WORD VECTORS FOR VARIATIONAL AUTOENCODING TOPIC MODELING

Kristyn Pantoja & Patrick Ding
Department of Statistics
Texas A&M University
College Station, TX 77840, USA
{kristynp,patrickding}@stat.tamu.edu

ABSTRACT

We investigate using pretrained word vectors for initializing the parameters of autoencoded topic models. Our experiments indicate the benefits of combining topic models and word embeddings in this way are mixed. Our results suggest the importance of joint training of topics and word embeddings.

1 INTRODUCTION

Topic models are a family of unsupervised learning algorithms for learning a set of topics in a corpus, where topics are probability distributions over a vocabulary. Traditional Bayesian topic models like LDA (Blei et al., 2003) are trained through MCMC methods, or else with mean field variational inference. Inspired by the inference approaches of the variational auto-encoder literature (Kingma & Welling, 2013), more recent works have sought to amortize the variational approximations by using inference networks to scalably to learn the latent variables.

Word embedding models (Mikolov et al., 2013b;a; Bojanowski et al., 2017) aim to represent words as low dimensional vectors that reflect semantic and syntactic similarity, as measured by the distance between vectors.

Most popular topic and word embedding models are task-specific, though there have been some attempts at combining the two classes of models. Das et al. (2015) learns topic models using pretrained word embeddings by treating documents as collections of word vectors drawn from topic specific Gaussian distributions. This model takes the word vectors as given. Meanwhile in Miao et al. (2017), the topic distributions over the words are parameterized with word vectors and topic vectors that are learned during training. However these are not word vectors per se, since the objective of the model does not explicitly encourage them to capture semantic similarity.

In this report we investigated the effect of combining topic and word embedding models by learning VAE topic models with topics parameterized with word and topic vectors, as in Miao et al. (2017). We initializing these word vectors with word embeddings pretrained on various corpora. We compared the quality of the learned models and found that the benefit of pretrained word vectors for learning topics is unclear. Our results motivate a proposed model for combining word embeddings and topic models.

2 BACKGROUND

2.1 TOPIC MODELS

One of the most influential topic models is the Latent Dirichlet Allocation (LDA) model of Blei et al. (2003). The generative model for a document d is:

$$\theta_d \sim Dirichlet(\alpha) \tag{1}$$

$$z_n|\theta_d \sim Categorical(\theta_d) \tag{2}$$

$$w_n|z_n, \beta \sim Categorical(\beta_{z_n}) \tag{3}$$

where θ_d is the document distribution over topics, z_n is the topic assignment of word w_n, and $\beta_k,\ k = 1 \ldots K$ is the distribution over words in topic k. Blei et al. (2003) proposed a mean-field variational EM algorithm for learning the latent variables and parameters of the model. Griffiths & Steyvers (2004) developed a collapsed Gibbs sampler for sampling the latent topic assignments.

There is recent work on using amortized variational inference techniques to scalably learn topic models. These approaches use the variational auto-encoder (VAE) framework of Kingma & Welling (2013), applying inference networks $q_\phi(h|x)$ to learn an approximate posterior distribution for hidden variables h. In contrast to the mean field approach amortized inference learns a shared set of neural network parameters to approximate the hidden variables, which sacrifices approximation expressiveness for scalability. Learning the parameters of both the model and the inference network is efficient due to use of the reparameterization trick, which allows for efficient Monte Carlo estimation of gradients of the evidence lower bound.

The Autoencoded Variational Inference for Topic Model (AVITM) of Srivastava & Sutton (2017) is one such VAE approach to learning LDA models. This approach makes two simplifications. The first is to sum out the latent assignments z to avoid dealing with reparameterization gradients for discrete variables. This leads to a word distribution $w_n|\beta, \theta \sim Categorical(\beta\theta)$. Now θ is the only latent variable for which we need a posterior approximation. The second simplification is to impose a Laplace approximation to the Dirichlet prior on θ in the softmax basis. This allows for use of the reparameterization trick since the approximation for the prior $p(\theta|\alpha)$ is a logistic normal with mean vector μ_1 and diagonal covariance Σ_1:

$$\mu_{1k} = \log \alpha_k - \frac{1}{K} \sum_i \alpha_i \tag{4}$$

$$\Sigma_{1kk} = \frac{1}{\alpha_k}\left(1 - \frac{2}{K}\right) + \frac{1}{K^2} \sum_i \frac{1}{\alpha_i} \tag{5}$$

AVITM employs a two inference networks f_μ and f_Σ to parameterize the logistic normal variational approximation $q(\theta|w) = \mathcal{LN}(f_\mu(d,\delta), f_\Sigma(d,\delta))$, where d is a bag of words representation of a document and δ are the parameters of the inference networks. Samples from $q(\theta|d)$ for estimating reparameterization gradients can be calculated by drawing $\epsilon \sim N(0, I)$ and computing $\theta = \sigma(\mu_0 + \Sigma_0^{1/2}\epsilon)$, where $\mu_0 = f_\mu(d,\delta)$, $\Sigma_0 = f_\Sigma(d,\delta)$, and $\sigma(\cdot)$ is the softmax function. The evidence lower bound is:

$$L = \sum_{i=1}^{D} \Bigg[- \left(\frac{1}{2} \left(Tr(\Sigma_1^{-1}\Sigma_0) + (\mu_1 - \mu_0)^T \Sigma_1^{-1} (\mu_1 - \mu_0) - K + \log \frac{|\Sigma_1|}{|\Sigma_0|} \right) \right) \tag{6}$$

$$+ \mathbb{E}_{\epsilon \sim N(0,1)} \left[d_i^T \log(\sigma(\beta)\sigma(\mu_0 + \Sigma_0^{1/2}\epsilon)) \right] \Bigg] \tag{7}$$

Miao et al. (2017) introduces three different generative models, each with a different neural network parameterization of θ: the Gaussian Softmax distribution (GSM), Gaussian Stick-Breaking distribution (GSB), and the Recurrent Stick-Breaking Process (RSB). Topic distributions over words are generated by introducing word vectors v and topic vectors t and letting $\beta = \sigma(vt^T)$.

In this paper, we focus on the first structure, the Gaussian Softmax, in which $\theta = \sigma(W_1 z)$, and consider how the word embeddings from a variety of language models might affect the learned topics, β.

2.2 Word embedding

Word embeddings are vector representations of words in a vocabulary of size V that capture the words' semantic and syntactical meanings. The positions of these vectors in the latent vector space is such that cosine similarity between these word vectors can be used to indicate how distant their corresponding words are in meaning. There are a variety of unsupervised methods for learning such language models from a corpus of text data.

One of the most popular word embedding approaches is Word2Vec by Mikolov et al. (2013a) in particular, their skip-gram model. the skip-gram method looks at a window around the word of interest during training. However, the goal in this method is, for some center word and its context window, look at how well the center words representation can predict the words in the context window:

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0}\log p(w_{t+j}|w_t)$$

$$p(w_{t+j}|w_t) = \frac{\exp({v'_{w_{t+j}}}^T v_{w_t})}{\sum_{w=1}^{W}\exp({v'_w}^T v_{w_t})}$$

Here, the center word is w_t, so, given the center words representation, we want to maximize the average log probability of the words in the context, which are w_{t+j}, for all j specified by the window size, c.

Bojanowski et al. (2017) proposed Fasttext, which extends the skip-gram model to incorporate subword information. They generalize the skip gram objective as

$$\sum_{t=1}^{T}\left[\sum_{c\in\mathbb{C}_t}\ell(s(w_t, w_c)) + \sum_{n\in\mathbb{N}_{t,c}}\ell(-s(w_t, n))\right] \tag{8}$$

where word w_t predicts the presence of context words w_c in its context $\mathbb{C}_t$, $\mathbb{N}_{t,c}$ is the set of negative samples for estimating the log likelihood of the term involving w_t and w_c, $\ell(x) = \log(1 + e^{-x})$, and s is a score function for mapping pairs of word and context words to a real number. We recover the skip-gram model in the case where $s(w_t, w_c) = u_{w_t}^T v_{w_c}$.

The authors incorporate subword information by representing words as bags of character n-grams. Each n-gram g gets a vector representation $\boldsymbol{z}_g$, and the word vector u_w is the sum of its character n-gram vectors. Then the scoring function becomes

$$s(w, c) = \sum_{g\in\mathbb{G}_w}\boldsymbol{z}_g^T\boldsymbol{v}_c \tag{9}$$

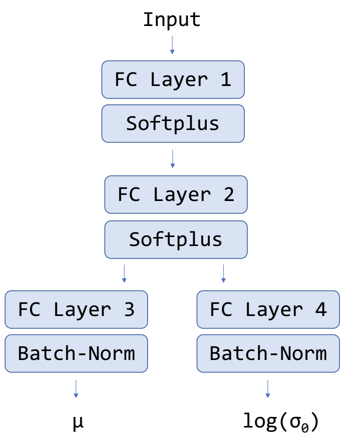
where $\mathbb{G}_w$ is the set of n-grams appearing in word w. This approach allows words to share representation through shared n-grams. Furthermore it allows estimation of word vectors for words not in the training set, since these new words and composed of the same character n-grams as words that are in the training set. The authors average the learned n-gram vectors of new words to arrive at a new word vector.

3 Model details

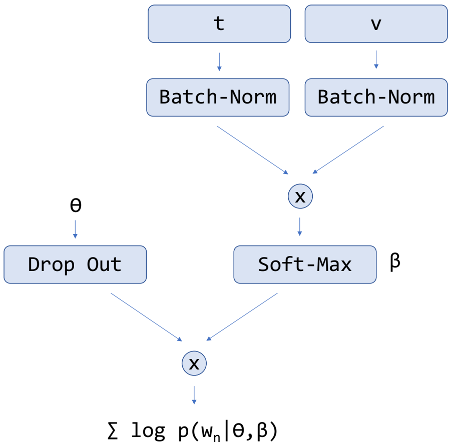
Miao et al. (2017) introduced a method that incorporated word vectors v and topic vectors t that were used to learn β in a neural network setting by letting the topic distributions $\beta = \sigma(vt^T)$, where ν is the $V \times H$ matrix of word vectors and t is the $K \times H$ is the matrix of topic vectors. In our work, we wanted to see if substituting the word vectors v with word embeddings from language models would improve the topic distributions. Our VAE topic model, which we refer to as GSM, was set up as follows:

As our input, we take the word counts for each word in the vocabulary, of size V, of a given document d in the corpus of size D. We pass it through a fully connected layer to reduce the input from dimension V to a space of smaller dimension, d_1, followed by a Softplus layer. Then we reduce the dimension further from d_1 to d_2 by passing to another fully connected layer and Softplus layer.

Similar to Srivastava & Sutton (2017), the encoder in the VAE generates θ_d, for each document d in D by drawing from the variational distribution logistic normal distribution $\mathcal{LN}(\mu(d), \Sigma_0(d))$, where $\mu(d)$ and $\log(\sigma_0(d))$ are K-vectors each learned from an inference network, and $\Sigma(d) = diag\{\sigma_0(d)\}$. To do this, we draw $\epsilon \sim \mathcal{N}(0, 1)$ and apply the reparametrization trick to get $z = \mu(d) + \epsilon e^{\sigma_0(d)}$ so that $z \sim \mathcal{N}(\mu(d), \Sigma_0(d))$. Then, we let $\theta = \sigma(z)$ so that $\theta \sim \mathcal{LN}(\mu(d), \Sigma(d))$. The architecture of the encoder is depicted in Figure 1a.



(a) The inference network.



(b) The generative model.

The per-word decoder likelihood is $\log p(w_n|\beta, \theta) = [\log \sigma(\beta)\theta]_{w_n}$, where $\beta = vt^T$. The document level likelihood is $\sum_n^N \log p(w_n|\beta, \theta)$. See Figure 1b for the architecture of the decoder.

For a baseline comparison, we use NVLDA with the same encoder. The per-word decoder for NVLDA is $\log p(w_n|\beta, \theta) = \log([\sigma(\beta\theta)]_{w_n})$.

The loss for the GSM and NVLDA models is the negative of the evidence lower bound.

As in Srivastava & Sutton (2017) we use batch-norm (Ioffe & Szegedy, 2015) and dropout layers to improve convergence.

4 Experiments

4.1 Data

We run our experiments on the dataset 20 News Groups. We removed English stop words, words fewer than three characters, non-alphabetical words, and very infrequent and frequent words. This led to a vocabulary of size 1739. We split the dataset into training and test sets, with 11314 and 7532 observations, respectively.

4.2 Hyperparameters

In our experiments, we chose the following parameters for the architectures of the GSM model and the baseline NVLDA model, see Table 1 and Table 2.

Table 1: Model parameters

Parameter	**Description**	**Value**
V	Vocabulary size	1739
K	Number of topics	50
α	Dirichlet prior parameter	1
H	Embedding dimension	300

Table 2: Encoder & decoder parameters

Parameter	**Dimension**
FC Layer 1	$\mathbb{R}^V \to \mathbb{R}^{100}$
FC Layer 2	$\mathbb{R}^{100} \to \mathbb{R}^{100}$
FC Layer 3	$\mathbb{R}^{100} \to \mathbb{R}^K$
FC Layer 4	$\mathbb{R}^{100} \to \mathbb{R}^K$
v	$\mathbb{R}^{V \times H}$
t	$\mathbb{R}^{K \times H}$

4.3 Optimization

We used the ADAM optimizer of Kingma & Ba (2014) with hyperparameters set according to the settings of Srivastava & Sutton (2017).

4.4 Pretrained word embeddings

We pretrained word embeddings in the VAE topic models in the following ways:

1. Training word2vec skip-gram model on 20newsgroup data
2. Training word2vec cbow model on 20newsgroup data
3. Training fasttext skip-gram model on 20newsgroup data
4. Training fasttext cbow model on 20newsgroup data
5. Using pretrained fastext skip-gram model on Wikipedia dataset

For each of these pretraining methods we initialize v in the VAE topic model to the values of the learned word vectors. v is then learned along with the other parameters. For all models the dimension of the embeddings H is set to 300.

4.5 Results

For each combination of model and pretrained word embeddings (or no embeddings), we trained the model for 80 epochs. We repeated the training process 5 times to account for random initializations. We evaluated the quality of the learned topics for each model using the normalized pointwise mutual information (NPMI) topic coherence metric of Lau et al. (2014), averaged over the 50 topics. This metric requires choosing the top n words as input; we chose $n = 20$. We also calculated the perplexity of the models on the held-out subset of the data.

The mean coherence scores over the 5 runs are given in Table 3. We see that pretraining v using word2vec or fasttext does not lead to improved performance over not pretraining v, or even not modeling topics with word vectors at all, in terms of mean coherence score.

In contrast pretraining v with word vectors led to improved perplexity in most cases. In Table 4 we see that NVLDA, the VAE topic model without word vectors, performs worst. GSM LDA, the VAE topic model with vectors that are not pretrained, underperforms the VAE topic models that have their word vectors pretrained with the 20newsgroup data Figure 3 shows that the outperformance of word vector pretraining is consistent across model fits.

Pretraining on word vectors learned on the outside Wikipedia corpus (the model called GSM fasttext Wiki) led to very poor performance in terms of both coherence and perplexity (Table 3 and Table 4). The perplexity is so high because in some runs the model failed to converge to good optimum, which can be seen in the loss over the 80 epochs in Figure 4a.

Table 3: Comparing models' coherences. GSM LDA has word vectors which are not initialized using pretrained values. NVLDA does not have word vectors.

Model	Mean Coherence	Standard Error
GSM w2v-sg 20News	117	3.8
GSM fasttext-sg 20News	118	3.7
GSM w2v-cbow 20News	115	3.2
GSM fasttext-bow 20News	116	3.5
GSM fasttext Wiki	100	7.4
GSM	118	3.3
NVLDA	117	4.2

Table 4: Comparing models' perplexities. Smaller is better.

Model	Mean Perplexity	Standard Error
GSM w2v-sg 20News	868	1.6
GSM fasttext-sg 20News	870	1.5
GSM w2v-cbow 20News	871	1.0
GSM fasttext-bow 20News	875	1.7
GSM fasttext Wiki	2296132600	1257643168
GSM	893	1.8
NVLDA	911	1.6

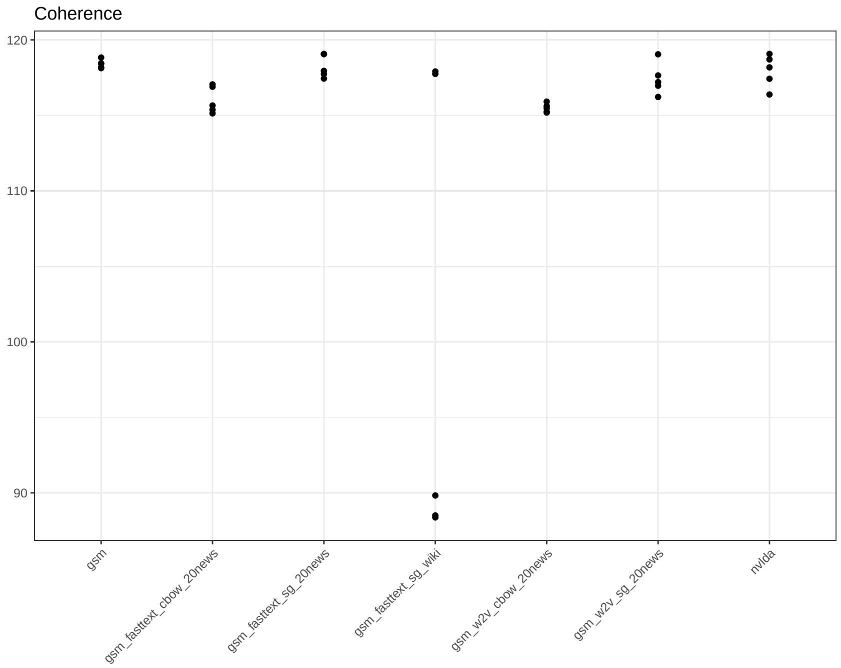


Figure 2: Normalized pointwise mutual information topic coherence metric. Larger is better.

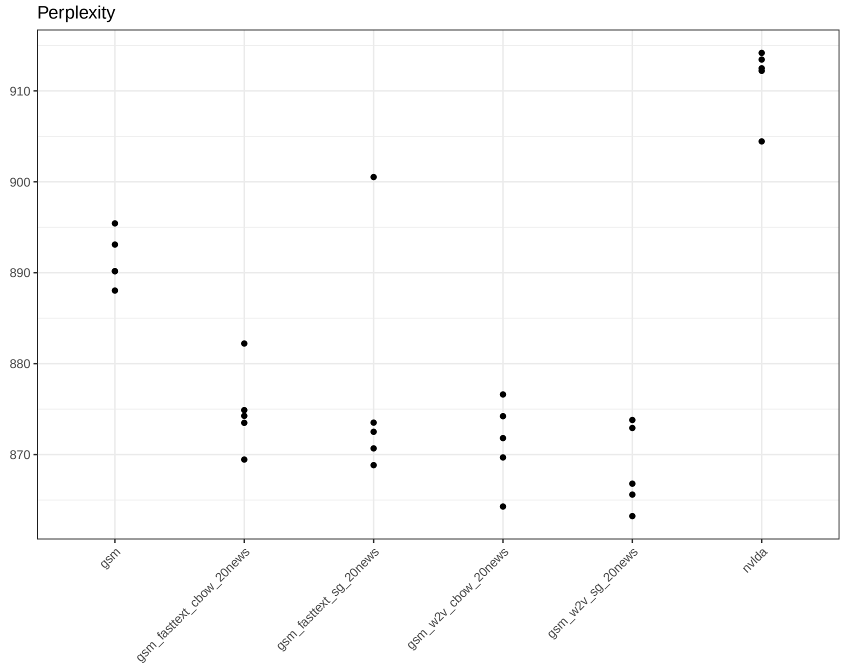
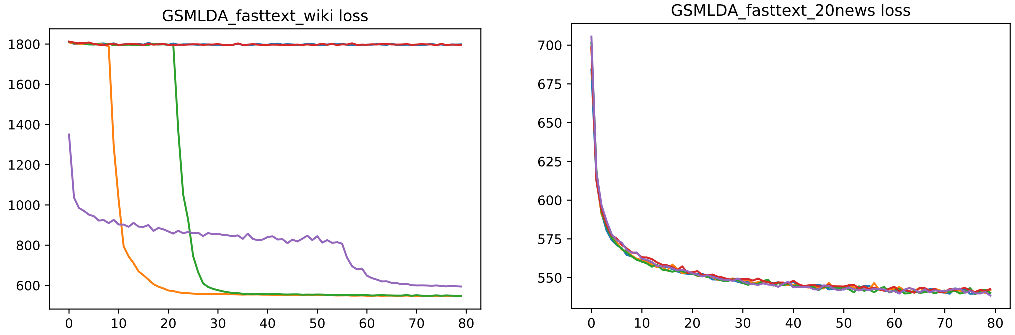


Figure 3: Perplexity on held out data. Smaller is better. We excluded the results of pretraining fasttext vectors on the outside Wikipedia dataset because for some model runs the perplexity was extremely large. See Table 4.



(a) Average loss per epoch for GSM LDA pretrained on external Wikipedia corpus.

(b) Average loss per epoch for GSM LDA pretrained on 20 News Groups. Losses for other models are similar.

5 DISCUSSION

5.1 IMPLICATIONS FOR WORD EMBEDDINGS AND TOPIC MODELS

Unfortunately pretraining word vectors does not improve coherence for VAE topic models. However, incorporating pretrained word vectors seemed to lead to better held-out perplexities. Perhaps pretraining can lead to better generalization.

Using word vectors pretrained on an external corpus led to especially poor topic learning. This isn't too surprising for coherence, since this metric is defined in terms of training corpus specific word co-occurrence frequencies. However the held-out perplexity of the model trained in this way was also very poor, and sometimes the model to failed to learn. This suggests that pretraining in this way really is detrimental for topic modeling. External semantic information seems to be harmful if it does not reflect corpus-specific word co-occurrence.

5.2 PRIOR WORK ON JOINTLY LEARNING WORD EMBEDDINGS AND TOPIC MODELS

In our experiments we did not consider other architectures for the encoder or decoder, nor did we consider other parameter choices for the optimizer. We also found that some of the models were very sensitive to initialization. In the future, we hope to explore other architectures and parameter choices, and consider reasons why the initialization is so important.

Our experimental results suggest that combining topic models and word embedding requires joint learning of topics and word vectors. Moody (2016) and Shi et al. (2017) introduce two ways to jointly learn word embeddings and latent topics.

Moody (2016) models both word embeddings and the topic distributions over the words by incorporating the skip-gram objective with the document level log likelihood into the loss. This enables learning interpretable topics as well as semantically meaningful word vectors. However no comparisons are provided with standard topic and word embedding models, so it is unclear if this approach provides an advantage versus learning the models separately.

Shi et al. (2017) introduces another method that learns word embeddings and latent topics together, in a way that addresses the issue of polysemy of words. For a model with K topics, each word has K different embedding vectors, reflecting the different meanings of the word depending on the topic. The Skip-Gram probability of context words w_{t+j} given the center word w_t is modified to include the topics of the words:

$$p(w_{t+j}|w_t, z_t, z_{t+j}) = \frac{\exp(V_{w_{t+j},z_{t+j}} \cdot U_{w_t,z_t})}{\sum_{w' \in \Lambda} \exp(V_{w',z_{t+j}} \cdot U_{w_t,z_t})} \tag{10}$$

They introduce two models, STE-Same and STE-Diff, where the STE-Same uses the Skip-Gram approach to learn the word representations for a pair of words assuming the same topic assignment z (in equation 10, $z_t = z_{t+j}$), and STE-Diff uses the Skip-Gram approach to learn the pair's word representations assuming independent topic assignments. The former leads to more coherent topics, while the latter learns better word embeddings. Unlike the typical topic representations as a list of the top n words which maximize $p(w_n|z)$, each topic is represented by a ranked list of bi-grams (w_{t+j}, w_t) which maximize $p(w_{t+j}|z, w_t)$.

5.3 PROPOSED MODEL

Ideally we would like to have meaningful word vectors as well as interpretable topics of the form $p(w|z, \beta)$. Brazinskas et al. (2017) presents a Bayesian skip gram model that would more easily incorporate topic assignments than the usual skip gram model. Their Bayesian skip gram model has the following generative model:

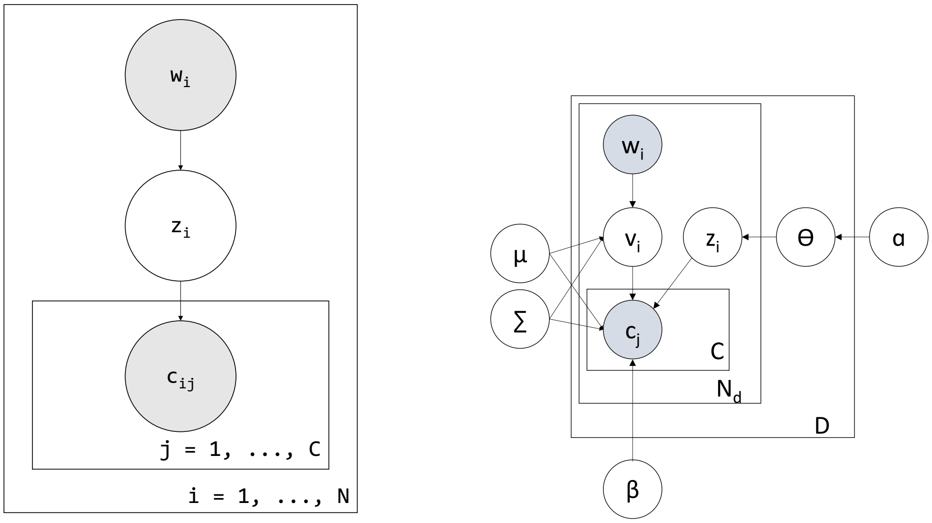
$$z_i \sim p(z|w_i) \qquad i = 1 \dots N \tag{11}$$

$$c_{ij} \sim p(c|z_i, w_i) \qquad j = 1 \dots C \tag{12}$$

where N is the number of words, C is the number of context words, z_i is the embedding vector for word i, c_{ij} is the jth context word of w_i. The z_i are drawn from normal distributions for each unique vocabulary word, and are token-level word vectors for each token in the corpus. Thus each unique vocabulary word is represented by a normal distribution. The probability of the context word given the token vector uses the unigram distribution $p(\cdot)$ in the scoring function for the skip-gram likelihood:

$$c_j|v_i \sim \frac{\text{Normal}(v_i; \mu_{c_i(j)}, \Sigma_{c_i(j)})p(c_i(j))}{\sum_{k=1}^{V} \text{Normal}(v_i; \mu_k, \Sigma_k)p(c_i(j))} \qquad j = 1, \ldots, C. \tag{13}$$

See Figure 5a for a graphical model respresentation.



(a) Bayesian skip gram of Brazinskas et al. (2017).

(b) Proposed bayesian topic skip gram model.

In our proposed model, each context word depends on the topic assigned to that context and the token vector drawn from the corresponding type's Normal distribution. The full specification of the model is:

$$\begin{aligned}
\theta_d &\sim \text{Dirichlet}(\alpha) && d = 1, \ldots, D \\
z_i|\theta_d &\sim \text{Categorical}(\theta_d) && i = 1, \ldots, N_d \\
v_i|w_i &\sim \text{Normal}(\mu_{w_i}, \Sigma_{w_i}) \\
w_i|z_i, \beta &\sim \text{Categorical}(\beta_{\cdot, z_i}) \\
c_j|v_i, z_i &\sim \frac{\text{Normal}(v_i; \mu_{c_i(j)}, \Sigma_{c_i(j)})\beta_{c_i(j), z_i}}{\sum_{k=1}^{V} \text{Normal}(v_i; \mu_k, \Sigma_k)\beta_{k, z_i}} && j = 1, \ldots, C. \qquad (13)
\end{aligned}$$

Where i is a token word in document d having context window c_i which contains tokens $c_i(j)$, for $j = 1, \ldots, C$. See Figure 5b for the graphical model representation.

This results in the following context level evidence lower bound, after marginalizing out the topic assignments z and using a Laplace approximation for the prior on θ:

$$\begin{aligned}
L(\Theta) &= \sum_{j=1}^{C} E_{q_\phi(v, \theta|c_i, w_i)}[\log p(c_i(j)|v_i, \theta)] \\
&\quad - KL(q(v|c_i, w_i)q(\theta|c_i, w_i)||p(v_i|w_i)p(\theta)) \\
p(c_i(j)|v_i, \theta) &= [\tilde{\beta} \cdot \theta]_{c_i(j)} \\
\tilde{\beta}_{c_i(j), h} &= \frac{\text{Normal}(v_i; \mu_{c_i(j)}, \Sigma_{c_i(j)})\beta_{c_i(j), h}}{\sum_{k=1}^{V} \text{Normal}(v_i; \mu_k, \Sigma_k)\beta_{k, h}} && h = 1, \ldots, K
\end{aligned}$$

We use a Monte Carlo estimate for the denominator of $\tilde{\beta}_{c_i(j), h}$.

In the second term, the KL divergence can be written as two KL divergences: one KL divergence is between two normal distributions, as in Miao et al. (2017), which has a closed form; the other KL divergence is between two logistic normal distributions, as in Brazinskas et al. (2017), which also has a closed form.

By using β instead of the unigram distribution in the scoring function (equation 13), we introduce a dependence between the word embedding and the topics so that they can be jointly learned. This will preserve the interpretability of topics in the LDA model, while also incorporating word embeddings.

References

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3(Jan):993–1022, 2003. ISSN ISSN 1533-7928. URL http://www.jmlr.org/papers/v3/blei03a.html.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, 5: 135–146, 2017. URL http://aclweb.org/anthology/Q17-1010.

Arthur Brazinskas, Serhii Havrylov, and Ivan Titov. Embedding Words as Distributions with a Bayesian Skip-gram Model. *arXiv:1711.11027 [cs]*, November 2017. URL http://arxiv.org/abs/1711.11027. arXiv: 1711.11027.

Rajarshi Das, Manzil Zaheer, and Chris Dyer. 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)*, pp. 795–804, Beijing, China, 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-1077. URL http://aclweb.org/anthology/P15-1077.

T. L. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(Supplement 1):5228–5235, April 2004. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.0307752101. URL http://www.pnas.org/cgi/doi/10.1073/pnas.0307752101.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pp. 448–456. JMLR.org, 2015. URL http://dl.acm.org/citation.cfm?id=3045118.3045167.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. URL http://dblp.uni-trier.de/db/journals/corr/corr1412.html#KingmaB14.

Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. *arXiv:1312.6114 [cs, stat]*, December 2013. URL http://arxiv.org/abs/1312.6114. arXiv: 1312.6114.

Jey Han Lau, David Newman, and Timothy Baldwin. 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*, pp. 530–539, Gothenburg, Sweden, April 2014. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/E14-1056.

Yishu Miao, Edward Grefenstette, and Phil Blunsom. Discovering Discrete Latent Topics with Neural Variational Inference. In *International Conference on Machine Learning*, pp. 2410–2419, July 2017. URL http://proceedings.mlr.press/v70/miao17a.html.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. *arXiv:1301.3781 [cs]*, January 2013a. URL http://arxiv.org/abs/1301.3781. arXiv: 1301.3781.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed Representations of Words and Phrases and their Compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger (eds.), *Advances in Neural Information Processing Systems 26*, pp. 3111–3119.

Curran Associates, Inc., 2013b. URL http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf.

Christopher E. Moody. Mixing Dirichlet Topic Models and Word Embeddings to Make lda2vec. *arXiv:1605.02019 [cs]*, May 2016. URL http://arxiv.org/abs/1605.02019. arXiv: 1605.02019.

Bei Shi, Wai Lam, Shoaib Jameel, Steven Schockaert, and Kwun Ping Lai. Jointly Learning Word Embeddings and Latent Topics. *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR '17*, pp. 375–384, 2017. doi: 10.1145/3077136.3080806. URL http://arxiv.org/abs/1706.07276. arXiv: 1706.07276.

Akash Srivastava and Charles Sutton. Autoencoding Variational Inference For Topic Models. *arXiv:1703.01488 [stat]*, March 2017. URL http://arxiv.org/abs/1703.01488. arXiv: 1703.01488.