Extracting Topics with Simultaneous Word Co-occurrence and Semantic Correlation Graphs: Neural Topic Modeling for Short Texts

Yiming Wang[1,2] **, Ximing Li**[1,2] *†**, Xiaotang Zhou**[3]**, Jihong Ouyang**[1,2]*
[1]College of Computer Science and Technology, Jilin University, China
[2]Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, China
[3]School of Computer Science and Engineering, Changchun University of Technology, China
{yimingw17, liximing86}@gmail.com
zhouxiaotang@ccut.edu.cn, ouyj@jlu.edu.cn

Abstract

Short text nowadays has become a more fashionable form of text data, *e.g.,* Twitter posts, news titles, and product reviews. Extracting semantic topics from short texts plays a significant role in a wide spectrum of NLP applications, and neural topic modeling is now a major tool to achieve it. Motivated by learning more coherent and semantic topics, in this paper we develop a novel neural topic model named Dual Word Graph Topic Model (DWGTM), which extracts topics from simultaneous word co-occurrence and semantic correlation graphs. To be specific, we learn word features from the global word co-occurrence graph, so as to ingest rich word co-occurrence information; we then generate text features with word features, and feed them into an encoder network to get topic proportions per-text; finally, we reconstruct texts and word co-occurrence graph with topical distributions and word features, respectively. Besides, to capture semantics of words, we also apply word features to reconstruct a word semantic correlation graph computed by pre-trained word embeddings. Upon those ideas, we formulate DWGTM in an auto-encoding paradigm and efficiently train it with the spirit of neural variational inference. Empirical results validate that DWGTM can generate more semantically coherent topics than baseline topic models.

1 Introduction

The topic modeling family targets at learning latent topic representations from text document collections (Blei, 2012). During the past decades, it has been extensively applied in many tasks of natural language processing, *e.g.,* sentiment analysis (Lin and He, 2009), summarization (Ma et al., 2012) and classification (Zeng et al., 2018), to name just a few. Conventional topic models such as Latent Dirichlet Allocation (LDA) (Blei et al., 2003) are often inferred by approximate inference methods, *e.g.,* mean-field variational inference (Jordan et al., 1999) and collapsed Gibbs sampling (Griffiths and Steyvers, 2004), which require model-specific derivations. The recent inference method with neural networks, such as Variational Auto-Encoder (VAE) (Kingma and Welling, 2014), works in a black-box manner, providing a more generic and flexible solution to topic models beyond traditional approximate inference methods. Broadly speaking, the models inferred with neural networks are referred to as **neural topic models**, and they have been recently drawn much more attention from the natural language processing community (Zhu et al., 2018; Burkhardt and Kramer, 2019; Dieng et al., 2020; Wu et al., 2020).

Unfortunately, whether for conventional or neural topic models, they tend to perform poorly on **short text**, a more fashionable and significant form of text data, *e.g.,* Twitter posts, news titles, and product reviews. The main reason is that short texts lack document-level word co-occurrences, known as the sparsity problem, which hinders models to capture coherent word patterns. Many conventional topic models have been developed to handle short texts. For example, given very few words per-text, Dirichlet Multinomial Mixture (DMM) (Nigam et al., 2000; Yin and Wang, 2014) constrains that each text covers a signal topic. Biterm Topic Model (BTM) (Yan et al., 2013; Cheng et al., 2014) directly learns topics from corpus-level word co-occurrence patterns. Recently, there are also few attempts of neural topic models aiming to address the sparsity problem of short texts. GraphBTM (Zhu et al., 2018) extracts topics from word graphs of randomly drawn mini-corpus. Negative sampling and Quantization Topic Model (NQTM) (Wu et al., 2020) applies a topic distribution quantization method to pursue peakier topic proportions of texts. As reported in (Wu et al., 2020), those neu-

* Corresponding Author
† Contributing equally with the first author.

Findings of the Association for Computational Linguistics: EMNLP 2021, pages 18–27
November 7–11, 2021.

ral topic models can empirically induce coherent topics from short texts.

Motivated by learning more coherent and semantic topics, in this paper we develop a novel neural topic model for short texts, namely **D**ual **W**ord **G**raph **T**opic **M**odel (**DWGTM**). As the name suggests, in DWGTM we apply two word graphs, including the **word co-occurrence graph** constructed by aggregating word co-occurrence patterns of each text to alleviate the sparsity problem, and the **word semantic correlation graph** generated by using the pre-trained word embeddings to capture the semantic information of words. Specifically, we formulate DWGTM in an auto-encoding paradigm with four main components: (1) We encode the word co-occurrence graph as word features by applying a Graph Convolutional Network (GCN) module (Kipf and Welling, 2016a). (2) For each text, we construct its feature with corresponding word features, and encode it as topic proportion. (3) Reconstruct texts with topical distributions. (4) Reconstruct the two word graphs with word features. With the word semantic correlation graph, DWGTM can output topics that are associated with the semantic information of words. Besides, we propose a novel topic quality metric to measure the semantic coherence of learned topics, namely **T**opical **S**emantics **C**oherence (**TSC**). We conduct extensive experiments to evaluate DWGTM, and empirical results indicate that DWGTM can learn more semantically coherent topics than existing baseline models.

In a nutshell, the major contributions of this paper are listed below:

- We propose a novel neural topic model **DWGTM** for short texts, extracting topics from simultaneous word co-occurrence and semantic correlation graphs.
- We propose a novel topic quality metric called **TSC**, which measures the semantic coherence of learned topics.
- On three benchmark datasets of short texts, **DWGTM** empirically outputs more semantically coherent topics than strong baseline models.

2 Related Work

In this section, we briefly review related works on conventional topic modeling of short texts and neural topic modeling.

2.1 Topic Modeling for Short Texts

Short texts lack the document-level word co-occurrence information, making conventional topic models such as LDA (Blei et al., 2003) much less effective. To resolve this issue of short text, existing models mainly adopt the methodology of word co-occurrence enrichment (Yan et al., 2013; Yin and Wang, 2014; Quan et al., 2015; Zuo et al., 2016a,b; Li et al., 2016, 2018; Shi et al., 2018; Li et al., 2019a,b, 2020a). First, one straightforward way is to generate long pseudo-texts by adaptively aggregating short texts and then learn topics from them by applying LDA. Several representatives (Quan et al., 2015; Zuo et al., 2016a; Li et al., 2018) jointly estimate long pseudo-texts and topics, however, they are often time consuming as well as sensitive to the number of long pseudo-texts. Second, another mainstream is to extract more word co-occurrences at the corpus level. The BTM (Yan et al., 2013; Cheng et al., 2014) directly induces topics from all word co-occurrence patterns of the corpus. Semantics-assisted Non-negative Matrix Factorization (SeaNMF) (Shi et al., 2018) regards each word type as a pseudo-text consisting of the words that co-occur with it in the same short text, and learns topics with those auxiliary word type pseudo-texts. Additionally, other attempts (Li et al., 2016, 2019a) upgrade existing models, *e.g.,* DMM and BTM, by further leveraging auxiliary knowledge or techniques such as word semantic correlations measured by pre-trained word embeddings (Mikolov et al., 2013; Pennington et al., 2014). In contrast to aforementioned models, our DWGTM is built on the framework of neural variational inference with GCN (Kipf and Welling, 2016a), enabling to effectively extract topics with word co-occurrence patterns.

2.2 Neural Topic Modeling

Along the new research line of integrating VAE (Kingma and Welling, 2014), a number of neural topic models have been proposed. Generally, the basic idea of neural topic modeling is to apply neural networks as topic encoders to induce topic representations of texts, and reconstruct texts with topical distributions. Benefiting from the effectiveness and flexibility of neural networks in unsupervised representation learning, neural topic models can induce more significant topics from texts. Nowadays, the representatives include Neural Variational Document Model (NVDM) (Miao et al., 2016), Product

of expert LDA (ProdLDA) (Srivastava and Sutton, 2017), and Embedded Topic Model (ETM) (Dieng et al., 2020), *etc.* Besides these "naive" neural variants of LDA, many other models have been investigated by applying (1) various neural modules to the topic encoder, *e.g.,* recurrent module (Rezaee and Ferraro, 2020), attention mechanism (Li et al., 2020b), and graphical connection (Zhu et al., 2018; Yang et al., 2020), and (2) new learning paradigms, *e.g.,* adversarial training (Wang et al., 2019), reinforcement learning (Gui et al., 2019), and lifelong learning (Gupta et al., 2020). However, despite their effectiveness on normal long texts, those models suffer from the sparsity problem of short texts (Zeng et al., 2018).

To our knowledge, there are only a few neural topic models for addressing the sparsity problem of short texts (Zeng et al., 2018; Zhu et al., 2018; Wu et al., 2020). Inspired by BTM (Yan et al., 2013), the GraphBTM method (Zhu et al., 2018) directly learns topics from the aggregated word co-occurrence patterns of randomly generated mini-corpus. The NQTM method (Wu et al., 2020) is based on the assumption that the peakier topic proportions of texts are more appropriate for modeling short texts as demonstrated in DMM (Yin and Wang, 2014), To achieve this, it applies a topic distribution quantization method, and meanwhile it adopts a negative sampling step to avoid repetitive topics. Orthogonal to those models, our DWGTM further employs the pre-trained word embeddings to capture the semantic information of words, so as to output more semantically coherent topics.

3 The Proposed DWGTM Model

In this section, we introduce the proposed **D**ual **W**ord **G**raph **T**opic **M**odel (**DWGTM**). For convenience, the important notations used in this paper are summarized in Table 1.

3.1 Overview of DWGTM

The topic modeling family such as LDA (Blei et al., 2003) refers to the probabilistic model that describes the generative process of documents. Basically, it posits totally k topics $\boldsymbol{\phi}_{1:k}$, each of which is a multinomial distribution over the vocabulary, and each document is represented by a topic proportion $\boldsymbol{\theta}$. Given a corpus $\mathcal{D}$ consisting of n documents $\mathbf{x}_{1:n}$, the main goal of topic modeling is to estimate topics $\boldsymbol{\phi}_{1:k}$ and topic proportions $\boldsymbol{\theta}_{1:n}$ from $\mathcal{D}$. However, it is commonly intractable to accurately estimate $\{\boldsymbol{\phi}, \boldsymbol{\theta}\}$ for short texts, due to the lack of document-level word co-occurrences, known as the sparsity problem.

Table 1: Notation summary.

Notation	Description
n	number of texts
v	vocabulary size
k	number of topics
$\mathbf{x}$	word frequency vector of text
$\mathcal{G}^c$	word co-occurrence graph
$\mathcal{G}^s$	word semantic correlation graph
$\mathbf{z}^w$	word feature
$\mathbf{z}^t$	latent text feature
$\boldsymbol{\phi}$	topic distribution
$\boldsymbol{\theta}$	topic proportion of text
$\mathbf{W}^c$	learnable parameter of WCG-Encoder
$\mathbf{W}^t$	learnable parameter of TP-Encoder

To effectively handle short texts, we propose a novel neural topic model named **DWGTM** by not only leveraging the corpus-level word co-occurrence information to address the sparsity problem (Yan et al., 2013), but also capturing word semantic correlations measured by pre-trained word embeddings (Mikolov et al., 2013; Pennington et al., 2014). Specifically, as depicted in Fig.1, DWGTM consists of the four components in the auto-encoding manner. (1) **WCG-Encoder**: We construct a global word co-occurrence graph $\mathcal{G}^c$, and then encode $\mathcal{G}^c$ as word features $\mathbf{z}^w_{1:v}$, where v denotes the vocabulary size. (2) **TP-Encoder**: We construct latent text features $\mathbf{z}^t_{1:n}$ by using $\mathbf{z}^w_{1:v}$, and then encode $\mathbf{z}^t_{1:n}$ as topic proportions $\boldsymbol{\theta}_{1:n}$. (3) **Text-Decoder**: We reconstruct the texts $\mathbf{x}_{1:n}$ with $\boldsymbol{\theta}_{1:n}$ and topics $\boldsymbol{\phi}_{1:k}$. (4) **DualWG-Decoder**: We reconstruct $\mathcal{G}^c$ with $\mathbf{z}^w_{1:v}$. Meanwhile, to further capture semantic information of words, we construct a word semantic correlation graph $\mathcal{G}^s$ by using pre-trained word embeddings, and reconstruct $\mathcal{G}^s$ with also $\mathbf{z}^w_{1:v}$. In the following part, we introduce each component of DWGTM in more details.

3.2 WCG-Encoder

Given a corpus $\mathcal{D}$, we first construct a word co-occurrence graph $\mathcal{G}^c = (\mathcal{V}, \mathcal{E}^c)$, where $\mathcal{V}$ and $\mathcal{E}^c$ denote the sets of word nodes and word co-occurrence edges, respectively. That is, the graph can be represented by the co-occurrence adjacency matrix $\mathbf{A}^c \in \mathbb{R}^{v \times v}$, where each element $\mathbf{A}^c_{ij}$ denotes the count of words w_i and w_j co-occurring in the same text. The WCG-encoder targets at encoding $\mathbf{A}^c$ as word features $\mathbf{Z}^w = [\mathbf{z}^w_1, \cdots, \mathbf{z}^w_v]^\top$, so that more

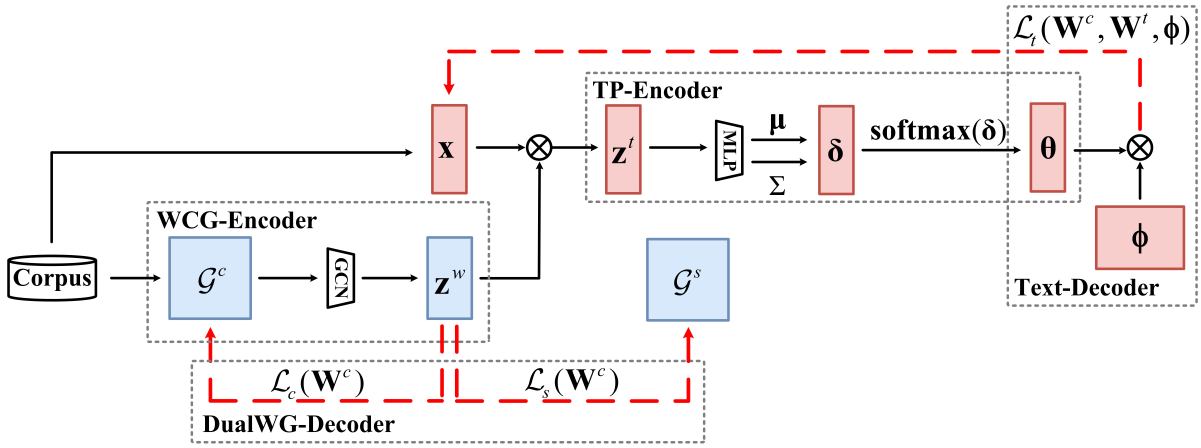

Figure 1: Overall model structure of DWGTM with four components, *i.e.,* **WCG-Encoder**, **TP-Encoder**, **Text-Decoder**, and **DualWG-Decoder**.

frequently co-occurring words tend to have more similar word features. This is achieved by applying a GCN module parameterized by $\mathbf{W}^c$:

$$\mathbf{Z}^w = f_{\text{GCN}}(\mathbf{A}^c; \mathbf{W}^c) \tag{1}$$

Following (Kipf and Welling, 2016a), each layer of the GCN module is formulated below:

$$\mathbf{Z}^w_{(l)} = \psi(\tilde{\mathbf{A}}^c \mathbf{Z}^w_{(l-1)} \mathbf{W}^c_{(l)}), \qquad l = 1, \cdots, l^c, \tag{2}$$

where l^c is the number of layers; $\mathbf{W}^c = \{\mathbf{W}^c_{(l)}\}_{l=1}^{l^c}$ are the learnable parameters; $\mathbf{Z}^w_{(0)}$ is initialized by the identity matrix $\mathbf{I}_v$ with the shape of v; $\psi(\cdot)$ denotes the Tanh activation function; $\tilde{\mathbf{A}}^c = \mathbf{D}^{-\frac{1}{2}}(\mathbf{A}^c + \mathbf{I}_v)\mathbf{D}^{-\frac{1}{2}}$ is the the symmetrically normalized adjacency matrix; and $\mathbf{D}$ denotes the degree matrix of $\mathbf{A}^c + \mathbf{I}_v$.

3.3 TP-Encoder

Naturally, the resulting word features $\mathbf{Z}^w$ learned from $\mathcal{G}^c$ are rich in global word co-occurrence information. Accordingly, we can use $\mathbf{Z}^w$ to generate latent text features $\mathbf{z}^t_{1:n}$, enabling to alleviate the sparsity problem of short texts. For each short text, the latent text feature can be easily obtained by aggregating its corresponding word features, formulated below:

$$\mathbf{z}^t_d = (\mathbf{Z}^w)^\top \frac{\mathbf{x}_d}{|\mathbf{x}_d|}, \qquad d = 1, \cdots, n, \tag{3}$$

where $\mathbf{x}_d$ and $|\mathbf{x}_d|$ denote the word frequency vector of the d^{th} document and its total number of word tokens, respectively.

The TP-Encoder aims at encoding $\mathbf{z}^t_{1:n}$ as topic proportions $\boldsymbol{\theta}_{1:n}$. Inspired by (Miao et al., 2016; Dieng et al., 2020), we apply the VAE-like paradigm with logistic-normal prior distribution. Specifically, suppose that for each short text the topic proportion is drawn from a logistic-normal prior as follows:

$$\boldsymbol{\delta}_d \sim \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0); \quad \boldsymbol{\theta}_d = \textbf{softmax}(\boldsymbol{\delta}_d), \quad d = 1, \cdots, n, \tag{4}$$

where $\boldsymbol{\delta}$ can be regarded as the unnormalized topic proportion; and $\mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$ denotes a Gaussian prior probability. We apply a fully-connected module, *a.k.a.,* variational inference network (Dieng et al., 2020), which ingests each latent text feature $\mathbf{z}^t_d$ and outputs the mean $\boldsymbol{\mu}_d$ and covariance $\boldsymbol{\Sigma}_d$ of the unnormalized topic proportion $\boldsymbol{\delta}_d$, formulated below:

$$\mathbf{H}_{(l)} = \rho(\mathbf{W}^t_{(l)} \mathbf{H}_{(l-1)}), \quad l = 1, \cdots, l^t. \tag{5}$$

$$\boldsymbol{\mu}_d = \mathbf{W}^t_{\boldsymbol{\mu}} \cdot \mathbf{H}_{(l^t)} \tag{6}$$

$$\boldsymbol{\Sigma}_d = \mathbf{W}^t_{\boldsymbol{\Sigma}} \cdot \mathbf{H}_{(l^t)}. \tag{7}$$

where l^t denotes the number of layers; $\mathbf{W}^t = \{\{\mathbf{W}^t_{(l)}\}_{l=1}^{l^t}, \mathbf{W}^t_{\boldsymbol{\mu}}, \mathbf{W}^t_{\boldsymbol{\Sigma}}\}$ are the learnable parameters; $\mathbf{H}_{(0)}$ is initialized by $\mathbf{z}^t_d$; and $\rho(\cdot)$ denotes the Tanh activation function. We then compute the topic proportion $\boldsymbol{\theta}_d$ by leveraging the reparameterization trick (Kingma and Welling, 2014):

$$\boldsymbol{\theta}_d = \textbf{softmax}(\boldsymbol{\mu}_d + \boldsymbol{\Sigma}_d \odot \epsilon), \quad \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_k) \tag{8}$$

where $\odot$ denotes element-wise product; and ϵ is a sample drawn from the Gaussian $\mathcal{N}(\mathbf{0}, \mathbf{I}_k)$. Due to the space limitation, we omit background descriptions of this VAE-like paradigm and reparameterization, and refer the readers to more details in (Kingma and Welling, 2014; Mnih and Gregor, 2014; Rezende et al., 2014; Miao et al., 2016).

Remark. Strictly speaking, the reparameterization trick (*i.e.,* Eq.8) is really meant for forming the Monte Carlo approximation of the variational objective (Kingma and Welling, 2014), and it should even be described in the decoding process. We kindly emphasize that we introduce Eq.8 as a step of encoding $\boldsymbol{\theta}$ for the sake of a more intuitive expression for the TP-Encoder.

3.4 Text-Decoder

In this component, we reconstruct the texts $\mathbf{x}_{1:n}$ with topic proportions $\boldsymbol{\theta}_{1:n}$ and topics $\boldsymbol{\phi}_{1;k}$. Following the spirit of VAE derivation (Kingma and Welling, 2014), the reconstruction loss of $\mathbf{x}_{1:n}$ consists of a log marginal likelihood term and a KL-divergence regularier as follows:

$$\mathcal{L}_t(\mathbf{W}^c, \mathbf{W}^t, \boldsymbol{\phi}) = -\log p(\mathbf{x}) + \mathcal{R}_{\mathrm{KL}}, \quad (9)$$

We adhere to the generative assumption of LDA-like models, therefore the marginal likelihood term of texts can be formulated below:

$$p(\mathbf{x}) = \prod_{d=1}^{n} \prod_{i \in \mathbf{x}_d} \sum_{t=1}^{k} \boldsymbol{\theta}_{dt} \boldsymbol{\phi}_{ti}, \quad (10)$$

where $\boldsymbol{\theta}_{i:n}$ are computed by Eq.8. Second, the KL-divergence regularizer admits a closed-form expression as follows:

$$\mathcal{R}_{\mathrm{KL}} = -\frac{1}{2} \sum_{d=1}^{n} (1 + \log |\boldsymbol{\Sigma}_d| - \boldsymbol{\mu}_d^\top \boldsymbol{\mu}_d - \mathbf{Tr}(\boldsymbol{\Sigma}_d)), \quad (11)$$

where $\mathbf{Tr}(\cdot)$ denotes the trace of a matrix.

3.5 DualWG-Decoder

As its name suggests, the aim of DualWG-Decoder is two-fold: applying the word features $\mathbf{z}^w_{1:v}$ to reconstruct the word co-occurrence graph $\mathcal{G}^c$ and also an auxiliary word semantic correlation graph $\mathcal{G}^s$.

Reconstruction of $\mathcal{G}^c$. Following (Kipf and Welling, 2016b), we apply an inner product decoder with word features. Accordingly, the reconstruction loss is formulated as follows:

$$\mathcal{L}_c(\mathbf{W}^c) = -\sum_{\{i,j\} \in \mathcal{E}^c} \mathbf{A}^c_{ij} \log \sigma\left((\mathbf{z}^w_i)^\top \mathbf{z}^w_j\right), \quad (12)$$

where $\sigma(\cdot)$ denotes the Sigmoid function.

Reconstruction of $\mathcal{G}^s$. Besides extracting topics by applying the word co-occurrence statistics, We expect to take the semantic information of words into consideration, so as to generate more semantically coherent topics (Li et al., 2016, 2019a). To achieve this, we construct a word semantic correlation graph $\mathcal{G}^s = (\mathcal{V}, \mathcal{E}^s)$, where $\mathcal{E}^s$ denotes the set of word semantic correlation edges. Let $\mathbf{A}^s \in \mathbb{R}^{v \times v}$ be the corresponding adjacency matrix, where each element $\mathbf{A}^s_{ij}$ reflects the cosine similarity between pre-trained GloVe embeddings[1] of words w_i and w_j. To be specific, it is formulated as follows:

$$\mathbf{A}^s_{ij} = \begin{cases} \gamma_{ij}, & \text{if } \gamma_{ij} > \gamma^* \\ 0, & \text{otherwise} \end{cases}, \quad (13)$$

where $\gamma_{ij} = \cos(\mathbf{g}_i, \mathbf{g}_j)$ denotes the cosine similarity; the notation $\mathbf{g}$ specifies the pre-trained GloVe word embedding; and γ^* is a word semantic correlation threshold.

We reconstruct $\mathcal{G}^s$ by encouraging the resulting word features to capture word semantic correlations. Accordingly, the reconstruction loss of $\mathcal{G}^s$ can be formulated below:

$$\mathcal{L}_s(\mathbf{W}^c) = \sum_{\{i,j\} \in \mathcal{E}^s} \| \cos(\mathbf{z}^w_i, \mathbf{z}^w_j) - \gamma_{ij} \|_2^2, \quad (14)$$

where $\| \cdot \|_2$ denotes the ℓ_2 norm.

3.6 Full Objective of DWGTM

We now outline the full objective of DWGTM. Except the reconstruction losses of $\mathbf{x}_{1:n}$, $\mathcal{G}^c$, and $\mathcal{G}^s$, we also incorporate the following entropy regularization term to encourage peakier topic proportions:

$$\mathcal{R}_{\mathrm{E}} = -\sum_{d=1}^{n} \sum_{t=1}^{k} \boldsymbol{\theta}_{dt} \log \boldsymbol{\theta}_{dt} \quad (15)$$

Finally, we can reach the full objective with respect to the learnable parameters $\{\mathbf{W}^c, \mathbf{W}^t, \boldsymbol{\phi}\}$ as follows:

$$\mathcal{L}(\mathbf{W}^c, \mathbf{W}^t, \boldsymbol{\phi}) = \mathcal{L}_t(\mathbf{W}^c, \mathbf{W}^t, \boldsymbol{\phi}) + \lambda_1 \mathcal{L}_c(\mathbf{W}^c) + \lambda_2 \mathcal{L}_s(\mathbf{W}^c) + \lambda_3 \mathcal{R}_{\mathrm{E}}, \quad (16)$$

where λ_1, λ_2, and λ_3 are the scale parameters.

4 Experiment

Datasets. In the experiments, we select three benchmark short text datasets: *Trec*,[2] *Google-*

[1] https://nlp.stanford.edu/projects/glove/

[2] http://cogcomp.cs.illinois.edu/Data/QA/QC

Table 2: Statistics of short text datasets after preprocessing. n: number of short texts. v: vocabulary size. $\widehat{n}$: average document length. l: number of categories.

Dataset	n	v	$\widehat{n}$	l
Trec	4,198	989	3.2	6
GoogleNews	9,284	659	3.7	152
YahooAnswer	22,937	1297	3.8	10

News,[3] and *YahooAnswer*.[4]For all datasets, we remove digits and words with term frequencies less than 20. Stop words and non-english words are filtered out by NLTK.[5] For clarity, the statistics of those datasets are listed in Table 2.

Baseline Topic Models. We select 8 existing baselines, including 4 conventional topic models and 4 neural topic models. Following their original papers, the important implementation details of all baselines are described below.

- **LDA**[6] (Blei et al., 2003): The model is inferred by variational inference, and the Dirichlet priors for topic proportions and topic distributions are set to 0.1 and 0.01, respectively.
- **DMM**[7] (Yin and Wang, 2014): The two Dirichlet priors are set as $50/k$ and 0.01, respectively.
- **BTM**[8] (Yan et al., 2013): The two Dirichlet priors are set as 0.01 and 0.001, respetively.
- **G**eneralized **P**ólya **U**rn **DMM** (**GPUDMM**)[9] (Li et al., 2016): The two Dirichlet priors are set as $50/k$ and 0.01, respectively; and the similarity threshold is set as 0.8.
- **NVDM**[10] (Miao et al., 2016): The model applies a 2-layer MLP encoder with 500 hidden neurons.
- **ProdLDA**[11] (Srivastava and Sutton, 2017): The model applies a 3-layer MLP encoder with 100 hidden neurons.
- **GraphBTM**[12] (Zhu et al., 2018): The model applies a 3-layer GCN encoder with 100 hidden neurons and samples 3 documents as a mini-corpus.
- **NQTM**[13] (Wu et al., 2020): The model applies a 3-layer MLP encoder with 100 hidden neurons and the word sample size for negative sampling is set as 20.

For DWGTM, we apply a 2-layer GCN WCG-Encoder and a 3-layer MLP TP-Encoder, where the hidden neurons of both encoders are set as 100-300-400-300-k. To avoid posterior collapsing, we adopt 0.4 dropout, batch normalization, and a shallower 1-layer Text-Decoder. The threshold γ^* is set to 0.6 for *Trec*, and 0.8 for *GoogleNews* and *YahooAnswer*. Scale parameters are set as $\lambda_1 = 0.1, \lambda_2 = 0.1, \lambda_3 = 1$. The number of epochs is 900 and mini-batch size is 200. To construct $\mathcal{G}^s$, we employ the 300-dimensional GloVe[14] embeddings trained on Wikipedia2014 and Gigaword5. For fair comparisons, the baselines requiring word embeddings use the same GloVe embeddings.

Evaluation Metrics. To evaluate the topic quality, we adopt two metrics: Topic Coherence (TC) and Topical Semantics Coherence (TSC).

First, TC is the most popular topic quality metric that measures the co-occurrence statistics between top-m words of topics. Here, we compute the TC scores with the public TC project of *Palmetto*,[15] where, especially, the setting of C_V is applied. Second, we propose a novel metric named TSC to measure the semantic coherence of topics. Analogy to TC, we suppose that higher similarities between top-m words of topics imply better semantic coherence for topics. Accordingly, TSC can be defined as follows:

$$\textbf{TSC} = \frac{2}{km^2}\sum_{t=1}^{k}\sum_{(w_i,w_j)\in\Omega_t}\frac{\cos(e_{w_i}, e_{w_j})+1}{2}, \tag{17}$$

where Ω_t is the top-m words of the t^{th} topic; and e_{w_i} and e_{w_j} denote the pre-trained word embeddings of w_i and w_j, respectively.

[3] https://news.google.com/
[4] https://answers.yahoo.com
[5] https://nltk.org
[6] https://github.com/blei-lab/lda-c
[7] https://github.com/jackyin12/GSDMM
[8] https://github.com/xiaohuiyan/BTM
[9] https://github.com/NobodyWHU/GPUDMM
[10] https://github.com/ysmiao/nvdm
[11] https://github.com/akashgit/autoencoding_vi_for_topic_models
[12] https://github.com/valdersoul/GraphBTM
[13] https://github.com/BobXWu/NQTM
[14] https://nlp.stanford.edu/projects/glove/
[15] https://github.com/dice-group/Palmetto

Table 3: Results (mean ± std) of TC. The best scores are displayed in boldface.

Model	Trec		GoogleNews		YahooAnswer		AvgRank
	$k = 20$	$k = 50$	$k = 20$	$k = 50$	$k = 20$	$k = 50$	
LDA	.368 ± .001	.361 ± .002	.382 ± .004	.381 ± .002	.377 ± .004	.377 ± .003	6.50
DMM	.372 ± .006	.379 ± .006	.408 ± .003	.393 ± .004	.383 ± .007	.373 ± .004	4.00
BTM	.365 ± .010	.370 ± .001	.395 ± .004	.385 ± .002	.367 ± .007	.367 ± .008	7.00
GPUDMM	.377 ± .005	.379 ± .005	.403 ± .007	.391 ± .003	.374 ± .005	.375 ± .003	4.50
NVDM	.343 ± .006	.349 ± .006	.390 ± .008	.395 ± .006	.383 ± .010	.377 ± .008	5.67
ProdLDA	.372 ± .022	.376 ± .005	.369 ± .010	.372 ± .007	.396 ± .014	.391 ± .013	5.17
GraphBTM	.354 ± .001	.340 ± .001	.359 ± .001	.383 ± .001	.384 ± .001	.371 ± .001	7.50
NQTM	.397 ± .006	.379 ± .007	.416 ± .004	.391 ± .004	.404 ± .015	**.397 ± .006**	2.17
DWGTM	**.402 ± .006**	**.392 ± .007**	**.419 ± .008**	**.403 ± .002**	**.406 ± .001**	.389 ± .015	1.33

Table 4: Results (mean ± std) of TSC. The best scores are displayed in boldface.

Model	Trec		GoogleNews		YahooAnswer		AvgRank
	$k = 20$	$k = 50$	$k = 20$	$k = 50$	$k = 20$	$k = 50$	
LDA	.396 ± .003	.393 ± .003	.386 ± .004	.398 ± .005	.485 ± .003	.456 ± .002	6.83
DMM	.442 ± .002	**.421 ± .004**	.406 ± .004	.413 ± .002	.509 ± .004	.491 ± .003	3.00
BTM	.418 ± .004	.414 ± .005	.407 ± .006	.415 ± .004	**.522 ± .008**	**.513 ± .004**	2.83
GPUDMM	.441 ± .007	.420 ± .003	.396 ± .006	.415 ± .003	.513 ± .004	.499 ± .003	3.00
NVDM	.400 ± .005	.400 ± .003	.387 ± .006	.382 ± .004	.460 ± .011	.449 ± .004	6.83
ProdLDA	.349 ± .004	.348 ± .003	.408 ± .010	**.423 ± .005**	.370 ± .015	.374 ± .017	6.17
GraphBTM	.348 ± .001	.356 ± .001	.363 ± .001	.370 ± .001	.399 ± .001	.370 ± .001	8.67
NQTM	.428 ± .012	.417 ± .003	.401 ± .004	.411 ± .002	.499 ± .011	.479 ± .006	4.50
DWGTM	**.453 ± .007**	.418 ± .005	**.412 ± .005**	.406 ± .004	.516 ± .007	.479 ± .040	2.83

Specially, we describe several details of metrics. (1) For both metrics, higher scores indicate better performance. (2) We fix m to 10 in all evaluations. (3) For fair comparisons, we employ the pre-trained word2vec[16] embeddings to compute TSC, instead of the GloVe embeddings that have been used in some of comparing models.

4.1 Topic Quality Results

We independently run each comparing model 5 times, then report the average scores of TC and TSC in Tables 3 and 4. In terms of TC, it can be clearly seen that our DWGTM can achieve higher scores than baseline models in most cases. First, DWGTM outperforms the neural competitors GraphBTM and NQTM, which also focus on handling short texts. Second, the TC scores of DWGTM are higher than those conventional topic models in all settings, where the results demonstrate the GCN WCG-Encoder can better capture the word co-occurrence information from the corpora. In terms of TSC, our DWGTM gets competitive scores, and ranks the first averagely. Comparing with neural topic models, DWGTM can achieve higher scores in most cases, where more importantly it beats the most art NQTM. Surprisingly, conventional short text topic models, *e.g.,* DMM, BTM, and GPUDMM, can achieve competitive TSC scores with DWGTM, and even perform better than NQTM. The possible reason is that those shallow models capture similar semantic information to word2vec, *i.e.,* the word embeddings used to compute TSC scores in the experiments. Specially, we kindly indicate that a potential problem of DWGTM is the TSC degradation with more topics compared to conventional topic models. We will further investigate this problem.

4.2 Topic Visualization

For qualitative evaluations, we show the top-10 words of two selected topics about *politics* and *credit* across *YahooAnswer*. As presented in Table 5, we can observe that DWGTM can effectively learn informative word patterns from corpora, where the top topical words are exactly associated with *politics* and *credit*, being consistent in the judgment of human-beings to some extent. In contrast to baseline models, the topics learned by DWGTM seem more coherent, where some of baselines often generate several less informative words, *e.g.,* for the topic of *credit*, {"best", "bad", "long"} in LDA, {"old", "weight", "stomach"} in NVDM, and {"salt", "water", "ice"} in NQTM for the topic of *credit*. Besides, we also observe that the top-10 words of GPUDMM and DWGTM contain semantically related words. This implies

[16] https://wikipedia2vec.github.io/wikipedia2vec/pretrained/

Table 5: Visualization of the top-10 words of two example topics learned by comparing models

Model	Top-10 word list
LDA	many bush people president ok war come world end times best way take really long get credit good place bad
DMM	bush think president war people iraq us world george like get need find school job know credit go money want
BTM	bush people president think us war states united america george find get need credit know good money free online help
GPUDMM	bush think president war people world iraq george **americans american** get credit find much money need business pay home company
NVDM	united states president bush school want high really know george rid get old back mean weight credit yr stomach loan
ProdLDA	president marriage movie iraq every form get need remember cancer uick money development wont wants longer sports care base treat
GraphBTM	immigration music bike republicans three operation step named income god illegal install stand female affect true turn accept choice pay
NQTM	president united states george bush god war clinton iraq nuclear credit card debt salt bank water ice loan green interview
DWGTM	iraq bush **democrats** clinton george president america war us **democracy** money **debt pay** credit **tax** company **taxes loan** income **paying**

applying pre-trained word embeddings can effectively capture semantics for topic modeling.

4.3 Ablative Study

We conduct an ablative study to evaluate whether the two reconstruction losses of the DualWG-Decoder and also the entropy regularization of θ (*i.e.,* Eq.15) have positive effects on topic extraction. To achieve this, we examine three simplified versions that independently remove the loss of $\mathcal{G}^c$ ($\lambda_1 = 0$), the loss of $\mathcal{G}^s$ ($\lambda_2 = 0$), and the entropy regularization term ($\lambda_3 = 0$).

We show the topic quality results of different versions of DWGTM on *Trec* when $k = 20$. As shown in Table 6, we can observe that the full DWGTM method outperforms all three simplified versions, indicating that all three components have positive effects on topic extraction. Specifically, DWGTM w/o the losses of $\mathcal{G}^c$ and $\mathcal{G}^s$ (*i.e.,* $\lambda_1 = 0$ and $\lambda_2 = 0$) lead to TSC deficiency over 0.01, indicating that the two reconstruction processes in the DualWG-Decoder can help capturing the semantic information of words. Besides, the gain of DWGTM over the version without the entropy regularization (*i.e.,* $\lambda_3 = 0$) shows more significant validity. This coincides with the fact that the entropy regularization tends to compute peakier topic proportions, which are beneficial for extracting topics from short texts with extremely limited words.

Specially, we have evaluated different values of $\{\lambda_1, \lambda_2, \lambda_3\}$ and also the threshold γ^* from the range $\{0.1, 0.2, \cdots, 1\}$ in the early experiments. The results show that $\{\lambda_1, \lambda_2\}$ and λ_3 perform better with smaller and larger values, respectively; and γ^* performs relatively stable with different values. Due to the space limitation, we omit the detailed results and will show them in the next version.

Table 6: Results of the ablative study.

Metric	DWGTM	$\lambda_1 = 0$	$\lambda_2 = 0$	$\lambda_3 = 0$
TC	0.402	0.397	0.395	0.393
TSC	0.453	0.439	0.440	0.447

5 Conclusion

In this paper, we develop a novel neural topic model for short texts, called DWGTM. The proposed DWGTM model extracts topics by simultaneously applying the word co-occurrence graph and word semantic correlation graph. Specifically, it consists of four main components: (1) Encode the word co-occurrence graph as word features. (2) Generate text features with word features, and encode them as topic proportions. (3) Reconstruct the texts with topical distributions. (4) Reconstruct both graphs with word features. We also propose a novel metric to evaluate the semantic coherence of topics, called TSC. Empirically, the effectiveness of DWGTM was validated on three benchmark datasets of short texts. We show that the topics learned by DWGTM can simultaneously capture meaningful patterns and semantic correlations of words.

Acknowledgements

The research is supported by the National Natural Science Foundation of China (NSFC) (No.61876071) and Scientific and Technological Developing Scheme of Jilin Province (No.20180201003SF, No.20190701031GH) and Energy Administration of Jilin Province (No.3D516L921421) and Key Program of Science and Technology Research during the 13th Five-Year Plan Period, the Educational Department of Jilin Province of China (Grant No. JJKH20200677KJ).

References

David M Blei. 2012. Probabilistic topic models. *Communications of The ACM*, 55(4):77–84.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Sophie Burkhardt and Stefan Kramer. 2019. Decoupling sparsity and smoothness in the Dirichlet variational autoencoder topic model. *Journal of Machine Learning Research*, 20(131):1–27.

Xueqi Cheng, Xiaohui Yan, Yanyan Lan, and Jiafeng Guo. 2014. BTM: topic modeling over short texts. *IEEE Transactions on Knowledge and Data Engineering*, 26(12):2928–2941.

Adji B Dieng, Francisco J R Ruiz, and David M Blei. 2020. Topic modeling in embedding spaces. *Transactions of the Association for Computational Linguistics*, 8:439–453.

Thomas L Griffiths and Mark Steyvers. 2004. Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America*, 101(suppl 1):5228–5235.

Lin Gui, Jia Leng, Gabriele Pergola, Yu Zhou, Ruifeng Xu, and Yulan He. 2019. Neural topic model with reinforcement learning. In *Empirical Methods in Natural Language Processing*, pages 3478–3483.

Pankaj Gupta, Yatin Chaudhary, Thomas Runkler, and Hinrich Schuetze. 2020. Neural topic modeling with continual lifelong learning. In *International Conference on Machine Learning*, pages 3907–3917. PMLR.

Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. 1999. An introduction to variational methods for graphical models. *Machine Learning*, 37(2):105–161.

Diederik P Kingma and Max Welling. 2014. Auto-encoding variational Bayes. In *International Conference on Learning Representations*.

Thomas N. Kipf and Max Welling. 2016a. Variational graph auto-encoders. In *NIPS Workshop on Bayesian Deep Learning*.

Thomas N. Kipf and Max Welling. 2016b. Variational graph auto-encoders. In *NIPS Workshop on Bayesian Deep Learning*.

Changchun Li, Ximing Li, Jihong Ouyang, and Yiming Wang. 2020a. Semantics-assisted Wasserstein learning for topic and word embeddings. In *IEEE International Conference on Data Mining*, pages 292–301.

Chenliang Li, Haoran Wang, Zhiqian Zhang, Aixin Sun, and Zongyang Ma. 2016. Topic modeling for short texts with auxiliary word embeddings. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 165–174.

Shuangyin Li, Yu Zhang, and Rong Pan. 2020b. Bidirectional recurrent attentional topic model. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 14(6):1–30.

Ximing Li, Changchun Li, Jinjin Chi, and Jihong Ouyang. 2018. Short text topic modeling by exploring original documents. *Knowledge and Information Systems*, 56(2):443–462.

Ximing Li, Ang Zhang, Changchun Li, Lantian Guo, Wenting Wang, and Jihong Ouyang. 2019a. Relational biterm topic model: Short text topic modeling using word embeddings. *The Computer Journal*, 62(3):359–372.

Ximing Li, Jiaojiao Zhang, and Jihong Ouyang. 2019b. Dirichlet multinomial mixture with variational manifold regularization: topic modeling over short texts. In *AAAI Conference on of Artificial Intelligence*, pages 7884–7891.

Chenghua Lin and Yulan He. 2009. Joint sentiment/ topic model for sentiment analysis. In *ACM international Conference on Information and Knowledge Management*, pages 375–384.

Zongyang Ma, Aixin Sun, Quan Yuan, and Gao Cong. 2012. Topic-driven reader comments summarization. In *ACM international Conference on Information and Knowledge Management*, pages 265–274.

Yishu Miao, Lei Yu, and Phil Blunsom. 2016. Neural variational inference for text processing. In *International Conference on Machine Learning*, pages 1727–1736.

Tomas Mikolov, Kai Chen, Greg S Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *International Conference on Learning Representations*.

Andriy Mnih and Karol Gregor. 2014. Neural variational inference and learning in belief networks. In *International Conference on Machine Learning*, pages 1791–1799.

Kamal Nigam, Andrew Kachites McCallum, Sebastian Thrun, and Tom Mitchell. 2000. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2–3):103–134.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. GloVe: global vectors for word representation. In *Empirical Methods in Natural Language Processing*, pages 1532–1543.

Xiaojun Quan, Chunyu Kit, Yong Ge, and Sinno Jialin Pan. 2015. Short and sparse text topic modeling via self-aggregatio. In *International Joint Conference on Artificial Intelligence*, pages 2270–2276.

Mehdi Rezaee and Francis Ferraro. 2020. A discrete variational recurrent topic model without the reparametrization trick. *arXiv preprint arXiv:2010.12055*.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. 2014. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, pages 1278–1286.

Tian Shi, Kyeongpil Kang, Jaegul Choo, and Chandan K. Reddy. 2018. Short-text topic modeling via non-negative matrix factorization enriched with local word-context correlations. In *The Web Conference*, pages 1105–1114.

Akash Srivastava and Charles A. Sutton. 2017. Autoencoding variational inference for topic models. In *International Conference on Learning Representations*.

Rui Wang, Deyu Zhou, and Yulan He. 2019. Atm: Adversarial-neural topic model. *Information Processing & Management*, 56(6):102098.

Xiaobao Wu, Chunping Li, Yan Zhu, and Yishu Miao. 2020. Short text topic modeling with topic distribution quantization and negative sampling decoder. In *Empirical Methods in Natural Language Processing*, pages 1772–1882.

Xiaohui Yan, Jiafeng Guo, Yanyan Lan, and Xueqi Cheng. 2013. A biterm topic model for short text. In *International Conference on World Wide Web*, pages 1445–1456.

Liang Yang, Fan Wu, Junhua Gu, Chuan Wang, Xiaochun Cao, Di Jin, and Yuanfang Guo. 2020. Graph attention topic modeling network. In *Proceedings of The Web Conference 2020*, pages 144–154.

Jianhua Yin and Jianyong Wang. 2014. A Dirichlet multinomial mixture model-based approach for short text clustering. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 233–242.

Jichuan Zeng, Jing Li, Yan Song, Cuiyun Gao, Michael R Lyu, and Irwin King. 2018. Topic memory networks for short text classification. In *Empirical Methods in Natural Language Processing*, pages 3120–3131.

Qile Zhu, Zheng Feng, and Xiaolin Li. 2018. GraphBTM: Graph enhanced autoencoded variational inference for biterm topic model. In *Empirical Methods in Natural Language Processing*, pages 4663–4672.

Yuan Zuo, Junjie Wu, Hui Zhang, Hao Lin, Fei Wang, Ke Xu, and Hui Xiong. 2016a. Topic modeling of short texts: A pseudo-document view. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Minin*, pages 2105–2114.

Yuan Zuo, Jichang Zhao, and Ke Xu. 2016b. Word network topic model: a simple but general solution for short and imbalanced texts. *Knowledge and Information Systems*, 48(2):379–398.