Improving Topic Models with Latent Feature Word Representations

Dat Quoc Nguyen[1], **Richard Billingsley**[1], **Lan Du**[1] and **Mark Johnson**[1,2]
[1] Department of Computing, Macquarie University, Sydney, Australia
[2] Santa Fe Institute, Santa Fe, New Mexico, USA
dat.nguyen@students.mq.edu.au, {richard.billingsley, lan.du, mark.johnson}@mq.edu.au

Abstract

Probabilistic topic models are widely used to discover latent topics in document collections, while latent feature vector representations of words have been used to obtain high performance in many NLP tasks. In this paper, we extend two different Dirichlet multinomial topic models by incorporating latent feature vector representations of words trained on very large corpora to improve the word-topic mapping learnt on a smaller corpus. Experimental results show that by using information from the external corpora, our new models produce significant improvements on topic coherence, document clustering and document classification tasks, especially on datasets with few or short documents.

1 Introduction

Topic modeling algorithms, such as Latent Dirichlet Allocation (Blei et al., 2003) and related methods (Blei, 2012), are often used to learn a set of latent topics for a corpus, and predict the probabilities of each word in each document belonging to each topic (Teh et al., 2006; Newman et al., 2006; Toutanova and Johnson, 2008; Porteous et al., 2008; Johnson, 2010; Xie and Xing, 2013; Hingmire et al., 2013).

Conventional topic modeling algorithms such as these infer document-to-topic and topic-to-word distributions from the co-occurrence of words within documents. But when the training corpus of documents is small or when the documents are short, the resulting distributions might be based on little evidence. Sahami and Heilman (2006) and Phan et al. (2011) show that it helps to exploit external knowledge to improve the topic representations. Sahami and Heilman (2006) employed web search results to improve the information in short texts. Phan et al. (2011) assumed that the small corpus is a sample of topics from a larger corpus like Wikipedia, and then use the topics discovered in the larger corpus to help shape the topic representations in the small corpus. However, if the larger corpus has many irrelevant topics, this will "use up" the topic space of the model. In addition, Petterson et al. (2010) proposed an extension of LDA that uses external information about word similarity, such as thesauri and dictionaries, to smooth the topic-to-word distribution.

Topic models have also been constructed using latent features (Salakhutdinov and Hinton, 2009; Srivastava et al., 2013; Cao et al., 2015). Latent feature (LF) vectors have been used for a wide range of NLP tasks (Glorot et al., 2011; Socher et al., 2013; Pennington et al., 2014). The combination of values permitted by latent features forms a high dimensional space which makes it is well suited to model topics of very large corpora.

Rather than relying solely on a multinomial or latent feature model, as in Salakhutdinov and Hinton (2009), Srivastava et al. (2013) and Cao et al. (2015), we explore how to take advantage of both latent feature and multinomial models by using a latent feature representation trained on a large external corpus to supplement a multinomial topic model estimated from a smaller corpus.

Our main contribution is that we propose two new latent feature topic models which integrate latent feature word representations into two Dirichlet

multinomial topic models: a Latent Dirichlet Allocation (LDA) model (Blei et al., 2003) and a one-topic-per-document Dirichlet Multinomial Mixture (DMM) model (Nigam et al., 2000). Specifically, we replace the topic-to-word Dirichlet multinomial component which generates the words from topics in each Dirichlet multinomial topic model by a two-component mixture of a Dirichlet multinomial component and a latent feature component.

In addition to presenting a sampling procedure for the new models, we also compare using two different sets of pre-trained latent feature word vectors with our models. We achieve significant improvements on topic coherence evaluation, document clustering and document classification tasks, especially on corpora of short documents and corpora with few documents.

2 Background

2.1 LDA model

The Latent Dirichlet Allocation (LDA) topic model (Blei et al., 2003) represents each document d as a probability distribution $\boldsymbol{\theta}_d$ over topics, where each topic z is modeled by a probability distribution $\boldsymbol{\phi}_z$ over words in a fixed vocabulary W.

As presented in Figure 1, where α and β are hyper-parameters and T is number of topics, the generative process for LDA is described as follows:

$$\boldsymbol{\theta}_d \sim \mathrm{Dir}(\alpha) \qquad z_{d_i} \sim \mathrm{Cat}(\boldsymbol{\theta}_d)$$
$$\boldsymbol{\phi}_z \sim \mathrm{Dir}(\beta) \qquad w_{d_i} \sim \mathrm{Cat}(\boldsymbol{\phi}_{z_{d_i}})$$

where Dir and Cat stand for a Dirichlet distribution and a categorical distribution, and z_{d_i} is the topic indicator for the i^{th} word w_{d_i} in document d. Here, the topic-to-word Dirichlet multinomial component generates the word w_{d_i} by drawing it from the categorical distribution $\mathrm{Cat}(\boldsymbol{\phi}_{z_{d_i}})$ for topic z_{d_i}.

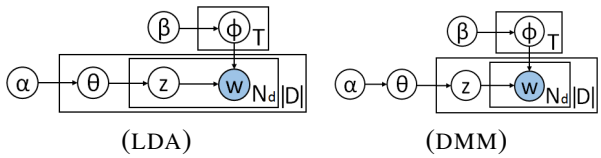


Figure 1: Graphical models of LDA and DMM

We follow the Gibbs sampling algorithm for estimating LDA topic models as described by Griffiths and Steyvers (2004). By integrating out $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$, the algorithm samples the topic z_{d_i} for the current i^{th} word w_{d_i} in document d using the conditional distribution $\mathrm{P}(z_{d_i} \mid \boldsymbol{Z}_{\neg d_i})$, where $\boldsymbol{Z}_{\neg d_i}$ denotes the topic assignments of all the other words in the document collection D, so:

$$\mathrm{P}(z_{d_i} = t \mid \boldsymbol{Z}_{\neg d_i}) \propto (N^t_{d_{\neg i}} + \alpha)\frac{N^{t,w_{d_i}}_{\neg d_i} + \beta}{N^t_{\neg d_i} + V\beta} \quad (1)$$

Notation: $N^{t,w}_d$ is the rank-3 tensor that counts the number of times that word w is generated from topic t in document d by the Dirichlet multinomial component, which in section 2.1 belongs to the LDA model, while in section 2.2 belongs to the DMM model. When an index is omitted, it indicates summation over that index (so N_d is the number of words in document d).

We write the subscript $\neg d$ for the document collection D with document d removed, and the subscript $\neg d_i$ for D with just the i^{th} word in document d removed, while the subscript $d_{\neg i}$ represents document d without its i^{th} word. For example, $N^t_{\neg d_i}$ is the number of words labelled a topic t, ignoring the i^{th} word of document d.

V is the size of the vocabulary, $V = |W|$.

2.2 DMM model for short texts

Applying topic models for short or few documents for text clustering is more challenging because of data sparsity and the limited contexts in such texts. One approach is to combine short texts into long pseudo-documents before training LDA (Hong and Davison, 2010; Weng et al., 2010; Mehrotra et al., 2013). Another approach is to assume that there is only one topic per document (Nigam et al., 2000; Zhao et al., 2011; Yin and Wang, 2014).

In the Dirichlet Multinomial Mixture (DMM) model (Nigam et al., 2000), each document is assumed to only have one topic. The process of generating a document d in the collection D, as shown in Figure 1, is to first select a topic assignment for the document, and then the topic-to-word Dirichlet multinomial component generates all the words in the document from the same selected topic:

$$\boldsymbol{\theta} \sim \mathrm{Dir}(\alpha) \qquad z_d \sim \mathrm{Cat}(\boldsymbol{\theta})$$
$$\boldsymbol{\phi}_z \sim \mathrm{Dir}(\beta) \qquad w_{d_i} \sim \mathrm{Cat}(\boldsymbol{\phi}_{z_d})$$

Yin and Wang (2014) introduced a collapsed Gibbs sampling algorithm for the DMM model in

which a topic z_d is sampled for the document d using the conditional probability $\mathrm{P}(z_d \mid \mathbf{Z}_{\neg d})$, where $\mathbf{Z}_{\neg d}$ denotes the topic assignments of all the other documents, so:

$$\mathrm{P}(z_d = t \mid \mathbf{Z}_{\neg d}) \propto (M^t_{\neg d} + \alpha) \frac{\Gamma(N^t_{\neg d} + V\beta)}{\Gamma(N^t_{\neg d} + N_d + V\beta)} \prod_{w \in W} \frac{\Gamma(N^{t,w}_{\neg d} + N^w_d + \beta)}{\Gamma(N^{t,w}_{\neg d} + \beta)} \quad (2)$$

Notation: $M^t_{\neg d}$ is the number of documents assigned to topic t excluding the current document d; Γ is the Gamma function.

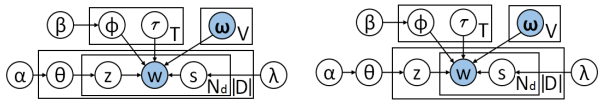
2.3 Latent feature vector models

Traditional count-based methods (Deerwester et al., 1990; Lund and Burgess, 1996; Bullinaria and Levy, 2007) for learning real-valued latent feature (LF) vectors rely on co-occurrence counts. Recent approaches based on deep neural networks learn vectors by predicting words given their window-based context (Collobert and Weston, 2008; Mikolov et al., 2013; Pennington et al., 2014; Liu et al., 2015).

Mikolov et al. (2013)'s method maximizes the log likelihood of each word given its context. Pennington et al. (2014) used back-propagation to minimize the squared error of a prediction of the log-frequency of context words within a fixed window of each word. Word vectors can be trained directly on a new corpus. In our new models, however, in order to incorporate the rich information from very large datasets, we utilize pre-trained word vectors that were trained on external billion-word corpora.

3 New latent feature topic models

In this section, we propose two novel probabilistic topic models, which we call the LF-LDA and the LF-DMM, that combine a latent feature model with either an LDA or DMM model. We also present Gibbs sampling procedures for our new models.



(LF-LDA) (LF-DMM)

Figure 2: Graphical models of our combined models

In general, LF-LDA and LF-DMM are formed by taking the original Dirichlet multinomial topic models LDA and DMM, and replacing their topic-to-word Dirichlet multinomial component that generates words from topics with a two-component mixture of a topic-to-word Dirichlet multinomial component and a latent feature component.

Informally, the new models have the structure of the original Dirichlet multinomial topic models, as shown in Figure 2, with the addition of two matrices $\boldsymbol{\tau}$ and $\boldsymbol{\omega}$ of latent feature weights, where $\boldsymbol{\tau}_t$ and $\boldsymbol{\omega}_w$ are the latent-feature vectors associated with topic t and word w respectively.

Our latent feature model defines the probability that it generates a word given the topic as the categorical distribution CatE with:

$$\mathrm{CatE}(w \mid \boldsymbol{\tau}_t \boldsymbol{\omega}^\top) = \frac{\exp(\boldsymbol{\tau}_t \cdot \boldsymbol{\omega}_w)}{\sum_{w' \in W} \exp(\boldsymbol{\tau}_t \cdot \boldsymbol{\omega}_{w'})} \quad (3)$$

CatE is a categorical distribution with log-space parameters, i.e. $\mathrm{CatE}(w \mid \boldsymbol{u}) \propto \exp(u_w)$. As $\boldsymbol{\tau}_t$ and $\boldsymbol{\omega}_w$ are (row) vectors of latent feature weights, so $\boldsymbol{\tau}_t \boldsymbol{\omega}^\top$ is a vector of "scores" indexed by words. $\boldsymbol{\omega}$ is fixed because we use pre-trained word vectors.

In the next two sections 3.1 and 3.2, we explain the generative processes of our new models LF-LDA and LF-DMM. We then present our Gibbs sampling procedures for the models LF-LDA and LF-DMM in the sections 3.3 and 3.4, respectively, and explain how we estimate $\boldsymbol{\tau}$ in section 3.5.

3.1 Generative process for the LF-LDA model

The LF-LDA model generates a document as follows: a distribution over topics $\boldsymbol{\theta}_d$ is drawn for document d; then for each i^{th} word w_{d_i} (in sequential order that words appear in the document), the model chooses a topic indicator z_{d_i}, a binary indicator variable s_{d_i} is sampled from a Bernoulli distribution to determine whether the word w_{d_i} is to be generated by the Dirichlet multinomial or latent feature component, and finally the word is generated from the chosen topic by the determined topic-to-word model. The generative process is:

$$\begin{aligned} &\boldsymbol{\theta}_d \sim \mathrm{Dir}(\alpha) && z_{d_i} \sim \mathrm{Cat}(\boldsymbol{\theta}_d) \\ &\boldsymbol{\phi}_z \sim \mathrm{Dir}(\beta) && s_{d_i} \sim \mathrm{Ber}(\lambda) \\ &w_{d_i} \sim (1 - s_{d_i})\mathrm{Cat}(\boldsymbol{\phi}_{z_{d_i}}) + s_{d_i}\mathrm{CatE}(\boldsymbol{\tau}_{z_{d_i}} \boldsymbol{\omega}^\top) \end{aligned}$$

where the hyper-parameter λ is the probability of a word being generated by the latent feature topic-to-word model and $\mathrm{Ber}(\lambda)$ is a Bernoulli distribution with success probability λ.

3.2 Generative process for the LF-DMM model

Our LF-DMM model uses the DMM model assumption that all the words in a document share the same topic. Thus, the process of generating a document in a document collection with our LF-DMM is as follows: a distribution over topics $\boldsymbol{\theta}$ is drawn for the document collection; then the model draws a topic indicator z_d for the entire document d; for every i^{th} word w_{d_i} in the document d, a binary indicator variable s_{d_i} is sampled from a Bernoulli distribution to determine whether the Dirichlet multinomial or latent feature component will be used to generate the word w_{d_i}, and finally the word is generated from the same topic z_d by the determined component. The generative process is summarized as:

$$\begin{aligned} &\boldsymbol{\theta} \sim \mathrm{Dir}(\alpha) && z_d \sim \mathrm{Cat}(\boldsymbol{\theta}) \\ &\boldsymbol{\phi}_z \sim \mathrm{Dir}(\beta) && s_{d_i} \sim \mathrm{Ber}(\lambda) \\ &w_{d_i} \sim (1 - s_{d_i})\mathrm{Cat}(\boldsymbol{\phi}_{z_d}) + s_{d_i}\mathrm{CatE}(\boldsymbol{\tau}_{z_d}\boldsymbol{\omega}^\top) \end{aligned}$$

3.3 Inference in LF-LDA model

From the generative model of LF-LDA in Figure 2, by integrating out $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$, we use the Gibbs sampling algorithm (Robert and Casella, 2004) to perform inference to calculate the conditional topic assignment probabilities for each word. The outline of the Gibbs sampling algorithm for the LF-LDA model is detailed in Algorithm 1.

Algorithm 1: An approximate Gibbs sampling algorithm for the LF-LDA model

Initialize the word-topic variables z_{d_i} using the LDA sampling algorithm
for *iteration iter = 1, 2, ...* **do**
  for *topic t = 1, 2, ..., T* **do**
    $\boldsymbol{\tau}_t = \arg\max_{\boldsymbol{\tau}_t} \mathrm{P}(\boldsymbol{\tau}_t \mid \boldsymbol{Z}, \boldsymbol{S})$
  for *document d = 1, 2, ..., |D|* **do**
    for *word index i = 1, 2, ..., N_d* **do**
      sample z_{d_i} and s_{d_i} from $\mathrm{P}(z_{d_i} = t, s_{d_i} \mid \boldsymbol{Z}_{\neg d_i}, \boldsymbol{S}_{\neg d_i}, \boldsymbol{\tau}, \boldsymbol{\omega})$

Here, $\boldsymbol{S}$ denotes the distribution indicator variables for the whole document collection D. Instead of sampling $\boldsymbol{\tau}_t$ from the posterior, we perform MAP estimation as described in the section 3.5.

For sampling the topic z_{d_i} and the binary indicator variable s_{d_i} of the i^{th} word w_{d_i} in the document d, we integrate out s_{d_i} in order to sample z_{d_i} and then sample s_{d_i} given z_{d_i}. We sample the topic z_{d_i} using the conditional distribution as follows:

$$\begin{aligned} &\mathrm{P}(z_{d_i} = t \mid \boldsymbol{Z}_{\neg d_i}, \boldsymbol{\tau}, \boldsymbol{\omega}) \\ &\propto (N^t_{d_{\neg i}} + K^t_{d_{\neg i}} + \alpha) \\ &\left((1-\lambda)\frac{N^{t,w_{d_i}}_{\neg d_i} + \beta}{N^t_{\neg d_i} + V\beta} + \lambda \mathrm{CatE}(w_{d_i} \mid \boldsymbol{\tau}_t \boldsymbol{\omega}^\top)\right) \end{aligned} \tag{4}$$

Then we sample s_{d_i} conditional on $z_{d_i} = t$ with:

$$\mathrm{P}(s_{d_i}{=}s \mid z_{d_i}{=}t) \propto \begin{cases} (1-\lambda)\frac{N^{t,w_{d_i}}_{\neg d_i} + \beta}{N^t_{\neg d_i} + V\beta} \text{ for } s = 0 \\ \lambda\ \mathrm{CatE}(w_{d_i} | \boldsymbol{\tau}_t \boldsymbol{\omega}^\top) \text{ for } s = 1 \end{cases} \tag{5}$$

Notation: Due to the new models' mixture architecture, we separate out the counts for each of the two components of each model. We define the rank-3 tensor $K^{t,w}_d$ as the number of times a word w in document d is generated from topic t by the latent feature component of the generative LF-LDA or LF-DMM model.

We also extend the earlier definition of the tensor $N^{t,w}_d$ as the number of times a word w in document d is generated from topic t by the Dirichlet multinomial component of our combined models, which in section 3.3 refers to the LF-LDA model, while in section 3.4 refers to the LF-DMM model. For both tensors K and N, omitting an index refers to summation over that index and negation $\neg$ indicates exclusion as before. So $N^w_d + K^w_d$ is the total number of times the word type w appears in the document d.

3.4 Inference in LF-DMM model

For the LF-DMM model, we integrate out $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$, and then sample the topic z_d and the distribution selection variables $\boldsymbol{s}_d$ for document d using Gibbs sampling as outlined in Algorithm 2.

Algorithm 2: An approximate Gibbs sampling algorithm for the LF-DMM model

Initialize the word-topic variables z_{d_i} using the DMM sampling algorithm
for *iteration iter = 1, 2, ...* **do**
  for *topic t = 1, 2, ..., T* **do**
    $\boldsymbol{\tau}_t = \arg\max_{\boldsymbol{\tau}_t} \mathrm{P}(\boldsymbol{\tau}_t \mid \boldsymbol{Z}, \boldsymbol{S})$
  for *document d = 1, 2, ..., |D|* **do**
    sample z_d and $\boldsymbol{s}_d$ from $\mathrm{P}(z_d = t, \boldsymbol{s}_d \mid \boldsymbol{Z}_{\neg d}, \boldsymbol{S}_{\neg d}, \boldsymbol{\tau}, \boldsymbol{\omega})$

As before in Algorithm 1, we also use MAP estimation of $\boldsymbol{\tau}$ as detailed in section 3.5 rather than

sampling from the posterior. The conditional distribution of topic variable and selection variables for document d is:

$$\begin{aligned}
&\mathrm{P}(z_d = t, \boldsymbol{s}_d \mid \boldsymbol{Z}_{\neg d}, \boldsymbol{S}_{\neg d}, \boldsymbol{\tau}, \boldsymbol{\omega}) \\
&\propto \lambda^{K_d}\,(1-\lambda)^{N_d}\,(M^t_{\neg d}+\alpha)\frac{\Gamma(N^t_{\neg d}+V\beta)}{\Gamma(N^t_{\neg d}+N_d+V\beta)} \\
&\prod_{w\in W}\frac{\Gamma(N^{t,w}_{\neg d}+N^w_d+\beta)}{\Gamma(N^{t,w}_{\neg d}+\beta)}\prod_{w\in W}\mathrm{CatE}(w \mid \boldsymbol{\tau}_t\,\boldsymbol{\omega}^\top)^{K^w_d}
\end{aligned} \tag{6}$$

Unfortunately the ratios of Gamma functions makes it difficult to integrate out $\boldsymbol{s}_d$ in this distribution P. As z_d and $\boldsymbol{s}_d$ are not independent, it is computationally expensive to directly sample from this distribution, as there are $2^{(N^w_d+K^w_d)}$ different values of $\boldsymbol{s}_d$. So we approximate P with a distribution Q that factorizes across words as follows:

$$\begin{aligned}
&Q(z_d = t, \boldsymbol{s}_d \mid \boldsymbol{Z}_{\neg d}, \boldsymbol{S}_{\neg d}, \boldsymbol{\tau}, \boldsymbol{\omega}) \\
&\propto\quad \lambda^{K_d}\,(1-\lambda)^{N_d}\,(M^t_{\neg d}+\alpha) \\
&\prod_{w\in W}\left(\frac{N^{t,w}_{\neg d}+\beta}{N^t_{\neg d}+V\beta}\right)^{N^w_d}\prod_{w\in W}\mathrm{CatE}(w \mid \boldsymbol{\tau}_t\,\boldsymbol{\omega}^\top)^{K^w_d}
\end{aligned} \tag{7}$$

This simpler distribution Q can be viewed as an approximation to P in which the topic-word "counts" are "frozen" within a document. This approximation is reasonably accurate for short documents. This distribution Q simplifies the coupling between z_d and $\boldsymbol{s}_d$. This enables us to integrate out $\boldsymbol{s}_d$ in Q. We first sample the document topic z_d for document d using $Q(z_d)$, marginalizing over s_d:

$$\begin{aligned}
&Q(z_d = t \mid \boldsymbol{Z}_{\neg d}, \boldsymbol{\tau}, \boldsymbol{\omega}) \\
&\propto\quad (M^t_{\neg d}+\alpha)\prod_{w\in W}\left(\begin{array}{l}(1-\lambda)\frac{N^{t,w}_{\neg d}+\beta}{N^t_{\neg d}+V\beta} \\ +\,\lambda\,\mathrm{CatE}(w \mid \boldsymbol{\tau}_t\,\boldsymbol{\omega}^\top)\end{array}\right)^{(N^w_d+K^w_d)}
\end{aligned} \tag{8}$$

Then we sample the binary indicator variable s_{d_i} for each i^{th} word w_{d_i} in document d conditional on $z_d = t$ from the following distribution:

$$Q(s_{d_i}{=}s \mid z_d = t) \propto \begin{cases} (1-\lambda)\frac{N^{t,w_{d_i}}_{\neg d}+\beta}{N^t_{\neg d}+V\beta} \text{ for } s = 0 \\ \lambda\,\mathrm{CatE}(w_{d_i} \mid \boldsymbol{\tau}_t\,\boldsymbol{\omega}^\top) \text{ for } s = 1 \end{cases} \tag{9}$$

3.5 Learning latent feature vectors for topics

To estimate the topic vectors after each Gibbs sampling iteration through the data, we apply regularized maximum likelihood estimation. Applying MAP estimation to learn log-linear models for topic models is also used in SAGE (Eisenstein et al., 2011) and SPRITE (Paul and Dredze, 2015). However, unlike our models, those models do not use latent feature word vectors to characterize topic-word distributions. The negative log likelihood of the corpus L under our model factorizes topic-wise into factors L_t for each topic. With L_2 regularization[1] for topic t, these are:

$$\begin{aligned}
L_t = -\sum_{w\in W}K^{t,w}\Big(\boldsymbol{\tau}_t\cdot\boldsymbol{\omega}_w - \log\big(\sum_{w'\in W}\exp(\boldsymbol{\tau}_t\cdot\boldsymbol{\omega}_{w'})\big)\Big) \\
+\,\mu\parallel\boldsymbol{\tau}_t\parallel_2^2
\end{aligned} \tag{10}$$

The MAP estimate of topic vectors $\boldsymbol{\tau}_t$ is obtained by minimizing the regularized negative log likelihood. The derivative with respect to the j^{th} element of the vector for topic t is:

$$\begin{aligned}
\frac{\partial L_t}{\partial \boldsymbol{\tau}_{t,j}} = -\sum_{w\in W}K^{t,w}\Big(\boldsymbol{\omega}_{w,j} - \sum_{w'\in W}\boldsymbol{\omega}_{w',j}\mathrm{CatE}(w' \mid \boldsymbol{\tau}_t\boldsymbol{\omega}^\top)\Big) \\
+\,2\mu\boldsymbol{\tau}_{t,j}
\end{aligned} \tag{11}$$

We used L-BFGS[2](Liu and Nocedal, 1989) to find the topic vector $\boldsymbol{\tau}_t$ that minimizes L_t.

4 Experiments

To investigate the performance of our new LF-LDA and LF-DMM models, we compared their performance against baseline LDA and DMM models on topic coherence, document clustering and document classification evaluations. The topic coherence evaluation measures the coherence of topic-word associations, i.e. it directly evaluates how coherent the assignment of words to topics is. The document clustering and document classification tasks evaluate how useful the topics assigned to documents are in clustering and classification tasks.

Because we expect our new models to perform comparatively well in situations where there is little data about topic-to-word distributions, our experiments focus on corpora with few or short documents. We also investigated which values of λ perform well, and compared the performance when using two different sets of pre-trained word vectors in these new models.

4.1 Experimental setup

4.1.1 Distributed word representations

We experimented with two state-of-the-art sets of pre-trained word vectors here.

[1]The L_2 regularizer constant was set to $\mu = 0.01$.

[2]We used the L-BFGS implementation from the Mallet toolkit (McCallum, 2002).

Google word vectors[3] are pre-trained 300-dimensional vectors for 3 million words and phrases. These vectors were trained on a 100 billion word subset of the Google News corpus by using the Google Word2Vec toolkit (Mikolov et al., 2013). Stanford vectors[4] are pre-trained 300-dimensional vectors for 2 million words. These vectors were learned from 42-billion tokens of Common Crawl web data using the Stanford GloVe toolkit (Pennington et al., 2014).

We refer to our LF-LDA and LF-DMM models using Google and Stanford word vectors as **w2v-LDA**, **glove-LDA**, **w2v-DMM** and **glove-DMM**.

4.1.2 Experimental datasets

We conducted experiments on the 20-Newsgroups dataset, the TagMyNews news dataset and the Sanders Twitter corpus.

The 20-Newsgroups dataset[5] contains about 19,000 newsgroup documents evenly grouped into 20 different categories. The TagMyNews news dataset[6] (Vitale et al., 2012) consists of about 32,600 English RSS news items grouped into 7 categories, where each news document has a news title and a short description. In our experiments, we also used a news title dataset which consists of just the news titles from the TagMyNews news dataset.

Each dataset was down-cased, and we removed non-alphabetic characters and stop-words found in the stop-word list in the Mallet toolkit (McCallum, 2002). We also removed words shorter than 3 characters and words appearing less than 10 times in the 20-Newsgroups corpus, and under 5 times in the TagMyNews news and news titles datasets. In addition, words not found in both Google and Stanford vector representations were also removed.[7] We refer to the cleaned 20-Newsgroups, TagMyNews news and news title datasets as **N20**, **TMN** and **TMNtitle**, respectively.

We also performed experiments on two subsets of the N20 dataset. The **N20short** dataset consists of all documents from the N20 dataset with less than 21 words. The **N20small** dataset contains 400 documents consisting of 20 randomly selected documents from each group of the N20 dataset.

Dataset	#g	#docs	#w/d	V
N20	20	18,820	103.3	19,572
N20short	20	1,794	13.6	6,377
N20small	20	400	88.0	8,157
TMN	7	32,597	18.3	13,428
TMNtitle	7	32,503	4.9	6,347
Twitter	4	2,520	5.0	1,390

Table 1: Details of experimental datasets. #g: number of ground truth labels; #docs: number of documents; #w/d: the average number of words per document; V: the number of word types

Finally, we also experimented on the publicly available Sanders Twitter corpus.[8] This corpus consists of 5,512 Tweets grouped into four different topics (Apple, Google, Microsoft, and Twitter). Due to restrictions in Twitter's Terms of Service, the actual Tweets need to be downloaded using 5,512 Tweet IDs. There are 850 Tweets not available to download. After removing the non-English Tweets, 3,115 Tweets remain. In addition to converting into lowercase and removing non-alphabetic characters, words were normalized by using a lexical normalization dictionary for microblogs (Han et al., 2012). We then removed stop-words, words shorter than 3 characters or appearing less than 3 times in the corpus. The four words *apple*, *google*, *microsoft* and *twitter* were removed as these four words occur in every Tweet in the corresponding topic. Moreover, words not found in both Google and Stanford vector lists were also removed.[9] In all our experiments, after removing words from documents, any document with a zero word count was also removed from the corpus. For the Twitter corpus, this resulted in just 2,520 remaining Tweets.

4.1.3 General settings

The hyper-parameter β used in baseline LDA and DMM models was set to 0.01, as this is a common setting in the literature (Griffiths and Steyvers,

[3] Download at: https://code.google.com/p/word2vec/

[4] Download at: http://www-nlp.stanford.edu/projects/glove/

[5] We used the "all-terms" version of the 20-Newsgroups dataset available at http://web.ist.utl.pt/acardoso/datasets/ (Cardoso-Cachopo, 2007).

[6] The TagMyNews news dataset is unbalanced, where the largest category contains 8,200 news items while the smallest category contains about 1,800 items. Download at: http://acube.di.unipi.it/tmn-dataset/

[7] 1366, 27 and 12 words were correspondingly removed out of the 20-Newsgroups, TagMyNews news and news title datasets.

[8] Download at: http://www.sananalytics.com/lab/index.php

[9] There are 91 removed words.

2004). We set the hyper-parameter $\alpha = 0.1$, as this can improve performance relative to the standard setting $\alpha = \frac{50}{T}$, as noted by Lu et al. (2011) and Yin and Wang (2014).

We ran each baseline model for 2000 iterations and evaluated the topics assigned to words in the last sample. For our models, we ran the baseline models for 1500 iterations, then used the outputs from the last sample to initialize our models, which we ran for 500 further iterations.

We report the mean and standard deviation of the results of ten repetitions of each experiment (so the standard deviation is approximately 3 standard errors, or a 99% confidence interval).

4.2 Topic coherence evaluation

This section examines the quality of the topic-word mappings induced by our models. In our models, topics are distributions over words. The topic coherence evaluation measures to what extent the high-probability words in each topic are semantically coherent (Chang et al., 2009; Stevens et al., 2012).

4.2.1 Quantitative analysis

Newman et al. (2010), Mimno et al. (2011) and Lau et al. (2014) describe methods for automatically evaluating the semantic coherence of sets of words. The method presented in Lau et al. (2014) uses the normalized pointwise mutual information (NPMI) score and has a strong correlation with human-judged coherence. A higher NPMI score indicates that the topic distributions are semantically more coherent. Given a topic t represented by its top-N topic words $w_1, w_2, ..., w_N$, the NPMI score for t is:

$$\text{NPMI-Score}(t) = \sum_{1 \leqslant i < j \leqslant N} \frac{\log \frac{\mathrm{P}(w_i, w_j)}{\mathrm{P}(w_i)\mathrm{P}(w_j)}}{-\log \mathrm{P}(w_i, w_j)} \qquad (12)$$

where the probabilities in equation (12) are derived from a 10-word sliding window over an external corpus.

The NPMI score for a topic model is the average score for all topics. We compute the NPMI score based on top-15 most probable words of each topic and use the English Wikipedia[10] of 4.6 million articles as our external corpus.

Figures 3 and 4 show NPMI scores computed for the LDA, w2v-LDA and glove-LDA models on the N20short dataset for 20 and 40 topics. We see that $\lambda = 1.0$ gives the highest NPMI score. In other words, using only the latent feature model produces the most coherent topic distributions.

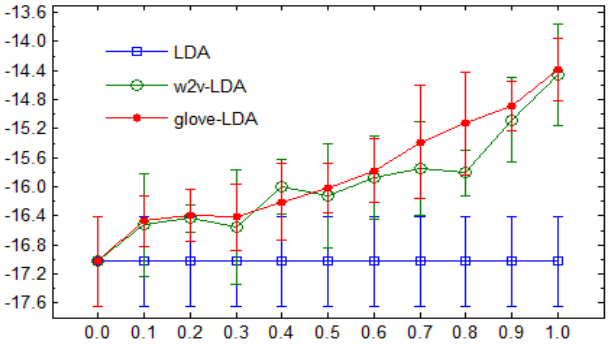


Figure 3: NPMI scores (mean and standard deviation) on the N20short dataset with 20 topics, varying the mixture weight λ from 0.0 to 1.0.

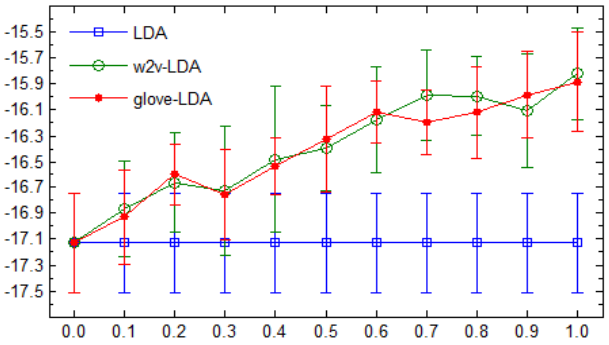


Figure 4: NPMI scores on the N20short dataset with 40 topics, varying the mixture weight λ from 0.0 to 1.0.

Data	Method	$\lambda = 1.0$			
		T=6	T=20	T=40	T=80
N20	LDA	-16.7 ± 0.9	-11.7 ± 0.7	-11.5 ± 0.3	-11.4 ± 0.4
	w2v-LDA	-14.5 ± 1.2	-9.0 ± 0.8	-10.0 ± 0.5	-10.7 ± 0.4
	glove-LDA	**-11.6** ± 0.8	**-7.4** ± 1.0	**-8.3** ± 0.7	**-9.7** ± 0.4
	Improve.	5.1	4.3	3.2	1.7
N20small	LDA	-18.4 ± 0.6	-16.7 ± 0.6	-17.8 ± 0.4	-17.6 ± 0.3
	w2v-LDA	**-12.0** ± 1.1	**-12.7** ± 0.7	-15.5 ± 0.4	**-16.3** ± 0.3
	glove-LDA	-13.0 ± 1.1	-12.8 ± 0.7	**-15.0** ± 0.5	-16.6 ± 0.2
	Improve.	6.4	4.0	2.8	1.3

Table 2: NPMI scores (mean and standard deviation) for N20 and N20small datasets. The *Improve.* row denotes the absolute improvement accounted for the best result produced by our latent feature model over the baselines.

Tables 2, 3 and 4 present the NPMI scores produced by the models on the other experimental datasets, where we vary[11] the number of topics in steps from 4 to 80. Tables 3 and 4 show that the DMM model performs better than the LDA model on

[10] We used the Wikipedia-articles dump of July 8, 2014.

[11] We perform with $T = 6$ on the N20 and N20small datasets as the 20-Newsgroups dataset could be also grouped into 6 larger topics instead of 20 fine-grained categories.

Data	Method	$\lambda = 1.0$			
		T=7	T=20	T=40	T=80
TMN	LDA	-17.3 ± 1.1	-12.7 ± 0.8	-12.3 ± 0.5	-13.0 ± 0.3
	w2v-LDA	-14.7 ± 1.5	-12.8 ± 0.8	-12.2 ± 0.5	-13.1 ± 0.2
	glove-LDA	**-13.0** ± 1.8	**-9.7** ± 0.7	**-11.5** ± 0.5	**-12.9** ± 0.4
	Improve.	4.3	3.0	0.8	0.1
TMN	DMM	-17.4 ± 1.5	-12.2 ± 1.0	-10.6 ± 0.6	-11.2 ± 0.4
	w2v-DMM	**-11.5** ± 1.6	-7.0 ± 0.7	**-5.8** ± 0.5	**-5.8** ± 0.3
	glove-DMM	-13.4 ± 1.5	**-6.2** ± 1.2	-6.6 ± 0.5	-6.3 ± 0.5
	Improve.	5.9	6.0	4.8	5.4
TMNtitle	LDA	-17.2 ± 0.8	-15.4 ± 0.7	-15.3 ± 0.3	-15.6 ± 0.3
	w2v-LDA	-14.2 ± 1.0	-14.0 ± 0.7	**-15.0** ± 0.3	**-14.9** ± 0.4
	glove-LDA	**-13.9** ± 0.9	**-13.4** ± 0.7	-15.2 ± 0.5	-15.2 ± 0.2
	Improve.	3.3	2.0	0.3	0.7
TMNtitle	DMM	-16.5 ± 0.9	-13.6 ± 1.0	-13.1 ± 0.5	-13.7 ± 0.3
	w2v-DMM	**-9.6** ± 0.6	**-7.5** ± 0.8	**-8.1** ± 0.4	-9.7 ± 0.4
	glove-DMM	-10.9 ± 1.3	-8.1 ± 0.5	**-8.1** ± 0.5	**-9.1** ± 0.3
	Improve.	5.6	6.1	5.0	4.6

Table 3: NPMI scores for TMN and TMNtitle datasets.

Data	Method	$\lambda = 1.0$			
		T=4	T=20	T=40	T=80
Twitter	LDA	-8.5 ± 1.1	-14.5 ± 0.4	-15.1 ± 0.4	-15.9 ± 0.2
	w2v-LDA	-7.3 ± 1.0	**-13.2** ± 0.6	**-14.0** ± 0.3	**-14.1** ± 0.3
	glove-LDA	**-6.2** ± 1.6	-13.9 ± 0.6	-14.2 ± 0.4	-14.2 ± 0.2
	Improve.	2.3	1.3	1.1	1.8
Twitter	DMM	-5.9 ± 1.1	-10.4 ± 0.7	-12.0 ± 0.3	-13.3 ± 0.3
	w2v-DMM	-5.5 ± 0.7	-10.5 ± 0.5	-11.2 ± 0.5	**-12.5** ± 0.1
	glove-DMM	**-5.1** ± 1.2	**-9.9** ± 0.6	**-11.1** ± 0.3	**-12.5** ± 0.4
	Improve.	0.8	0.5	0.9	0.8

Table 4: NPMI scores for Twitter dataset.

the TMN, TMNtitle and Twitter datasets. These results show that our latent feature models produce significantly higher scores than the baseline models on all the experimental datasets.

Google word2vec vs. Stanford glove word vectors: In general, our latent feature models obtain competitive NPMI results in using pre-trained Google word2vec and Stanford glove word vectors for a large value of T, for example $T = 80$. With small values of T, for example $T \leq 7$, using Google word vectors produces better scores than using Stanford word vectors on the small N20small dataset of normal texts and on the short text TMN and TMNtitle datasets. However, the opposite pattern holds on the full N20 dataset. Both sets of the pre-trained word vectors produce similar scores on the small and short Twitter dataset.

4.2.2 Qualitative analysis

This section provides an example of how our models improve topic coherence. Table 5 compares the top-15 words[12] produced by the baseline DMM model and our w2v-DMM model with $\lambda = 1.0$ on the TMNtitle dataset with $T = 20$ topics.

In table 5, topic 1 of the DMM model consists of words related to "nuclear crisis in Japan" together with other unrelated words. The w2v-DMM model produced a purer topic 1 focused on "Japan earthquake and nuclear crisis," presumably related to the "Fukushima Daiichi nuclear disaster." Topic 3 is about "oil prices" in both models. However, all top-15 words are qualitatively more coherent in the w2v-DMM model. While topic 4 of the DMM model is difficult to manually label, topic 4 of the w2v-DMM model is about the "Arab Spring" event.

Topics 5, 19 and 14 of the DMM model are not easy to label. Topic 5 relates to "entertainment", topic 19 is generally a mixture of "entertainment" and "sport", and topic 14 is about "sport" and "politics." However, the w2v-DMM model more clearly distinguishes these topics: topic 5 is about "entertainment", topic 19 is only about "sport" and topic 14 is only about "politics."

4.3 Document clustering evaluation

We compared our models to the baseline models in a document clustering task. After using a topic model to calculate the topic probabilities of a document, we assign every document the topic with the highest probability given the document (Cai et al., 2008; Lu et al., 2011; Xie and Xing, 2013; Yan et al., 2013). We use two common metrics to evaluate clustering performance: *Purity* and *normalized mutual information* (NMI): see (Manning et al., 2008, Section 16.3) for details of these evaluations. Purity and NMI scores always range from 0.0 to 1.0, and higher scores reflect better clustering performance.

Figures 5 and 6 present Purity and NMI results obtained by the LDA, w2v-LDA and glove-LDA models on the N20short dataset with the numbers of topics T set to either 20 or 40, and the value of the mixture weight λ varied from 0.0 to 1.0.

We found that setting λ to 1.0 (i.e. using only the latent features to model words), the glove-LDA produced 1%+ higher scores on both Purity and NMI results than the w2v-LDA when using 20 topics. However, the two models glove-LDA and w2v-LDA returned equivalent results with 40 topics where they

[12] In the baseline model, the top-15 topical words output from the 1500^{th} sample are similar to top-15 words from the 2000^{th} sample if we do not take the order of the most probable words into account.

Topic 1									Topic 3		
InitDMM	Iter=1	Iter=2	Iter=5	Iter=10	Iter=20	Iter=50	Iter=100	Iter=500	InitDMM	Iter=50	Iter=500
japan	japan	japan	japan	japan	japan	japan	japan	japan	u.s.	prices	prices
nuclear	nuclear	nuclear	nuclear	nuclear	nuclear	nuclear	nuclear	nuclear	oil	sales	sales
u.s.	u.s.	u.s.	u.s.	u.s.	u.s.	plant	u.s.	u.s.	japan	oil	oil
crisis	**russia**	crisis	plant	plant	plant	u.s.	plant	plant	prices	u.s.	u.s.
plant	radiation	china	crisis	radiation	**quake**	**quake**	**quake**	**quake**	stocks	stocks	profit
china	**nuke**	**russia**	radiation	crisis	radiation	radiation	radiation	radiation	sales	profit	stocks
libya	**iran**	plant	china	china	crisis	**earthquake**	**earthquake**	**earthquake**	profit	japan	japan
radiation	crisis	radiation	**russia**	**nuke**	**nuke**	**tsunami**	**tsunami**	**tsunami**	fed	rise	rise
u.n.	china	**nuke**	**nuke**	**russia**	china	**nuke**	**nuke**	**nuke**	rise	**gas**	**gas**
vote	libya	libya	**power**	**power**	**tsunami**	crisis	crisis	crisis	growth	growth	growth
korea	plant	**iran**	u.n.	u.n.	**earthquake**	**disaster**	**disaster**	**disaster**	wall	**profits**	shares
europe	u.n.	u.n.	**iran**	**iran**	**disaster**	**plants**	**oil**	**power**	street	shares	**price**
government	**mideast**	**power**	**reactor**	**earthquake**	**power**	**power**	**plants**	**oil**	china	**price**	**profits**
election	**pakistan**	**pakistan**	**earthquake**	**reactor**	**reactor**	**oil**	**power**	**japanese**	fall	**rises**	**rises**
deal	**talks**	**talks**	libya	**quake**	**japanese**	**japanese**	**tepco**	**plants**	shares	**earnings**	**earnings**

Topic 4			Topic 5			Topic 19			Topic 14		
InitDMM	Iter=50	Iter=500	InitDMM	Iter=50	Iter=500	InitDMM	Iter=50	Iter=500	InitDMM	Iter=50	Iter=500
egypt	libya	libya	critic	**dies**	star	nfl	nfl	nfl	nfl	law	law
china	egypt	egypt	corner	star	sheen	idol	draft	**sports**	court	bill	texas
u.s.	**mideast**	**iran**	office	**broadway**	**idol**	draft	**lockout**	draft	law	governor	bill
mubarak	**iran**	**mideast**	video	**american**	**broadway**	american	players	players	bill	texas	governor
bin	**opposition**	**opposition**	game	**idol**	show	show	**coach**	**lockout**	wisconsin	senate	senate
libya	**leader**	**protests**	star	lady	**american**	film	**nba**	**football**	players	union	union
laden	u.n.	**leader**	lady	gaga	gaga	season	**player**	**league**	judge	**obama**	**obama**
france	**protests**	**syria**	gaga	show	**tour**	sheen	sheen	n.f.l.	governor	wisconsin	budget
bahrain	**syria**	u.n.	show	**news**	**cbs**	n.f.l.	**league**	**player**	union	budget	wisconsin
air	**tunisia**	**tunisia**	weekend	critic	**hollywood**	back	n.f.l.	**baseball**	house	**state**	**immigration**
report	**protesters**	**chief**	sheen	**film**	**mtv**	top	**coaches**	**court**	texas	**immigration**	**state**
rights	**chief**	**protesters**	box	**hollywood**	lady	star	**football**	**coaches**	lockout	**arizona**	**vote**
court	**asia**	mubarak	park	**fame**	**wins**	charlie	**judge**	**nflpa**	budget	**california**	**washington**
u.n.	**russia**	**crackdown**	takes	**actor**	**charlie**	players	**nflpa**	**basketball**	peru	**vote**	**arizona**
war	**arab**	bahrain	man	**movie**	**stars**	men	**court**	**game**	senate	**federal**	**california**

Table 5: Examples of the 15 most probable topical words on the TMNtitle dataset with $T = 20$. InitDMM denotes the output from the 1500^{th} sample produced by the DMM model, which we use to initialize the w2v-DMM model. Iter=1, Iter=2, Iter=3 and the like refer to the output of our w2v-DMM model after running 1, 2, 3 sampling iterations, respectively. The words found in InitDMM and not found in Iter=500 are underlined. Words found by the w2v-DMM model but not found by the DMM model are in **bold**.

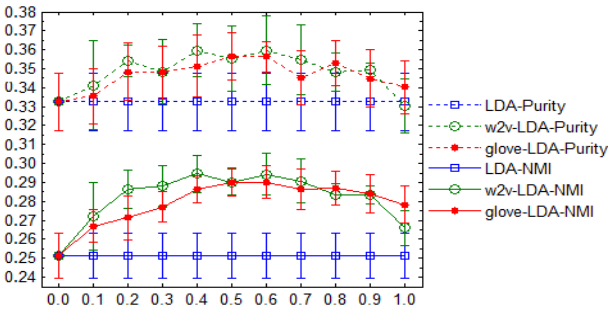


Figure 5: Purity and NMI results (mean and standard deviation) on the N20short dataset with number of topics T = 20, varying the mixture weight λ from 0.0 to 1.0.

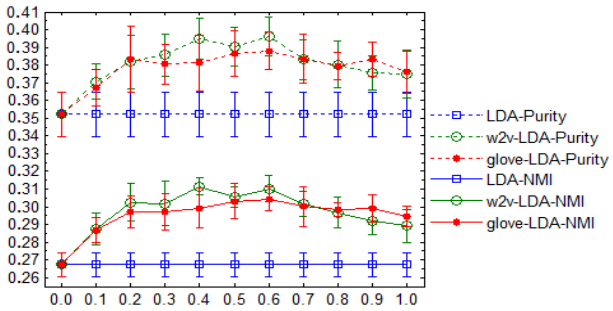


Figure 6: Purity and NMI results on the N20short dataset with number of topics $T = 40$, varying the mixture weight λ from 0.0 to 1.0.

gain 2%+ absolute improvement[13] on the two Purity and NMI against the baseline LDA model.

By varying λ, as shown in Figures 5 and 6, the w2v-LDA and glove-LDA models obtain their best results at $\lambda = 0.6$ where the w2v-LDA model does slightly better than the glove-LDA. Both models significantly outperform their baseline LDA models; for example with 40 topics, the w2v-LDA model attains 4.4% and 4.3% over the LDA model on Purity and NMI metrics, respectively.

We fix the mixture weight λ at 0.6, and report experimental results based on this value for the rest of this section. Tables 6, 7 and 8 show clustering results produced by our models and the baseline models on the remaining datasets with different numbers

[13]Using the Student's t-Test, the improvement is significant ($p < 0.01$).

Data	Method	Purity				NMI			
		T=6	T=20	T=40	T=80	T=6	T=20	T=40	T=80
N20	LDA	0.293 ± 0.002	0.573 ± 0.019	**0.639** ± 0.017	**0.646** ± 0.005	0.516 ± 0.009	0.582 ± 0.009	**0.557** ± 0.007	**0.515** ± 0.003
	w2v-LDA	0.291 ± 0.002	0.569 ± 0.021	0.616 ± 0.017	0.638 ± 0.006	0.500 ± 0.008	0.563 ± 0.009	0.535 ± 0.008	0.505 ± 0.004
	glove-LDA	**0.295** ± 0.001	**0.604** ± 0.031	0.632 ± 0.017	0.638 ± 0.007	**0.522** ± 0.003	**0.596** ± 0.012	0.550 ± 0.010	0.507 ± 0.003
	Improve.	0.002	0.031	-0.007	-0.008	0.006	0.014	-0.007	-0.008
N20small	LDA	0.232 ± 0.011	0.408 ± 0.017	0.477 ± 0.015	0.559 ± 0.018	0.376 ± 0.016	0.474 ± 0.013	0.513 ± 0.009	0.563 ± 0.008
	w2v-LDA	0.229 ± 0.005	**0.439** ± 0.015	**0.516** ± 0.024	**0.595** ± 0.016	0.406 ± 0.023	**0.519** ± 0.014	**0.548** ± 0.017	**0.585** ± 0.009
	glove-LDA	**0.235** ± 0.008	0.427 ± 0.022	0.492 ± 0.022	0.579 ± 0.011	**0.436** ± 0.019	0.504 ± 0.020	0.527 ± 0.013	0.576 ± 0.006
	Improve.	0.003	0.031	0.039	0.036	0.06	0.045	0.035	0.022

Table 6: Purity and NMI results (mean and standard deviation) on the N20 and N20small datasets with $\lambda = 0.6$. *Improve.* row denotes the difference between the best result obtained by our model and the baseline model.

Data	Method	Purity				NMI			
		T=7	T=20	T=40	T=80	T=7	T=20	T=40	T=80
TMN	LDA	0.648 ± 0.029	0.717 ± 0.009	**0.721** ± 0.003	0.719 ± 0.007	0.436 ± 0.019	0.393 ± 0.008	0.354 ± 0.003	0.320 ± 0.003
	w2v-LDA	**0.658** ± 0.020	0.716 ± 0.012	0.720 ± 0.008	**0.725** ± 0.004	0.446 ± 0.014	0.399 ± 0.006	0.355 ± 0.005	**0.325** ± 0.003
	glove-LDA	**0.658** ± 0.034	**0.722** ± 0.007	0.719 ± 0.008	**0.725** ± 0.006	**0.448** ± 0.017	**0.403** ± 0.004	**0.356** ± 0.004	0.324 ± 0.004
	Improve.	0.01	0.005	-0.001	0.006	0.012	0.01	0.002	0.005
TMN	DMM	0.637 ± 0.029	0.699 ± 0.015	0.707 ± 0.014	0.715 ± 0.009	0.445 ± 0.024	0.422 ± 0.007	0.393 ± 0.009	0.364 ± 0.006
	w2v-DMM	0.623 ± 0.020	0.737 ± 0.018	**0.760** ± 0.010	0.772 ± 0.005	0.426 ± 0.015	0.428 ± 0.009	0.405 ± 0.006	0.378 ± 0.003
	glove-DMM	**0.641** ± 0.042	**0.749** ± 0.011	0.758 ± 0.008	**0.776** ± 0.006	**0.449** ± 0.028	**0.441** ± 0.008	**0.408** ± 0.005	**0.381** ± 0.003
	Improve.	0.004	0.05	0.053	0.061	0.004	0.019	0.015	0.017
TMNtitle	LDA	0.572 ± 0.014	0.599 ± 0.015	0.593 ± 0.011	0.580 ± 0.006	0.314 ± 0.008	0.262 ± 0.006	0.228 ± 0.006	0.196 ± 0.003
	w2v-LDA	0.579 ± 0.020	0.619 ± 0.015	**0.611** ± 0.007	0.598 ± 0.004	0.321 ± 0.012	0.279 ± 0.006	**0.239** ± 0.005	**0.210** ± 0.002
	glove-LDA	**0.584** ± 0.026	**0.623** ± 0.012	0.600 ± 0.008	**0.601** ± 0.004	**0.322** ± 0.015	**0.280** ± 0.004	0.235 ± 0.006	0.209 ± 0.003
	Improve.	0.012	0.024	0.018	0.021	0.008	0.018	0.011	0.014
TMNtitle	DMM	0.558 ± 0.015	0.600 ± 0.010	0.634 ± 0.011	0.658 ± 0.006	0.338 ± 0.012	0.327 ± 0.006	0.304 ± 0.004	0.271 ± 0.002
	w2v-DMM	0.552 ± 0.022	0.653 ± 0.012	0.678 ± 0.007	0.682 ± 0.005	0.314 ± 0.016	0.325 ± 0.006	0.305 ± 0.004	**0.282** ± 0.003
	glove-DMM	**0.586** ± 0.019	**0.672** ± 0.013	**0.679** ± 0.009	**0.683** ± 0.004	**0.343** ± 0.015	**0.339** ± 0.007	**0.307** ± 0.004	**0.282** ± 0.002
	Improve.	0.028	0.072	0.045	0.025	0.005	0.012	0.003	0.011

Table 7: Purity and NMI results on the TMN and TMNtitle datasets with the mixture weight $\lambda = 0.6$.

Data	Method	Purity				NMI			
		T=4	T=20	T=40	T=80	T=4	T=20	T=40	T=80
Twitter	LDA	0.559 ± 0.020	0.614 ± 0.016	0.626 ± 0.011	0.631 ± 0.008	0.196 ± 0.018	0.174 ± 0.008	0.170 ± 0.007	0.160 ± 0.004
	w2v-LDA	**0.598** ± 0.023	**0.635** ± 0.016	**0.638** ± 0.009	**0.637** ± 0.012	**0.249** ± 0.021	**0.191** ± 0.011	0.176 ± 0.003	**0.167** ± 0.006
	glove-LDA	0.597 ± 0.016	**0.635** ± 0.014	0.637 ± 0.010	**0.637** ± 0.007	0.242 ± 0.013	**0.191** ± 0.007	**0.177** ± 0.007	0.165 ± 0.005
	Improve.	0.039	0.021	0.012	0.006	0.053	0.017	0.007	0.007
Twitter	DMM	0.523 ± 0.011	0.619 ± 0.015	0.660 ± 0.008	0.684 ± 0.010	0.222 ± 0.013	0.213 ± 0.011	0.198 ± 0.008	0.196 ± 0.004
	w2v-DMM	**0.589** ± 0.017	0.655 ± 0.015	**0.668** ± 0.008	0.694 ± 0.009	0.243 ± 0.014	0.215 ± 0.009	**0.203** ± 0.005	0.204 ± 0.006
	glove-DMM	0.583 ± 0.023	**0.661** ± 0.019	0.667 ± 0.009	**0.697** ± 0.009	**0.250** ± 0.020	**0.223** ± 0.014	0.201 ± 0.006	**0.206** ± 0.005
	Improve.	0.066	0.042	0.008	0.013	0.028	0.01	0.005	0.01

Table 8: Purity and NMI results on the Twitter dataset with the mixture weight $\lambda = 0.6$.

of topics. As expected, the DMM model is better than the LDA model on the short datasets of TMN, TMNtitle and Twitter. For example with 80 topics on the TMNtitle dataset, the DMM achieves about 7+% higher Purity and NMI scores than LDA.

New models vs. baseline models: On most tests, our models score higher than the baseline models, particularly on the small N20small dataset where we get 6.0% improvement on NMI at $T = 6$, and on the short text TMN and TMNtitle datasets we obtain 6.1% and 2.5% higher Purity at $T = 80$. In addition, on the short and small Twitter dataset with $T = 4$, we achieve 3.9% and 5.3% improvements in Purity and NMI scores, respectively. Those results show that an improved model of topic-word mappings also improves the document-topic assignments.

For the small value of $T \leq 7$, on the large datasets of N20, TMN and TMNtitle, our models and baseline models obtain similar clustering results. However, with higher values of T, our models perform better than the baselines on the short TMN and TMNtitle datasets, while on the N20 dataset, the baseline LDA model attains a slightly higher clustering results than ours. In contrast, on the short and small Twitter dataset, our models obtain considerably better clustering results than the baseline models with a small value of T.

Google word2vec vs. Stanford glove word vectors: On the small N20short and N20small datasets, using the Google pre-trained word vectors produces

higher clustering scores than using Stanford pre-trained word vectors. However, on the large datasets N20, TMN and TMNtitle, using Stanford word vectors produces higher scores than using Google word vectors when using a smaller number of topics, for example $T \leq 20$. With more topics, for instance $T = 80$, the pre-trained Google and Stanford word vectors produce similar clustering results. In addition, on the Twitter dataset, both sets of pre-trained word vectors produce similar results.

4.4 Document classification evaluation

Unlike the document clustering task, the document classification task evaluates the distribution over topics for each document. Following Lacoste-Julien et al. (2009), Lu et al. (2011), Huh and Fienberg (2012) and Zhai and Boyd-graber (2013), we used Support Vector Machines (SVM) to predict the ground truth labels from the topic-proportion vector of each document. We used the WEKA's implementation (Hall et al., 2009) of the fast Sequential Minimal Optimization algorithm (Platt, 1999) for learning a classifier with ten-fold cross-validation and WEKA's default parameters. We present the macro-averaged F_1 score (Manning et al., 2008, Section 13.6) as the evaluation metric for this task.

Just as in the document clustering task, the mixture weight $\lambda = 0.6$ obtains the highest classification performances on the N20short dataset. For example with $T = 40$, our w2v-LDA and glove-LDA obtain F_1 scores at 40.0% and 38.9% which are 4.5% and 3.4% higher than F_1 score at 35.5% obtained by the LDA model, respectively.

We report classification results on the remaining experimental datasets with mixture weight $\lambda = 0.6$ in tables 9, 10 and 11. Unlike the clustering results, the LDA model does better than the DMM model for classification on the TMN dataset.

Data	Method	$\lambda = 0.6$			
		T=6	T=20	T=40	T=80
N20	LDA	0.312 ± 0.013	0.635 ± 0.016	**0.742** ± 0.014	0.763 ± 0.005
	w2v-LDA	**0.316** ± 0.013	0.641 ± 0.019	0.730 ± 0.017	**0.768** ± 0.004
	glove-LDA	0.288 ± 0.013	**0.650** ± 0.024	0.733 ± 0.011	0.762 ± 0.006
	Improve.	0.004	0.015	-0.009	0.005
N20small	LDA	0.204 ± 0.020	0.392 ± 0.029	0.459 ± 0.030	0.477 ± 0.025
	w2v-LDA	**0.213** ± 0.018	**0.442** ± 0.025	**0.502** ± 0.031	**0.509** ± 0.022
	glove-LDA	0.181 ± 0.011	0.420 ± 0.025	0.474 ± 0.029	0.498 ± 0.012
	Improve.	0.009	0.05	0.043	0.032

Table 9: F_1 scores (mean and standard deviation) for N20 and N20small datasets.

New models vs. baseline models: On most evaluations, our models perform better than the baseline models. In particular, on the small N20small and Twitter datasets, when the number of topics T is equal to number of ground truth labels (i.e. 20 and 4 correspondingly), our w2v-LDA obtains $5^+\%$ higher F_1 score than the LDA model. In addition, our w2v-DMM model achieves 5.4% and 2.9% higher F_1 score than the DMM model on short TMN and TMNtitle datasets with $T = 80$, respectively.

Data	Method	$\lambda = 0.6$			
		T=7	T=20	T=40	T=80
TMN	LDA	0.658 ± 0.026	0.754 ± 0.009	0.768 ± 0.004	0.778 ± 0.004
	w2v-LDA	0.663 ± 0.021	0.758 ± 0.009	**0.769** ± 0.005	**0.780** ± 0.004
	glove-LDA	**0.664** ± 0.025	**0.760** ± 0.006	0.767 ± 0.003	0.779 ± 0.004
	Improve.	0.006	0.006	0.001	0.002
TMN	DMM	0.607 ± 0.040	0.694 ± 0.026	0.712 ± 0.014	0.721 ± 0.008
	w2v-DMM	0.607 ± 0.019	0.736 ± 0.025	**0.760** ± 0.011	0.771 ± 0.005
	glove-DMM	**0.621** ± 0.042	**0.750** ± 0.011	0.759 ± 0.006	**0.775** ± 0.006
	Improve.	0.014	0.056	0.048	0.054
TMNtitle	LDA	0.564 ± 0.015	0.625 ± 0.011	0.626 ± 0.010	0.624 ± 0.006
	w2v-LDA	0.563 ± 0.029	**0.644** ± 0.010	**0.643** ± 0.007	0.640 ± 0.004
	glove-LDA	**0.568** ± 0.028	**0.644** ± 0.010	0.632 ± 0.008	**0.642** ± 0.005
	Improve.	0.004	0.019	0.017	0.018
TMNtitle	DMM	0.500 ± 0.021	0.600 ± 0.015	0.630 ± 0.016	0.652 ± 0.005
	w2v-DMM	0.528 ± 0.028	0.663 ± 0.008	0.682 ± 0.008	**0.681** ± 0.006
	glove-DMM	**0.565** ± 0.022	**0.680** ± 0.011	**0.684** ± 0.009	**0.681** ± 0.004
	Improve.	0.065	0.08	0.054	0.029

Table 10: F_1 scores for TMN and TMNtitle datasets.

Data	Method	$\lambda = 0.6$			
		T=4	T=20	T=40	T=80
Twitter	LDA	0.526 ± 0.021	0.636 ± 0.011	0.650 ± 0.014	0.653 ± 0.008
	w2v-LDA	**0.578** ± 0.047	0.651 ± 0.015	0.661 ± 0.011	**0.664** ± 0.010
	glove-LDA	0.569 ± 0.037	**0.656** ± 0.011	**0.662** ± 0.008	0.662 ± 0.006
	Improve.	0.052	0.02	0.012	0.011
Twitter	DMM	0.469 ± 0.014	0.600 ± 0.021	0.645 ± 0.009	0.665 ± 0.014
	w2v-DMM	**0.539** ± 0.016	0.649 ± 0.016	0.656 ± 0.007	0.676 ± 0.012
	glove-DMM	0.536 ± 0.027	**0.654** ± 0.019	**0.657** ± 0.008	**0.680** ± 0.009
	Improve.	0.07	0.054	0.012	0.015

Table 11: F_1 scores for Twitter dataset.

Google word2vec vs. Stanford glove word vectors: The comparison of the Google and Stanford pre-trained word vectors for classification is similar to the one for clustering.

4.5 Discussion

We found that the topic coherence evaluation produced the best results with a mixture weight $\lambda = 1$, which corresponds to using topic-word distributions defined in terms of the latent-feature word vectors. This is not surprising, since the topic coherence evaluation we used (Lau et al., 2014) is based on word co-occurrences in an external corpus (here, Wikipedia), and it is reasonable that the billion-word corpora used to train the latent feature word vectors are more useful for this task than the much smaller topic-modeling corpora, from which the topic-word multinomial distributions are trained.

On the other hand, the document clustering and document classification tasks depend more strongly on possibly idiosyncratic properties of the smaller topic-modeling corpora, since these evaluations reflect how well the document-topic assignments can group or distinguish documents within the topic-modeling corpus. Smaller values of λ enable the models to learn topic-word distributions that include an arbitrary multinomial topic-word distribution, enabling the models to capture idiosyncratic properties of the topic-modeling corpus. Even in these evaluations we found that an intermediate value of $\lambda = 0.6$ produced the best results, indicating that better word-topic distributions were produced when information from the large external corpus is combined with corpus-specific topic-word multinomials. We found that using the latent feature word vectors produced significant performance improvements even when the domain of the topic-modeling corpus was quite different to that of the external corpus from which the word vectors were derived, as was the case in our experiments on Twitter data.

We found that using either the Google or the Stanford latent feature word vectors produced very similar results. As far as we could tell, there is no reason to prefer either one of these in our topic modeling applications.

5 Conclusion and future work

In this paper, we have shown that latent feature representations can be used to improve topic models. We proposed two novel latent feature topic models, namely LF-LDA and LF-DMM, that integrate a latent feature model within two topic models LDA and DMM. We compared the performance of our models LF-LDA and LF-DMM to the baseline LDA and DMM models on topic coherence, document clustering and document classification evaluations. In the topic coherence evaluation, our model outperformed the baseline models on all 6 experimental datasets, showing that our method for exploiting external information from very large corpora helps improve the topic-to-word mapping. Meanwhile, document clustering and document classification results show that our models improve the document-topic assignments compared to the baseline models, especially on datasets with few or short documents.

As an anonymous reviewer suggested, it would be interesting to identify exactly how the latent feature word vectors improve topic modeling performance. We believe that they provide useful information about word meaning extracted from the large corpora that they are trained on, but as the reviewer suggested, it is possible that the performance improvements arise because the word vectors are trained on context windows of size 5 or 10, while the LDA and DMM models view documents as bags of words, and effectively use a context window that encompasses the entire document. In preliminary experiments where we train latent feature word vectors from the topic-modeling corpus alone using context windows of size 10 we found that performance was degraded relative to the results presented here, suggesting that the use of a context window alone is not responsible for the performance improvements we reported here. Clearly it would be valuable to investigate this further.

In order to use a Gibbs sampler in section 3.4, the conditional distributions needed to be distributions we can sample from cheaply, which is not the case for the ratios of Gamma functions. While we used a simple approximation, it is worth exploring other sampling techniques that can avoid approximations, such as Metropolis-Hastings sampling (Bishop, 2006, Section 11.2.2).

In order to compare the pre-trained Google and Stanford word vectors, we excluded words that did not appear in both sets of vectors. As suggested by anonymous reviewers, it would be interesting to learn vectors for these unseen words. In addition, it is worth fine-tuning the seen-word vectors on the dataset of interest.

Although we have not evaluated our approach on very large corpora, the corpora we have evaluated on do vary in size, and we showed that the gains from our approach are greatest when the corpora are small. A drawback of our approach is that it is slow on very large corpora. Variational Bayesian inference may provide an efficient solution to this problem (Jordan et al., 1999; Blei et al., 2003).

Acknowledgments

This research was supported by a Google award through the Natural Language Understanding

Focused Program, and under the Australian Research Council's *Discovery Projects* funding scheme (project numbers DP110102506 and DP110102593). The authors would like to thank the three anonymous reviewers, the action editor and Dr. John Pate at the Macquarie University, Australia for helpful comments and suggestions.

References

Christopher M. Bishop. 2006. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc.

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

David M. Blei. 2012. Probabilistic Topic Models. *Communications of the ACM*, 55(4):77–84.

John A. Bullinaria and Joseph P. Levy. 2007. Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior Research Methods*, 39(3):510–526.

Deng Cai, Qiaozhu Mei, Jiawei Han, and Chengxiang Zhai. 2008. Modeling Hidden Topics on Document Manifold. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, pages 911–920.

Ziqiang Cao, Sujian Li, Yang Liu, Wenjie Li, and Heng Ji. 2015. A Novel Neural Topic Model and Its Supervised Extension. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 2210–2216.

Ana Cardoso-Cachopo. 2007. Improving Methods for Single-label Text Categorization. PhD Thesis, Instituto Superior Tecnico, Universidade Tecnica de Lisboa.

Jonathan Chang, Sean Gerrish, Chong Wang, Jordan L. Boyd-graber, and David M. Blei. 2009. Reading Tea Leaves: How Humans Interpret Topic Models. In *Advances in Neural Information Processing Systems 22*, pages 288–296.

Ronan Collobert and Jason Weston. 2008. A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning. In *Proceedings of the 25th International Conference on Machine Learning*, pages 160–167.

Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. 1990. Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science*, 41(6):391–407.

Jacob Eisenstein, Amr Ahmed, and Eric Xing. 2011. Sparse Additive Generative Models of Text. In *Proceedings of the 28th International Conference on Machine Learning*, pages 1041–1048.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Domain Adaptation for Large-Scale Sentiment Classification: A Deep Learning Approach. In *Proceedings of the 28th International Conference on Machine Learning*, pages 513–520.

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

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The WEKA Data Mining Software: An Update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18.

Bo Han, Paul Cook, and Timothy Baldwin. 2012. Automatically Constructing a Normalisation Dictionary for Microblogs. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 421–432.

Swapnil Hingmire, Sandeep Chougule, Girish K. Palshikar, and Sutanu Chakraborti. 2013. Document Classification by Topic Labeling. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 877–880.

Liangjie Hong and Brian D. Davison. 2010. Empirical Study of Topic Modeling in Twitter. In *Proceedings of the First Workshop on Social Media Analytics*, pages 80–88.

Seungil Huh and Stephen E. Fienberg. 2012. Discriminative Topic Modeling Based on Manifold Learning. *ACM Transactions on Knowledge Discovery from Data*, 5(4):20:1–20:25.

Mark Johnson. 2010. PCFGs, Topic Models, Adaptor Grammars and Learning Topical Collocations and the Structure of Proper Names. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1148–1157.

Michael I. Jordan, Zoubin Ghahramani, Tommi S. Jaakkola, and Lawrence K. Saul. 1999. An Introduction to Variational Methods for Graphical Models. *Machine Learning*, 37(2):183–233.

Simon Lacoste-Julien, Fei Sha, and Michael I. Jordan. 2009. DiscLDA: Discriminative Learning for Dimensionality Reduction and Classification. In *Advances in Neural Information Processing Systems 21*, pages 897–904.

Han Jey 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*, pages 530–539.

D. C. Liu and J. Nocedal. 1989. On the Limited Memory BFGS Method for Large Scale Optimization. *Mathematical Programming*, 45(3):503–528.

Yang Liu, Zhiyuan Liu, Tat-Seng Chua, and Maosong Sun. 2015. Topical Word Embeddings. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 2418–2424.

Yue Lu, Qiaozhu Mei, and ChengXiang Zhai. 2011. Investigating task performance of probabilistic topic models: an empirical study of PLSA and LDA. *Information Retrieval*, 14:178–203.

Kevin Lund and Curt Burgess. 1996. Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instruments, & Computers*, 28(2):203–208.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press.

Andrew McCallum. 2002. MALLET: A Machine Learning for Language Toolkit.

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.

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 26*, pages 3111–3119.

David Mimno, Hanna M. 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.

David Newman, Chaitanya Chemudugunta, and Padhraic Smyth. 2006. Statistical Entity-Topic Models. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 680–686.

David Newman, Jey Han Lau, Karl Grieser, and Timothy Baldwin. 2010. Automatic Evaluation of Topic Coherence. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 100–108.

Kamal Nigam, AK McCallum, S Thrun, and T Mitchell. 2000. Text Classification from Labeled and Unlabeled Documents Using EM. *Machine learning*, 39:103–134.

Michael Paul and Mark Dredze. 2015. SPRITE: Generalizing Topic Models with Structured Priors. *Transactions of the Association for Computational Linguistics*, 3:43–57.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543.

James Petterson, Wray Buntine, Shravan M. Narayanamurthy, Tibério S. Caetano, and Alex J. Smola. 2010. Word Features for Latent Dirichlet Allocation. In *Advances in Neural Information Processing Systems 23*, pages 1921–1929.

Xuan-Hieu Phan, Cam-Tu Nguyen, Dieu-Thu Le, Le-Minh Nguyen, Susumu Horiguchi, and Quang-Thuy Ha. 2011. A Hidden Topic-Based Framework Toward Building Applications with Short Web Documents. *IEEE Transactions on Knowledge and Data Engineering*, 23(7):961–976.

John C. Platt. 1999. Fast Training of Support Vector Machines using Sequential Minimal Optimization. In Bernhard Schölkopf, Christopher J. C. Burges, and Alexander J. Smola, editors, *Advances in kernel methods*, pages 185–208.

Ian Porteous, David Newman, Alexander Ihler, Arthur Asuncion, Padhraic Smyth, and Max Welling. 2008. Fast Collapsed Gibbs Sampling for Latent Dirichlet Allocation. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 569–577.

Christian P. Robert and George Casella. 2004. *Monte Carlo Statistical Methods (Springer Texts in Statistics)*. Springer-Verlag New York, Inc.

Mehran Sahami and Timothy D. Heilman. 2006. A Web-based Kernel Function for Measuring the Similarity of Short Text Snippets. In *Proceedings of the 15th International Conference on World Wide Web*, pages 377–386.

Ruslan Salakhutdinov and Geoffrey Hinton. 2009. Replicated Softmax: an Undirected Topic Model. In *Advances in Neural Information Processing Systems 22*, pages 1607–1614.

Richard Socher, John Bauer, Christopher D. Manning, and Ng Andrew Y. 2013. Parsing with Compositional Vector Grammars. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 455–465.

Nitish Srivastava, Ruslan Salakhutdinov, and Geoffrey Hinton. 2013. Modeling Documents with a Deep

Boltzmann Machine. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, pages 616–624.

Keith Stevens, Philip Kegelmeyer, David Andrzejewski, and David Buttler. 2012. Exploring Topic Coherence over Many Models and Many Topics. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 952–961.

Yee W Teh, David Newman, and Max Welling. 2006. A Collapsed Variational Bayesian Inference Algorithm for Latent Dirichlet Allocation. In *Advances in Neural Information Processing Systems 19*, pages 1353–1360.

Kristina Toutanova and Mark Johnson. 2008. A Bayesian LDA-based Model for Semi-Supervised Part-of-speech Tagging. In *Advances in Neural Information Processing Systems 20*, pages 1521–1528.

Daniele Vitale, Paolo Ferragina, and Ugo Scaiella. 2012. Classification of Short Texts by Deploying Topical Annotations. In *Proceedings of the 34th European Conference on Advances in Information Retrieval*, pages 376–387.

Jianshu Weng, Ee-Peng Lim, Jing Jiang, and Qi He. 2010. TwitterRank: Finding Topic-sensitive Influential Twitterers. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, pages 261–270.

Pengtao Xie and Eric P. Xing. 2013. Integrating Document Clustering and Topic Modeling. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, pages 694–703.

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.

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.

Ke Zhai and Jordan L. Boyd-graber. 2013. Online Latent Dirichlet Allocation with Infinite Vocabulary. In *Proceedings of the 30th International Conference on Machine Learning*, pages 561–569.

Wayne Xin Zhao, Jing Jiang, Jianshu Weng, Jing He, Ee-Peng Lim, Hongfei Yan, and Xiaoming Li. 2011. Comparing Twitter and Traditional Media Using Topic Models. In *Proceedings of the 33rd European Conference on Advances in Information Retrieval*, pages 338–349.