone-link-today" have the same problem. Topics learned by LDAHW are still close to each other and hard to be linearly separated. LDAHGW mixes up topic "job" and topic "egypt" in " 'egypt'-'jobs'-job-free-'jan25' ". TWTM and TWDA show us linearly separable topics, a few of them has prominent themes. They can learn topic "jobs" and topic "music", but the other topics are still not clear. ATM and HGTM both can



(a) LDA (b) LDAH (c) LDAHW (d) LDAHGW

(e) TWTM (f) TWDA (g) ATM (h) HGTM

Fig. 6. Topic distributions from different topic models mapped in a 2D map on Tweet2011. The number of topics is 5 for each model.

give linearly separable topic distributions. Furthermore, HGTM learns more semantically clear topics, such as "girl-shit-feel-man-ass" for slang, topic "people-world-today-news-egypt" for politics and world affairs, and topic "song-tonight-listen-free-music" for music. Interestingly, we find that the "slang" topic is close to topics about politics and world affairs in the 2D map. The model discovers that people always express emotion with slang when discussing popular news.

4.4.3 Quality Evaluation

We further investigate the content of topics to study the quality of topics and representative hashtags discovered by our model. The topics that are well readable, easily understandable and easy to distinguish show the rationality of latent topic structures. Table 6 gives examples of 4 topics (out of 60) learned by HGTM-L on Tweet2011 and Tweet2015, respectively. We use one word with the highest probability to denote that topic.

Referring to the results in Table 6(a), the top 10 words and top 10 hashtags are highly related to the specific topic. Taking topic "EGYPT" on Tweet2011 for example, we discover the most important key words, such as "egypt", "people", "obama", "mubarak" and "police". Meanwhile, HGTM finds highly related hashtags, such as "#egypt", "#jan25", et al. For topic "SONG", the top-10 words are much prominent and precise about music things. "#nowplaying", "#np", "#music", "#lastfm" and "#soundcloud" are typical hashtags related to music. It also helps new users who are not familiar with Twitter to guess that "#nowplaying" is the expanded form of "#np". Topic "GAME" is oriented towards popular sport events "Super Bowl" at that time. The topic contains the common words we use when discussing and goes along with a hashtag list showing some favorites, such as "#nfl" (the National Football League), "#steelers" (the Pittsburgh Steelers) and "#packers" (the Green Bay Packers). The discussion about news of heavy snow shows us many connections with cold weather and people's feelings (see topic "SNOW"). Besides, we discover that some functional hashtags, such as "#fb" (this hashtag is used by people who have installed the Selective Twitter Update application on Facebook. Tweets ending in "#fb" are automatically imported to Facebook), dominate in multiple topics. Even though hashtag "#fb" isn't a clear topic indicator, it tells hot topics in Facebook and their relationships. This situation disappears on the keyword-based data set Tweet2015. For Tweet2015, HGTM can discover coherent topics during the period from June 17th to June 23rd, 2015, such as topic "SHOOTING" related to gunshots and topic "JAMES" related to 2015 NBA Finals. Therefore, hashtag graph information benefits both latent topic structure modeling and hashtag semantic mining, which can help further applications on Twitter, such as information organization and retrieval.

5 Discusion and Conclusion

Uncovering topics within tweets has become a vital task for widespread content analysis and social media mining. Different from modeling normal text, tweet mining has suffered a great deal of sparseness and informality problems. In this work, we consider that users have provided hashtags as a powerful and valuable data source in the vast amount of tweets on the web. This paper presents HGTM that first introduces the hashtag relation graphs as weakly-supervised information for tweet semantic modeling. We demonstrate that hashtag graphs contain reliable information to bridge semantically-related words in sparse short texts.

HGTM can enhance semantic relations between tweets and reduce noise at the same time. Compared to single document-oriented topic models (e.g., LSA, LDA, ATM, TWTM, TWDA), HGTM has a better ability to capture semantic relations between words with or without co-occurrence by utilizing the wisdom of crowds from user-generated hashtags. The model provides a more robust solution for tweet modeling than aggregation strategies with traditional topic models. We also prove that LDA framework inherently can not benefit from hashtag graphs. We achieve significant improvement on the performance of content mining tasks, such as tweet clustering, hashtag clustering and hashtag classification. HGTM discovers more readable and distinguishable topics than the stat-of-the-art models as well.

This paper shows one effective alternative of utilizing user-contributed hashtags for tweet topic modeling to handle both sparseness and noise in tweets. However, there are still many questions which need to be explored. For example, we would like to explore reasonable and effective ways of combining multi-modal hashtag relations for tweet modeling and to model time-

(a) Results on Tweet2011

		EGYPT	SONG	GAME	SNOW
Tweet2011	WORDS	egypt people obama mubarak egyptian police protest president state news	song listen album play radio feat sound live mix music	game super bowl show fan year tonight green play team	snow home today feel morning wind weather cold tomorrow feb
	HASHTAGS	#egypt #25-Jan #tcot #news #sotu #p2 #fb #mubarak #dearjohn #tahrir	#nowplaying #np #music #lastfm #fb #soundcloud #itunes #blogtalkradio #listeningto #pandora	#superbowl #fb #steelers #nfl #packers #sb45 #bears #jets #sotu #nowplaying	#weather #fb #wdisplay #nowplaying #tcyasi #news #realestate #np #egypt #travel

(b) Results on Tweet2015

		SHOOTING	VIDEO	JAMES	JOB
Tweet2015	WORDS	shooting church family killed tragedy prayer victim violence white black	video madonna new music bitch amp radio tune kanya feat	james lebron nba draft state golden final warrior cleveland player	job team hiring looking apply detail opening view manager new
	HASHTAGS	#charlestonshooting #charleston #blacklivesmatter #dylannroof #prayersforcharleston #ameshooting #prayforcharleston #tcot #p2 #confederate	#nowplaying #madonna #bitchimmadonna #ma #listenlive #soundcloud #makemoney #aidansnewvideo #music #internetmarketing	#nba #sports #nbafinals2015 #news #basketball #nbafinals #lebron #football #warriors #nfl	#jobs #job #hiring #tweetmyjobs #hr #itjob #local #sydney #news #denver

TABLE 6
An illustration of 4 topics from a 60-topic solution. Each topic is shown with the top-10 words and hashtags that have the highest probability conditioned on that topic.

sensitive hashtag relations. The resulting model is highly scalable and could be used in a number of real-world applications, such as hashtag recommendation, short text retrieval, and event detection.

ACKNOWLEDGMENTS

This research is supported by the National Natural Science Foundation of China (No. 61105049 and No. 61300166), the National Science Fund for Distinguished Young Scholars (No. 61222210), the State Key Program of National Natural Science of China (No. 61432011), the Open Project Foundation of Information Technology Research Base of Civil Aviation Administration of China (No. CAAC-ITRB-201303 and No. CAAC-ITRB-201408), the Natural Science Foundation of Tianjin (No. 14JCQNJC00600), the Science and Technology Planning Project of Tianjin (No. 13ZCZDGX01098) and the Tianjin Key Laboratory of Cognitive Computing and Application.

REFERENCES

[1] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet Allocation," *the Journal of machine Learning research*, vol. 3, pp. 993–1022, 2003.

[2] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman, "Indexing by latent semantic analysis," *JASIS*, vol. 41, no. 6, pp. 391–407, 1990.

[3] S. Vieweg, A. L. Hughes, K. Starbird, and L. Palen, "Microblogging during two natural hazards events: What Twitter may contribute to situational awareness," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '10. New York, NY, USA: ACM, 2010, pp. 1079–1088.

[4] Y. Chen, H. Amiri, Z. Li, and T.-S. Chua, "Emerging topic detection for organizations from microblogs," in *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '13. New York, NY, USA: ACM, 2013, pp. 43–52.

[5] J. Chen, R. Nairn, L. Nelson, M. Bernstein, and E. Chi, "Short and tweet: Experiments on recommending content from information streams," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '10. New York, NY, USA: ACM, 2010, pp. 1185–1194.

[6] K. Tao, F. Abel, Q. Gao, and G.-J. Houben, "TUMS: Twitter-based user modeling service," in *The Semantic Web: ESWC 2011 Workshops*, ser. Lecture Notes in Computer Science, R. Garca-Castro, D. Fensel, and G. Antoniou, Eds. Springer Berlin Heidelberg, 2012, vol. 7117, pp. 269–283.

[7] A. Dong, R. Zhang, P. Kolari, J. Bai, F. Diaz, Y. Chang, Z. Zheng, and H. Zha, "Time is of the essence: Improving recency ranking using Twitter data," in *Proceedings of the 19th International Conference on World Wide Web*, ser. WWW '10. New York, NY, USA: ACM, 2010, pp. 331–340.

[8] T. Hofmann, "Probabilistic latent semantic indexing," in *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '99. New York, NY, USA: ACM, 1999, pp. 50–57.

[9] L. Hong and B. D. Davison, "Empical study of topic modeling in Twitter," in *Proceedings of the First Workshop on Social Media Analytics*, ser. SOMA '10. New York, NY, USA: ACM, 2010, pp. 80–88.

[10] R. Mehrotra, S. Sanner, W. Buntine, and 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*, ser. SIGIR '13. New York, NY, USA: ACM, 2013, pp. 889–892.

[11] K. W. Lim and W. Buntine, "Twitter opinion topic model: Extracting product opinions from tweets by leveraging hashtags and sentiment lexicon," in *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, ser. CIKM '14. New York, NY, USA: ACM, 2014, pp. 1319–1328.

[12] Y. Wang, J. Liu, J. Qu, Y. Huang, J. Chen, and X. Feng, "Hashtag graph based topic model for tweet mining," in *Data Mining (ICDM), 2014 IEEE International Conference on*, Dec 2014, pp. 1025–1030.

[13] D. Blei and J. Lafferty, "Correlated topic models," *Advances in neural information processing systems*, vol. 18, p. 147, 2006.

[14] D. Cai, Q. Mei, J. Han, and C. Zhai, "Modeling hidden topics on document manifold," in *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, ser. CIKM '08. New York, NY, USA: ACM, 2008, pp. 911–920.

[15] C. Chemudugunta, P. Smyth, and M. Steyvers, "Modeling general and specific aspects of documents with a probabilistic topic model," in *NIPS*, B. Schlkopf, J. Platt, and T. Hoffman, Eds. MIT Press, 2006, pp. 241–248. [Online]. Available: http://dblp.uni-trier.de/db/conf/nips/nips2006.html#ChemuduguntaSS06

[16] C. Lin and Y. He, "Joint sentiment/topic model for sentiment analysis," in *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, ser. CIKM '09. New York, NY, USA: ACM, 2009, pp. 375–384.

[17] J. Weng, E.-P. Lim, J. Jiang, and Q. He, "Twitterrank: Finding topic-sensitive influential twitterers," in *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, ser. WSDM '10. New York, NY, USA: ACM, 2010, pp. 261–270.

[18] X. Yan, J. Guo, S. Liu, X.-q. Cheng, and Y. Wang, "Clustering short text using ncut-weighted non-negative matrix factorization," in *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, ser. CIKM '12. New York, NY, USA: ACM, 2012, pp. 2259–2262.

[19] X. Yan, J. Guo, Y. Lan, and X. Cheng, "A biterm topic model for short texts," in *Proceedings of the 22Nd International Conference on World*

Wide Web, ser. WWW '13. Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee, 2013, pp. 1445–1456.

[20] X. Hu, L. Tang, and H. Liu, "Enhancing accessibility of microblogging messages using semantic knowledge," in *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, ser. CIKM '11. New York, NY, USA: ACM, 2011, pp. 2465–2468.

[21] W. Zhao, J. Jiang, J. Weng, J. He, E.-P. Lim, H. Yan, and X. Li, "Comparing Twitter and traditional media using topic models," in *Advances in Information Retrieval*, ser. Lecture Notes in Computer Science, P. Clough, C. Foley, C. Gurrin, G. Jones, W. Kraaij, H. Lee, and V. Mudoch, Eds. Springer Berlin Heidelberg, 2011, vol. 6611, pp. 338–349.

[22] T. Lin, W. Tian, Q. Mei, and H. Cheng, "The dual-sparse topic model: Mining focused topics and focused terms in short text," in *Proceedings of the 23rd International Conference on World Wide Web*, ser. WWW '14. New York, NY, USA: ACM, 2014, pp. 539–550.

[23] D. Ramage, D. Hall, R. Nallapati, and C. D. Manning, "Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1*, ser. EMNLP '09. Stroudsburg, PA, USA: Association for Computational Linguistics, 2009, pp. 248–256.

[24] D. Ramage, S. Dumais, and D. Liebling, "Characterizing microblogs with topic models," in *International AAAI Conference on Weblogs and Social Media*, vol. 5, no. 4, 2010, pp. 130–137.

[25] X. Si and M. Sun, "Tag-LDA for scalable real-time tag recommendation," *Journal of Computational Information Systems*, vol. 6, no. 8, pp. 23–31, 2009.

[26] D. Ramage, C. D. Manning, and S. Dumais, "Partially labeled topic models for interpretable text mining," in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '11. New York, NY, USA: ACM, 2011, pp. 457–465.

[27] D. M. Mimno and A. McCallum, "Topic models conditioned on arbitrary features with Dirichlet-multinomial regression," in *Proceedings of the 24th Conference in Uncertainty in Artificial Intelligence*, 2008, pp. 411–418.

[28] S. Li, J. Li, and R. Pan, "Tag-weighted topic model for mining semi-structured documents," in *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, ser. IJCAI '13. AAAI Press, 2013, pp. 2855–2861.

[29] S. Li, G. Huang, R. Tan, and R. Pan, "Tag-weighted Dirichlet Allocation," in *Proceedings of the 13th International Conference on Data Mining*, ser. ICDM'13, vol. 0. Los Alamitos, CA, USA: IEEE Computer Society, 2013, pp. 438–447.

[30] M. Rosen-Zvi, C. Chemudugunta, T. Griffiths, P. Smyth, and M. Steyvers, "Learning author-topic models from text corpora," *ACM Trans. Inf. Syst.*, vol. 28, no. 1, pp. 4:1–4:38, Jan. 2010.

[31] M. Steyvers, P. Smyth, M. Rosen-Zvi, and T. Griffiths, "Probabilistic author-topic models for information discovery," in *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '04. New York, NY, USA: ACM, 2004, pp. 306–315.

[32] F. S. Tsai, "A tag-topic model for blog mining," *Expert Syst. Appl.*, vol. 38, no. 5, pp. 5330–5335, May 2011.

[33] H. Deng, J. Han, B. Zhao, Y. Yu, and C. X. Lin, "Probabilistic topic models with biased propagation on heterogeneous information networks," in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '11. New York, NY, USA: ACM, 2011, pp. 1271–1279.

[34] X. Chen, M. Zhou, and L. Carin, "The contextual focused topic model," in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '12. New York, NY, USA: ACM, 2012, pp. 96–104.

[35] T. Griffiths, "Gibbs sampling in the generative model of Latent Dirichlet Allocation," Stanford University, Tech. Rep., 2002.

[36] A. Drummond, Z. Vagena, and C. Jermaine, "Topic models for feature selection in document clustering," in *Proceedings of the 2013 SIAM International Conference on Data Mining*. SIAM, 2013, pp. 521–529.

[37] P. Xie and E. P. Xing, "Integrating document clustering and topic modeling," *CoRR*, vol. abs/1309.6874, 2013.

[38] R. Huang, G. Yu, Z. Wang, J. Zhang, and L. Shi, "Dirichlet process mixture model for document clustering with feature partition," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 25, no. 8, pp. 1748–1759, Aug 2013.

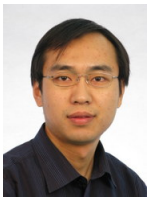
[39] D. M. Romero, B. Meeder, and J. Kleinberg, "Differences in the mechanics of information diffusion across topics: idioms, political hashtags, and complex contagion on Twitter," in *Proceedings of the 20th international conference on World wide web*, ser. WWW '11. New York, NY, USA: ACM, 2011, pp. 695–704.

[40] T. L. Griffiths, M. Steyvers, D. M. Blei, and J. B. Tenenbaum, "Integrating topics and syntax," in *In Advances in Neural Information Processing Systems 17*. MIT Press, 2005, pp. 537–544.

[41] D. M. Blei, "Probabilistic topic models," *Commun. ACM*, vol. 55, no. 4, pp. 77–84, Apr. 2012.

[42] D. Newman, J. H. Lau, K. Grieser, and T. Baldwin, "Automatic evaluation of topic coherence," in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, ser. HLT '10. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010, pp. 100–108.

Yuan Wang received her BS degree in software engineering at Nankai University, Tianjin China, in 2011. She is currently working toward the Ph.D. degree in the College of Computer and Control Engineering at Nankai University. Her current research interests include web search, social media, text mining and machine learning.



Jie Liu is an associate professor in the College of Computer and Control Engineering at Nankai University. His research interests include machine learning, pattern recognition, information retrieval and data mining. He has published several papers on CIKM, ICDM, PAKDD, APWEB, IPM, WWWJ, Soft Computing, etc. He is the coauthor of the best student paper for ICMLC 2013. He has owned the second place in the international ICDAR Book Structure Extraction Competition in 2012 and 2013. He has visited the University of California, Santa Cruz from Sept. 2007 to Sept. 2008. He has visited the Microsoft Research Asia from Aug. 2012 to Feb. 2013. Prior to joining Nankai Univeristy, he obtained his Ph. D. in computer science at Nankai Univeristy.

Yalou Huang is a professor in the College of Software at Nankai University. His primary research covers data mining, information retrieval and intelligent robotics. He has published several papers on SIGIR, WWW, CIKM, ICDM, IPM, WWWJ, Soft Computing, etc. He received the MS degree in computer science and the PhD degree in control engineering from Nankai University, Tianjin, China, in 1990 and 1993, respectively.

Xia Feng is a professor in Civil Aviation University of China, Tianjin, China. Her research interests include text mining, data mining, aviation intelligent information processing. She received her BS degree in information processing and PhD degree in computer science at Nankai University, Tianjin China, in 1991 and 2005, respectively.