Combating Label Sparsity in Short Text Topic Modeling via Nearest Neighbor Augmentation

Anonymous ACL submission

Abstract

Extracting semantic topics from short texts presents a significant challenge in the field of data mining. While efforts have been made to mitigate data sparsity issue, the limited length of short documents also results in the absence of semantically relevant words, causing biased evidence lower bound and incomplete labels for likelihood maximization. We refer to this issue as the *label sparsity* problem. To combat this problem, we propose kNNTM, a neural short text topic model that incorporates a k-Nearest-Neighbor-based label completion algorithm by augmenting the reconstruction label with k nearest documents to complement these relevant but unobserved words. Furthermore, seeking a precise reflection of distances between documents, we propose a fused multi-view distances metric that takes both local word similarities and global topic semantics into consideration. Extensive experiments on multiple public short-text datasets show that kNNTM model outperforms the state-of-the-art baseline models and can derive both high-quality topics and document representations.

1 Introduction

Depiste the success of topic models in numerous NLP tasks (Boyd-Graber et al., 2017) for uncovering the underlying semantic concepts (Blei et al., 2003), traditional topic models often suffer from poor performances when applied to short text contents, e.g., social media posts and news headlines (Yan et al., 2013). This deficiency can be attributed to the lack of word co-occurrence information due to the limited length for a single short document, known as the *data sparsity* problem (Murshed et al., 2022).

Many topic models have been developed to overcome the data sparsity issue. The Dirichlet Multinomial Mixture (DMM) model (Yin and Wang, 2014; Li et al., 2016, 2017) constraints that each short text is generated by a single topic. Biterm Topic Model (Yan et al., 2013; Cheng et al., 2014) utilizes the rich corpus-level word co-occurrence patterns for inferring topics. And some works (Mehrotra et al., 2013; Quan et al., 2015; Zuo et al., 2016) aggregate semantically similar texts into long pseudo-documents. Recently, with the developments of neural topic models (NTMs) (Srivastava and Sutton, 2017), there are also attempts to mitigate the data sparsity issue by utilizing biterm graph (Zhu et al., 2018) and topic quantization techniques (Wu et al., 2020, 2022).

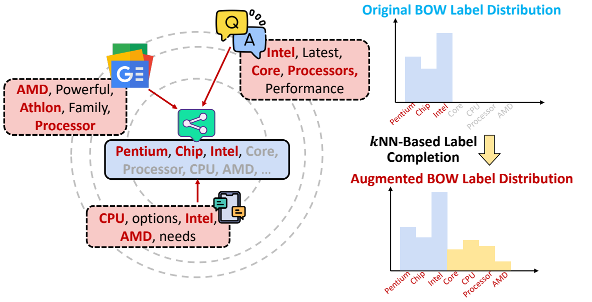
AMD, Powerful, Athlon, Family, Processor
Intel, Latest, Core, Processors, Performance
Pentium, Chip, Intel, Core, Processor, CPU, AMD, ...
CPU, options, Intel, AMD, needs
Original BOW Label Distribution
Pentium
Chip
Intel
Core
CPU
Processor
AMD
*k*NN-Based Label Completion
Augmented BOW Label Distribution
Pentium
Chip
Intel
Core
CPU
Processor
AMD

Figure 1: A motivating example of label sparsity issue in short text topic modeling and kNN-based label completion of the unobserved words.

Though these above works have achieved good performances and mitigated the data sparsity issue to some extent, there are still problems that they ignore. Under the variational autoencoding framework, current mainstream NTMs are optimized by the maximum likelihood objective, which is achieved by maximizing the evidence lower bound (ELBO). However, the limited length of short texts results that only a few words get described in a document, while many other semantically relevant words remain unobserved (Zhang and Lauw, 2022). As shown in the motivating example in Figure 1, the document in the center talks about CPU chips and contains words like 'chip' and 'intel'. However, many semantically related words like 'core', 'processor', and 'cpu' remain uncovered. Such 'incomplete' short documents will lead to a biased evidence lower bound as the possibilities of those unobserved but relevant words are completely ignored, resulting in a sub-optima optimization of the maximum likelihood. To be more specific, the absence of these relevant words leads to an incomplete target for the reconstruction objective during variational autoencoding, which makes the probabilities of the absent words get inappropriately suppressed and results in biased training signals. This problem, different from the data sparsity problem of the input data, is referred to as the ***label sparsity*** problem in this paper.

Inspired by the above observation, we propose to explicitly **augment the reconstruction target** in short text NTM with semantically related words to provide unbiased training signals. One direct approach to derive these words is to leverage the similarities between pre-trained word embeddings. However, word embeddings trained with general corpora may not capture the word co-occurrence patterns from a specific domain. Moreover, simply relying on word similarities ignores the context information on the document level. In this paper, we propose *kNNTM*, a short text topic modeling framework, which incorporates a **k-Nearest-Neighbor-based label completion algorithm** by aggregating k documents semantically closest to the target document to augment its reconstruction label. As illustrated in Figure 1, documents with semantically relevant words are retrieved by k-nearest neighbor searching, and the label distribution gets augmented by complementing the probabilities of originally unobserved words. The kNN approach is shown to be effective in multiple fields for information supplementation and data completion, like machine translation (Khandelwal et al., 2021), healthcare prediction (Zhang et al., 2021), and computer vision (Yu et al., 2021). In our scenario of short text topic modeling, aggregating k nearest documents for label completion helps to make full use of the word co-occurrence information and document relations from the original dataset.

However, one remaining challenge of the kNN-based label completion is to seek a proper distance metric that could precisely reflect the similarities between short documents with scarce context. A good metric should reflect both the word-level similarities and the global semantic resemblance. Therefore, we propose a multi-view distance metric by fusing the distances from the input space and the semantic space to leverage both local and global similarities information. The distance metric in the input space depicts the local word similarity, which is defined with the optimal transport distance between bag-of-words distributions, with cost functions built upon word similarities from both general corpora and the specific dataset. And the metric in the hidden space reflects the global semantic resemblance, which is defined through the lens of topic semantics with the similarities between document-topic distributions. With the fused multi-view distance metric, we can take various factors into account when evaluating the distances between documents, and provide a reasonable distance metric for the kNN algorithm.

Our contributions are summarized as follows:

- We identify the label sparsity problem in short text neural topic modeling, and propose a novel topic modeling framework, kNNTM, to combat this issue by a kNN-based label completion algorithm with similar document aggregation.
- We propose a fused multi-view distance metric that takes both global and local semantic similarities into consideration to support the kNN label completion algorithm.
- Extensive experiments are conducted on three short text datasets, and both quantitative and qualitative results demonstrate that kNNTM outperforms state-of-the-art baselines, and could generate high-quality topics and meaningful document representations.

2 Related Works

Neural Topic Modeling With the recent developments of neural variational inference (Kingma and Welling, 2014; Rezende et al., 2014), many neural topic models (NTMs) are proposed for higher scalability and easier inference. NVDM (Miao et al., 2016) and ProdLDA (Srivastava and Sutton, 2017) are two representative works, which leverage Gaussian and logistic normal distribution as approximations of the Dirichlet prior. And many subsequent NTMs have been investigated. Some focus on improving the encoder network, e.g., recurrent networks (Rezaee and Ferraro, 2020), graph neural networks (Yang et al., 2020; Xie et al., 2021). Some works aim for a better approximation of the Dirichlet prior, e.g., Wasserstein autoencoders (Nan et al., 2019), reject sampling (Burkhardt and Kramer, 2019), and Weibull distribution (Zhang et al., 2018). Some works attempt to find new training paradigms, e.g., adversarial training (Wang et al., 2019, 2020; Hu et al., 2020), and optimal transport (Zhao et al., 2021; Wang et al., 2022). The most recent work of NTM to the best of our knowledge is ECRTM (Wu et al., 2023), which incorporates an embedding clustering regularization on the topic and word embeddings. Despite their success in modeling topics on normal long texts, these works still suffer from the sparsity issue of short texts.

Topic Models for Short Text Conventional short text topic models can be mainly categorized into three classes. The Dirichlet Multinomial Mixture (DMM) models (Yin and Wang, 2014; Li et al., 2016, 2017) assume that each short text is generated by a single topic, thus reducing the complexity for topic inference. Biterm Topic Model (BTM) (Yan et al., 2013; Cheng et al., 2014) utilizes the rich corpus-level word co-occurrence patterns and splits the entire dataset into numerous biterms. Self-aggregation models (Mehrotra et al., 2013; Quan et al., 2015; Zuo et al., 2016) tend to aggregate semantically similar short text into long pseudo-documents to infer topics.

Another line of research focuses on neural topic modeling for short texts. GraphBTM (Zhu et al., 2018) generalizes the BTM model and performs variation autoencoding on biterm graph from randomly-sampled mini-corpus. NQTM (Wu et al., 2020) shares similar insights with DMM and quantizes document-topic distributions to obtain peakier distributions. And the TSCTM model (Wu et al., 2022) further improves upon NQTM by introducing a contrastive loss on quantized distributions. MCTM (Zhang and Lauw, 2022) focuses on short texts in the variable-length corpus and learns a semantics predictor based on long documents within the corpus. However, current neural topic models for short text mainly focus on mitigating the data sparsity problem from the input side, yet ignore the insufficient training signals for the reconstruction objective, namely the label sparsity issue, brought by the limited length of a single document.

3 Methodology

3.1 Problem Formulation

Consider a corpus $\mathcal{D}$ with N_D documents, where each document d contains N_d words $\{x_{d_1}, \ldots, x_{d_{N_d}}\}$ belonging to a vocabulary of size V. The target is to discover K topics from the corpus. Each topic is defined as a distribution $\beta_k \in \Delta^V$ over the words in the vocabulary, namely the topic-word distribution. Meanwhile, for each input document, the model should also infer a distribution over the topics, i.e., the document-topic distribution, denoted as $\theta \in \Delta^K$.

3.2 Model Architecture

We choose the Quantization Topic Model (QTM) (Wu et al., 2020, 2022), as the basic model architecture for kNNTM. QTM is a VAE-based neural topic model that quantizes topic representations for peakier topic distributions. Here we briefly introduce the model architecture. For more detailed implementations, please refer to the original paper (Wu et al., 2020).

3.2.1 Text encoder

The text encoder takes document d in the form of bag-of-words as the input x^d, and produces corresponding hidden topic representation $h^d \in \mathbb{R}^K$, and the topic representation is further normalized into a probability simplex to obtain the document-topic distribution $\theta^d \in \Delta^K$ by a softmax function $\theta^d = \text{softmax}\left(h^d\right)$.

3.2.2 Topic Quantization

The document-topic distributions is further quantized to alleviate the data sparsity problem, The quantized distribution is defined as

$$\theta_q^d = \boldsymbol{e}_k, \text{ where } k = \text{argmin}_j \left\|\theta^d - \boldsymbol{e}_j\right\|_2, \tag{1}$$

where $\boldsymbol{e} = (\boldsymbol{e}_1, \boldsymbol{e}_2, ..., \boldsymbol{e}_K) \in \mathbb{R}^{K \times K}$ are K preset quantization prototypes. These prototypes are initialized as different one-hot vectors and get optimized during training.

3.2.3 Decoder and Objective Function

The decoder network consists of topic-word distributions $\boldsymbol{\beta}$, and tries to reconstruct the observed texts with the quantized distributions Let x^d denote the bag-of-words form of a document d, then the reconstruction objective for topic models is

$$\mathcal{L}_{\text{RECON}}\left(x^d\right) = -x^{d^\top} \log\left(\text{softmax}\left(\boldsymbol{\beta}\theta_q^d\right)\right). \tag{2}$$

Besides the reconstruction loss, the QTM leverages a regularizer constraining the distances between original and quantized distributions,

$$\mathcal{L}_{\text{REG}}(\theta^d) = \left\|\text{sg}(\theta^d) - \theta_q^d\right\|_2 + \lambda \left\|\text{sg}(\theta_q^d) - \theta^d\right\|_2, \tag{3}$$

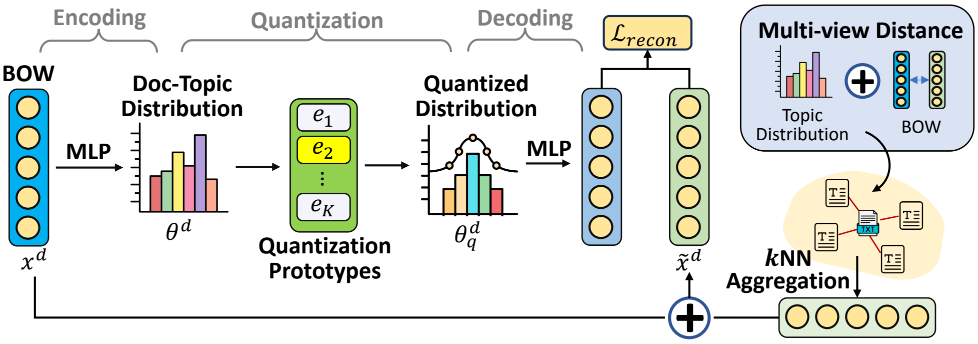

Figure 2: The overall structure of kNNTM. The left side is the model architecture including encoding, quantization, and decoding. The right side is the kNN-based label completion module with a fused multi-view distance metric.

where $\text{sg}(\cdot)$ is a stop gradient operation, and λ is set to 0.1 following (Wu et al., 2020, 2022). The final objective function for the model is

$$\mathcal{L}_{\text{TM}}(x^d) = \mathcal{L}_{\text{RECON}}(x^d) + \mathcal{L}_{\text{REG}}(\theta^d) \tag{4}$$

3.3 kNN-based Label Completion

The core difference between kNNTM and other neural topic models is the kNN-based label completion module. As discussed in the Introduction and Figure 1, short text modeling faces the label sparsity problem. The limited length of short texts makes some semantically related words unobserved in the short document. The probabilities of these words are ignored and lead to a biased evidence lower bound during optimization. When optimizing the neural topic model with $\mathcal{L}_{\text{TM}}$, the probabilities of the observed words get encouraged, while the probabilities of those unobserved but relevant words get discouraged as the predicted probability vector get normalized by the softmax function, leading to biased training signal and suboptimal optimization. Therefore, a label completion module is needed to derive the hidden relevant words and construct an unbiased label for the reconstruction objective.

Motivated by the above thoughts, we propose the kNN-based label completion algorithm. Given a document x^d, we find its n_k nearest neighbors with a distance metric, $\text{dist}(\cdot,\cdot)$ (we leave the design of the metric to the next section). The set of n_k nearest neighbors is denoted as $\mathcal{N}_{x^d}$, and the reconstruction label is augmented with a coefficient α as

$$\tilde{x}^d = x^d + \alpha * \frac{1}{n_k} \sum_{x_j \in \mathcal{N}_{x^d}} x_j. \tag{5}$$

3.4 Fused Multi-View Distance Metric

To perform an effective kNN algorithm, a reliable distance metric is required to precisely measure the similarities between short documents. Here we propose a fused multi-view distance, which fuses distances from both the input BoW space and hidden topical semantic space and takes information from both local word-level relations and global topic-level similarities.

3.4.1 Distance from the Input Space

As the original form of the input document, the bag-of-word vectors naturally contain the information for evaluating the distances between documents. However, directly comparing the bag-of-word vectors would result in a bad distance metric, as the hidden semantic relations between words are not explored. Documents with different but highly relevant word sets would be considered dissimilar without considering the hidden relation between words. Therefore, we propose to evaluate the distance between two bag-of-words vectors with a word semantic-based optimal transport (OT) distance. Firstly we introduce the OT distance between two probability vectors $\boldsymbol{a} \in \Delta^{D^a}$ and $\boldsymbol{b} \in \Delta^{D^b}$, which is defined as:

$$\text{dist}_{\mathbf{M}}^{OT}(\boldsymbol{a}, \boldsymbol{b}) := \min_{\gamma \in U(\boldsymbol{a},\boldsymbol{b})} \langle \gamma, \mathbf{M} \rangle, \tag{6}$$

where $\langle \cdot, \cdot \rangle$ denotes the Frobenius dot-product, $U(\boldsymbol{a}, \boldsymbol{b})$ denotes the transport polytope of $\boldsymbol{a}$ and $\boldsymbol{b}$, $U(\boldsymbol{a}, \boldsymbol{b}) := \{\gamma \in \mathbb{R}_+^{D^a \times D^b} \mid \gamma \mathbf{1} = \boldsymbol{a}, \gamma^\top \mathbf{1} = \boldsymbol{b}\}$, and $\mathbf{M} \in \mathbb{R}_{\geq 0}^{D^a \times D^b}$ is a cost matrix indicating the transportation cost between probability vectors.

Therefore, with an appropriate cost matrix $\mathbf{M} \in \mathbb{R}_{\geq 0}^{V \times V}$ depicting the semantic similarities between words, the OT distances will become a suitable distance metric for gauging the distance between two bag-of-words vectors, namely $\text{dist}_{\mathbf{M}}^{OT}(\hat{x}^{d_i}, \hat{x}^{d_j})$, where $\hat{x}^d$ is the normalized bag-of-words vector.

We propose two perspectives to build cost matrix $\mathbf{M}$. The first aspect is to leverage the word embeddings pre-trained with general corpora. The cosine similarities between pre-trained word embeddings have been proven to be highly effective in reflecting the semantic similarities between words. The cost matrix is built as

$$\mathbf{M}^{cos}_{i,j} = (1.0 - s_{\cos}(w_i, w_j)) * 0.5, \tag{7}$$

where $s_{\cos}(\cdot)$ is the cosine similarity function, and w_i is the i-th word in the vocabulary.

Though effective, word embeddings pre-trained with general corpora may not be able to capture the co-occurrence patterns in a specific domain. Hence, we propose another way to build the cost matrix $\mathbf{M}$ with the word co-occurrences of the current corpus. Concretely, the cost matrix is built as

$$\mathbf{M}^{coo}_{i,j} = 1.0 - (p(w_i|w_j) + p(w_j|w_i))/2, \tag{8}$$

where $p(w_i|w_j)$ is the conditional probability of word w_i given w_j, and is calculated as $p(w_i|w_j) = df(w_i, w_j)/df(w_j)$, where $df(w_i, w_j)$ is the frequency that the word w_i and w_j co-occur.

Finally, to leverage both the rich information from general corpora and specific patterns from the current dataset, we fuse the two OT distances to formulate the distance from input space as:

$$\begin{aligned}\text{dist}^{\text{BoW}}(x^{d_i}, x^{d_j}) = \rho * \text{dist}^{OT}_{\mathbf{M}^{cos}}(\hat{x}^{d_i}, \hat{x}^{d_j}) \\ + (1.0 - \rho) * \text{dist}^{OT}_{\mathbf{M}^{coo}}(\hat{x}^{d_i}, \hat{x}^{d_j}).\end{aligned} \tag{9}$$

where ρ is a fusing hyper-parameter.

3.4.2 Distance from the Semantic Space

Besides the bag-of-words vector, the topic distribution for each document is also an effective semantic representation of documents. The metric defined on the bag-of-words vectors mainly utilizes **local** word-level similarities, whereas the metric defined on the topic distributions can depict semantic distances between documents from a **global** view, since the topics reveal the hidden semantic structures of the entire corpus. Incorporating the metric defined on topic distributions would encourage topically similar documents to fall into the neighborhood of the target document.

However, one important problem is that the document-topic distributions keep evolving during training and cannot be pre-computed beforehand. It is also impractical to go through the entire dataset to compute the distributions whenever the nearest neighbors are needed due to excessively high time costs. Therefore, we maintain a memory bank $\{m_1, ...m_{N_D}\}$, to store the most recent document-topic distributions. Entries within the memory bank get updated every time the document-topic distribution are computed during training,

$$m_d = \theta^d_t, \tag{10}$$

where θ^d_t is the document-topic distribution for document d computed in the t-th iteration during training. And the distance from the topical semantic space is defined as

$$\text{dist}^{\text{Topic}}(x^{d_i}, x^{d_j}) = \|\theta^{d^i} - m^{d_j}\|_2. \tag{11}$$

3.4.3 Multi-View Distance Fusion

To get a balanced distance metric considering both similarities in the BoW and the topic spaces, we fuse the above two distances with a hyper-parameter η as

$$\begin{aligned}\text{dist}^{\text{Fuse}}(x^{d_i}, x^{d_j}) = \eta * \text{dist}^{\text{Topic}}(x^{d_i}, x^{d_j}) \\ + (1.0 - \eta) * \text{dist}^{\text{BoW}}(x^{d_i}, x^{d_j}).\end{aligned} \tag{12}$$

3.5 Training Procedure and Objective

To stabilize the training process, we adopt a two-phase training strategy. We first pre-train the topic model with the objective in Eq.4 without label augmentation for P epochs to get more accurate document-topic distributions for distance calculation. After pre-training, we use the augmented label from Eq.5 for the reconstruction loss,

$$\mathcal{L}_{\text{kNN}}(x^d) = -\tilde{x}^{d^\top} \log\left(\text{softmax}(\boldsymbol{\beta}\theta^d_q)\right). \tag{13}$$

and the final training objective is

$$\mathcal{L}_{\text{kNNTM}}(x^d) = \mathcal{L}_{\text{kNN}}(x^d) + \mathcal{L}_{\text{REG}}(\theta^d). \tag{14}$$

The overall structure of kNNTM is shown in Fig.2, and we provide the detailed training procedure in the Algorithm 1 in Appendix A.

4 Experiment Settings

4.1 Datasets

In the experiments, we use three public benchmark short text datasets: 1) **GoogleNews** with titles of over 10,000 news articles categorized into 152 clusters, 2) **Snippet** consisting of over 10,000 web search results across 8 domains, 3) **StackOverflow** with 20,000 question titles from 20 different tags.

We utilized the aforementioned datasets provided by the STTM library[1] (Qiang et al., 2020). Additionally, we further filter out words with a frequency below 3 and documents with a length less than 2. Please refer to Appendix B.1 for the detailed statistics for each dataset after preprocessing.

[1] https://github.com/qiang2100/STTM

4.2 Baseline Methods

We compare our model with the following state-of-the-art baselines: **prodLDA** (Srivastava and Sutton, 2017), a prominent work of NTM with black-box neural variational inference; **WLDA** (Nan et al., 2019), a NTM with the Wasserstein autoencoder framework; **ECRTM** (Wu et al., 2023), a NTM with an topic embedding clustering regularization, which is the state-of-the-art NTM for normal long documents; **NQTM** (Wu et al., 2020), a neural short text topic model with topic distribution quantization and negative sampling; **MCTM** (Zhang and Lauw, 2022), a NTM that predicts missing semantics for short documents based on other long documents in variable-length corpora; **TSCTM** (Wu et al., 2022), a state-of-the-art short text neural topic model based on NQTM with a contrastive objective on quantized distributions.

4.3 Implementation Datails

We follow the settings for hyper-parameters shared with (Wu et al., 2022), including learning rate, batch size, epoch number, etc. And for hyper-parameters exclusive to our method, we conduct grid search to determine the optimal values. Please refer to Appendix B.3 for detailed settings.

5 Experimental Results

We evaluate the topic models from two perspectives: topic-word distribution and document-topic distribution. For the former, we assess the quality of topics based on coherence and diversity. Regarding the latter, we utilize the performances from the clustering task as previous studies (Zhao et al., 2021; Wang et al., 2022). To verify models' effectiveness under different topic numbers, following previous work (Wu et al., 2022), we conducted experiments under 50 and 100 topics, respectively.

5.1 Topic Quality

Metric Following previous work in topic modeling (Dieng et al., 2020; Wu et al., 2022), we evaluate the quality of learned topics from two perspectives, **Topic Coherence** and **Topic Diversity**.

For topic coherence, we adopt a widely-used coherence metric, C_V (Röder et al., 2015), which is shown to be better than other coherence metrics like UMASS (Mimno et al., 2011) and NPMI (Aletras and Stevenson, 2013) and have been adopted by many works in short text topic modeling (Wu et al., 2020; Wang et al., 2021; Wu et al., 2022). We use the well-adopted library *Palmetto*[2] to compute C_V with Wikipedia texts as the reference corpus. For topic diversity, we employ the Topic Uniqueness (TU) for evaluation (Nan et al., 2019; Dieng et al., 2020). which is defined as the proportion of unique words among all the topical words.

Moreover, as pointed out in (Wu et al., 2020), there exists a trade-off relation between the coherence and diversity metrics. Higher TU scores tend to cause lower C_V scores and vice versa. To provide a more comprehensive metric for topic quality, following previous work (Dieng et al., 2020), we adopt the Topic Quality (TQ) metric as the product of the topic diversity and coherence score,

$$TQ = C_V * TU. \tag{15}$$

We take the top 15 words with the highest probabilities of each topic for the aforementioned metrics following (Wu et al., 2022).

Results The results are shown in Table 1. From the results, we could find that our kNNTM model outperforms or achieves compatible performances with the start-of-the-art baselines, which proves the existence of the label sparsity problem and the effectiveness of our solution. We can find the kNNTM achieves high TU scores under many settings, which indicates that labels augmented by multiple documents bring more diverse information for topic optimization. Furthermore, in terms of the comprehensive evaluation metric, Topic Quality (TQ), kNNTM outperforms the baseline models in four distinct settings and attains comparable results in the remaining two scenarios. This underscores the capability of our model to effectively strike a balance between topic coherence and diversity, resulting in the extraction of high-quality topics that exhibit both coherence and diversity.

It is worth noting that MCTM model achieves the highest C_V scores under almost every setting, whereas its TU scores are notably diminished. This indicates that a set of coherent words frequently repeats across MCTM's topics. Therefore, in spite of some coherent topics being discovered, many of those are repetitive and uninformative, hence making its TU and TQ scores hardly comparable with other methods.

5.2 Text Clustering

Metric To evaluate the quality of document-topic distributions, we leverage the short text clustering

[2] https://github.com/dice-group/Palmetto

Model			prodLDA	WLDA	ECRTM	NQTM	MCTM	TSCTM	kNNTM
GoogleNews	$K = 50$	C_V	0.313±0.008	0.305±0.008	0.302±0.003	0.300±0.002	**0.361±0.011**	0.313±0.002	0.312±0.004
		TU	0.936±0.022	0.882±0.012	0.876±0.050	0.972±0.003	0.556±0.026	**0.996±0.003**	0.995±0.004
		TQ	0.294±0.013	0.269±0.008	0.265±0.016	0.312±0.004	0.201±0.004	**0.312±0.004**	0.311±0.002
		top-Purity	0.333±0.016	0.388±0.023	0.521±0.032	0.393±0.015	0.262±0.019	0.570±0.019	**0.581±0.008**
		top-NMI	0.372±0.007	0.621±0.013	0.773±0.017	0.634±0.008	0.486±0.010	0.773±0.013	**0.799±0.004**
	$K = 100$	C_V	0.322±0.006	0.308±0.007	0.304±0.005	0.302±0.003	**0.356±0.002**	0.302±0.006	0.305±0.007
		TU	0.786±0.019	0.662±0.012	0.923±0.036	0.943±0.018	0.614±0.081	0.971±0.003	**0.980±0.004**
		TQ	0.252±0.009	0.204±0.004	0.280±0.009	0.285±0.006	0.219±0.030	0.293±0.005	**0.299±0.006**
		top-Purity	0.366±0.007	0.484±0.011	0.333±0.083	0.567±0.030	0.180±0.049	0.766±0.007	**0.786±0.007**
		top-NMI	0.382±0.003	0.676±0.007	0.547±0.067	0.712±0.012	0.397±0.059	0.842±0.004	**0.868±0.002**
Snippet	$K = 50$	C_V	0.349±0.003	0.329±0.008	0.322±0.003	0.339±0.011	0.352±0.026	0.348±0.007	**0.366±0.004**
		TU	0.987±0.003	0.848±0.033	0.981±0.005	0.974±0.007	0.750±0.028	0.994±0.002	**0.998±0.002**
		TQ	0.345±0.003	0.279±0.004	0.316±0.004	0.331±0.012	0.264±0.019	0.346±0.006	**0.365±0.004**
		top-Purity	0.503±0.017	0.586±0.026	0.751±0.015	0.630±0.026	0.443±0.008	0.712±0.009	**0.762±0.007**
		top-NMI	0.172±0.008	0.273±0.010	**0.444±0.008**	0.302±0.016	0.177±0.015	0.381±0.010	0.427±0.006
	$K = 100$	C_V	0.327±0.007	0.326±0.006	0.320±0.011	0.309±0.006	**0.355±0.013**	0.329±0.004	0.341±0.006
		TU	0.950±0.001	0.669±0.006	**0.981±0.017**	0.928±0.005	0.581±0.045	0.948±0.003	0.979±0.002
		TQ	0.310±0.006	0.218±0.004	0.314±0.010	0.286±0.007	0.206±0.009	0.312±0.003	**0.334±0.006**
		top-Purity	0.477±0.005	0.635±0.005	0.392±0.092	0.682±0.005	0.421±0.026	0.759±0.009	**0.819±0.005**
		top-NMI	0.132±0.002	0.298±0.003	0.241±0.076	0.325±0.004	0.157±0.013	0.387±0.005	**0.436±0.001**
StackOverflow	$K = 50$	C_V	0.259±0.001	0.279±0.005	0.284±0.016	0.268±0.004	**0.320±0.003**	0.284±0.002	0.284±0.006
		TU	0.865±0.009	0.804±0.025	0.924±0.023	0.915±0.003	0.492±0.009	**0.952±0.004**	0.950±0.007
		TQ	0.224±0.002	0.224±0.003	0.262±0.009	0.245±0.003	0.158±0.004	**0.271±0.003**	0.269±0.006
		top-Purity	0.227±0.003	0.443±0.003	0.319±0.024	0.433±0.035	0.290±0.019	0.576±0.007	**0.607±0.010**
		top-NMI	0.074±0.002	0.296±0.001	0.258±0.014	0.298±0.029	0.280±0.013	0.423±0.007	**0.463±0.004**
	$K = 100$	C_V	0.253±0.002	0.283±0.008	0.266±0.002	0.276±0.004	**0.307±0.012**	0.273±0.005	0.264±0.002
		TU	0.672±0.011	0.615±0.033	0.801±0.033	0.795±0.007	0.586±0.003	0.808±0.012	**0.833±0.002**
		TQ	0.170±0.004	0.174±0.001	0.216±0.010	0.210±0.003	0.180±0.006	**0.220±0.007**	**0.220±0.002**
		top-Purity	0.167±0.005	0.406±0.008	0.099±0.002	0.467±0.038	0.281±0.008	0.571±0.010	**0.616±0.014**
		top-NMI	0.046±0.003	0.267±0.006	0.086±0.004	0.308±0.029	0.273±0.001	0.393±0.003	**0.440±0.010**

Table 1: The results for metrics of topic quality and text clustering on three datasets under 50 and 100 topics. The best-performing method is highlighted in **bold** and the second best method is underlined. We run each model 3 times with different random seeds and report the mean the standard deviation.

task following (Wang et al., 2022; Wu et al., 2022), and report the Purity and Normalized Mutual Information (NMI) (Schütze et al., 2008), where document labels are used during evaluation. Specifically, to compute Purity and NMI, following previous works (Zhao et al., 2021; Wu et al., 2022), we directly take the most significant topic as the cluster assignment for each document, and the metrics are denoted as **top-Purity** and **top-NMI**. Moreover, we also calculate the results with the cluster assignments from K-Means algorithm, which could be found in Appendix D.

Results We report the results of text clustering Table 1. From the results, we can find out that the kNNTM model consistently outperforms all baseline models under almost every setting. The performances indicate that kNNTM can infer high-quality document-topic distributions which accurately reflect the semantics of documents.

5.3 Ablation Studies

To analyze the effects of different modules of kNNTM, we compare kNNTM with its following variants: 1) kNNTM-w/o-kNN: kNNTM without kNN label completion module, degenerating to the basic QTM model. 2) kNNTM-w/o-Topic: kNNTM with a distance metric without considering distances from the hidden topic space. 3) kNNTM-w/o-BoW: kNNTM with a distance metric without considering the distances from the input BoW space. 4) kNNTM-w/-sim: kNNTM without kNN algorithm, and complementing labels directly with words based on pre-trained word embeddings.

Methods	C_V	TU	TQ	top-Purity	top-NMI
kNNTM	0.341	0.979	**0.334**	**0.819**	**0.436**
w/o-kNN	0.323	0.947	0.306	0.753	0.385
w/o-Topic	0.340	0.967	0.328	0.809	0.427
w/o-BoW	0.320	**0.986**	0.315	0.799	0.420
w/-sim	**0.343**	0.969	0.332	0.793	0.413

Table 2: Ablation Studies on Snippet Dataset.

The ablation studies are conducted on the Snippet dataset under 100 topics. The results are reported in Table 2, and the standard deviations are shown in Appendix Table 7 due to space limit. The effectiveness of the kNN label completion module is proved by the improvement from kNNTM-w/o-kNN to the original kNNTM. The decreases in kNNTM-w/o-Topic and kNNTM-w/o-BoW in-

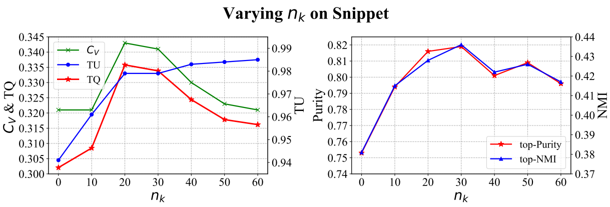
Varying n_k on Snippet

Figure 3: Sensitivity analysis on neighbor number n_k.

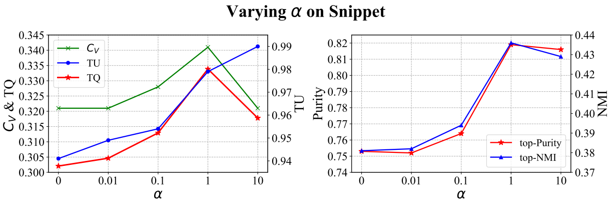
Varying α on Snippet

Figure 4: Sensitivity analysis on the coefficient α.

dicate the importance of both views in the fused distance metric. Moreover, while C_V sightly increases on kNNTM-w/-sim, the decreases of the performances on TU, TQ, and text clustering metrics indicate the kNN-based completion method would lead to higher topic quality and better topic distributions as it can better utilize the word patterns from current dataset.

5.4 Sensitivity Analysis

We conduct sensitivity analysis on two important hyper-parameters of kNNTM, the number of aggregated neighbors n_k, and the coefficient α balancing the original label and the augmented label.

As shown in Figure 3, as n_k gradually increases, the coherence score C_V and the overall topic quality TQ increase initially and then decline when n_k gets too large. A similar phenomenon can be found on those metrics about text clustering, but the TU score keeps increasing. The reason might be that increasing n_k will bring more diverse documents during label completion and further increase the diversity of learned topics. But more diverse documents will introduce more noisy words that are unrelated to the current document, and finally degrades the overall performance of the topic model.

For the coefficient α, a similar phenomenon could be found in Figure 4. As α increases, the training objective of the model puts more attention on the retrieved neighbor documents, whereas it can dominate the probabilities of the original documents once α gets too large, which can also degrade the model performance.

Models	Topic Word Examples
prodLDA	pentium amd intel chip athlon core processors processor disney walt newsgroups drama graduation quotations usenet time hiv aids boxing prevention horse racing goalkeeper epidemic academy nuclear oscar awards weapons weapon military award
WLDA	medical treatment hospital care surgery health mental patient **wikipedia wiki encyclopedia** psychological commercial natural simple law **wikipedia wiki encyclopedia** disambiguation participants retrieved article literally tickets paris french tennis inventory roland garros france
ECRTM	wikipedia wiki retrieved encyclopedia real-time simple aesthetics meanings hiv prevention cdc aids respiratory nida resp nanotechnology duo pentium athlon amd processor itanium cores cpu naval commander navy weapons nuclear carlisle force fleet
NQTM	memory upgrade upgrades virtual ddr machine ram cache income tax salary interview effective monster skills mobile force navy air mil military units fleet personnel film producer encyclopedia wiki wikipedia rugby consisting states
MCTM	ucsd **physicist physics** mathematics sociology **astrophysics anthropology** **physics aesthetics** sociology **anthropology** mathematics **physicist astrophysics** **astrophysics physicist** predicting predictions discoveries eia gsfc geophysics **physics** economics movies pentium ucsd **aesthetics** bollywood **astrophysics**
TSCTM	academy awards oscar winners nominees annual oscars award duo processor anandtech core intel imac xeon chips navy commander force mil fleet military naval air physics theoretical quantum particle solid mechanics reasoning quant
kNNTM	intel duo itanium chip imac xeon core processor navy mil commander force corps naval nuclear fleet hiv aids unaids prevention ucsf aidsinfo influenza epidemic academy winners nominees annual awards oscar nominee oscars

Table 3: Visualization of topics learned by different methods on Snippet dataset.

5.5 Topic Visualization

For qualitative evaluations of topics, we show the examples of topic words yielded by different baselines and our kNNTM model on the Snippet dataset in Table 3. We can observe that baseline models with lower TU, such as WLDA and MCTM, generate some repetitive topics with repeated words, such as "wikipedia", "encyclopedia", "physics", "physicist" and "astrophysics". Such repetitive sets of coherent words will lead to abnormally high results on C_V scores, making it unfair to compare with other methods. However, we can see that kNNTM only generates a single coherent topic for each corresponding topic and its topic quality is apparently higher.

6 Conclusions

In this paper, we identify the *label sparsity* problem in short text topic modeling, resulting from the inherently limited document length. Subsequently, we design an novel neural short text topic modeling framework dubbed kNNTM, which mitigates the label sparsity problem with a kNN label completion module that aggregates semantically similar documents to augment the reconstruction labels. To effectively find similar documents, a fused multi-view distance metric is proposed considering both local word similarities and global document semantics. Extensive experiments show that kNNTM outperforms the baselines and can generate high-quality topic and document representations.

Limitations

One limitation of the proposed kNNTM model is the time complexity of computing the OT distances. The original OT distance metric is known to have high complexity, and computing distances between each pair of documents also increases the time overhead. The inherent small text lengths and the accelerated algorithms for OT metric can help alleviate this issue, and the computations of the distances between different text pairs can be easily parallelized, but when dealing with excessively large datasets, kNNTM still faces high time cost. For future work, we hope to design a sampling strategy for kNNTM , aiming to restrict the nearest neighbor searching to a limited number of candidate documents instead of the entire dataset, and thus lowering the time cost of our model.

Ethics Statement

We comply with the ACL Code of Ethics. Our method is proposed to enhance short text topic modelling, and we believe our model would not cause significant social risks if applied appropriately.

References

Nikolaos Aletras and Mark Stevenson. 2013. Evaluating topic coherence using distributional semantics. In *Proceedings of the 10th international conference on computational semantics (IWCS 2013)–Long Papers*, pages 13–22.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022.

Jordan L Boyd-Graber, Yuening Hu, David Mimno, et al. 2017. *Applications of topic models*, volume 11. Now Publishers Incorporated.

Sophie Burkhardt and Stefan Kramer. 2019. Decoupling sparsity and smoothness in the dirichlet variational autoencoder topic model. *J. Mach. Learn. Res.*, 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 JR Ruiz, and David M Blei. 2020. Topic modeling in embedding spaces. *Transactions of the Association for Computational Linguistics*, 8:439–453.

Xuemeng Hu, Rui Wang, Deyu Zhou, and Yuxuan Xiong. 2020. Neural topic modeling with cycle-consistent adversarial training. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9018–9030.

Urvashi Khandelwal, Angela Fan, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2021. Nearest neighbor machine translation. In *International Conference on Learning Representations*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Diederik P Kingma and Max Welling. 2014. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Chenliang Li, Yu Duan, Haoran Wang, Zhiqian Zhang, Aixin Sun, and Zongyang Ma. 2017. Enhancing topic modeling for short texts with auxiliary word embeddings. *ACM Transactions on Information Systems (TOIS)*, 36(2):1–30.

Chenliang Li, Haoran Wang, Zhiqian Zhang, Aixin Sun, and Zongyang Ma. 2016. Topic modeling for short texts with auxiliary word embeddings. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 165–174.

Rishabh Mehrotra, Scott Sanner, Wray Buntine, and Lexing Xie. 2013. Improving lda topic models for microblogs via tweet pooling and automatic labeling. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 889–892.

Yishu Miao, Lei Yu, and Phil Blunsom. 2016. Neural variational inference for text processing. In *International conference on machine learning*, pages 1727–1736. PMLR.

David Mimno, Hanna Wallach, Edmund Talley, Miriam Leenders, and Andrew McCallum. 2011. Optimizing semantic coherence in topic models. In *Proceedings of the 2011 conference on empirical methods in natural language processing*, pages 262–272.

Belal Abdullah Hezam Murshed, Suresha Mallappa, Jemal Abawajy, Mufeed Ahmed Naji Saif, Hasib Daowd Esmail Al-Ariki, and Hudhaifa Mohammed Abdulwahab. 2022. Short text topic modelling approaches in the context of big data: taxonomy, survey, and analysis. *Artificial Intelligence Review*, pages 1–128.

Feng Nan, Ran Ding, Ramesh Nallapati, and Bing Xiang. 2019. Topic modeling with wasserstein autoencoders. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6345–6381.

Jipeng Qiang, Zhenyu Qian, Yun Li, Yunhao Yuan, and Xindong Wu. 2020. Short text topic modeling techniques, applications, and performance: a survey.

IEEE Transactions on Knowledge and Data Engineering, 34(3):1427–1445.

Xiaojun Quan, Chunyu Kit, Yong Ge, and Sinno Jialin Pan. 2015. Short and sparse text topic modeling via self-aggregation. In *24th International Joint Conference on Artificial Intelligence, IJCAI 2015*, pages 2270–2276. AAAI Press/International Joint Conferences on Artificial Intelligence.

Mehdi Rezaee and Francis Ferraro. 2020. A discrete variational recurrent topic model without the reparametrization trick. *Advances in neural information processing systems*, 33:13831–13843.

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. PMLR.

Michael Röder, Andreas Both, and Alexander Hinneburg. 2015. Exploring the space of topic coherence measures. In *Proceedings of the eighth ACM international conference on Web search and data mining*, pages 399–408.

Hinrich Schütze, Christopher D Manning, and Prabhakar Raghavan. 2008. *Introduction to information retrieval*, volume 39. Cambridge University Press Cambridge.

Akash Srivastava and Charles Sutton. 2017. Autoencoding variational inference for topic models. In *5th International Conference on Learning Representations*.

Dongsheng Wang, Dandan Guo, He Zhao, Huangjie Zheng, Korawat Tanwisuth, Bo Chen, and Mingyuan Zhou. 2022. Representing mixtures of word embeddings with mixtures of topic embeddings. In *International Conference on Learning Representations*.

Rui Wang, Xuemeng Hu, Deyu Zhou, Yulan He, Yuxuan Xiong, Chenchen Ye, and Haiyang Xu. 2020. Neural topic modeling with bidirectional adversarial training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 340–350.

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

Yiming Wang, Ximing Li, Xiaotang Zhou, and Jihong Ouyang. 2021. Extracting topics with simultaneous word co-occurrence and semantic correlation graphs: neural topic modeling for short texts. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 18–27.

Xiaobao Wu, Xinshuai Dong, Thong Nguyen, and Anh Tuan Luu. 2023. Effective neural topic modeling with embedding clustering regularization. *arXiv preprint arXiv:2306.04217*.

Xiaobao Wu, Chunping Li, Yan Zhu, and Yishu Miao. 2020. Short text topic modeling with topic distribution quantization and negative sampling decoder. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1772–1782.

Xiaobao Wu, Anh Tuan Luu, and Xinshuai Dong. 2022. Mitigating data sparsity for short text topic modeling by topic-semantic contrastive learning. *arXiv preprint arXiv:2211.12878*.

Qianqian Xie, Jimin Huang, Pan Du, Min Peng, and Jian-Yun Nie. 2021. Graph topic neural network for document representation. In *Proceedings of the Web Conference 2021*, pages 3055–3065.

Xiaohui Yan, Jiafeng Guo, Yanyan Lan, and Xueqi Cheng. 2013. A biterm topic model for short texts. In *Proceedings of the 22nd 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 *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 233–242.

Xumin Yu, Yongming Rao, Ziyi Wang, Zuyan Liu, Jiwen Lu, and Jie Zhou. 2021. Pointr: Diverse point cloud completion with geometry-aware transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12498–12507.

Chaohe Zhang, Xin Gao, Liantao Ma, Yasha Wang, Jiangtao Wang, and Wen Tang. 2021. Grasp: generic framework for health status representation learning based on incorporating knowledge from similar patients. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 715–723.

Delvin Ce Zhang and Hady Lauw. 2022. Meta-complementing the semantics of short texts in neural topic models. *Advances in Neural Information Processing Systems*, 35:29498–29511.

Hao Zhang, Bo Chen, Dandan Guo, and Mingyuan Zhou. 2018. WHAI: Weibull hybrid autoencoding inference for deep topic modeling. In *International Conference on Learning Representations*.

He Zhao, Dinh Phung, Viet Huynh, Trung Le, and Wray Buntine. 2021. Neural topic model via optimal transport. In *International Conference on Learning Representations*.

Qile Zhu, Zheng Feng, and Xiaolin Li. 2018. Graphbtm: Graph enhanced autoencoded variational inference for biterm topic model. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 4663–4672.

Yuan Zuo, Junjie Wu, Hui Zhang, Hao Lin, Fei Wang, Ke Xu, and Hui Xiong. 2016. Topic modeling of short texts: A pseudo-document view. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 2105–2114.

A kNNTM Algorithm Framework

Algorithm 1 The training procedure of kNNTM framework.

```
Input: the input corpus $\mathcal{D}$, topic number $K$, pre-training epoch number $P$, total epoch number $T$, the number of nearest neighbors $n_k$, hyperparameters $\alpha$, $\eta$, $\rho$.
Output: $K$ topic-word distributions $\beta_k$, $N_D$ document-topic distribution $\theta^d$
for $epoch$ from 1 to $T$ do
    for a random batch of $B$ documents do
        $\mathcal{L}_{batch} \leftarrow 0$;
        for each document $d$ in the batch do
            compute the topic distribution $\theta^d$;
            if $epoch \leq P$ then
                $\mathcal{L}_{batch} \leftarrow \mathcal{L}_{batch} + \mathcal{L}_{\mathrm{TM}}$ by Eq.4;
            else
                get augmented reconstruction label $\tilde{x}^d$ by Eq.5 with distance in Eq.12;
                $\mathcal{L}_{batch} \leftarrow \mathcal{L}_{batch} + \mathcal{L}_{\mathrm{kNNTM}}$ by Eq.14;
            end if
            update $m_d$ with $\theta^d$ by Eq.10;
        end for
        update model parameters with $\nabla\mathcal{L}_{batch}$
    end for
end for
```

B Experimental Details

B.1 Dataset Statistics

We conduct our experiments on the following public short text datasets:

GoogleNews: The GoogleNews dataset is from the Google News site and includes the titles of over 10,000 news articles categorized into 152 clusters.

Snippet: The Snippet dataset consists of over 10,000 search results from web across 8 different domains, obtained using predefined phrases.

StackOverflow: The StackOverflow dataset is sourced from the challenge data released by Kaggle. The dataset we use is a subset of 20,000 question titles randomly selected from 20 different tags from the original dataset.

We present the detailed statistics of the above three datasets after preprocessing in Table 4.

Datasets	Number of documents	Average length	Number of categories	Vocabulary size
GoogleNews	11019	5.75	152	3476
Snippet	12294	14.42	8	4720
StackOverflow	16392	5.02	20	2300

Table 4: Statistics of 3 datasets after preprocessing.

Datasets	n_k	α	η	ρ
GoogleNews	20	1.0	0.5	0.5
Snippet	30	1.0	0.2	0.6
StackOverflow	30	0.5	0.4	0.4

Table 5: Hyper-parameters for different datasets.

B.2 Baselines

Here we provide brief introductions to the baseline methods compared in this paper.

prodLDA: prodLDA (Srivastava and Sutton, 2017) is a prominent work in neural topic models. It employs black-box neural variational inference and approximates the Dirichlet prior via a logistic normal distribution.

WLDA: WLDA (Nan et al., 2019) utilizes the Wasserstein autoencoder framework for neural topic modeling and directly enforces the Dirichlet prior through Maximum Mean Discrepancy.

ECRTM: To the best of our knowledge, ECRTM (Wu et al., 2023) is the current state-of-the-art neural topic model for normal long documents. It incorporates an embedding clustering regularization that encourages word embeddings to cluster around topic embeddings.

NQTM: NQTM (Wu et al., 2020) proposes learning peakier topic distributions and discovering better topics through topic distribution quantization and negative sampling.

MCTM: MCTM (Zhang and Lauw, 2022) focuses on variable-length corpora and utilizes meta-learning to train a missing semantics predictor for short documents based on other long documents.

TSCTM: TSCTM (Wu et al., 2022) is a state-of-the-art short text neural topic model. It builds upon NQTM and introduces a contrastive objective on quantized distributions.

B.3 Implementation Details

Regarding the training environment, our method is implemented using PyTorch 1.12.1 with Python

Model			prodLDA	WLDA	ECRTM	NQTM	MCTM	TSCTM	kNNTM
GoogleNews	$K = 50$	km-Purity	0.333±0.016	0.447±0.022	0.535±0.030	0.494±0.018	0.542±0.025	0.595±0.017	**0.612±0.006**
		km-NMI	0.372±0.007	0.664±0.012	0.789±0.019	0.706±0.007	0.705±0.008	0.793±0.009	**0.825±0.004**
	$K = 100$	km-Purity	0.364±0.003	0.607±0.007	0.502±0.008	0.660±0.011	0.433±0.079	0.769±0.003	**0.788±0.005**
		km-NMI	0.381±0.003	0.731±0.004	0.690±0.012	0.756±0.003	0.581±0.056	0.845±0.004	**0.871±0.003**
Snippet	$K = 50$	km-Purity	0.503±0.017	0.617±0.028	0.761±0.015	0.667±0.017	0.673±0.003	0.723±0.012	**0.775±0.004**
		km-NMI	0.172±0.008	0.298±0.012	**0.440±0.009**	0.332±0.013	0.328±0.008	0.390±0.011	0.438±0.006
	$K = 100$	km-Purity	0.476±0.004	0.677±0.014	0.805±0.010	0.699±0.006	0.683±0.026	0.761±0.010	**0.821±0.007**
		km-NMI	0.132±0.002	0.322±0.003	**0.437±0.008**	0.339±0.003	0.331±0.015	0.390±0.005	0.436±0.001
StackOverflow	$K = 50$	km-Purity	0.227±0.003	0.465±0.007	0.404±0.042	0.452±0.041	0.428±0.017	0.586±0.005	**0.613±0.009**
		km-NMI	0.074±0.005	0.323±0.003	0.290±0.033	0.315±0.033	0.320±0.012	0.433±0.007	**0.468±0.003**
	$K = 100$	km-Purity	0.166±0.005	0.440±0.005	0.451±0.026	0.480±0.038	0.401±0.006	0.574±0.009	**0.615±0.011**
		km-NMI	0.046±0.003	0.296±0.004	0.350±0.018	0.317±0.029	0.300±0.007	0.397±0.002	**0.443±0.008**

Table 6: The results for metrics of text clustering on three datasets under 50 and 100 topics. The best-performing method is highlighted in **bold** and the second best method is underlined. We run each model 3 times with different random seeds and report the mean the standard deviation.

Methods	C_V	TU	TQ	top-Purity	top-NMI
kNNTM	0.341±0.006	0.979±0.002	**0.334±0.006**	**0.819±0.005**	**0.436±0.001**
w/o-kNN	0.323±0.008	0.947±0.009	0.306±0.010	0.753±0.001	0.385±0.005
w/o-Topic	0.340±0.008	0.967±0.002	0.328±0.008	0.809±0.009	0.427±0.003
w/o-BoW	0.320±0.006	**0.986±0.002**	0.315±0.006	0.799±0.007	0.420±0.002
w/-sim	**0.343±0.008**	0.969±0.005	0.332±0.009	0.793±0.013	0.413±0.006

Table 7: Ablation Studies on Snippet Dataset with 100 topics. We run each model 3 times with different random seeds and report the mean the standard deviation.

3.9.16, and the experiments are conducted on four GeForce RTX 2080Ti GPUs. Regarding the model architecture, the encoder network consists of a 3-layer MLP, and we set the hidden layer's dimension to 128. Training is performed using the Adam optimizer (Kingma and Ba, 2014) with a learning rate of 0.002. We use a batch size (B) of 200, 20 pre-training epochs (P), and a total of 200 epochs (T). For other hyper-parameters, please refer to Table 5. We use grid search to determine the value of the above hyperparameters. And for all baselines, we follow the hyperparameter settings reported in their original papers. Additionally, we employ 300-dimensional GloVe embeddings as pre-trained word embeddings for all the methods that require word embeddings.

C Metrics for Topic-Word Distribution

C.1 Topic Coherence

We use C_V as the topic coherence metric in our experiments. For a topic z with T words $\{x_1, x_2, ..., x_T\}$, the definition of C_V is

$$C_V(z) = \frac{1}{T} \sum_{i=1}^{T} s_{\cos}\left(v_{\text{NPMI}}\left(x_i\right), \boldsymbol{v}_{\text{NPMI}}\left(\boldsymbol{x}_{1:T}\right)\right), \tag{C.1}$$

where $s_{\cos}(\cdot)$ is the cosine similarity between two vectors, and the v_{NPMI} vector is defined as

$$\begin{aligned} v_{\text{NPMI}}\left(x_i\right) &= \left\{\text{NPMI}\left(x_i, x_j\right)\right\}_{j=1,\ldots,T} \\ \boldsymbol{v}_{\text{NPMI}}\left(\boldsymbol{x}_{1:T}\right) &= \left\{\sum_{i=1}^{T} \text{NPMI}\left(x_i, x_j\right)\right\}_{j=1,\ldots,T} . \end{aligned} \tag{C.2}$$

And the NPMI indicates the Normalized Pointwise Mutual Information between words and is calculated as

$$\text{NPMI}\left(x_i, x_j\right) = \frac{\log \frac{p(x_i, x_j)+\epsilon}{p(x_i)p(x_j)}}{-\log\left(p\left(x_i, x_j\right)+\epsilon\right)}, \tag{C.3}$$

where $p(x_i, x_j)$ is the co-occurrence probability within a reference corpus.

C.2 Topic Diversity

We use Topic Uniqueness (TU) as the metric for topic diversity. The TU of K topics $\{z_1, z_2, ..., z_K\}$ could be calculated as:

$$TU = \frac{1}{K} \sum_{i=1}^{K} \frac{1}{T} \sum_{j=1}^{T} \frac{1}{\text{cnt}\left(x_j^{z_i}\right)}, \tag{C.4}$$

where $\text{cnt}(x_i)$ indicates number of times that the word x_i appears in all topics.

D More Results on Text Clustering

Metrics To evaluated the models on text clustering method, besides top-Purity and top-NMI, we also apply the K-Means algorithm to assign clusters to different documents. We set cluster number set as the topic number K and apply the KMeans algorithm on all the document-topic distribution vectors. The metrics are denoted as **km-Purity** and **km-NMI**.

Results We show the results in Table 6. From the results, we could draw the same conclusions as in section 5.2. Our kNNTM model outperforms all baseline models under almost every setting, and achieves compatible results in a few scenarios. That indicates kNNTM possesses the ability to obtain high quality document-topic distributions and derive the hidden semantics for each document.

E More Results of Ablation Studies

Due to space limit, the standard deviations of the results in the Ablation Studies section are not reported in the main paper. Here we provide the main and the standard deviation of the results in Table 7, corresponding to Table 2.