A Word Embedding Topic Model for Robust Inference of Topics and Visualization

Sanuj Kumar
sanujkr@nmsu.edu
New Mexico State University
Las Cruces, USA

Tuan M. V. Le
tuanle@nmsu.edu
New Mexico State University
Las Cruces, USA

ABSTRACT

Probabilistic topic models for semantic visualization are useful for discovering and visualizing latent topics in document collections. In these models, the inference of topics and visualization is largely based on word co-occurrences within documents. Therefore, when documents in a corpus are short in length, these models may not achieve good results due to the sparsity of word co-occurrences. In this paper, we propose a word embedding topic model (WTM) that is robust to data sparsity when detecting topics and generating visualization of short texts. Extensive experiments conducted on four real-world datasets show that WTM is more effective in dealing with short texts than state-of-the-art models.

KEYWORDS

topic modeling, short texts, visualization

ACM Reference Format:
Sanuj Kumar and Tuan M. V. Le. 2021. A Word Embedding Topic Model for Robust Inference of Topics and Visualization. In *The First International Conference on AI-ML-Systems (AIMLSystems '21), October 21–23, 2021, Bangalore, India.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3486001.3486002

1 INTRODUCTION

Probabilistic topic models for semantic visualization such as PLSV [6] and its variants [10, 11] are useful for discovering and visualizing latent topics in document collections. Different from the traditional topic models like Probabilistic Latent Semantic Analysis (PLSA) [5] and Latent Dirichlet Allocation (LDA) [1], PLSV jointly learns a set of latent topics and embeds both documents and topics in a 2D or 3D visualization space for visualizing as a scatterplot. Such visualization and topic modeling can support exploratory tasks by allowing the user to easily understand and interpret a huge and complex set of documents.

In this paper, we consider the problem of visualization and topic modeling over short texts (e.g., tweets, search snippets, or status updates which can be as short as 140 characters). Semantic visualization models such as PLSV may work well for normal-length documents, however, they may be not ideal methods for visualizing and extracting topics of such short texts. The main reason is that in these models the inference of topics and visualization is largely based on word co-occurrences within documents. Therefore, when the documents are short, these models may not achieve good results due to the sparsity of word co-occurrences.

There have been several works on topic modeling for short texts. PTM [25] uses the concept of pseudo document to aggregate short texts to deal with data sparsity. Other methods such as GPUDMM [14], ETM [3] rely on word embeddings that can be learned from an external large corpus to enrich the learned topic model over short texts. However, none of these models produces a visualization of the corpus while learning the topics. Using these models for visualization, one needs to follow a two-step approach that first performs topic modeling and then uses a dimensionality reduction method such as t-SNE [15] to embed the topic distributions into the visualization space. Although this approach is straightforward, the results can be mixed because there are two different objective functions optimized in two separate steps.

In the context of visualization and topic modeling, the most related work to our work is GaussianSV [12]. GaussianSV relies on auxiliary word embeddings to build a joint model for generating both topic model and visualization of short texts. However, GaussianSV does not scale well to large datasets. Therefore, in this work, we aim to propose a novel word embedding topic model called WTM and develop an Auto-Encoding Variational Bayes based [8] inference algorithm that can scale well to large datasets. To deal with the sparsity of word co-occurrences, WTM leverages semantic information of word embeddings to enrich the learned topic model and visualization. WTM assumes that topics have representations in both visualization space and word embedding space. WTM ties together the visualization coordinates, topic embeddings, and word embeddings into a single generative process to explain the generation of documents. WTM models the word distribution of topic using a log-linear model that takes the matrix multiplication of the word embedding matrix and the topic embedding as inputs. The word distribution is akin to the likelihood of words used in the continuous bag-of-words (CBOW) model for learning word embeddings [16, 17]. The main difference is that the topic embedding is used as the context vector instead. In CBOW, the context vector is formed by summing the context embedding vectors of surrounding words.

We summarize our contributions as follows:

- We propose a novel word embedding topic model called WTM for visualization and topic modeling of short texts.
- We develop an Auto-Encoding Variational Bayes based inference algorithm for WTM that can scale well to large datasets.
- We conduct extensive experiments on four real-world short-text datasets. The extensive experiments show that our method

AIMLSystems '21, October 21–23, 2021, Bangalore, India

ACM ISBN 978-1-4503-8594-7/21/10…$15.00
https://doi.org/10.1145/3486001.3486002

scales well to large datasets and it outperforms state-of-the-art methods in terms of k-NN accuracy and topic coherence in most settings.

2 RELATED WORK

Topic modeling for Short Texts. Besides traditional topic models such as PLSA [5] and LDA [1], several topic models have been proposed for short texts. The first approach is to aggregate short texts into pseudo-documents before applying topic modeling such as PTM [25], SATM [22]. In the second approach, we can rely on auxiliary word embeddings to enrich the learned topics. There are several proposed models in this category such as GPUDMM [14], GPUPDMM [13], LFDMM [18], GaussianLDA [2], ETM [3]. Among these methods, ETM is closer to our work where it also uses the idea of modeling the word distribution of topic using a log-linear model that takes the matrix multiplication of the word embedding matrix and the topic embedding as inputs. A more detailed review on topic models for short texts can be found in [21].

Topic modeling and Visualization. The above methods are not designed for visualization. For visualization, we need to use a dimensionality reduction method such as t-SNE [15] to visualize the documents' topic proportions. This two-step approach for topic modeling and visualization may not be ideal. Therefore, there have been proposed joint models for this problem such as PLSV[6] and its variants [10, 11]. These models work well for normal-length documents but they are not developed for short-texts. Recently, GaussianSV [12] has been proposed to tackle this problem for short texts. GaussianSV is the most related work to our proposed model. It is different from us where it replaces LDA's categorical distributions over words with multivariate Gaussian distributions on the embedding space.

3 THE PROPOSED WORD EMBEDDING TOPIC MODEL

3.1 Generative Process

WTM is a generative probabilistic model of documents and visualization. In its generative process, we assume that each document n has a coordinate x_n in a 2D or 3D visualization space (i.e., $x_n \in \mathbb{R}^D$, $D = 2$ or 3). Let $\mathcal{V}$ be a finite vocabulary from documents and $V = |\mathcal{V}|$ is the size of the vocabulary. A word in the vocabulary is represented by an embedding vector in the semantic space of words. Let $\omega_v \in \mathbb{R}^P$ be the embedding vector of word v in the vocabulary where P is the dimensionality[1] of the word embedding. We use Ω to indicate the $P \times V$ word embedding matrix where its column v is the word embedding ω_v of the word v in the vocabulary.

In our model, a topic is represented by three latent representations in three different spaces. In the same semantic space of words, a topic z is represented as a topic embedding vector $\tau_z \in \mathbb{R}^P$ that is a distributed representation of the topic z. In the word simplex space, a topic z is represented as a distribution over the vocabulary β_z. In the visualization space, we assign to each topic z a coordinate $\phi_z \in \mathbb{R}^D$. By using this representation, we can visualize topics together with the documents in the same visualization space. The distances between documents, topics will reflect the topic distributions of the documents. We tie together the three representations as follows.

Following [3], we model the word distribution β_z of topic z using a log-linear model that takes the matrix multiplication of the word embedding matrix Ω and the topic embedding τ_z as inputs. More specifically,

$$\beta_z = \text{softmax}(\Omega^\top \tau_z + b) \tag{1}$$

The word distribution β_z is akin to the likelihood of words used in the continuous bag-of-words (CBOW) model for learning word embeddings [16, 17]. The main difference is that the topic embedding τ_z is used as the context vector instead. In CBOW, the context vector is formed by summing the context embedding vectors of surrounding words. By using topic embedding in Eq. 1, the words are not drawn from the context of surrounding words, but from the document context. Different from [3], we introduce the bias term b in Eq. 1 because we observe that it significantly improves the performance of the model.

For the topic embedding τ_z, a multi-layer perceptron (MLP) f is introduced for transforming ϕ_z to τ_z, i.e., $\tau_z = f(\phi_z)$. This transformation allows the learned topic embeddings are faithfully displayed in the visualization space. The topic distribution θ_n of a document n is defined using a softmax function over its distances to all topics:

$$\theta_{nz} = p(z|x_n, \mathbf{\Phi}) = \frac{\exp\left(-\frac{1}{2}\|x_n - \phi_z\|^2\right)}{\sum_{z'=1}^{Z} \exp\left(-\frac{1}{2}\|x_n - \phi_{z'}\|^2\right)} \tag{2}$$

here $\|x_n - \phi_z\|$ is the Euclidean distance between document n and topic ϕ_z. In Eq. 2, the topic probability is high when the document is close to that topic in the visualization space. This is useful for visual sense-making of text corpora because the users can quickly see the topics of documents in the visualization.

Putting everything together, WTM assumes the following process to generate documents and visualization. The graphical model of WTM is shown in Figure 1.

(1) For each topic $z = 1, \cdots, Z$:
  (a) Compute its semantic embedding vector: $\tau_z = f(\phi_z)$
  (b) Compute its word distribution: $\beta_z = \text{softmax}(\Omega^\top \tau_z)$
(2) For each document $n = 1, \cdots, N$:
  (a) Draw a document coordinate: $x_n \sim \text{Normal}(\mathbf{0}, \gamma \boldsymbol{I})$
  (b) For each word w_{nm} in document n:
    (i) Draw a topic: $z \sim \text{Multi}\left(\{p(z|x_n, \mathbf{\Phi})\}_{z=1}^{Z}\right)$
    (ii) Draw a word: $w_{nm} \sim \text{Multi}(\beta_z)$

In Steps 1a and 1b, we compute, for each topic z, its embedding vector τ_z and its word distribution β_z (Eq. 1). Step 2a is for drawing the coordinate x_n of a document n. Here x_n has a Gaussian prior:

$$p(x_n|\gamma) = \left(\frac{1}{2\pi\gamma}\right)^{D/2} \exp\left(-\frac{\|x_n\|^2}{2\gamma}\right) \tag{3}$$

For each document, we compute its distances to all topics. The topic distribution is then computed as in Eq. 2. In Step 2b(i), for each word of document n, we draw its topic z based on the document topic distribution $p(z|x_n, \mathbf{\Phi})$. Finally, in Step 2b(ii), we draw a word based on the word distribution β_z. The computation of β_z in Eq. 1 is based on pre-fitted word embeddings that are learned from a large corpus

[1] In our experiments, $P = 300$

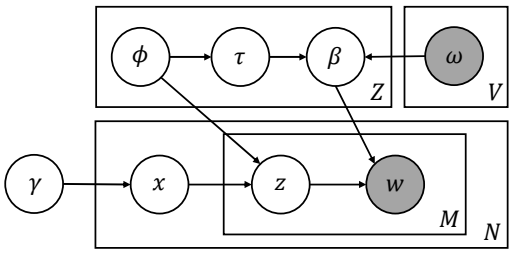

Figure 1: Graphical Model of WTM

by a dedicated algorithm such as the skip-gram model [17]. The word embeddings can be obtained by training a skip-gram model on a large external corpus if the domains of the two corpora are overlapped or on the corpus itself if it is large enough. We assume that by supplementing short texts with semantic information in word embeddings learned from a dedicated algorithm, our model is more robust to the data sparsity, as shown by the experiments in Section 4.

3.2 Autoencoding Variational Inference

Given a corpus of documents $\mathcal{W} = \{w_1, ..., w_N\}$, we want to estimate the parameters of WTM including document coordinates $\{x_n\}_{n=1}^N$, topic coordinates $\{\phi_z\}_{z=1}^Z$, word distributions of topics $\{\beta_z\}_{z=1}^Z$, and topic embeddings $\{\tau_z\}_{z=1}^Z$. Let $\boldsymbol{\chi} \in \mathbb{R}^{N\times D}$ is the document coordinate matrix where the row n is the visualization coordinate x_n of document n, $\boldsymbol{\Phi} \in \mathbb{R}^{Z\times D}$ be the topic coordinate matrix where the row z is the visualization coordinate ϕ_z of topic z, $\boldsymbol{\beta} \in \mathbb{R}^{Z\times V}$ be the topic-word probability matrix where the row z is the word distribution β_z of topic z, and $\boldsymbol{\tau} \in \mathbb{R}^{Z\times P}$ is the topic embedding matrix where its row z is the embedding vector τ_z of topic z. Let $\boldsymbol{\Psi} = \langle \boldsymbol{\chi}, \boldsymbol{\Phi}, \boldsymbol{\beta}, \boldsymbol{\tau} \rangle$. A document n is represented as a row vector of word counts: $\mathbf{w}_n \in \mathbb{Z}_{\geq}^{|\mathcal{V}|}$ and $\mathbf{w}_n^v$ is the number of occurrences of word $v \in \mathcal{V}$ in the document. The log marginal likelihood of the corpus is given by:

$$p(\mathcal{W}) = \sum_{n=1}^{N} \log p\left(\mathbf{w}_n|\gamma, \boldsymbol{\Phi}, \boldsymbol{\beta}\right) \tag{4}$$

where the marginal likelihood of a document is as follows:

$$\begin{aligned} p\left(\mathbf{w}_n|\gamma, \boldsymbol{\Phi}, \boldsymbol{\beta}\right) &= \int_x \left(\prod_{v=1}^{V} \left(\sum_{z=1}^{Z} p(v|z, \boldsymbol{\beta}) p\left(z|x, \boldsymbol{\Phi}\right) \right)^{\mathbf{w}_n^v} \right) p(x|\gamma) dx \\ &= \int_x \left(\prod_{v=1}^{V} p\left(v|x, \boldsymbol{\Phi}, \boldsymbol{\beta}\right)^{\mathbf{w}_n^v} \right) p(x|\gamma) dx \end{aligned} \tag{5}$$

here $\boldsymbol{\beta}, p\left(z|x, \boldsymbol{\Phi}\right)$ are computed using Eq. 1 and Eq. 2 respectively. To estimate the model parameters, we maximize the log marginal likelihood in Eq. 4 based on the Auto-Encoding Variational Bayes (AEVB) approach [8, 23]. For the simplicity of deriving AEVB for our model, we collapse the discrete latent topic assignment variable $\boldsymbol{z}$, as in [19, 23]. More specifically, we have the following lower bound to the marginal log-likelihood (ELBO) of a document:

$$\begin{aligned} &\mathcal{L}\left(\eta|\gamma, \boldsymbol{\Phi}, \boldsymbol{\beta}\right) = \\ &- \mathbb{D}_{\mathrm{KL}}\left[q(x|\mathbf{w}_n, \eta) \| p(x|\gamma)\right] + \mathbb{E}_{q(x|\mathbf{w}_n, \eta)}\left[\log p\left(\mathbf{w}_n|x, \boldsymbol{\Phi}, \boldsymbol{\beta}\right)\right] \end{aligned} \tag{6}$$

where $p(x|\gamma)$ is the prior distribution of document coordinate x (Eq. 3), $q(x|\mathbf{w}_n, \eta) = \text{Normal}\left(\boldsymbol{\mu}_n, \Sigma_n\right)$ is the variational distribution and $\boldsymbol{\mu}_n$, diagonal $\Sigma_n \in \mathbb{R}^D$ are outputs of the encoding feed-forward neural network with variational parameters η. The whole inference network architecture WTM is shown in Figure 2. The KL divergence between two Gaussians in Eq. 6 can be computed in a closed form as follows [7]:

$$\begin{aligned} &\mathbb{D}_{\mathrm{KL}}\left[q(x|\mathbf{w}_n, \eta) \| p(x|\gamma)\right] \\ &= \frac{1}{2}\left(\mathrm{tr}\left((\gamma \boldsymbol{I})^{-1} \Sigma_n \right) + \left(-\boldsymbol{\mu}_n\right)^\top (\gamma \boldsymbol{I})^{-1} \left(-\boldsymbol{\mu}_n\right) - D + \log \frac{|\gamma \boldsymbol{I}|}{|\Sigma_n|} \right) \end{aligned} \tag{7}$$

To estimate the expectation w.r.t $q(x|\mathbf{w}_n, \eta)$ in Eq. 6, we sample $x^{(l)}$ from the posterior $q(x|\mathbf{w}_n, \eta)$ by using the reparameterization trick, i.e., $x^{(l)} = \boldsymbol{\mu}_n + \Sigma_n^{1/2} \boldsymbol{\epsilon}^{(l)}$ where $\boldsymbol{\epsilon}^{(l)} \sim \text{Normal}\left(\mathbf{0}, \boldsymbol{I}\right)$ [8]. The expectation can then be approximated as:

$$\mathbb{E}_{q(x|\mathbf{w}_n, \eta)}\left[\log p\left(\mathbf{w}_n|x, \boldsymbol{\Phi}, \boldsymbol{\beta}\right)\right] \approx \frac{1}{L} \sum_{l=1}^{L} \log p\left(\mathbf{w}_n|x^{(l)}, \boldsymbol{\Phi}, \boldsymbol{\beta}\right) \tag{8}$$

$$= \log\left(\theta_n^{(l)} \boldsymbol{\beta}\right) \mathbf{w}_n^T \tag{9}$$

where $\theta_n^{(l)} \in \mathbb{R}^Z$ is a row vector of topic distribution and $\theta_{nz}^{(l)} = p\left(z|x^{(l)}, \boldsymbol{\Phi}\right)$ is computed as in Eq. 2. For the whole corpus, the lower bound is then approximated as:

$$\begin{aligned} &\mathcal{L}(\boldsymbol{\Psi}) = \\ &\sum_{n=1}^{N} \left[-\frac{1}{2}\left(\mathrm{tr}\left((\gamma \boldsymbol{I})^{-1} \Sigma_n \right) + \left(-\boldsymbol{\mu}_n\right)^T (\gamma \boldsymbol{I})^{-1} \left(-\boldsymbol{\mu}_n\right) - D + \log \frac{|\gamma \boldsymbol{I}|}{|\Sigma_n|} \right) \right. \\ &\left. + \log\left(\hat{\theta}_n \boldsymbol{\beta}\right) \mathbf{w}_n^T \right] \end{aligned} \tag{10}$$

where $\boldsymbol{\mu}_n$, diagonal $\Sigma_n \in \mathbb{R}^D$ are outputs of the encoding feed-forward neural network as shown in the left part of the whole inference network of WTM in Figure 2. In the experiments, we set $H1 = H2 = H3 = H4 = 100$. The embedding dimensionality P is set to 300 and the visualization dimensionality is 2.

4 EXPERIMENTS

4.1 Datasets

We use four short-text datasets for assessing the performance of WTM. We remove stopwords and documents with less than 3 words. The first dataset is BBC[2] which contains 2,220 BBC news articles in 5 categories from 2004-2005 [4]. To keep it a short-text dataset, we only use titles and headlines of articles. The second dataset is SearchSnippet[3] which consists of 12,267 web searched snippets from 8 different domains [20]. The third dataset is YahooAnswers from the Yahoo! Answers Comprehensive Questions and Answers version 1.0 dataset [24]. YahooAnswers has 10 classes and 50,000

[2] http://mlg.ucd.ie/datasets/bbc.html
[3] http://jwebpro.sourceforge.net/data-web-snippets.tar.gz

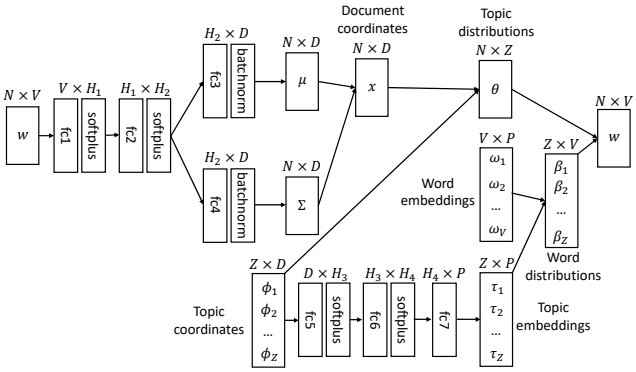

Figure 2: The Inference Network of WTM

documents (5000 documents per class). We only use the question title and the question content. The fourth dataset is AGNEWS which contains 119,938 AG's news articles from 4 classes. We only use the title and description fields [24]. The average document length is 12.02 words for BBC, 13.19 words for SearchSnippet, 10.75 for YahooAnswers, and 15.4 for AGNEWS.

4.2 Comparative Baselines

There are two classes of baselines. The first class includes methods for joint topic modeling and visualization: PLSV-VAE [4] [6, 19], GaussianSV[5] [12]. The second class includes methods for topic modeling only. For visualization, we need to use t-SNE[6] [15] to visualize the documents' topic proportions. We compare our method WTM with ProdLDA[7] + t-SNE [23], ETM[8] + t-SNE [3], PTM [9] + t-SNE [25], and GPUDMM [9] + t-SNE [14]. In our experiments, we use default parameter settings for all baselines. The batch size is set to 256 for BBC, 1000 for SearchSnippet, 5000 for YahooAnswers, and 10000 for AGNEWS. For word embeddings, we train skip-gram [17] on each corpus and use these for all of the methods that rely on word embeddings. We use dropout with probability $p = 0.2$ and $\gamma = 1$. The number of samples L per document is set to 1. We use Adam as our optimizing algorithm. The learning rate is set to 0.001. Models are trained with 1000 epochs. All the experiment results are averaged across five runs on a system with 64GB memory, an Intel(R) Xeon(R) CPU E5-2623v3, 16 cores at 3.00GHz. The GPU in use on this system is NVIDIA Quadro P2000 GPU with 1024 CUDA cores and 5 GB GDDR5.

4.3 Visualization Quantitative Evaluation

In this task, we quantitatively measure the quality of the visualization by employing the k-nearest neighbors (k-NN) accuracy in the visualization space [6, 15]. We rely on the labels of documents and a k-NN classifier is used to classify documents using their visualization coordinates. A good visualization should group documents of the same label together. Therefore, it will yield a high classification

[4]https://github.com/dangpnh2/plsv_vae
[5]https://github.com/tuanlvm/GaussianSV
[6]https://github.com/DmitryUlyanov/Multicore-TSNE
[7]https://github.com/akashgit/autoencoding_vi_for_topic_models
[8]https://github.com/adjidieng/ETM
[9]https://github.com/qiang2100/STTM

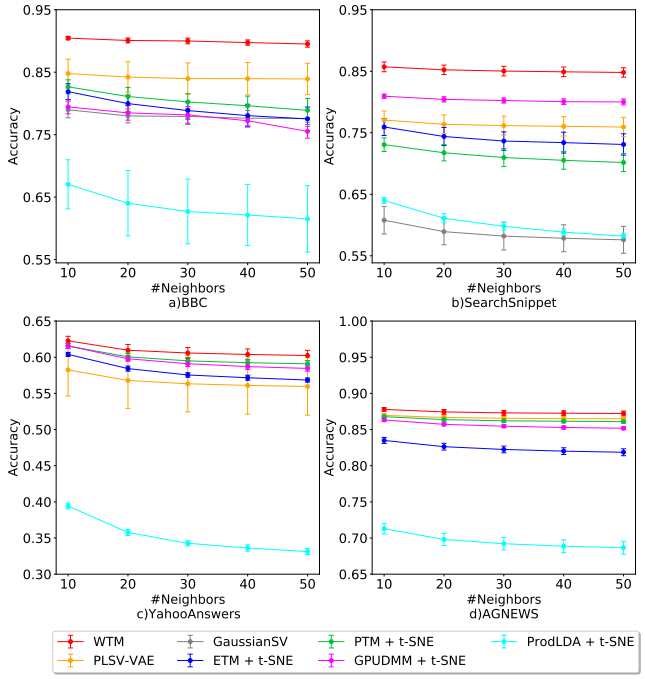

Figure 3: k-NN accuracy in the visualization space. Vary k with $Z = 50$ topics. GaussianSV is not shown in c), d) because it does not return any results even after 48 hours of running

accuracy in the visualization space. Figures 3 and 4 show k-NN accuracy of all models across datasets when we vary the k nearest neighbors ($Z = 50$ topics) and when we vary the number of topics Z ($k = 50$) respectively. It can be seen from Figure 3 that our method significantly outperforms other baselines across all settings and datasets. Among methods for joint topic modeling and visualization, WTM achieves better accuracy than PLSV-VAE because PLSV-VAE is not designed for short texts. In some settings, PLSV-VAE is better than the short-text topic models with t-SNE. This may be because PLSV-VAE is a joint model, therefore it can sometimes beat the two steps approach for short texts. WTM is better than GaussianSV while it runs much faster than GaussianSV, as shown in Section 4.6. GaussianSV are not shown in Figures 3c and 3d because it does not scale well to these large datasets. It does not return any results even after 48 hours of running. For other baselines that rely on t-SNE for visualization, their accuracies are lower than WTM. This shows that the two-step approach for visualization is not ideal. Results in Figure 4 also show that WTM is stable across different numbers of topics and beats all the baselines in most of the settings.

4.4 Topic Coherence

In this task, we aim to show that while achieving good visualization results, WTM also generates a good topic model. We evaluate the quality of topic models in terms of topic coherence. We use the Normalized Pointwise Mutual Information (NPMI) [9] estimated based on a large external corpus. We use Wikipedia 7-gram dataset created from the Wikipedia[10] dump data as of June 2008 version 12. The

[10]https://nlp.cs.nyu.edu/wikipedia-data/

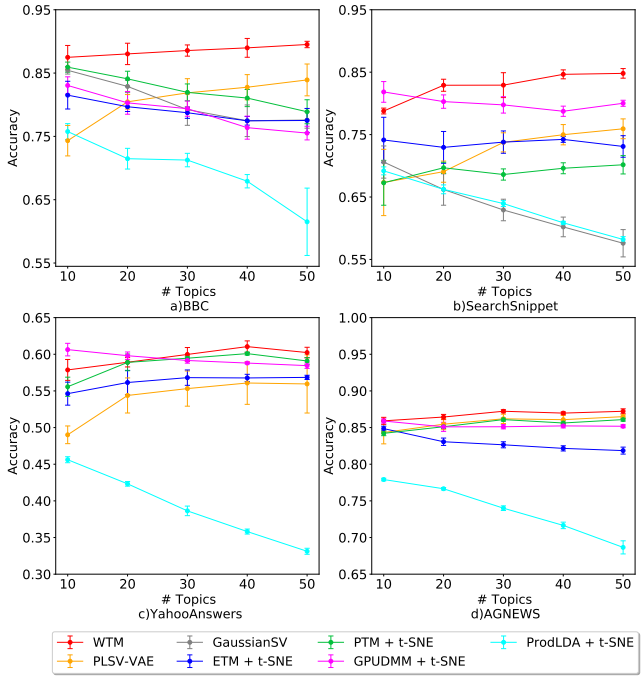

Figure 4: k-NN accuracy in the visualization space. Vary topics Z with $k = 50$. GaussianSV is not shown in c), d) because it does not return any results even after 48 hours of running

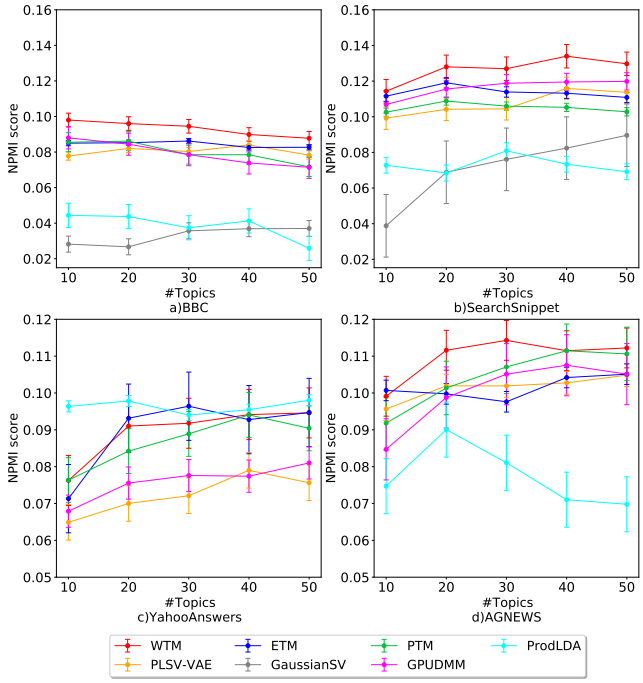

Figure 5: Topic coherence based on NPMI with different number of topics Z. GaussianSV are not shown in c), d) because it does not return any results even after 48 hours of running

Normalized Pointwise Mutual Information of a given pair of words is computed as $NPMI(w_i, w_j) = \log \frac{p(w_i, w_j)}{p(w_i)p(w_j)} / (-\log p(w_i, w_j))$.

For each topic, we compute its NMPI based on the average of the pairwise NPMI of its top 10 words. For each method, we average NPMI of its topics and report the results. Figure 5 shows the NPMI scores of all the models with respect to different numbers of topics. Except for YahooAnswers in Figure 5c where WTM achieves an NPMI score as good as those by other baselines, for other datasets, WTM can achieve better results. This shows that WTM can find good topics while producing better visualization. In Table 1, we show some example topics found by WTM in AGNEWS dataset for a qualitative evaluation of topic quality.

4.5 Short Texts Visualization Examples

In this section, we show some visualization examples by different methods. Due to space constraints, we only show the visualizations by our method and the next three best methods on SearchSnippet (Figure 6), YahooAnswers (Figure 7), and AGNEWS (Figure 8). It can be seen that the classes in PTM's and GPUDMM's visualization are often mixed together. In Figure 6b, PLSV mixes some documents of *health* and *business* together while WTM can separate well the two classes. In Figure 8b, while *business* and *sports* are not well separated in PLSV, WTM can differentiate these two classes.

4.6 Running Time Comparison

We measure the average running time by different methods. Figure 9 shows the results. GaussianSV does not scale well to large datasets. We do not show GaussianSV in Figures 9c, 9d because it does not

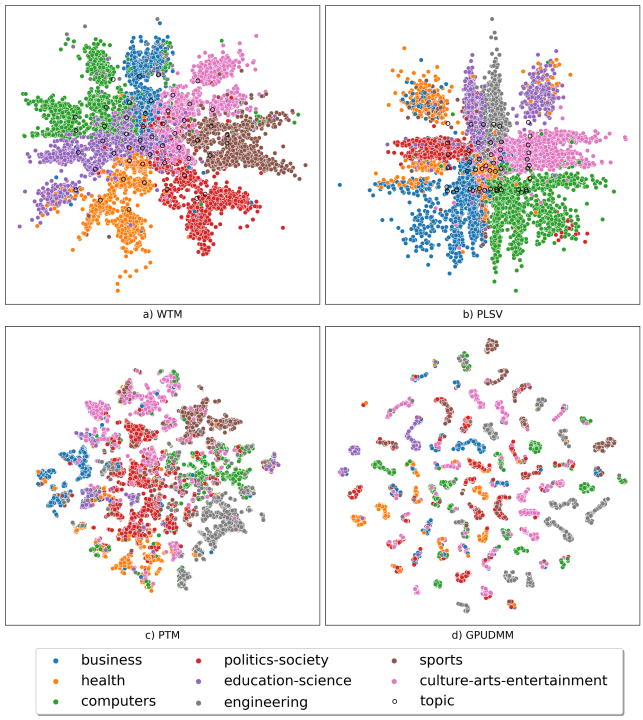

Figure 6: Visualization of SearchSnippet by different methods ($Z = 50$ topics)

Table 1: Example topics found by WTM in AGNEWS dataset

Sci/Tech			Business		Sports		World		
Topic27	**Topic30**	**Topic35**	**Topic6**	**Topic28**	**Topic29**	**Topic43**	**Topic5**	**Topic23**	**Topic37**
microsoft	space	computer	rate	quarter	game	game	baghdad	people	minister
search	nasa	apple	reserve	profit	team	bowl	killed	killed	election
window	scientist	processor	federal	sale	player	team	iraqi	police	prime
software	moon	intel	interest	third	league	season	iraq	official	president
google	earth	company	price	percent	sport	coach	city	reuters	iraq

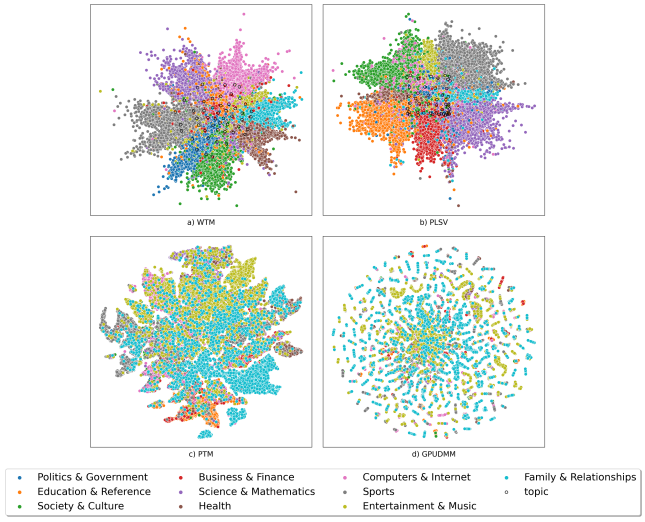

Figure 7: Visualization of YahooAnswers by different methods ($Z = 50$ topics)

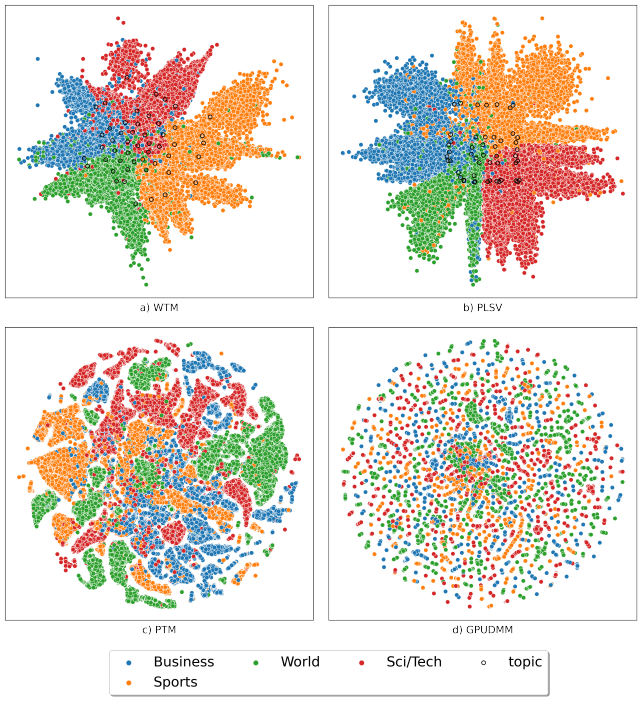

Figure 8: Visualization of AGNEWS by different methods ($Z = 50$ topics)

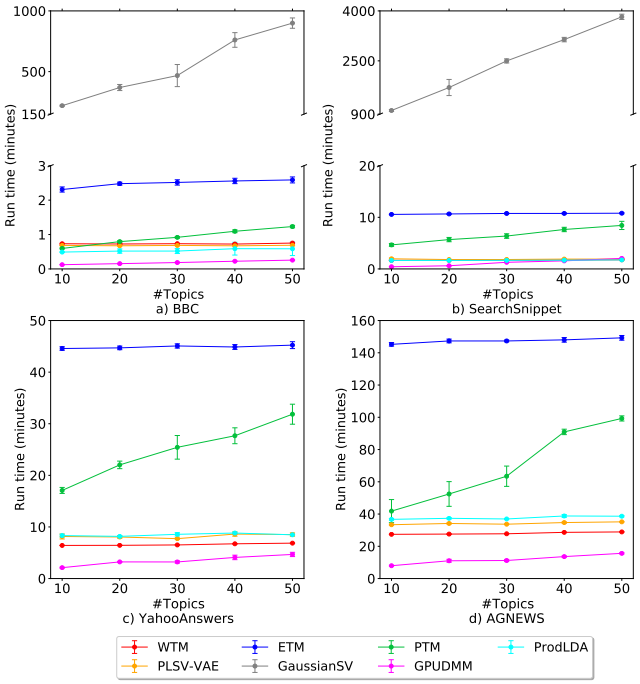

Figure 9: Running time comparison

return any results even after 48 hours of running. For two-step approach methods, PTM, ETM, ProdLDA, and GPUDMM, we do not count the running time of t-SNE. We can see from the figure that WTM scales well to very large datasets.

5 CONCLUSION

We propose WTM, a word embedding topic model for jointly generating topic model and visualization of short texts. To deal with the sparsity of word co-occurrences, WTM leverages semantic information of word embeddings to enrich the learned topic model and visualization. WTM assumes that topics have representations in both visualization space and word embedding space. WTM ties together the visualization coordinates, topic embeddings, and word embeddings into a single generative process to explain the generation of documents. The extensive experiments show that our method scales well to large datasets and it outperforms state-of-the-art methods in terms of k-NN accuracy and topic coherence in most settings.

ACKNOWLEDGMENTS

This research is sponsored by NSF #1757207 and NSF #1914635.

REFERENCES

[1] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3, Jan (2003), 993–1022.

[2] Rajarshi Das, Manzil Zaheer, and Chris Dyer. 2015. Gaussian lda for topic models with word embeddings. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 795–804.

[3] Adji B Dieng, Francisco JR Ruiz, and David M Blei. 2020. Topic modeling in embedding spaces. *Transactions of the Association for Computational Linguistics* 8 (2020), 439–453.

[4] Derek Greene and Pádraig Cunningham. 2006. Practical Solutions to the Problem of Diagonal Dominance in Kernel Document Clustering. In *Proc. 23rd International Conference on Machine learning (ICML'06)*. ACM Press, 377–384.

[5] Thomas Hofmann. 1999. Probabilistic Latent Semantic Analysis. In *UAI*.

[6] Tomoharu Iwata, Takeshi Yamada, and Naonori Ueda. 2008. Probabilistic latent semantic visualization: topic model for visualizing documents. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. 363–371.

[7] Adam Tauman Kalai, Ankur Moitra, and Gregory Valiant. 2010. Efficiently learning mixtures of two Gaussians. In *Proceedings of the forty-second ACM symposium on Theory of computing*. 553–562.

[8] 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*.

[9] Jey Han Lau, David Newman, and Timothy Baldwin. 2014. Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*. 530–539.

[10] Tuan MV Le and Hady W Lauw. 2014. Manifold learning for jointly modeling topic and visualization. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*.

[11] Tuan MV Le and Hady W Lauw. 2014. Semantic visualization for spherical representation. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 1007–1016.

[12] Tuan MV Le and Hady W Lauw. 2017. Semantic visualization for short texts with word embeddings. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. 2074–2080.

[13] 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 (2017), 1–30.

[14] 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*. 165–174.

[15] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, Nov (2008), 2579–2605.

[16] Tomas Mikolov, Kai Chen, Greg S. Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. In *Proceedings of Workshop at ICLR*.

[17] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.

[18] Dat Quoc Nguyen, Richard Billingsley, Lan Du, and Mark Johnson. 2015. Improving topic models with latent feature word representations. *Transactions of the Association for Computational Linguistics* 3 (2015), 299–313.

[19] Dang Pham and Tuan Le. 2020. Auto-Encoding Variational Bayes for Inferring Topics and Visualization. In *Proceedings of the 28th International Conference on Computational Linguistics*. 5223–5234.

[20] Xuan-Hieu Phan, Le-Minh Nguyen, and Susumu Horiguchi. 2008. Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In *Proceedings of the 17th international conference on World Wide Web*. 91–100.

[21] 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* (2020).

[22] Xiaojun Quan, Chunyu Kit, Yong Ge, and Sinno Jialin Pan. 2015. Short and sparse text topic modeling via self-aggregation. In *Twenty-fourth international joint conference on artificial intelligence*.

[23] Akash Srivastava and Charles A. Sutton. 2017. Autoencoding Variational Inference For Topic Models. In *ICLR*.

[24] Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems* 28 (2015), 649–657.

[25] 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*. 2105–2114.