Neurocomputing 383 (2020) 282–294

Contents lists available at ScienceDirect

Neurocomputing

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

Generation of topic evolution graphs from short text streams

Wang Gao[a,b,*], Min Peng[b,**], Hua Wang[c], Yanchun Zhang[c], Weiguang Han[b], Gang Hu[b], Qianqian Xie[b]

[a] School of Mathematics and Computer Science, Jianghan University, Wuhan, China
[b] School of Computer Science, Wuhan University, Wuhan, China
[c] Centre for Applied Informatics, Victoria University, Melbourne, Australia

ARTICLE INFO

Article history:
Received 9 January 2019
Revised 18 October 2019
Accepted 25 November 2019
Available online 5 December 2019

Communicated by Dr. Jing Jiang

MSC:
00-01
99-00

Keywords:
Topic evolution graph
Topic model
Short text mining
Word embedding

ABSTRACT

Topic evolution mining on short texts is an important research topic in natural language processing. Existing methods have been focused either on the topic evolution of normal documents or on the evolution of topics along a timeline. In this paper, we aim to generate topic evolutionary graphs from short texts, which not only capture the main topic timeline, but also reveal the correlations between related subtopics. Firstly, we propose an Encoder-only Transformer Language Model (ETLM) to quantify the relationship between words. Then we propose a novel topic model, referred as weighted Conditional random field regularized Correlated Topic Model (CCTM), which leverages semantic correlations to discover meaningful topics and topic correlations. Finally, topic evolutionary graphs are generated by an Online version of CCTM (OCCTM) to capture the evolutionary patterns of main topics and related subtopics. Experimental results on real-world datasets demonstrate our method outperforms baselines on quality of topics and presents motivated patterns for topic evolution mining.

1. Introduction

Short texts have become an important information source in modern society. Examples include instant messages, web page titles, text advertisements, image captions, tweets and questions in Q&A websites. The typical features of these texts are sparse, ambiguous and noisy. Understanding trending topics and tracking their evolution from a large number of unannotated short texts posed by the massive online corpus become fundamental to many applications, such as emerging topic detection [1], automatic summarization [2], sentiment analysis [3], and crisis management [4].

In recent years, the research on topic evolution has attracted wide attention in academic fields such as data mining and Natural Language Processing (NLP), which focuses on discovering what and how topics change over time from time-stamped document collections [5–7]. Most approaches of researching topic evolution are based on probabilistic topic models. Conventional topic models like Probabilistic Latent Semantic Analysis (PLSA) and Latent Dirichlet Allocate (LDA) [8] represent each document as a mixture of topics and each topic as a distribution over words. Incorporating time into these models becomes a popular choice for topic evolution. For instance, Wang et al. proposed an LDA-style model Topics Over Time (TOT) that first jointly models both word co-occurrences and timestamps in a probabilistic topic model [9]. TOT treats time as a continuous variable drawn from a Beta distribution and incorporates it into the model explicitly. Similarly, AlSumait et al. proposed an Online LDA (OLDA) topic model that automatically captures the temporal evolution of topics in data streams [10]. In OLDA, an evolutionary matrix for each topic can be used to capture the evolution of the topic over time. Except for these two models, there are many other topic models that can be used for topic evolution analysis such as Dynamic Topic Models (DTM) [11], non-parametric Topics Over Time (npTOT) [12] and so forth, all of which contribute to many document analysis tasks.

However, there are still many challenges in topic evolution mining. The first one is how to effectively reveal a set of high quality topics from short texts. Traditional topic models have achieved great successes on lengthy documents (*e.g.*, news articles and blogs), but they do not work well on short texts [13]. Since a short text contains only a small number of meaningful keywords, it is

* Corresponding author at: School of Computer Science, Wuhan University, Wuhan, China.
** Corresponding author.
E-mail addresses: gaowang@whu.edu.cn (W. Gao), pengm@whu.edu.cn (M. Peng), hua.wang@vu.edu.au (H. Wang), yanchun.zhang@vu.edu.au (Y. Zhang), han.wei.guang@whu.edu.cn (W. Han), hoogang@whu.edu.cn (G. Hu), xieq@whu.edu.cn (Q. Xie).

https://doi.org/10.1016/j.neucom.2019.11.077
0925-2312/

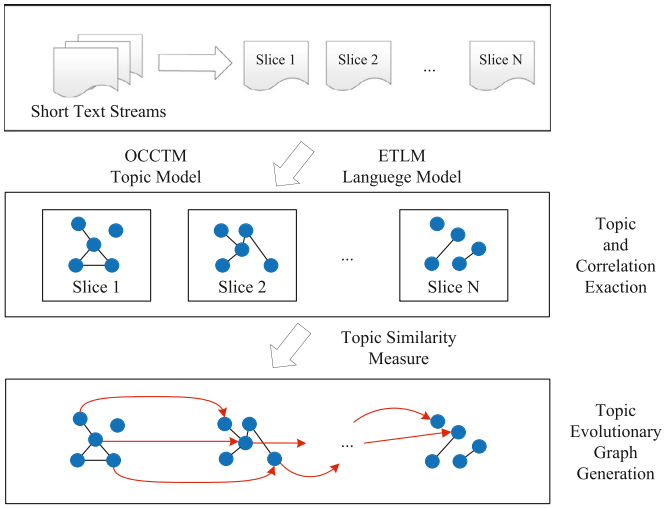


Fig. 1. Overview of the proposed framework for topic evolution graph generation.

difficult to capture the word co-occurrence information. A straightforward strategy is to utilize the internal semantic relationship of words to alleviate the problem of lacking word co-occurrence information [14]. Recently, several attempts leverage pre-trained word embeddings [15] to promote semantically related word pairs under the same topic during the sampling process [16–19]. Nevertheless, these methods ignore the quantifiable relationship between words and fix the amount of promotion (called promotion weight) for each semantically related word pair. Take a text "*Google has the web, Facebook has its app, and Apple has the iPhone.*" as an example. Both ("*Apple, Google*") and ("*Apple, iPhone*") are semantically related word pairs. These methods set the same weight to promote them under the same topic. However, "*Apple*" tends to have a higher probability appearing in the same company-related topic with "*Google*" than "*iPhone*", which depends on the context.

The second challenge is how to construct a topic evolutionary structure to help readers understand the whole evolution of a topic effectively. Existing methods are able to analyze the evolution of topics along a timeline. Zhu et al. proposed a Coherent Topic Hierarchy (CTH) to analyze the topic evolution process on microblog feeds, which is represented by the splitting and merging of local topics from beginning to end along a timeline [5]. Chen et al. designed a framework to effectively model emerging, evolving and fading topics for timely event analysis [7]. However, these approaches pay little attention to the correlation between topics in the same time slice. For a complex event, a single topic timeline fails to display the causality between related events at each time point [6]. As a result, it is difficult for readers to understand the interconnection between correlated topics and the whole evolution quickly.

In this paper, we design an effective framework to address the above challenges. Fig. 1 gives an overview of the proposed framework. We first build a novel topic model to extract high quality topics and topic correlations, referred as weighted Conditional random field regularized Correlated Topic Model (CCTM). Correlated Topic Model (CTM) extends LDA using a logistic-normal prior, which has been widely used to effectively discover correlation structures among latent topics [20–22]. The CCTM model utilizes semantic relatedness knowledge provided by word embeddings by using a Weighted Conditional Random Field (WCRF) [23] on the latent topic layer of CTM to improve the coherence of learned topics. In CCTM, two sets of potential functions are used for modeling both global and local semantic correlations. Unlike other topic models with word embeddings [16–19], the proposed model employs a language model with transformer self-attention [24] to dynamically adjust the promotion weight. Experiments on two real-world short text datasets with classic as well state-of-the-art baselines show the effectiveness of our model in terms of topic coherence and short text classification. In addition, an Online version of CCTM (OCCTM) is presented, which has the important characteristics of not growing in size with time, and it can deal with the dynamic change of vocabulary. Finally, we demonstrate that OCCTM can automatically generate topic evolutionary graphs that are able to capture the main topic timeline and related subtopics from a series of microblog feeds. The main contributions of this paper are summarized as follows.

(1) We propose an Encoder-only Transformer Language Model (ETLM) to dynamically adjust the promotion weight. The quantifiable relationship that ETLM learns from the entire corpus is a powerful complement to the semantic relationship provided by word embeddings learned from large external corpora.
(2) We propose a novel topic model CCTM to discover topics and topic correlations from short texts, and an Online version of CCTM (OCCTM) to automatically generate topic evolutionary graphs. To the best of our knowledge, this is the first work to integrate word correlation knowledge into CTM with the WCRF model.
(3) The experimental results on real-world short text datasets demonstrate the high quality of topics learned by CCTM, and the effectiveness of OCCTM in discovering meaningful patterns for topic evolution mining.

The paper is organized as follows. The second section covers related work. In the third section, we propose our models and present the inference details. Section 4 contains the experiments and finally, Section 5 concludes.

2. Related work

In this section, we briefly summarize related works on topic models on short texts and topic evolution analysis.

Topic models on short texts.

The typical feature of short texts is that they are short in length. Even if two words are semantically related, they are difficult to co-occur in the same short text, which hinders traditional topic models from extracting coherent topics on short text collections. A strategy to increase the word co-occurrence information per document is to aggregate short texts into regular-size texts, which are call pseudo-documents. By modeling the distribution of topics for pseudo-documents rather than short texts, standard topic models are expected to achieve superior performance. For instance, Weng et al. aggregated all tweets from the same user into a pseudo-document before applying the standard LDA model [25]. Other auxiliary information that has been used to merge short texts include named entities, timestamps, and hashtags [14,26,27]. One limitation of this strategy, however, is that such auxiliary metadata may not always be available or just too costly for collection. Therefore, the above methods cannot be easily applied to the more general form of short texts (*e.g.*, news titles). Gao et al. proposed a generalized method for aggregation of short texts against data sparsity by using the Embedding-based Minimum Average Distance (EMAD) [19]. However, their model is lack of the ability to figure out the correlated relationship between the extracted topics. By contrast, our model extends CTM to explicitly model the correlation structure among topics with a Gaussian covariance matrix.

Word embeddings learned from external sources, such as Wikipedia, are encoded with both syntactic and semantic information of words, which can be regarded as prior knowledge. Das et al. proposed Gaussian LDA that changes the generation process of LDA by generating word embeddings instead of textural

words, and models each topic as a multivariate Gaussian distribution over the word vector space [28]. Based on Gaussian LDA, Xun et al. incorporated the information of word embeddings into the Dirichlet Multinomial Mixture (DMM) model, and also introduced an alternative background mode to complement Gaussian topics [17]. Based on the DMM model, Li et al. proposed a novel topic model that promotes semantically related words under the same topic during the inference process by using the Generalized Pólya Urn (GPU) model [18]. Gao et al. proposed a Conditional Random Field Topic Model (CRFTM) to integrate word correlation knowledge provided by word embeddings into LDA [19]. However, the above methods do not take quantifiable relationships between words into consideration. Furthermore, since there are many differences between external corpora and short text collections we use, such word embeddings may bring some noise to the inference process. Compared with the existing methods of incorporating word embeddings into topic modeling, the proposed model employs a transformer-based language model learned from the entire corpus to quantify the relationship between words.

Topic evolution analysis.

The research on topic evolution analysis can be divided into two categories according to the topic tracking methods. One strategy is to exploit probabilistic topic models with considering temporal information to model documents and timestamps jointly in the topic generation process. Wang et al. proposed Topic Over Time (TOT) model that associates topics with a continuous distribution over timestamps [9]. Kawamae et al. proposed a TOT-style topic evolution model called Trend Analysis Model (TAM) which introduces a latent trend variable into each document [29]. AlSumait et al. proposed the Online LDA (OLDA) model for mining text streams [10]. Another strategy focuses on depicting the inherent evolution and connection between topics of adjacent time slices. Blei et al. proposed a Dynamic Topic Model (DTM) which models the temporal evolution of topics in time-stamped text data [11]. Ahmed et al. proposed infinite DTM (iDTM) that can automatically identify the number of topics [30]. Due to the length of short texts, these models suffer a lot from the data sparsity problem on short texts, resulting in inferior topic inferences.

Many recently proposed topic evolution models are more suitable for short texts. Wang et al. proposed a temporal-LDA (TM-LDA), which could be used for modeling the topic transitions in social media data by minimizing the prediction error on topic distribution in subsequent postings [31]. Twitter-Topic Tracking Model (Twitter-TTM) is a topic model based on Twitter-LDA for tracking topic trends and user interests from twitter [32]. Zhu et al. proposed a Coherent Topic Hierarchy (CTH) to discover the topic evolutionary structure on microblog feeds, which is represented by the splitting and merging of local topics from beginning to end along a timeline [5]. However, these models neglect the interpretation of learned topics and the correlation between topics. Compared with these methods, our model is able to efficiently extract coherent topics from short texts and discover meaningful patterns for topic evolutionary analysis.

3. Methodology

In this section, we first describe how to quantify the relationship between words by using ETLM. After that, we propose a new topic model CCTM to discover topics and topic correlations. CCTM makes semantically correlated words a higher probability to appear in the same topic by using a weighted conditional random field. The Online version of CCTM (OCCTM) is also proposed to create topic evolutionary graphs that present a graphical summary of the complex evolutionary relationships between topics.

3.1. ETLM

As we have mentioned, short text topic models like CRFTM and GPU-DMM [18,19], which exploit word embeddings to improve topic modeling over short texts, fix the promotion weight for each semantically related word pair. However, the promotion weight is very important because if two words are more correlated, they should be more likely to appear under the same topic.

In these models, the semantic relatedness between two words is measured by the cosine similarity between their vector representations, which are learned with the aim to retain words' global contextual information. Words with similar semantic and syntactic attributes are close to each other in the vector space, which means that they tend to replace each other in the same context. However, topic models are designed to implicitly capture word co-occurrence patterns to distill the topic structure. Therefore, we think the promotion weight should satisfy the following properties: if two words have a higher probability to co-occur in the same generative sentence, they are more related. Conversely, the relationship between them shall be weakened.

Recently, artificial neural networks have proven to be effective for learning the relationship between words for sentence generation [33]. Inspired by this, we find a neural probabilistic language model is a good choice to satisfy these properties. The reason is that when word w_i and previously observed words are given, the output of a language model can quantify the generation probability of word w_j. This probability is able to reflect the possibility of co-occurrence between two words (w_i, w_j) in a particular context. Therefore, the semantic correlated word pair with a greater generation probability is more likely to belong to the same topic.

Encouraged by recent work which utilizes the transformer for various tasks [24], we use a transformer architecture to train a language model. Given a sequence of words $w_{0:n} = w_0, w_1, \ldots, w_n$, a language model assigns a probability distribution by factoring out the joint probability as follows:

$$P(w_{0:n}) = P(w_0) \sum_{i=1}^{n} P(w_i|w_{0:i-1}). \tag{1}$$

To model the conditional probability $P(w_i|w_{0:i-1})$, we train a deep transformer model to process the word sequence $w_{0:\,n}$.

The transformer architecture is proposed for sequence-to-sequence tasks like machine translation, to replace traditional recurrent networks. Comparing a feature with all other features in the sequence to calculate self-attention is the main idea of its original architecture. Since this calculation cannot be performed efficiently using the original features directly, the model employs linear projections to map features to query (Q), key (K) and value (V) embeddings. The output of the query can be computed as an attention weighted sum of values V. The weights on the values are determined by the dot products of the query Q with all keys K. In sequences-to-sequences tasks, the query is the word being translated. The memory keys and values are linear projections of the input sequence and the previously generated output sequence, respectively. Since the transformer architecture contains no recurrence and no convolution, in order to incorporate positional information, a location embedding is also added to these representations.

In this paper, we use an encoder-only transformer architecture for the language model, which is a variant of the original transformer. The model exploits a multi-headed self-attention mechanism to encode the input context words, and position-wise feedforward layers are used to produce an output distribution over target words. Specifically, for the word sequence $w_{0:\,n}$, we use a standard language modeling objective to maximize the following likelihood:

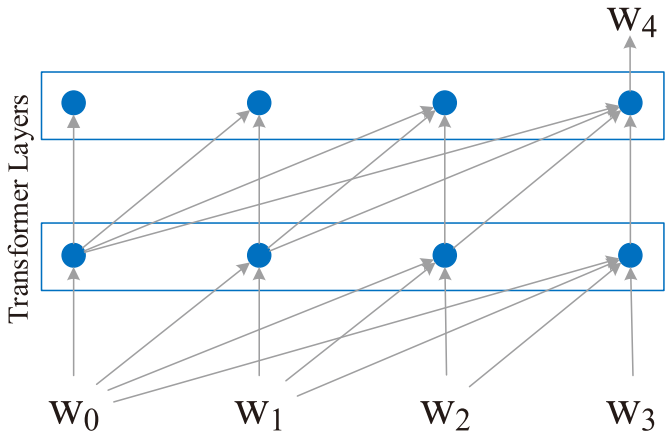


Fig. 2. ETLM with two transformer layers processing a four word sequence to predict w_4.

$$\mathcal{L}(w_{0:n}) = \sum_i LogP(w_i|w_{i-cw}, \ldots, w_{i-1}; \boldsymbol{\Theta}), \tag{2}$$

where cw is the size of the context window, and the conditional probability P is modeled using a transformer network with parameters $\boldsymbol{\Theta}$. These parameters are trained using stochastic gradient descent.

The input layer $x = \{w_i, w_{i+1}, \ldots, w_{i+cw}\}$ is the context vector of words, we can compute hidden and output layers with x:

$$\begin{aligned} h_0 &= \mathbf{W}_e x + \mathbf{W}_p \\ h_t &= transformer\_layer(h_{t-1}), \forall t \in [1, nl] \\ y &= \sigma(h_{nl}\mathbf{W}_e^T) \end{aligned} \tag{3}$$

where nl is the number of transformer layers. Two weight matrices $\mathbf{W}_e$ and $\mathbf{W}_p$ are the word embedding matrix and positional embedding matrix, respectively. $\sigma(\cdot)$ denotes softmax function and "*transformer_layer*" means a block containing two sub-layers. The first is a multi-head self-attention layer, and the second is a position-wise fully connected feed-forward network.

To ensure that the prediction for word w_i is only conditioned on words that appear before w_i, we need to prevent leftward information flow in the model. As a result, we employ a masked attention to mask our attention layers, and thus each position can only attend leftward. This is the same as the self-attention mechanism in the decoder of the original transformer model used for sequence-to-sequence tasks. Fig. 2 illustrates our transformer layer with the masked attention, which prevents information flow from right to left. The predictions for each word can depend only on the word that appeared earlier.

After training the model with the objective in Eq. (2), we can define the quantifiable relationship between w_i and w_j as $y_i(j)$, which is the jth value in y_i:

$$y_i(j) = P(w_j|w_{i-cw:i}), \tag{4}$$

where $y_i(j)$ denotes given a sequence of words $w_{i-cw:i}$, the generation probability of w_j under ETLM. The word sequence guarantees the previously observed context also have effects on the generation of w_j. For word w_i and w_j, the promotion weight ω_{ij} can be defined as:

$$\omega_{ij} = y_i(j) \times \gamma + 1, \tag{5}$$

where γ is used to avoid the value of ω_{ij} being too small.

3.2. CCTM

The standard CTM model provides an extension of LDA, which replaces the Dirichlet prior with a logistic-normal distribution for topic proportions and models topic correlation patterns with a Gaussian covariance matrix. Let us briefly outline related notation which is necessary for the later description of our new model based on CTM. The graphical model of CTM is shown in Fig. 3. The meaning of the notations is shown in Table 1.

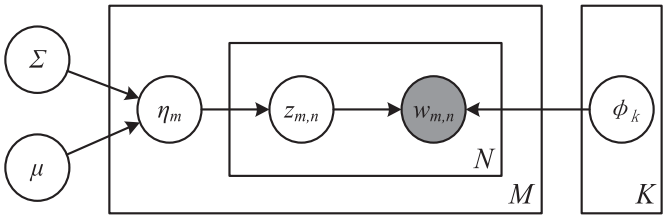


Fig. 3. The graphical model representation of CTM.

Table 1
Notation used in the CTM.

Name	Description
M	the number of documents in the corpus
$m = 1, 2, \ldots, M$	the index of an individual document in the corpus
N	the numbers of words in document m
$n = 1, 2, \ldots, N$	the index of a word in document m
K	the numbers of topics
$k = 1, 2, \ldots, K$	the index of a topic
w_{mn}	the unique word associated with the n_{th} token in document m
z_{mn}	the hidden topic of w_{mn}
$\boldsymbol{\phi}_k$	the topic-specific word distribution in topic k
$\boldsymbol{\eta}_m$	a K-dimensional vector, specifying topic priors for document m
μ and Σ	the mean and covariance matrix of a multivariate Gaussian distribution

The generative process of document m is described as follows:

(1) Draw $\boldsymbol{\eta_m} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$.
(2) Compute the document-specific topic proportions $\boldsymbol{\theta}_m$ using the logistic normal transformation as: $\theta_m^k = \frac{e^{\eta_m^k}}{\Sigma_{j=k}^K e^{\eta_m^j}}$.
(3) For each topic k
  a. Draw a word proportion $\phi_k \sim Dir(\boldsymbol{\beta})$
(4) For each word index $n = 1, 2, \ldots, N$:
  a. Draw a topic assignment $z_{mn} \sim Mult(\boldsymbol{\theta}_{\boldsymbol{\eta}_m})$.
  b. Draw the observed word $w_{mn} \sim Mult(\boldsymbol{\phi}_{z_{mn}})$,

where $Mult(\cdot)$ denotes the multinomial distribution.

Although CTM is able to discover the relationship between topics, it does not perform well in short texts [21]. Therefore, we first exploit EMAD proposed by our previous paper [19] to measure the dissimilarity between two short texts. Secondly, we implement the same clustering algorithm to merge short texts into lengthy pseudo-document. A detailed discussion of the EMAD and clustering algorithm is presented in [19]. In addition, following [16,18,19,22], the general word semantic relatedness knowledge based on word embeddings at word level can be used to improve topic modeling and correlation discovery at topic level. The proposed model extends the classic CTM model by imposing a WCRF on the latent topic layer, which aims at incorporating both global and local semantic relatedness knowledge in topic assignments. WCRF is a weighted variant of the conditional random fields that have shown to be effective in encoding various known relationships between observations [23]. Fig. 4 shows the graphical model of CCTM.

Global semantic correlation. Words that are semantically or syntactically similar to each other are more likely to belong to the same topic [18,19]. To achieve this, we measure the semantic relatedness between two words using the cosine distance between their word

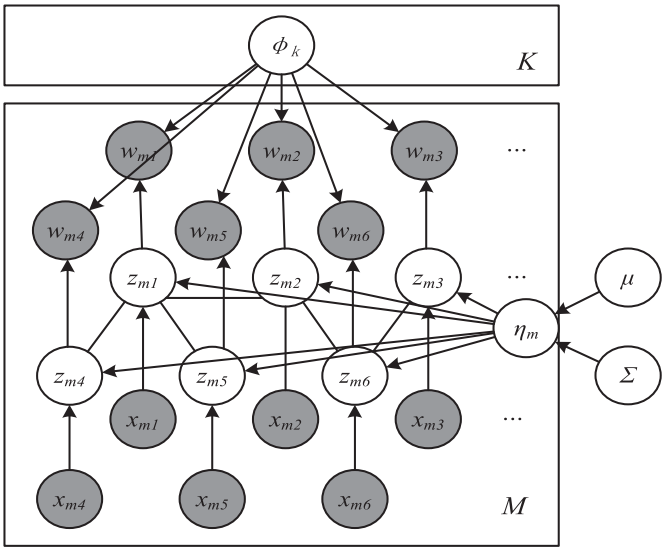


Fig. 4. The graphical model representation of CCTM.

embeddings. The basic idea is that if the distance between two words w_{mi} and w_{mj} in pseudo-document m is less than a threshold μ, we assume that (w_{mi}, w_{mj}) is a globally correlated word pair. Therefore, they should be given a higher probability to share the same topic. Specifically, the proposed model adds a WCRF on the topic layer of CTM. Given a pseudo-document m that contains N_m words $\{w_{mi}\}_{i=1}^{N_m}$, we examine each pair of words (w_{mi}, w_{mj}). If they are globally correlated, *i.e.*, $d(w_{mi}, w_{mj}) < \mu$, we create an undirected edge between their latent topic labels (z_{mi}, z_{mj}) with promotion weight ω_{ij}. As shown in Fig. 4, there are five edges {(z_{m1}, z_{m2}), (z_{m1}, z_{m4}), (z_{m1}, z_{m5}), (z_{m2}, z_{m6}), (z_{m3}, z_{m6})}. Unlike CRFTM, our model set the promotion weight based on ETLM, which is a fixed value in the CRFTM model.

Local semantic correlation. The word embedding learning method represents a word with all its possible meanings as a single vector, which may introduce bias into the process of topic inference. Therefore, we employ contextual words to effectively alleviate the semantic ambiguity in short text topic modeling. For each word $\{w_{mi}\}_{i=1}^{N_m}$ in m, its contextual words $\{x_{mi,p}\}_{p=1}^{P}$ can be defined as its P-nearest words in the current pseudo-document m based on the cosine similarity between their word embeddings. If the average difference between the distance of w_{mi} and its contextual words x_{mi}, and the distance of another word w_{mj} and x_{mi} is less than a threshold ε, *i.e.*, $\frac{1}{P}\sum_{p=1}^{P}|d(w_{mi}, x_{mi,p}) - d(w_{mj}, x_{mi,p})| < \varepsilon$, we assume that ($w_{mi}$, w_{mj}) is a locally correlated word pair. Given these contextual words, CCTM can associate the polysemous word with its most appropriate meaning. Therefore, words with no local semantic correlation should not share the same topic label, even if they are globally correlated.

In CCTM, the joint distribution of topic assignments $\{z_{mi}\}_{i=1}^{N_m}$ in pseudo-document m can be explicitly written as:

$$p(\mathbf{z_m} \mid \boldsymbol{\theta_m}, \mathbf{x_m}, \boldsymbol{\omega}) = \prod_{i=1}^{N_m} p(z_{mi} \mid \boldsymbol{\theta}_m)\boldsymbol{\Psi}(\boldsymbol{\omega}, z_{mi}, x_{mi}), \tag{6}$$

where $\boldsymbol{\Psi}(\cdot)$ represents the potential function that is used to model semantic correlations, and can be defined as:

$$\boldsymbol{\Psi}(\boldsymbol{\omega}, z_{mi}, x_{mi}) = exp\left(\frac{1}{A}\Big(\sum_{(mi,mj)\in E}\omega_{ij}f(z_{mi}, z_{mj}) + \sum_{(mi,mj)\in E}\omega_{ij}g(z_{mi}, z_{mj}, x_{mi})\Big)\right). \tag{7}$$

Topic label z_{mi} only relies on topic proportion distribution $\boldsymbol{\theta_m}$ in CTM and its variants. However, in the CCTM model, z_{mi} not only relies on $\boldsymbol{\theta_m}$, but also depends on the topic assignments of related words. In Eq. (7), E denotes all edges that connect the topic labels of related words and A represents a normalization term. Parameter ω_{ij} learned by ETLM dynamically adjusts the weight of promotion for each semantically correlated word w_{mj} when processing word w_{mi}. $f(\cdot)$ and $g(\cdot)$ represent the unary potential function and pairwise potential function, respectively. It is difficult to infer parameters due to the non-conjugacy problem between the logistic normal distribution and multinomial distribution especially for a complex graphical model as ours. We therefore design the following potential functions to handle $\boldsymbol{\Psi}(\cdot)$ as a constant, which can be calculated during the preprocessing stage. Meanwhile, we use the same Gibbs sampling method with data augmentation to sample logistic-normal parameters in the proposed model as [34].

We think the unary potential should satisfy the following properties, which encourages correlated words to share the same topic label. If the two topic assignments of a globally correlated word pair are different, the unary potential f generates a small value. If the two topic labels are the same, f produces a large value. Therefore, we define the unary potential as:

$$f(z_{mi}, z_{mj}) = \begin{cases} 1 & \text{if } z_{mi} = z_{mj} \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

Word embedding techniques such as word2vec [15] cannot discriminate among different meanings of a word. Mixing different meanings into a single representation may hinder the semantic understanding of our model. The pairwise potential g utilizes the local semantic correlation to disambiguate word sense. If globally correlated word pair (w_{mi}, w_{mj}) has no local semantic correlation, a penalty is added to Eq. (7) to mitigate the noise and bias caused by ambiguous words during the inference process. The pairwise potential is defined as:

$$g(z_{mi}, z_{mj}, x_{mi}) = \begin{cases} 0 & \text{if } \frac{1}{P}\sum_{p=1}^{P}|d(w_{mi}, x_{mi,p}) - d(w_{mj}, x_{mi,p})| < \varepsilon \\ -1 & \text{otherwise.} \end{cases} \tag{9}$$

Given the topic labels, the generation of words is the same as CTM. w_{mi} is generated from the topic-words multinomial distribution $\phi_{z_{mi}}$ corresponding to z_{mi}. The generative process of pseudo-document m is described as follows:

(1) Draw $\boldsymbol{\eta_m} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$.
(2) Draw a topic proportion $\theta_m^k = \frac{e^{\eta_m^k}}{\sum_{j=k}^{K} e^{\eta_m^j}}$.
(3) For each topic k
  a. Draw a word proportion $\phi_k \sim Dir(\boldsymbol{\beta})$
(4) For each pseudo-document m
  a. Draw a topic assignment $\mathbf{z}_m$ according to Eq. (6).
  b. Draw the observed word $w_{mi} \sim Mult(\phi_{z_{mi}})$

Parameter inference. Because of the conjugacy between multinomial likelihood and a Dirichlet prior, we first integrate out the topic-word distribution $\boldsymbol{\phi}$. The sampling process of topic assignments $\mathbf{z}$ is similar to the collapsed Gibbs sampling for LDA. Given $\boldsymbol{\eta} = \{\boldsymbol{\eta}\}_{m=1}^{M}$ and $\mathbf{z}_{-mn}$ which is the topic assignment without considering the current word w_{mn}, the topic assignment of each word is drawn iteratively as follows:

$$p(z_{mn} = k|\mathbf{z}_{-mn}, \mathbf{w}_{-mn}, \boldsymbol{\eta}) \propto \frac{C_{k,-n}^{w_{mn}} + \beta_{w_{mn}}}{\sum_{j=1}^{V} C_{k,-n}^{j} + \sum_{j=1}^{V}\beta_j} e^{\eta_m^k}\boldsymbol{\Psi}(\boldsymbol{\omega}, z_{mn} = k, x_{mn}), \tag{10}$$

where $C_k^{w_{mn}}$ is the number of times topic k being assigned to the word w_{mn} in the entire corpus, $C_{\cdot,-n}$ denotes that word n is ex-

cluded from the corresponding pseudo-document or topic, V is the size of the vocabulary.

When topic assignments $\mathbf{z}$ are given, it is difficult to directly sample the logistic normal parameters because of the non-conjugacy. To solve the non-conjugacy between the logistic-normal prior and multinomial topic mixing proportions, following the approaches in [35] and [22], we can sample the logistic normal parameters $\boldsymbol{\eta}$ with data augmentation techniques. For a pseudo-document m, the likelihood for η_m^k conditioned on $\boldsymbol{\eta}_m^{-k}$ is given below:

$$\begin{aligned}\ell(\eta_m^k|\boldsymbol{\eta}_m^{-k}) &= \prod_{n=1}^{N_m}\left(\frac{e^{\eta_m^k}}{\sum_i e^{\eta_m^i}}\right)^{z_{mn}^k}\left(1-\frac{e^{\eta_m^k}}{\sum_i e^{\eta_m^i}}\right)^{1-z_{mn}^k}\\ &= \frac{(e^{\rho_m^k})^{C_m^k}}{(1+e^{\rho_m^k})^{N_m}}\end{aligned} \tag{11}$$

where z_{mn}^k denotes the topic indicator that $z_{mn}^k = 1$ if word w_{mn} is assigned to the k_{th} topic, $\rho_m^k = \eta_m^k - \zeta_m^k$; $\zeta_m^k = \log(\sum_{j\neq k} e^{\eta_m^k})$; and $C_m^k = \sum_{n=1}^{N_m} z_{mn}^k$ denotes the number of words assigned to topic k in pseudo-document m. As a result, we obtain the conditional distribution as follow:

$$p\left(\eta_m^k|\boldsymbol{\eta}_m^{-k},\mathbf{z},\mathbf{w}\right)\propto\ell\left(\eta_m^k|\boldsymbol{\eta}_m^{-k}\right)\mathcal{N}\left(\eta_m^k|\mu_m^k,\sigma_k^2\right). \tag{12}$$

As for the prior part, it is a univariate normal distribution conditioned on the logistic normal prior in the current pseudo-document $\boldsymbol{\eta}_m^{-k}$. Therefore, when $\boldsymbol{\eta}_m^{-k}$, $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are given, we have:

$$\begin{aligned}\mu_m^k &= \mu_k - \boldsymbol{\Lambda}_{kk}^{-1}\boldsymbol{\Lambda}_{k-k}\left(\boldsymbol{\eta}_m^{-k}-\boldsymbol{\mu}_{-k}\right)\\ \sigma_k^2 &= \boldsymbol{\Lambda}_{kk}^{-1},\end{aligned} \tag{13}$$

where $\boldsymbol{\Lambda} = \boldsymbol{\Sigma}^{-1}$ is the precision matrix. Nevertheless, because of the non-conjugacy between logistic likelihood and the Gaussian prior, it is hard to calculate the likelihood $\ell(\eta_m^k|\boldsymbol{\eta}_m^{-k})$ and unable to directly draw samples η_m^k.

Based on a data augmentation representation with only one layer of auxiliary variables δ_m^k, we can solve this non-conjugacy problem and the likelihood $\ell(\eta_m^k|\boldsymbol{\eta}_m^{-k})$ can be expressed as follows:

$$\ell(\eta_m^k|\boldsymbol{\eta}_m^{-k}) = \frac{1}{2^{N_m}}e^{\kappa_m^k\rho_m^k}\int_0^\infty e^{-\frac{\delta_m^k(\rho_m^k)^2}{2}}p(\delta_m^k|N_m,0)d\delta_m^k, \tag{14}$$

where $\kappa_m^k = C_m^k - N_m/2$ and $p(\delta_m^k|N_m,0)$ is the Polya-Gamma distribution $\mathcal{PG}(N_m,0)$. Eq. (14) suggests that $p(\eta_m^k|\boldsymbol{\eta}_m^{-k},\mathbf{z},\mathbf{w})$ is a marginal distribution of the complete distribution:

$$\begin{aligned}&p(\eta_m^k,\delta_m^k|\boldsymbol{\eta}_m^{-k},\mathbf{z},\mathbf{w})\\ &\propto\frac{1}{2^{N_m}}exp\left(\kappa_m^k\rho_m^k-\frac{\delta_m^k(\rho_m^k)^2}{2}\right)p(\delta_m^k|N_m,0)\mathcal{N}(\eta_m^k|\mu_m^k,\sigma_k^2)\end{aligned} \tag{15}$$

Accordingly, we can draw a sample η_m^k from the joint distribution by discarding the augmented variable δ_m^k. The sampling procedure is detailed below.

For δ_m^k: the conditional distribution of the augmented variable is $p(\delta_m^k|\boldsymbol{\eta},\mathbf{z},\mathbf{w})\propto\exp(-\frac{\delta_m^k(\rho_m^k)^2}{2})p(\delta_m^k|N_m,0) = \mathcal{PG}(\delta_m^k;N_m,\rho_m^k)$ according to Eq. (15) and [35]. Following the work of [34], the time complexity of sampling method to draw Polya-Gamma random variables can be reduced to $O(1)$. Therefore, we can efficiently draw a sample of δ_m^k.

For η_m^k: according to Eq. (15), the posterior distribution $p(\eta_m^k|\boldsymbol{\eta}_m^{-k},\mathbf{z},\mathbf{w},\delta_m^k)$ can be expressed as:

$$p(\eta_m^k|\boldsymbol{\eta}_m^{-k},\mathbf{z},\mathbf{w},\delta_m^k)\propto exp\left(\kappa_m^k\eta_m^k-\frac{\delta_m^k(\eta_m^k)^2}{2}\right)\mathcal{N}(\eta_m^k|\mu_m^k,\sigma_k^2). \tag{16}$$

Conditioning on the auxiliary variable δ_m^k, Eq. (16) leads to a univariate Gaussian distribution $\mathcal{N}(\gamma_m^k,(\tau_m^k)^2)$, where the mean is $\gamma_m^k = (\tau_m^k)^2(\sigma_m^{-2}\mu_m^k + \kappa_m^k + \delta_m^k\zeta_m^k)$ and the variance is $(\tau_m^k)^2 = (\sigma_m^{-2}+\delta_m^k)^{-1}$. Therefore, η_m^k can be easily drawn from a univariate Gaussian distribution according to the auxiliary variable δ_m^k.

After that, we draw the logistic normal parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ from the conjugate Normal-Inverse-Wishart prior, $p_0(\boldsymbol{\mu},\boldsymbol{\Sigma}) = \mathcal{NIW}(\boldsymbol{\mu}_0,\rho,\kappa,W)$ where $(\boldsymbol{\mu}_0,\ \rho,\ \kappa,\ W)$ are hyper-parameters. Given $\{\boldsymbol{\eta}_m\}_{m=1}^M$, the hyper-parameters can be updated as:

$$\begin{aligned}\boldsymbol{\mu}_0' &= \frac{\rho}{\rho+M}\boldsymbol{\mu}_0 + \frac{\rho}{\rho+M}\overline{\boldsymbol{\eta}},\\ \rho' &= \rho + M,\\ \kappa' &= \kappa + M,\\ W' &= W + Q + \frac{\rho M}{\rho+M}(\overline{\boldsymbol{\eta}}-\boldsymbol{\mu}_0)(\overline{\boldsymbol{\eta}}-\boldsymbol{\mu}_0)^\top,\end{aligned} \tag{17}$$

where $\overline{\boldsymbol{\eta}}$ is the empirical mean of $\{\boldsymbol{\eta}_m\}_{m=1}^M$, and $Q = \Sigma_m(\boldsymbol{\eta}_m - \overline{\boldsymbol{\eta}})(\boldsymbol{\eta}_m - \overline{\boldsymbol{\eta}})^\top$.

3.3. Online CCTM

The CCTM model processes data in a single batch to learn topic assignments for each pseudo-document. To model the temporal evolution of topics in data streams, we require a topic evolution model that: (1) processes short text streams and periodically updates the model; (2) does not grow in size over time to ensure it can capture topic changes in the data streams. To achieve these two goals, we develop an online CCTM (OCCTM) that can dynamically construct evolutionary graphs of topics and subtopics from short text streams.

For time-stamped short text collections, we first assume that short texts are partitioned into time slices and arrive in chronological order. L is the span of time slices, which relys on the nature of the short text collection, *e.g.*, an hour, a day, or a year. We employ t_n to represent each time slice, and t_0 denotes the first time slice. Since we need a model that does not grow in size over time, when short texts in the new time slice arrive, short texts in the old time slice will be discarded. The reason is that storing the entire short text stream history data would result in an infinite growth of the model over time, and thus the online topic model will become less and less sensitive to topic changes.

Most offline topic models have fixed symmetric Dirichlet priors for the distributions of their variables. Nevertheless, it is too ideal to assume all distributions are under fixed priors for an online topic model, which ignores the history information [10,36]. Our model processes short text streams in an online fashion by resampling topic labels for new short texts using Dirichlet priors from a previously learned model. When new short texts arrive in the time slice t_{n+1}, we suppose that ϕ from the previous model in slice t_n is used to serve as Dirichlet prior $\boldsymbol{\beta}_{n+1}$ in the new model in slice t_{n+1}, and then OCCTM reassigns the topic assignments $\mathbf{z}$ for all pseudo documents according to Eq. (10).

Furthermore, as an online model that can extract topics changing over time, it is not appropriate to use a fixed vocabulary in each time slice. Dynamic changes in the vocabulary at different time slices are important for detecting emerging events in the short text stream. New words that appear for the first time in any time slice are likely to be associated with emerging events. Therefore, the OCCTM model regenerates the vocabulary for short texts in the time slice at every update. In the time slice t_{n+1}, for previously seen words, Dirichlet priors $\boldsymbol{\beta}_{n+1}$ can be expressed as follows:

$$\beta_{n+1}^{kw} = \frac{C_n^{kw}}{C_n}\times K\times V_{n+1}\times\beta_0, \tag{18}$$

where β_{n+1}^{kw} is the Dirichlet prior for word w in topic k in the time slice t_{n+1}, C_n^{kw} is the number of word w assigned to topic k in the

time slice t_n, C_n, K and V_{n+1} are the number of tokens in previously processed short texts, number of topics and number of vocabulary in the current slice. β_0 denotes the default Dirichlet prior for ϕ. The rationale of the normalization method is to maintain a constant sum of priors in the processing of different batches, *i.e.*, $\sum \boldsymbol{\beta} = K \times V_{n+1} \times \beta_0$. For new words that are assumed to have 0 count for all topics in the previous stream, we set $\beta_{n+1}^{kw} = \beta_0$.

Next, we discuss how to define the topic evolution relationship between adjacent time slices. The Kullback-Leibler divergence (KLD) can be employed to evaluate the similarity of a pair of topics, and it is used to construct topic evolution paths in many topic evolution models. Following [1,5], our model also utilizes KLD between the word distribution of each pair of topics to measure the evolution relationship in two adjacent time slices. For topic z_i and z_j in time slices t_n and t_{n+1} respectively, their similarity is defined as follows:

$$topic\_sim(z_i, z_j) = \frac{1}{2}(KLD(z_i||z_j) + KLD(z_j||z_i)), \tag{19}$$

where $KLD(z_i||z_j) = \sum_{w=1}^{V} \phi_{iw} \log(\phi_{iw}/\phi_{jw})$.

In the last step, a topic evolution graph can be constructed by our model. Firstly, we utilize OCCTM to extract coherent topics and topic correlations at each time slice. Secondly, the topic evolution relationship between adjacent time slices is established according to the KLD similarities. If $topic\_sim(z_i, z_j)$ is smaller than a specific threshold λ, then it denotes that topic z_j is a continuation of topic z_i.

4. Experiments

In this section, we conduct experiments to demonstrate the effectiveness of our model against six baseline methods. The performance in terms of topic coherence and short text classification are reported over two real short text datasets, *i.e.*, an English news dataset and a Q&A dataset. The experiment results validate the effectiveness of our model at discovering coherent topics and topic correlations from both short text datasets. Finally, we utilize OCCTM to generate a topic evolution graph from a microblog corpus.

4.1. Experimental setup

4.1.1. DataSets

In our experiments, CCTM is tested on two real-world short-text corpora. The information about these datasets are detailed as follows:

News. This dataset[1] consists 31,150 English news articles that are crawled from RSS feeds of three popular newspaper websites (usatoday.com, nyt.com, reuters.com). Categories are: business, sport, health, U.S., sci&tech, entertainment and world. This dataset has been used in a few studies [37]. In this dataset, we only retain news descriptions because they are typical short texts.

StackOverflow. This is a dataset[2] of 20,000 question titles with 20 different labels, which are randomly selected by [38] from the challenge data published on Kaggle.com.

For better training of topic models, we perform the following preprocessing on these two datasets: (1) remove stop words[3], punctuations and non-alphabetic characters; (2) convert letters to lowercase; (3) remove words with document frequency less than 3; (4) remove words with length less than 3 characters. We do not do stemming because many researchers found that stemmers produce no meaningful improvement in topic coherence, and actually reduce topic stability [39]. Significantly, ETLM is trained on the original datasets rather than an external corpus, which is more in line with the needs of our model. As we have mentioned, it attempts to learn the quantifiable relationship between words from the original corpus. This is a complement to the word semantic relationship provided by word embeddings learned from large external corpora. In addition, to be consistent with the topic model and better quantify the relationship between topic words, ETLM is trained in the preprocessed corpus.

4.1.2. Baseline methods

We compare the performance of the CCTM model with a long text topic model CTM, three typical methods for short text topic modeling and two variant of CCTM, which can be referred as follows.

- **CTM** replaces the Dirichlet prior in LDA with logistic-normal distribution to model the correlations among latent topics, and works as the basis of our model [20].
- **CRFTM** is a recently method for short text topic modeling which proposed a generalized solution to alleviate the sparsity problem by merging short texts into regular-sized pseudo-documents [19].
- **GPU-DMM** extracts semantic relatedness between words from word embeddings, and employs the generalized Pólya urn (GPU) model to improve coherence of topics by using these semantic relatedness information [18].
- **GLTM** is a novel word embedding-based topic model for short texts, which trains local word embeddings to capture context information for each word by using the continuous skip-gram model [16].
- **CCTM-W** is a variant of CCTM. The promotion weight ω is set to 1 for all semantically related word pairs in CCTM-W.
- **CCTM-L-W** is based on the CCTM model but only global semantic correlations is considered.

For GPU-DMM, CTM and CRFTM, we utilize the tools released by the authors. For GLTM, since the authors did not release the code, we implement its code in Java.

4.1.3. Evaluation measures

Topic coherence. How to evaluate the performance of topic models is still an open problem [40]. Conventionally, perplexity or the likelihood of held-out data can be used to evaluate the performance of topic models. Nevertheless, as shown in [41], these automated methods of evaluation are less correlated to human judgments. Recently, coherence measures [42,43] have been proposed to assess topic quality, which are more correlated to human interpretability.

In the following experiments, we employ the UCI [42] and UMass [43] topic coherence to measure the semantic coherence of topics learned by topic models, both of which have been proved to be consistent with human judgments. The general idea of these metrics is that words belonging to the same topic tend to co-occur within the same document. Given a topic k and its top N words $(w_1, w_2, \ldots, w_N)$ sorted by probability in descending order, the UCI coherence of k is calculated as follow.

$$UCI(k) = \frac{2}{N(N-1)} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} PMI(w_i, w_j), \tag{20}$$

$$PMI(w_i, w_j) = log \frac{p(w_i, w_j)}{p(w_i)p(w_j)}, \tag{21}$$

where $p(w_i)$ and $p(w_j)$ are the probabilities of words w_i and w_j appears in a sliding window respectively. $p(w_i, w_j)$ is the probability of words w_i and w_j co-occurring in the same sliding window. The

[1] http://acube.di.unipi.it/tmn-dataset/
[2] http://github.com/jacoxu/StackOverow
[3] Stop word list is from NLTK: http://www.nltk.org/

average UCI score of all topics will be used to evaluate the quality of the topic model. A higher UCI score implies better learned topics. As for UMass, the coherence of topic k is calculated as

$$UMass(k) = \frac{2}{N(N-1)} \sum_{i=2}^{N} \sum_{j=1}^{i-1} \frac{p(w_i, w_j)}{p(w_j)}, \quad (22)$$

where $p(w_i, w_j)$ and $p(w_j)$ are estimated based on document frequencies. UCI and UMass assess topic quality from a sliding window and document frequencies respectively, which can evaluate our models more comprehensively.

Classification measures. The document classification task is another popular evaluation of topic models. In this experiment, we consider the multi-class classification task for predicting the categories for test short texts to evaluate the quality of latent short text representations (a fixed set of topical feature $p(z|d)$) distilled by topic models. A high classification accuracy denotes the extracted topics are more discriminative and representative. Following [18], we also make use of summation over words (SW) representations to infer $p(z|d)$, which is a proper method for short text topic modeling:

$$p(z = k|d) \propto \sum_{w} p(z = k|w) p(w|d), \quad (23)$$

where $p(w|d)$ can be estimated by using the relative frequency of w in short text d.

For News and StackOverflow datasets, we exploit the available tool word2vec[4] trained on a large Google news dataset (about 100 billion words), which provides an embedding matrix for a vocabulary of about 3 million words or phrases. Each word or phrase is represented by a 300-dimensional embedding vector, trained using the approach in [15]. If a word does not have an embedding vector, the word is considered to have no semantic correlation knowledge.

For the baselines, if not explicitly mentioned, we choose the parameters according to their original papers. For all the methods in comparison, we set the hyper-parameters $\alpha = 50/K, \beta = 0.01$ and run 1,000 iterations of sampling. For GPU-DMM, GLTM, CRFTM and CCTM, we set $\mu = 0.3$ using a manual examination process proposed by [18], that is, words pairs with distance lower than 0.3 are labeled as correlated. The number of pseudo-documents in our method is set to $M = n/50$ where n is the number of short texts in the corpus. For CTM and our method, we set the hyper-parameters as $\rho = \kappa = 0.01M, \boldsymbol{\mu}_0 = 0$ and $W = \kappa I$. For CRFTM and our method, we set $\varepsilon = 0.1,\ P = 5$. For ETLM, $nl = 8, cw = 8$ and other parameters of the transformer architecture are set according to suggestion of the original literature [24]. For CCTM, we empirically set $\gamma = V/2$ where V denotes the size of vocabulary. In the following experiments, all results reported below are averaged on five runs. CTM, CRFTM and our models extract topics directly from pseudo-documents, while GPU-DMM and GLTM learn topics from the original short texts due to their intrinsic characteristics, which do not fit lengthy pseudo-documents very well. The statistical significance is based on the student t-test.

4.2. Experimental results

4.2.1. Topic evaluation by topic coherence

In this paper, we use Palmetto[5] as a quality measuring tool for topics to compute UCI and UMass coherence, which employs 3 million English Wikipedia articles as an external corpus.

Figs. 5 and 6 show UCI and UMass coherence of seven methods on two datasets with number of top words per topic $T = \{5, 10\}$ and number of topics $K = \{40, 60, 80\}$, respectively. It can be intuitively found from the experimental results on both datasets that CCTM achieves the best performance, and the improvement over CTM is statistical significance at 0.01 level. These observations validate that CCTM utilizes ETLM to quantify the relationship between words, which is beneficial to achieve higher topic quality. Furthermore, CCTM-W is the second best model in most cases and CCTM-L-W performs better than GLTM and GPU-DMM on both datasets. The experimental results demonstrate that the WCRF mechanism of CCTM ensures the topics distilled by the proposed models are more coherent and contain fewer ambiguous words. Although CRFTM also uses a conditional random field regularized model that encourages semantically related words to share the same topic label, it lacks an appropriate strategy to dynamically adjust the weight of promotion for each word pair, which may be the reason for its poorer results. Another observation is that CTM achieves a comparable performance with other state-of-the-art baselines. As discussed in [19], this may be due to the fact that short text aggregation contributes a lot to normal text topic models such as CTM when applied to short texts.

Table 2
Average classification accuracy of the 7 models on two datasets, with different number of topic K settings.

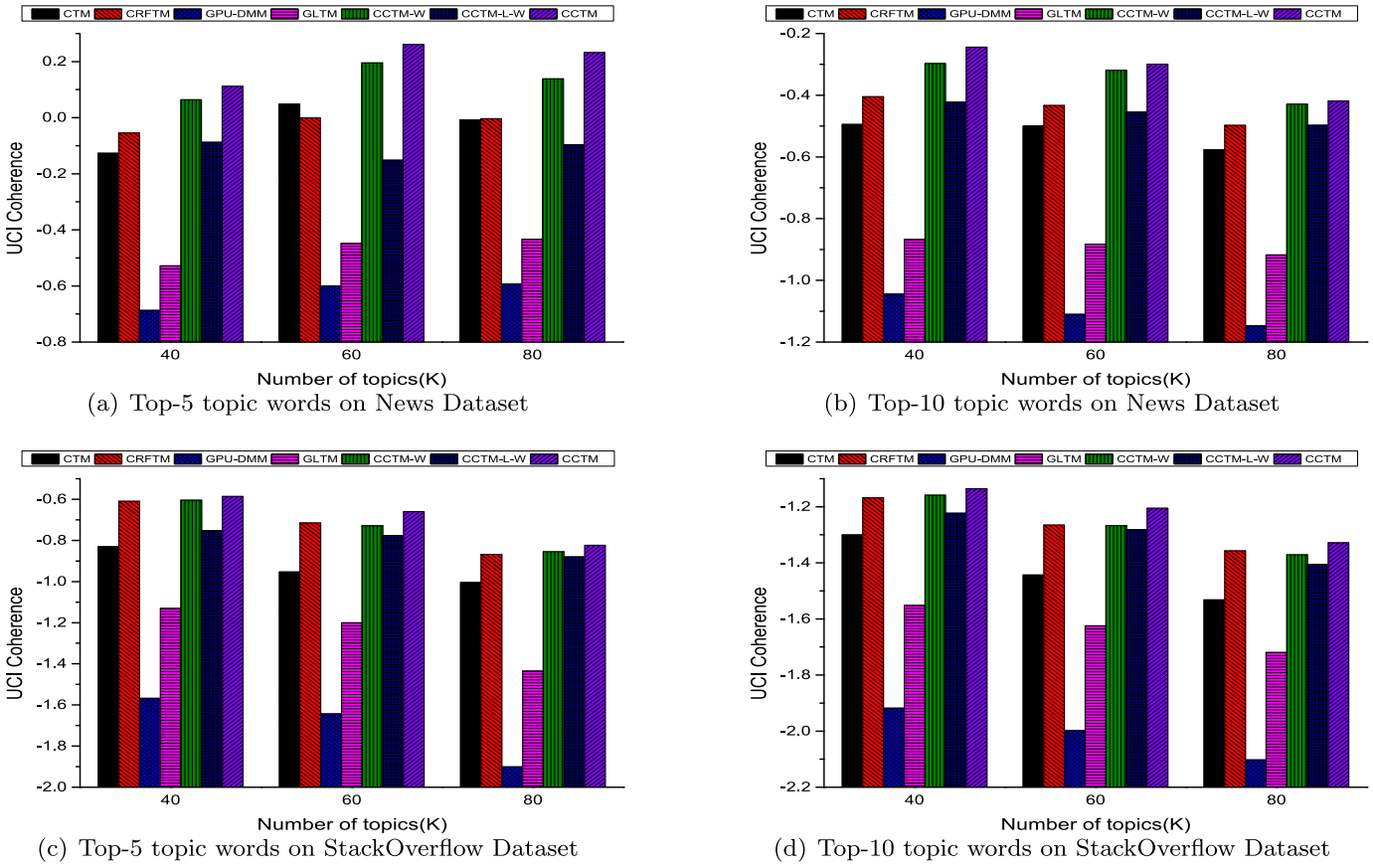
Dataset	Model	$K = 40$	$K = 60$	$K = 80$
News	CTM	76.03†	76.25†	76.94†
	CRFTM	75.99†	76.31†	77.40†
	GPU-DMM	74.20†	75.66†	77.10†
	GLTM	75.09†	76.20†	77.00†
	CCTM-W	76.29†	76.86†	77.45
	CCTM-L-W	76.00†	76.62†	77.20†
	CCTM	**76.42**	**77.10**	**77.61**
Stack Overflow	CTM	72.70†	75.16†	76.87†
	CRFTM	71.51†	75.71†	77.03†
	GPU-DMM	68.44†	70.98†	71.83†
	GLTM	68.26†	72.03†	73.13†
	CCTM-W	75.16†	76.59	**77.69**
	CCTM-L-W	74.43†	76.16†	77.00†
	CCTM	**75.71**	**76.94**	77.47

4.2.2. Topic evaluation by short text classification

Topics extracted by topic models can be regarded as a topic-level representation $p(z|d)$ of a short text. Therefore, the quality of the topics can be measured by the results of short text classification using their topic distribution. In this experiment, we compare the classification accuracy of our method with baselines. A better classification accuracy indicates that the topics distilled by the model are more representative and discriminative. We apply a linear kernel support vector machines classifier in sklearn with default parameter settings to classify these short texts. For each topic model, a five-fold cross validation is used to compute the classification accuracy.

Table 2 reports the average classification accuracy of each model on both datasets with number of topics $K = \{40, 60, 80\}$. The best results are highlighted in boldface, and † denotes the difference with the best result is statistically significant at 0.01 Level. Here, we make the following observations. The classification accuracy of all models increases with the number of topics. This may be because the large number of topics provides more features for training the classifier. On the News dataset, CCTM achieves the best classification accuracy across all settings. On the StackOverflow dataset, our model outperforms all other models in 2 out of 3 settings, and CCTM-W achieves the best performance in the remaining setting. Specifically, significant performance gains are achieved by CCTM over CTM on both datasets at 0.01 level. This validates the effectiveness of our model against baselines in learning semantic representations of short texts. In addition, CCTM-W performs better than CCTM-L-W on the two datasets, which demonstrates that using local semantic correlations to filter noise words is benefi-

[4] http://code.google.com/p/word2vec
[5] http://aksw.org/Projects/Palmetto.html



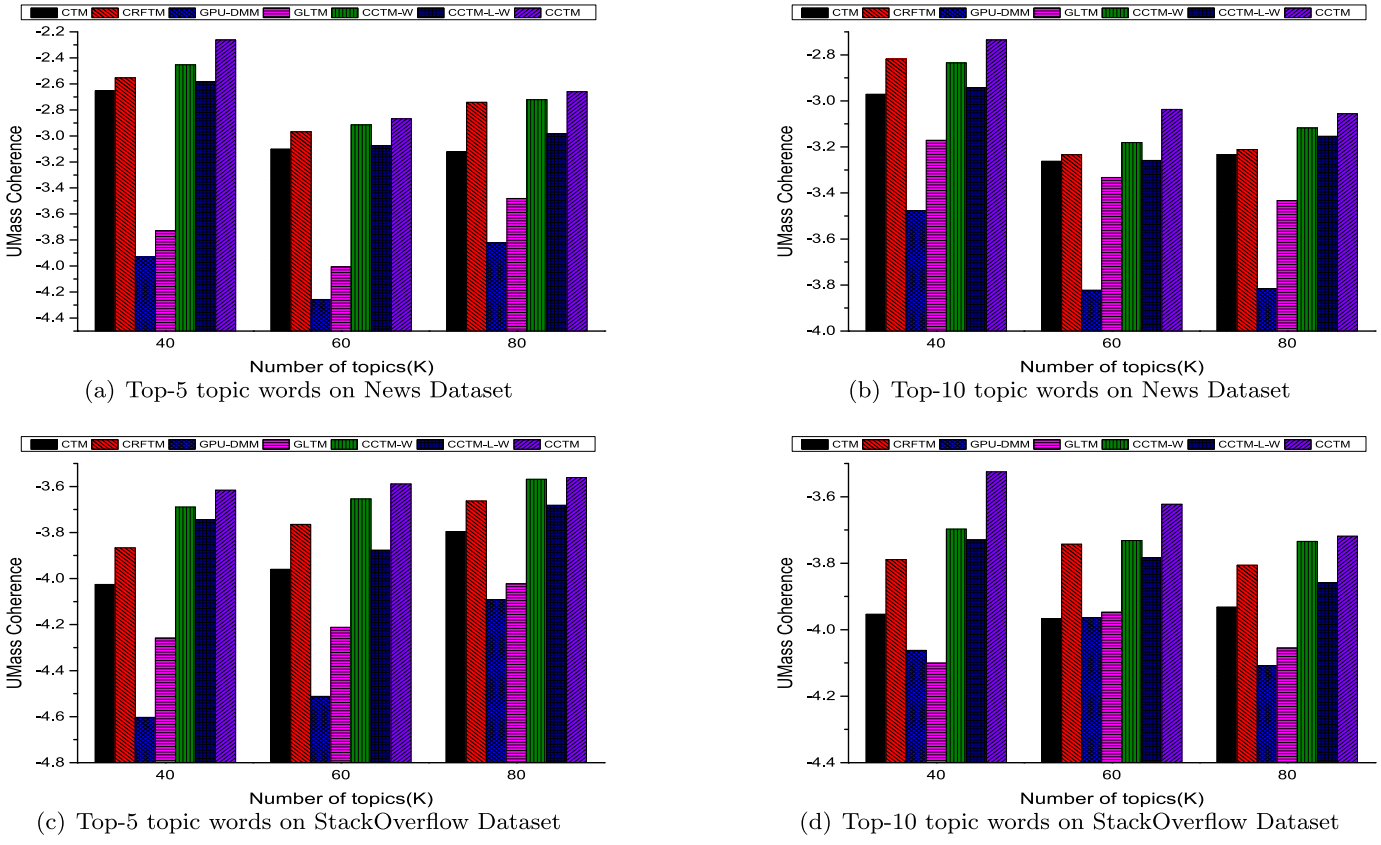
(a) Top-5 topic words on News Dataset

(b) Top-10 topic words on News Dataset

(c) Top-5 topic words on StackOverflow Dataset

(d) Top-10 topic words on StackOverflow Dataset

Fig. 5. UCI Coherence on Both Datasets.



(a) Top-5 topic words on News Dataset

(b) Top-10 topic words on News Dataset

(c) Top-5 topic words on StackOverflow Dataset

(d) Top-10 topic words on StackOverflow Dataset

Fig. 6. UMass Coherence on Both Datasets.

Table 3
Top 5 words of the 5 most incoherent topics learned by 5 topic models with number of topics $K = 80$ on the News dataset.

Model	Top-5 Words	Topic Coherence
CTM	day, championship, round, sports, masters	−4.81754
	play, little, going, back, man	−3.91692
	one, another, day, time, world	−3.46590
	next, announced, deal, tuesday, year	−3.30604
	people, killed, least, said, two	−2.75132
CRFTM	made, make, making, makes, every	−4.17365
	years, first, last, time, week	−3.06704
	one, two, three, four, another	−2.91289
	like, good, going, dont, want	−2.70013
	trial, former, case, fund, trading	−2.65227
GPU-DMM	new, said, one, first, years	−6.31985
	club, premier, league, forms, american	−6.08223
	said, friday, san, states, casino	−4.85316
	united, states, country, nations, world	−3.76618
	years, last, year, week, time	−3.05398
GLTM	north, south, east, following, came	−5.37266
	coach, roger, pitching, braves, said	−4.90567
	like, get, make, even, made	−3.45894
	back, left, right, first, lost	−3.10814
	championship, round, masters, lead, first	−2.89422
CCTM	war, international, defense, first, used	−2.94819
	coach, season, team, announced, head	−2.76058
	star, john, stars, michael, david	−2.32983
	homes, river, mississippi, residents, water	−2.24679
	school, charges, accused, sentenced, high	−1.75380

cial for short text topic modeling. GPU-DMM performs the worst among all models on both datasets even though they also exploit word embeddings. One possible reason is that the word semantic relationships in external corpora and our datasets are different. The general semantics knowledge from external corpora may not work well on extracting topics from these datasets.

4.2.3. Qualitative evaluation

To investigate how much benefit ETLM can bring to our model, Table 3 presents top 5 words of the 5 most incoherent topics learned by CTM and four word embedding-based topic models with number of topics $K = 80$ on the News dataset. From Table 3, we observe that these incoherent topics usually contain words that are close together in the vector space (*e.g.*, "*made*", "*make*" and "*making*" in CRFTM, "*years*","*year*" and "*week*" in GPU-DMM, "*north*", "*south*" and "*east*" in GLTM, etc.). CRFTM, GPU-DMM and GLTM will give these words a high probability to share the same topic assignment. However, these words may not frequently co-occur in the same document. Therefore, these topics are often difficult to interpret, and not particularly useful. CRFTM, GPU-DMM and GLTM promote the semantically related words under the same topic with the same weight, resulting in their lower topic coherence scores. By contrast, this is less common in CCTM. The reason is that CCTM uses ETLM to learn quantifiable relationships from the whole corpus, and dynamically adjust the promotion weight. The experimental results demonstrate that ETLM can capture the co-occurrence information between two words in a specific context, which helps to improve the topic model.

4.2.4. Topic correlations

To investigate the correlations between topics extracted by CCTM, we qualitatively evaluate our approach by visualizing each topic with their top words as well as the correlation patterns. To make the visualization more clearly, we only choose 2 categories from the News dataset whose topic words and correlations can be easily identified and defined. In the experiment, 3000 news descriptions are selected in the "sports" and "entertainment" categories respectively. We set the number of topics K to 10 because selecting a value of K that is too high will cause "over-clustering" of the dataset. For each topic, we pick up the top 5 words extracted by CCTM, and the detected topic correlation is denoted as a solid line between topics.

In Fig. 7, edges represent a correlation between two topics and the edge thickness denotes the magnitude of their correlation. We can easily figure that several topics are correlated to each other and exhibit obvious correlation structure. For instance, a topic about NBA (Topic 2) is correlated with a topic of NFL (Topic 6) and a topic related to films (Topic 8) is correlated with a topic about stars (Topic 4). Furthermore, the set of topics in the left region are mainly about sport and are interrelated closely, while their connections to entertainment topics shown in the middle part are weak. The experiment results demonstrate the effectiveness of CCTM in discovering topics and topic correlations.

4.2.5. Impact of the ε value

In this part, we investigate the impact of local semantic correlation threshold ε in CCTM (see Eq. (9)). Experiments for sensitivity analysis are conducted under the setting of $K = \{40, 60, 80\}$ and $T = 10$. The threshold ε controls the number of word pairs with local semantic correlations. A smaller ε usually indicates fewer word pairs have local semantic correlations, whereas a larger ε indicates more globally correlated word pairs will be assigned to the same topic.

Fig. 8 reports UCI and UMass topic coherence of setting different ε values under $K = \{40, 60, 80\}$ and $T = 10$ on the StackOverflow dataset. If ε is set to zero, the semantic correlation between words is ignored and CCTM is equivalent to CTM. From Fig. 8(a), we find significant performance gains are achieved by CCTM over CTM (*i.e.*, when $\varepsilon = 0$) with any positive ε by using the UCI topic coherence. Specifically, the best UCI scores are achieved by setting $\varepsilon = \{0.1, 0.075, 0.1\}$ when $K = \{40, 60, 80\}$. A larger ε value results in significant performance degradation since more globally correlated words are allocated to the same topic label. Therefore, the probability that irrelevant words are put into the same topic becomes higher. A similar observation holds when using the UMass coherence, as shown in Fig. 8(b). Based on the results, we set $\varepsilon = 0.1$ in our experiments.

4.2.6. Topic evolution graphs

In this section, we evaluate the effectiveness of the proposed model OCCTM with a real world dataset. Taking MH370 air disaster[6] as an example, we selected 8056 tweets from March 8, to March 15, 2014 in a microblog dataset. The microblog feeds are gathered by [5] when given the search keyword "MH370". The dataset was preprocessed by the authors, *e.g.*, Chinese word segmentation and stop words removal, and we train 100-dimensional word embeddings from 7 million Chinese Baike website[7]. In the experiment, we empirically set the number of topics to 5, the span of time slices to one day and the threshold λ to 3. Other parameter values are identical to those used in CCTM. Note that the parameter λ has a "zooming" effect on the topic evolution graph. A low value of λ indicates that only the strongest evolutionary transition can be seen, whereas a high value of λ will allow us to see some of the weaker ones.

Fig. 9 shows the part of the topic evolution graph and each topic node is labeled with its five most probable words. Green boxes illustrate the main topic evolution timeline and white boxes denote related subtopics. Some edges between subtopics are omitted to better understand the main topic correlations. As shown, the main topic evolve from discussing the missing aircraft on March 8, to praying for the passengers on March 9. Meanwhile,

[6] https://en.wikipedia.org/wiki/MalaysiaAirlinesFlight370
[7] http://baike.baidu.com/

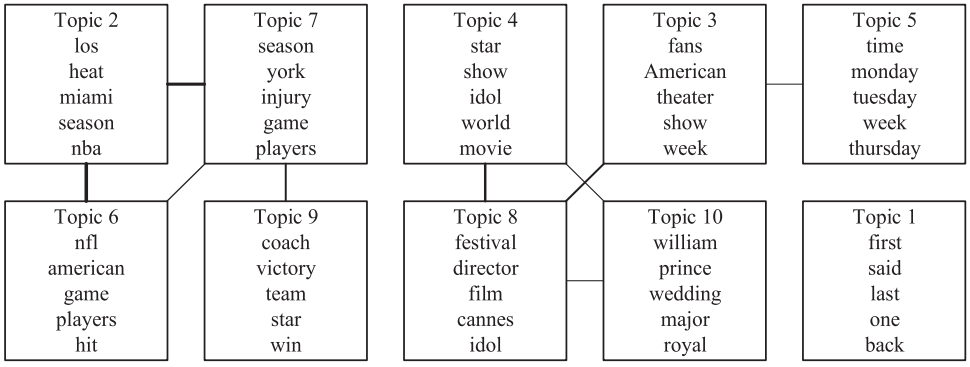


Fig. 7. Topic words and correlations obtained by our method from the News dataset.

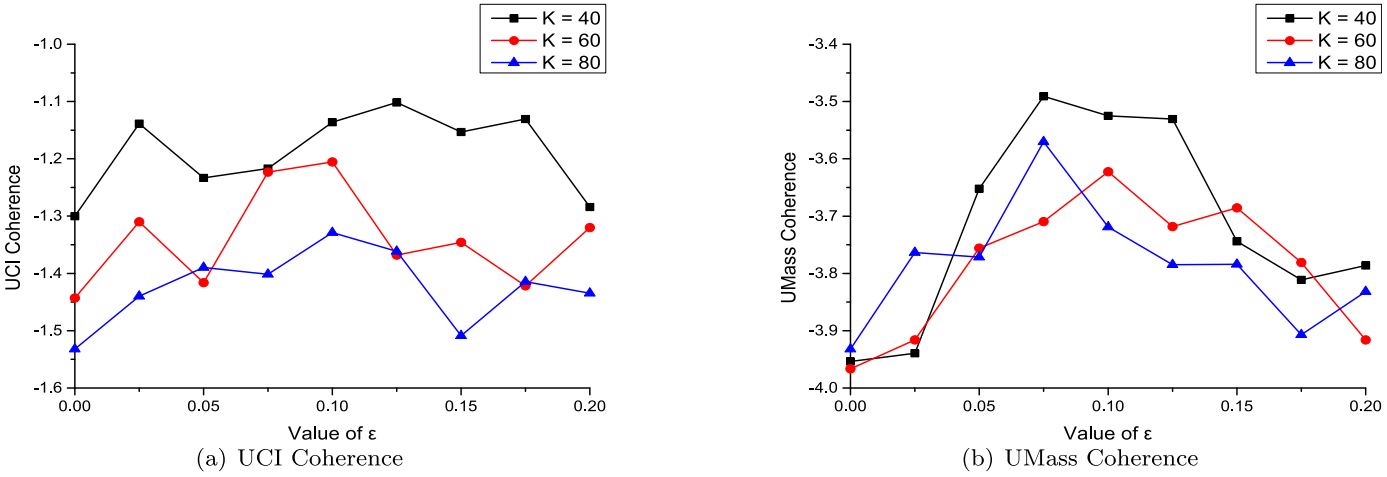


Fig. 8. Effect of threshold ε under the setting of $K = \{40, 60, 80\}$ and $T = 10$.

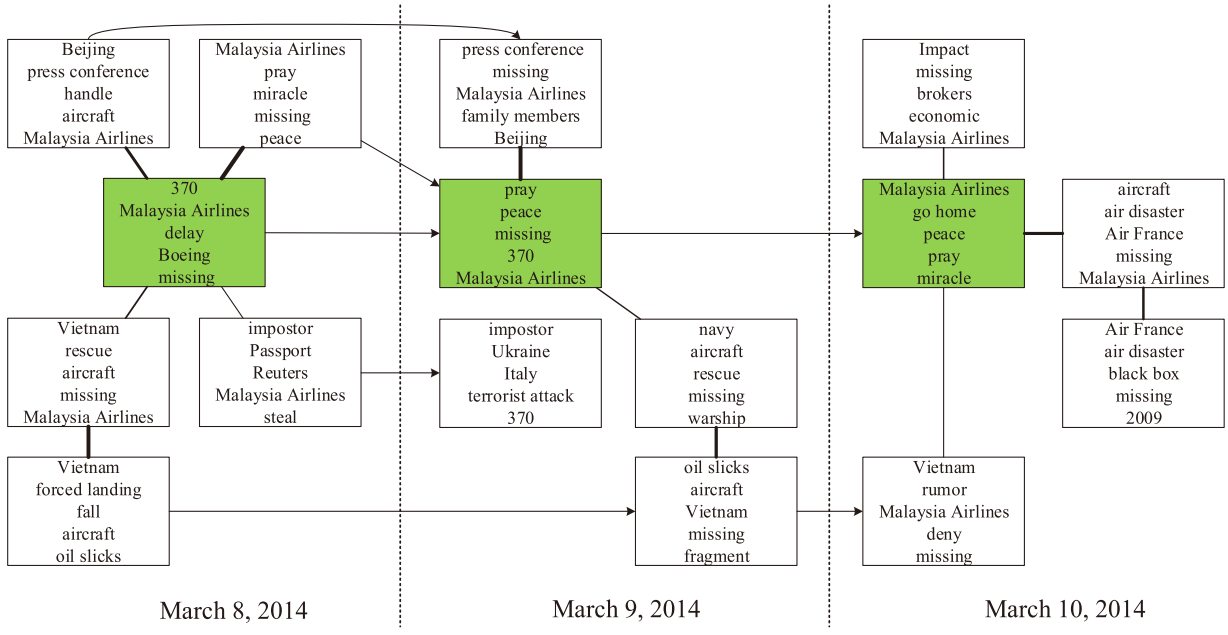


Fig. 9. A portion of the topic evolution graph of topic "MH370".

the subtopics "*the Vietnamese military found traces of oil*", "*the impostors boarded the plane*" and so forth, can help readers identify the development roadmap of the whole event.

5. Conclusion

In this paper, we propose a new framework for topic evolution mining on short texts. We first propose an Encoder-only Transformer Language Model (ETLM) to quantify the relationship between words. Next, we propose a novel topic model CCTM that incorporates global and local semantic correlations by using a Weighted Conditional Random Field (WCRF) to encourage semantically related words to share the same topic label. Finally, OCCTM is used to automatically find topics and topic correlations at each time slice, and construct topic evolutionary graphs. Experiments on real-world short text collections validate the effectiveness of our proposed methods. In the future, we will develop a parallel version of OCCTM in a distributed system, and evaluate it on more complete and large-scale datasets.

Declaration of Competing Interest

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

Acknowledgments

We would like to thank reviewers for their invaluable comments. This research was partially supported by the National Key R&D Program of China (No. 2018YFC1604000 / 2018YFC1604003) and National Science Foundation of China (NSFC, No. 61772382).

References

[1] 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.
[2] H. Fang, W. Lu, F. Wu, Y. Zhang, X. Shang, J. Shao, Y. Zhuang, Topic aspect-oriented summarization via group selection, Neurocomputing 149 (3) (2015) 1613–1619.
[3] S. Xiong, K. Wang, D. Ji, B. Wang, A short text sentiment-topic model for product reviews, Neurocomputing 297 (5) (2018) 1–13.
[4] Q. Deng, G. Cai, H. Zhang, Y. Liu, L. Huang, F. Sun, Enhancing situation awareness of public safety events by visualizing topic evolution using social media, in: Proceedings of the 19th Annual International Conference on Digital government Research: Governance in the Data Age, 2018, pp. 7:1–7:10.
[5] J. Zhu, X. Li, M. Peng, J. Huang, T. Qian, Coherent topic hierarchy: a strategy for topic evolutionary analysis on microblog feeds, in: Proceedings of the International Conference on Web-Age Information Management (WAIM), 2015, pp. 70–82.
[6] H. Zhou, H. Yu, R. Hu, Topic evolution based on the probabilistic topic model: a review, Front. Comput. Sci. 11 (5) (2017) 786–802.
[7] Y. Chen, H. Zhang, J. Wu, X. Wang, R. Liu, M. Lin, Modeling emerging, evolving and fading topics using dynamic soft orthogonal nmf with sparse representation, in: Proceedings of the IEEE International Conference on Data Mining (ICDM), 2015, pp. 61–70.
[8] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent dirichlet allocation, J. Mach. Learn. Res. 3 (2003) 993–1022.
[9] X. Wang, A. Mccallum, Topics over time: A non-markov continuous-time model of topical trends, in: Proceedings of the 12th ACM international conference on knowledge discovery and data mining (SIGKDD), 2006, pp. 424–433.
[10] L. Alsumait, D. Barbar, C. Domeniconi, On-line LDA: Adaptive topic models for mining text streams with applications to topic detection and tracking, in: Proceedings of the 8th IEEE International Conference on Data Mining, 2008, pp. 3–12.
[11] D.M. Blei, J.D. Lafferty, Dynamic topic models, in: Proceedings of the 23rd International Conference on Machine Learning (ICML), 2006, pp. 113–120.
[12] A. Dubey, A. Hefny, S. Williamson, E.P. Xing, A non-parametric mixture model for topic modeling over time, in: Proceedings of the SIAM International Conference on Data Mining, 2013, pp. 530–538.
[13] 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.
[14] L. Hong, B.D. Davison, Empirical study of topic modeling in twitter, in: Proceedings of the 1st Workshop on Social Media Analytics (SOMA), 2010, pp. 80–88.
[15] T. Mikolov, W.T. Yih, G. Zweig, Linguistic regularities in continuous space word representations, in: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL), 2013, pp. 746–751.
[16] W. Liang, R. Feng, X. Liu, Y. Li, X. Zhang, Gltm: a global and local word embedding-based topic model for short texts, IEEE Access 6 (2018) 43612–43621.
[17] G. Xun, V. Gopalakrishnan, F. Ma, Y. Li, J. Gao, A. Zhang, Topic discovery for short texts using word embeddings, in: Proceedings of the IEEE International Conference on Data Mining (ICDM), 2016, pp. 1299–1304.
[18] C. Li, Y. Duan, H. Wang, Z. Zhang, A. Sun, Z. Ma, Enhancing topic modeling for short texts with auxiliary word embeddings, ACM Trans. Inf. Syst. 36 (2) (2017) 11:1–11:30.
[19] W. Gao, M. Peng, H. Wang, Y. Zhang, Q. Xie, G. Tian, Incorporating word embeddings into topic modeling of short text, Knowl. Inf. Syst. 61 (2) (2019) 1123â1145.
[20] D.M. Blei, J.D. Lafferty, Correction: a correlated topic model of science, Ann. Appl. Stat. 1 (1) (2007) 17–35.
[21] W. Huang, Phrasectm: correlated topic modeling on phrases within Markov random fields, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL), 2018, pp. 521–526.
[22] G. Xun, Y. Li, W.X. Zhao, J. Gao, A. Zhang, A correlated topic model using word embeddings, in: Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI), 2017, pp. 4207–4213.
[23] J.D. Lafferty, A. Mccallum, F.C.N. Pereira, Conditional random fields: probabilistic models for segmenting and labeling sequence data, in: Proceedings of the 18th International Conference on Machine Learning (ICML), 2001, pp. 282–289.
[24] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L.u. Kaiser, I. Polosukhin, Attention is all you need, in: Proceedings of the Advances in Neural Information Processing Systems (NIPS), 2017, pp. 5998–6008.
[25] J. Weng, E.P. Lim, J. Jiang, Q. He, Twitterrank: Finding topic-sensitive influential twitterers, in: Proceedings of the 3rd ACM International Conference on Web Search and Data Mining (WSDM), 2010, pp. 261–270.
[26] 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: Proceedings of the 33rd European conference on information retrieval (ECIR), 2011, pp. 338–349.
[27] 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 Conference on Research and Development in Information Retrieval (SIGIR), 2013, pp. 889–892.
[28] R. Das, M. Zaheer, C. Dyer, Gaussian lda for topic models with word embeddings, in: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL), 2015, pp. 795–804.
[29] N. Kawamae, Trend analysis model: Trend consists of temporal words, topics, and timestamps, in: Proceedings of the 4th ACM International Conference on Web Search and Data Mining, 2011, pp. 317–326.
[30] A. Ahmed, E.P. Xing, Timeline: a dynamic hierarchical dirichlet process model for recovering birth/death and evolution of topics in text stream, in: Proceedings of the 26th Uncertainty in Artificial Intelligence (UAI), 2010, pp. 20–29.
[31] Y. Wang, E. Agichtein, M. Benzi, Tm-lda: efficient online modeling of latent topic transitions in social media, in: Proceedings of the 18th ACM international conference on knowledge discovery and data mining (SIGKDD), 2012, pp. 123–131.
[32] K. Sasaki, T. Yoshikawa, T. Furuhashi, Twitter-ttm: An efficient online topic modeling for twitter considering dynamics of user interests and topic trends, in: Proceedings of the 7th International Conference on Soft Computing and Intelligent Systems (SCIS), 2014, pp. 440–445.
[33] I.V. Serban, A. Sordoni, R. Lowe, L. Charlin, J. Pineau, A. Courville, Y. Bengio, A hierarchical latent variable encoder-decoder model for generating dialogues, in: Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI), 2017, pp. 3295–3301.
[34] J. Chen, J. Zhu, Z. Wang, X. Zheng, B. Zhang, Scalable inference for logistic-normal topic models, in: Proceedings of the Advances in Neural Information Processing Systems (Nips), 2013, pp. 2445–2453.
[35] N.G. olson, J.G. Scott, J. Windle, Bayesian inference for logistic models using polyaâgamma latent variables, J. Am. Stat. Assoc. 108 (504) (2013) 1339–1349.
[36] H.M. Wallach, D.M. Mimno, A. McCallum, Rethinking lda: why priors matter, Adv. Neural Inf. Process. Syst. 23 (2009) 1973–1981.
[37] X.H. Phan, L.M. Nguyen, S. Horiguchi, Learning to classify short and sparse text and web with hidden topics from large-scale data collections, in: Proceedings of the 17th International Conference on World Wide Web (WWW), 2008, pp. 91–100.
[38] J. Xu, P. Wang, G. Tian, B. Xu, J. Zhao, F. Wang, H. Hao, Short text clustering via convolutional neural networks, in: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL), 2015, pp. 62–69.
[39] A. Schofield, D. Mimno, Comparing apples to apple: the effects of stemmers on topic models, Trans. Assoc. Comput. Linguist. 4 (2016) 287–300.
[40] E.H. Ramirez, R. Brena, D. Magatti, F. Stella, Topic model validation, Neurocomputing 76 (1) (2012) 125–133.
[41] J. Chang, J. Boyd-Graber, S. Gerrish, C. Wang, D.M. Blei, Reading tea leaves: How humans interpret topic models, in: Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS), 2009, pp. 288–296.
[42] D. Newman, J.H. Lau, K. Grieser, T. Baldwin, Automatic evaluation of topic coherence, in: Proceedings of the Conference of the North American Chapter of

the Association for Computational Linguistics: Human Language Technologies (NAACL), 2010, pp. 100–108.
[43] 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 (EMNLP), 2011, pp. 262–272.

Wang Gao received M.S degree in Software Engineering from Wuhan University of Technology, China, in 2011. He is currently pursuing Ph.D. degree in the School of Computer Science at Wuhan University. His current research interests include natural language processing and information retrieval.

Ming Peng received the M.S and Ph.D degree from the Wuhan University, Wuhan, China, in 2002 and 2006. She is currently a Professor at School of Computer Science, Wuhan University. Currently, she works on machine comprehension, automatic text summarization and social network analysis. She is a member of the China Computer Federation (CCF).

Hua Wang is a full time Professor at the Centre for Applied Informatics, Victoria University. Before joining Victoria University, he was a Professor at the University of Southern Queensland (USQ) during 2011–2013. He obtained his Ph.D in Computer Science from USQ in 2004.

Yanchun Zhang is Professor and Director of the Centre for Applied Informatics in Victoria University. His current research interests include databases, data mining, health informatics, web information systems and web services.

Weiguang Han received the B.S degree from Wuhan University, China, in 2016. He is currently working toward the Ph.D degree in the School of Computer Science at Wuhan University. His research interests include machine comprehension and information retrieval.

Gang Hu received the M.S degree in signal and information processing from Yunnan Minzu University, Kunming, China, in 2016. He is currently working toward the Ph.D degree at Wuhan University. His current research interests include search-based software engineering and mining software repositories.

Qianqian Xie is a Ph.D candidate in School of Computer Science at Wuhan University, Wuhan, China. She received her bachelors degrees from Jiangxi Normal University. Her current research focus areas include natural language processing, machine learning and deep learning.