Neural Embedding Allocation

Neural Embedding Allocation: Distributed Representations of Topic Models

Kamrun Naher Keya*
University of Maryland,
Baltimore County, MD, USA

Yannis Papanikolaou**
Healx,
Cambridge, UK

James R. Foulds†
University of Maryland,
Baltimore County, MD, USA

We propose a method which uses neural embeddings to improve the performance of any given LDA-style topic model. Our method, called neural embedding allocation *(NEA), deconstructs topic models (LDA or otherwise) into interpretable vector-space embeddings of words, topics, documents, authors, and so on, by learning neural embeddings to mimic the topic model. We demonstrate that NEA improves coherence scores of the original topic model by smoothing out the noisy topics when the number of topics is large. Furthermore, we show NEA's effectiveness and generality in deconstructing and smoothing LDA, author-topic models, and the recent mixed membership skip-gram topic model and achieve better performance with the embeddings compared to several state-of-the-art models.*

1. Introduction

In recent years, methods for automatically learning representations of text data have become an essential part of the Natural Language Processing (NLP) pipeline. Word embedding models such as the skip-gram improve the performance of NLP methods by revealing the latent structural relationship between words (Mikolov et al. 2013a,b). These embeddings have proven valuable for a variety of NLP tasks such as statistical machine translation (Vaswani et al. 2013), part-of-speech tagging, chunking, and named entity recognition (Collobert et al. 2011). Since word vectors encode distributional information, the similarity relationships between the semantic meanings of the words are reflected in the similarity of the vectors (Sahlgren 2008).

* E-mail: kkeya1@umbc.edu.
** E-mail: yannis.papanikolaou@healx.io
† E-mail: jfoulds@umbc.edu

Action editor: Sameer Singh. Submission received: 7 August 2021; revised version received: 14 July 2022; accepted for publication: 5 August 2022.

On the other hand, topic models such as latent Dirichlet allocation (LDA) (Blei, Ng, and Jordan 2003) construct latent representations of topical themes and of documents. Unlike word embeddings, topic models recover human interpretable semantic themes in the corpus. However, since topic models represent words using only dictionary indices rather than using vector-space embeddings, they are not able to directly capture or leverage the nuanced/distinct similarity relationships between words that are afforded by such embeddings.

We therefore desire a unified method which gains the benefits of both word embeddings (nuanced semantic relationships) and topic models (interpretable topical themes) in a mutually informing manner. A number of models have been proposed which combine aspects of word embeddings and topic models, by modeling them conditionally or jointly (Das, Zaheer, and Dyer 2015; Liu et al. 2015; Nguyen et al. 2015; Moody 2016; Shi et al. 2017; Meng et al. 2020), or by using neural variational inference for topic models (Miao, Yu, and Blunsom 2016; Zhu, He, and Zhou 2020). These models have not yet supplanted standard word embedding and topic modeling techniques in most applications, perhaps due to their complexity.

More recently, transformer-based language models such as BERT (Devlin et al. 2019) and it's variant RoBERTa (Liu et al. 2019) have emerged as a powerful technique to improve over word embeddings with state-of-the-art performance at learning text representations for many tasks, but they do not aim to learn representations of topical semantics that are meaningful to humans. Another transformer-based autoregressive language model called GPT (Radford et al. 2018, 2019; Brown et al. 2020) can produce human-like text and also perform various NLP tasks such as text summarization, question answering, textual entailment, etc. However, Bender et al. 2021 criticized these large language models for the environmental impact of training and storing the models.

A more parsimonious approach, first used in mixed membership word embeddings (MMSG) (Foulds 2018), and subsequently in the embedded topic model (ETM) (Dieng, Ruiz, and Blei 2020) (proposed independently of this work) is to parameterize a topic model's categorical distributions via embeddings, thereby obtaining mutually informing topics and embeddings without complicated joint or conditional modeling. These two models improve performance over their corresponding topic models, but do not apply more generally.

Building on the aforementioned line of work, in this paper we propose a method for efficiently and accurately training a *general class of embedding-parameterized topic models*. Our approach, which we call *neural embedding allocation* (NEA), is to *deconstruct topic models* by reparameterizing them using vector-space embeddings. Given as input any arbitrary pre-trained topic model that is parameterized by categorical distributions, NEA outputs vector-space embeddings which encode its topical semantics. As well as learning effective embeddings, we demonstrate that the embeddings can be used to improve the quality of the topics. To this end, NEA uses the learned lower-dimensional representations to smooth out noisy topics. It outputs a smoothed version of the categorical topics which is typically less noisy and more semantically coherent than the original topic model.

We can view our NEA method as *learning to mimic a topic model with a skip-gram style embedding model* to reveal underlying semantic vector representations. Our approach is thus reminiscent of model distillation for supervised models (Buciluǎ, Caruana, and Niculescu-Mizil 2006; Hinton, Vinyals, and Dean 2015). Inspired by subtle connections between embedding and topic model learning algorithms, we train NEA by minimizing the KL-divergence to the data distribution of the corresponding topic model, using a

stream of simulated data from the model (subset of data are randomly drawn from the topic model's parameters). The resulting embeddings allow us to:

(1) *improve the coherence* of topic models by "smoothing out" noisy topics,
(2) *improve classification performance* by producing topic-informed document vectors, and
(3) construct embeddings and smoothed distributions over *general topic modeling variables* such as authors.

In short, **NEA takes any general off-the-shelf LDA-style topic model**, and improves it by **making it more coherent** and **extracting powerful latent representations**. Since it takes the pre-trained topic model as input, **NEA can achieves this for a range of sophisticated models that extend LDA**, such as those with additional latent variables, without complicating inference for the input topic model. This also enables it to be applied in use-cases where we are given a pre-trained topic model by someone else and we do not have access to the original data, e.g. due to privacy concerns.

NEA is related to the embedded topic model (ETM) (Dieng, Ruiz, and Blei 2020), a variant of neural network-based topic models, and to the mixed membership skip-gram (MMSG) (Foulds 2018), in that they each learn topic models which are parameterized by vector representations of both words and topics. However, while ETM and MMSG are *models* with specific architectures and associated special-purpose learning algorithms for those particular architectures, NEA is an *algorithm* which learns vector representations for *any LDA-style topic model*.

We show the benefits and generality of our NEA method by applying it to LDA, author-topic models (ATM) (Rosen-Zvi et al. 2004), and the mixed membership skip gram topic model (MMSGTM) (Foulds 2018). NEA is compatible with sublinear algorithms for topic models (Li et al. 2014) and embeddings (Mikolov et al. 2013a; Mnih and Kavukcuoglu 2013), thereby readily scaling to tens of thousands of topics, unlike previous topical embedding methods. To the best of our knowledge, NEA is the first general method for improving arbitrary LDA-style topic models via embeddings.[1]

2. Background

For completeness, and to establish notation, we provide necessary background on topic models and word embeddings.

2.1 Latent Dirichlet Allocation

Probabilistic topic models, for example, LDA (Blei, Ng, and Jordan 2003) use latent variables to encode co-occurrences between words in text corpora and other bag-of-words represented data. A simple way to model text corpora is using multinomial naive Bayes with a latent cluster assignment for each document, which is a multinomial distribution over words, called a *topic* $k \in \{1, ...K\}$. LDA topic models improve over naive Bayes using mixed membership, by relaxing the condition that all words in a document d belong to the same topic. In LDA's generative process, for each word w_{di} of a document d, a topic assignment z_{di} is sampled from document-topic distribution $\theta^{(d)}$

1 A very early version of this research was presented at the MASC-SLL 2018 symposium (Keya and Foulds 2018), pre-dating (Dieng, Ruiz, and Blei 2020).

followed by drawing the word from topic-word distribution $\phi^{(z_{di})}$ (see Table 1, bottom-right). Dirichlet priors encoded by α_k and β_w are used for these parameters, respectively.

2.2 Author Topic Model

The author-topic model (ATM) is a probabilistic model for both authors and topics which extends LDA to include authorship information (Rosen-Zvi et al. 2004). In the generative process of ATM, for each word w_{di} of a document d, an author assignment a_{di} is uniformly chosen from the set of authors A_d and then a topic assignment z_{di} is sampled from the author-topic distribution $\theta^{(a_{di})}$ followed by drawing the word from topic-word distribution $\phi^{(z_{di})}$ as follows:

- For each document d
  - For each word in the document w_{di}
    Draw $a_{di} \sim$ Uniform$(\frac{1}{|A_d|})$
    Draw $z_{di} \sim$ Discrete$(\theta^{(a_{di})})$
    Draw $w_{di} \sim$ Discrete$(\phi^{(z_{di})})$

Like for LDA, Dirichlet priors α_a and β_w are used for $\theta^{(a)}$ and $\phi^{(z)}$ parameters, respectively.

2.3 Word Embeddings

Traditional neural probabilistic language models predict words given their context words using a joint probability for sequences of words in a language (Bengio et al. 2003) based on distributed representations (Hinton et al. 1986) from neural network weights. Later, word embeddings were found to be useful for semantic representations of words, even without learning a full joint probabilistic language model. In particular, the skip-gram model is an effective method for learning better quality vector representations of words from big unstructured text data.

The skip-gram (Mikolov et al. 2013b) is a log-bilinear classifier for predicting words that occur in the context of other words in a document, where the context is typically defined to be a small window around the word. For a sequence of input training words, the objective of the skip-gram model is to maximizing the average log probability of the output context words given the input word. We can think of it as a certain parameterization of a set of discrete distributions, $p(w_c|w_i)$, where w_c is a context word and w_i is an "input" word, and both w_c and w_i range over the W words in the dictionary (see Table 1, top-left). In the simplest case, these discrete distributions have the form:

$$p(w_c|w_i) \propto exp({v'_{w_c}}^{\mathsf{T}} v_{w_i}) \; . \tag{1}$$

where, v'_{w_c} and v_{w_i} are vector embeddings of context words and input words, respectively, with dimensionality V.

2.4 MMSG Topic Model

We also consider our prior work, a topic model called the mixed membership skip-gram topic model (MMSGTM) (Foulds 2018), which combines ideas from topic models and word embeddings to recover domain specific embeddings for small data (e.g., 2,000 articles). The generative model for MMSGTM is:

- For each word w_i in the corpus
  - Sample a topic $z_i \sim \text{Discrete}(\theta^{w_i})$
  - For each word $w_c \in context(i)$
    Sample a context word
    $w_c \sim \text{Discrete}(\phi^{z_i})$.

Finally, the mixed membership skip-gram model (MMSG) is trained for word and topic embeddings with the topic assignments z as input and surrounding w_c as output. Since the MMSG training algorithm depends on the topic assignments for the whole corpus, it is not scalable for big data, unlike our proposed method NEA (introduced in Section 4.1). NEA is a general method which can be applied to train the MMSG model, and our experiments will demonstrate that it improves over the original MMSG training algorithm, as well as improving the MMSGTM's representations (see Section 5.3).

3. Connections Between Word Embeddings and Topic Models

According to the distributional hypothesis, words which typically occur in similar contexts are likely to have similar meanings (Sahlgren 2008). Hence, the skip-gram's conditional distributions over context words, and the vector representations which encode these distributions, are expected to be informative of the semantic relationships between words (Mikolov et al. 2013b). Similarly, Griffiths, Steyvers, and Tenenbaum (2007) modeled semantic relationships between words based on LDA, which they successfully used to solve a word association task. This suggests that topic models implicitly encode semantic relationships between words, motivating methods to recover this information, as we shall propose.

In this section, we first develop a bridge to connect word embeddings methods such as the skip-gram with topic models, which will inform our approach going forward. First, we will show how word embedding models such as the skip-gram can be reinterpreted as a version of a corresponding topic model. Then, we will show how the learning algorithm for the skip-gram model can be understood as learning to mimic this topic model. We will use this perspective to motivate our proposed NEA method in Section 4.

3.1 Interpreting Embedding Models as Topic Models

The relationship between the skip-gram and topic models goes beyond their common ability to recover semantic representations of words. The skip-gram (Mikolov et al. 2013b) and LDA (Blei, Ng, and Jordan 2003) models are summarized in Table 1 (top-left, bottom-right), where we have interpreted the skip-gram, which is discriminative, as a "conditionally generative" model. As the table makes clear, the skip-gram and LDA both model conditional discrete distributions over words; conditioned on an input word in the former, and conditioned on a topic in the latter. To relate the two models, we hence reinterpret the skip-gram's conditional distributions over words as "topics" $\phi^{(w_i)}$, and the input words w_i as observed cluster assignments, analogous to topic assignments z. From this perspective, the skip-gram can be understood as a particular "topic model," in which the "topics" are parameterized via embeddings, and are assumed to generate context words.

In more detail, Table 1 (top) shows how the skip-gram (top-left) can be reinterpreted as a certain parameterization of a fully supervised naive Bayes topic model (top-right), which Foulds (2018) calls the *(naive Bayes) skip-gram topic model (SGTM)*.

Table 1
"Generative" models of the skip-gram (top-left) and its analogous naive Bayes topic model (top-right), and the *neural embedding allocation* reparameterization of the LDA topic model (bottom).

	Embedding Models	Topic Models
	Skip-gram	Naive Bayes skip-gram topic model (SGTM)
Words|Input Word	• For each word in the corpus w_i – Draw input word $w_i \sim p_{data}(w_i)$ – For each word $w_c \in context(i)$ Draw $w_c \| w_i \propto exp({v'_{w_c}}^{\mathsf{T}} v_{w_i})$	• For each word in the corpus w_i – Draw input word $w_i \sim p_{data}(w_i)$ – For each word $w_c \in context(i)$ Draw $w_c \| w_i \sim$ Discrete$(\phi^{(w_i)})$
	Neural embedding allocation	Latent Dirichlet allocation
Words|Topics	• For each document d – For each word in the document w_{di} Draw $z_{di} \| d \sim$ Discrete$(\theta^{(d)})$ Draw $w_{di} \| z_{di} \propto \exp({v'_{w_{di}}}^{\mathsf{T}} \bar{v}_{z_{di}})$	• For each document d – For each word in the document w_{di} Draw $z_{di} \| d \sim$ Discrete$(\theta^{(d)})$ Draw $w_{di} \| z_{di} \sim$ Discrete$(\phi^{(z_{di})})$

These two models have the same assumed generative process, differing only in how they parameterize the distribution of context words w_c given input words w_i, i.e. $p(w_c|w_i)$. The skip-gram parameterizes this distribution using embeddings v'_{w_c} and v_{w_i} via a log-bilinear model, while the SGTM parameterizes it as a "topic," i.e. a discrete distribution over words, $\phi^{(w_i)}$. The models are equivalent up to this parameterization.

3.2 Interpreting Embedding Model Training as Mimicking a Topic Model

We next show how learning algorithms for the skip-gram are related to the SGTM. We will do this by introducing a variational interpretation of skip-gram training. This interpretation will provide a new perspective on the skip-gram learning algorithm: *training the skip-gram on dataset corresponds to learning to mimic the optimal SGTM for that dataset.* We first overview our argument before describing it in more precise detail.

3.2.1 Informal Summary of our Argument. It is well known that maximizing the log likelihood for a model is equivalent to minimizing the KL-divergence to the model's empirical data distribution, cf. Hinton (2002). When trained via maximum likelihood estimation (MLE), the skip-gram (SG) and its corresponding topic model both aim to approximate this same empirical data distribution. The skip-gram topic model (SGTM) can encode any set of conditional discrete distributions, and so its MLE recovers this distribution exactly. Thus, we can see that the skip-gram, trained via MLE, also aims to approximate the MLE skip-gram topic model in a variational sense.

3.2.2 More Mathematically Precise Argument. More formally, consider the joint distribution $p(w_c, w_i)$ obtained by augmenting the skip-gram SG and its topic model SGTM with the empirical input word distribution $p(w_i) = p_{data}(w_i)$:

$$p_{SG}(w_c, w_i; \mathbf{v}, \mathbf{v}') = p(w_c|w_i; \mathbf{v}, \mathbf{v}')p_{data}(w_i) \tag{2}$$

$$p_{SGTM}(w_c, w_i; \mathbf{\Phi}) = p(w_c|w_i; \mathbf{\Phi})p_{data}(w_i)\ . \tag{3}$$

It can readily be seen that

$$\begin{aligned} & D_{KL}(p_{data}(w_c|w_i)p_{data}(w_i)||p_{SG}(w_c, w_i; \mathbf{v}, \mathbf{v}')) \\ & \quad = -\sum_{w_c,w_i} \frac{N_{w_c,w_i}}{N_{w_i}} \frac{N_{w_i}}{N} \log p(w_c|w_i; \mathbf{v}, \mathbf{v}') + \text{const} \\ & \quad = -\sum_{w_c,w_i} \frac{N_{w_c,w_i}}{N} \log p(w_c|w_i; \mathbf{v}, \mathbf{v}') + \text{const} \,. \end{aligned} \tag{4}$$

By a similar argument, we also obtain

$$D_{KL}(p_{data}(w_c|w_i)||p_{SGTM}(w_c, w_i; \mathbf{\Phi})) = -\sum_{w_c,w_i} \frac{N_{w_c,w_i}}{N} \log p(w_c|w_i; \mathbf{\Phi}) + \text{const} \,. \tag{5}$$

Since the discrete topic distributions are unconstrained, this is minimized to zero at

$$\hat{\phi}_{w_c}^{(w_i)} = \frac{N_{w_c,w_i}}{N_{w_i}} = p_{data}(w_c|w_i) \,. \tag{6}$$

So maximizing the conditional log-likelihood for the skip-gram minimizes the KL-divergence to $p_{data}(w_c|w_i)p_{data}(w_i) = p_{SGTM}(w_c, w_i; \hat{\mathbf{\Phi}})$, where $\hat{\mathbf{\Phi}}$ is the MLE of the skip-gram topic model. Therefore, **the skip-gram is attempting to mimic the "optimal" skip-gram topic model**, by solving a variational inference problem which aims to make its distribution over input/output word pairs as similar as possible to that of the SGTM's MLE.[2]

While the above holds for maximum likelihood training, it should be noted that the skip-gram is more typically trained via negative sampling (NEG) (Mikolov et al. 2013b) or noise contrastive estimation (NCE) (Gutmann and Hyvärinen 2010, 2012), rather than MLE. These methods are however used for computational reasons, while maximum likelihood estimation is the gold-standard "ideal" training procedure which NEG and NCE aim to approximate. The NCE algorithm was derived as an approximate method for maximum likelihood estimation (Gutmann and Hyvärinen 2010, 2012), and negative sampling (NEG) (Mikolov et al. 2013b) can be understood as an approximate version of NCE (Dyer 2014). We can therefore view both NCE and NEG as approximately solving the same variational problem as MLE, with some bias in their solutions due to the approximations that they make. In this sense, our claim that "skip-gram training aims

2 With sufficiently high-dimensional vectors, the skip-gram will be able to solve this optimization problem exactly and perfectly reconstruct the MLE SGTM, assuming that a global optimum can be found. E.g., if $V = W$, the skip-gram can trivially encode any set of "topic" distributions Φ by setting each v'_{w_i} as a one-hot vector which selects a single dimension of the v_{w_c} embeddings that encodes $p(w_c|w_i)$. Thus, we can see that whenever $V \geq W$, the skip-gram can encode any topics, including those of the SGTM's MLE. Alternatively, if the skip-gram cannot encode the MLE topic model due to having too low dimensionality, which is more typically the case in practice, it will find a local optimum in the KL-divergence objective function. This is actually desirable, as the embeddings are forced to perform compression when encoding the topics, which forces the embeddings to capture patterns in the data, and hence encode meaningful information.

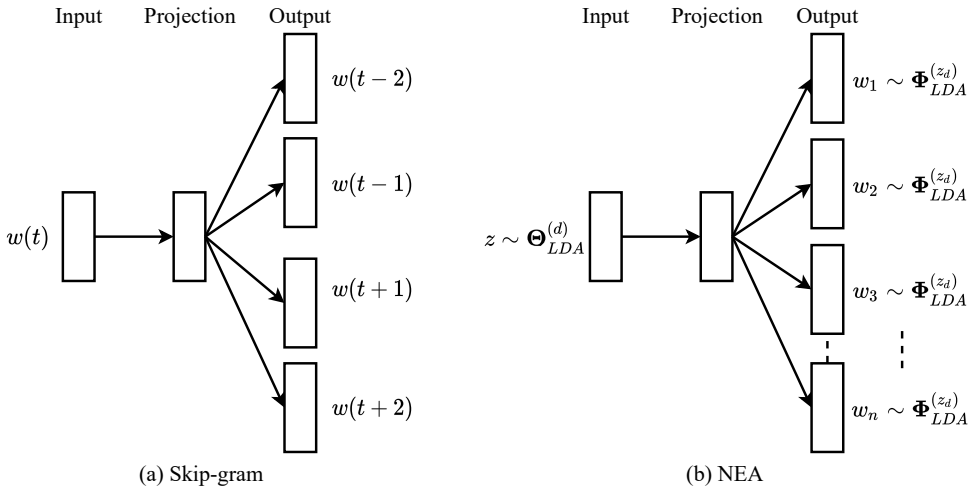

Figure 1
The model architecture of training (a) skip-gram, and (b) NEA. In skip-gram, the training objective is to learn word vector representations that are good at predicting the nearby (or context) words. Inspired by the skip-gram, we train NEA with simulated topic z and word w which are drawn from LDA parameters $\mathbf{\Theta}^{(d)}_{LDA}$ and $\mathbf{\Phi}^{(z_d)}_{LDA}$, respectively, for a uniformly random document d. The training objective to train NEA for LDA is to learn topic embeddings that are good at predicting simulated words.

to mimic a topic model" extends beyond the idealized MLE training procedure to the NEG and NCE implementations used in practice.[3]

4. Neural Embedding Allocation

We have seen that the skip-gram model (approximately) minimizes the KL-divergence to the distribution over data at the maximum likelihood estimate of its corresponding topic model. The skip-gram deconstructs its topic model into vector representations which aim to encode the topic model's distributions over words. We can view this as *learning to mimic a topic model with an embedding model*.

In the skip-gram model, the training objective is to learn word vector representations that are good at predicting the nearby (or context) words (Mikolov et al. 2013b) as shown in Figure 1 (a). The resulting vectors capture semantic relationships between words which were not directly available in its original "topic model" in which words were simply represented as dictionary indices. We therefore propose to *apply this same approach, deconstructing topic models into neural embedding models, to other topic models. By doing so, we aim to similarly extract vector representations which encode the semantics of words (and topics, etc) which were latent in the target topic model's parameters*.

3 As a side note, we can also see from Equation 6 that the SGTM and SG's MLEs can be completely computed using the input/output word co-occurrence count matrix as sufficient statistics. The skip-gram then has a global objective function that can be defined in terms of the word co-occurrence matrix, and the development of the GloVe model (Pennington, Socher, and Manning 2014) as an alternative with a global objective function seems unnecessary in hindsight. Levy and Goldberg (2014) further illustrated a closely related point by constructing global training objectives for NEG and NCE based on matrix factorization interpretations of these methods.

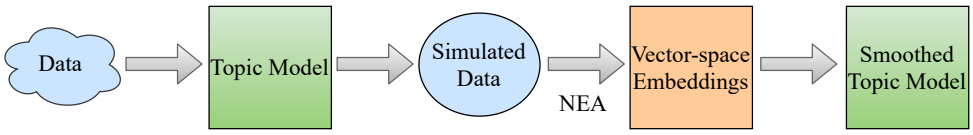

Figure 2
Schematic diagram of neural embedding allocation (NEA) framework to deconstruct and reconstruct topic model.

The resulting method, which we refer to as *neural embedding allocation* (NEA), corresponds to reparameterizing the discrete distributions in topic models with embeddings. The neural embedding model generally loses some model capacity relative to the topic model, but it provides *vector representations* which encode valuable similarity information between words.

As we shall see, NEA's reconstruction of the discrete distributions also *smooths out noisy estimates of the topics*, informed by the vectors' similarity patterns, mitigating overfitting in the topic model training. For example, we show the "generative" model for NEA in Table 1 (bottom-left), which reparameterizes the LDA model by topic vectors $\bar{v}_k$ and "output" word vectors v'_w that mimic LDA's topic distributions over words, $\phi^{(k)}$, by re-encoding them using log-bilinear models. In the generative model, $\theta^{(d)}$ draws a topic for a document and the topic vectors $\bar{v}_k$ are used as the input vectors to draw a word v'_w.[4] The schematic diagram of NEA framework to deconstruct and reconstruct (i.e. smooth out) a topic model is shown in Figure 2.

4.1 Training NEA for LDA

To train the NEA reconstruction of LDA, we start with pre-trained LDA parameters: document-topic distributions $\mathbf{\Theta}_{LDA}$, topic-word distributions $\mathbf{\Phi}_{LDA}$, and topic assignments Z. The model architecture of training NEA on LDA parameters is demonstrated in Figure 1 (b). Given the input LDA (or other) topic model, our ideal objective function to train NEA is $D_{KL}(p_{LDA}||p_{NEA})$. It can be seen that minimizing $D_{KL}(p_{LDA}||p_{NEA})$ is equivalent to maximizing $E_{p_{LDA}(w,z)}[p(w,z;\mathbf{V})]$. This suggests a procedure where minibatches are drawn from the topic model, and are used to update the parameters $\mathbf{V} = \{\mathbf{V}^{(W)\prime}, \bar{\mathbf{V}}^{(K)}\}$ via stochastic gradient descent. We construct minibatches of input topics z and target words w by repeatedly drawing a document index d uniformly at random, drawing a topic z from that document's $\mathbf{\Theta}^{(d)}_{LDA}$ and sampling a word w from drawn topic $\mathbf{\Phi}^{(z_d)}_{LDA}$. Then, we would take a gradient step on $\log p(w,z|\mathbf{V},b,\mathbf{\Theta}_{LDA}) = \log p(w|z,\mathbf{V},b) + \text{const}$ to update $\mathbf{V}$. However, as for other embedding models, normalization over the dictionary becomes a bottleneck in the stochastic gradient updates. Since noise-contrastive estimation (NCE) (Mnih and Kavukcuoglu 2013; Gutmann and Hyvärinen 2010, 2012) has been shown to be an asymptotically consistent estimator of the MLE in the number of noise samples (Gutmann and Hyvärinen 2012), it is a principled approximation of our $E_{p_{LDA}(\text{data})}[p(\text{data};\mathbf{V})]$ objective. In practice, however, we obtained better performance using negative sampling (NEG) (Mikolov et al. 2013b),

4 We can also consider a model variant where $\theta^{(d)}$ is reparameterized using a log-bilinear model, however we obtained better performance by constructing document vectors based on topic vectors, as below.

Algorithm 1 Training NEA for LDA

Input: W = #Words, K = # Topics, D = # Documents,
M= Mini-batch size, trained LDA model $\mathbf{\Theta}_{LDA}$, $\mathbf{\Phi}_{LDA}$, Z
Output: $\mathbf{\Phi}_{NEA}$ = encoded $\mathbf{\Phi}_{LDA}$, $\boldsymbol{V}^{(W)\prime}$, $\bar{\boldsymbol{V}}^{(K)}$, and $\boldsymbol{V}^{(D)}$ are word, topic, and document embeddings, respectively

Embeddings steps:

- For each iteration t: //in practice, use mini-batches
  - Draw a document, $d \sim unif(D)$
  - Draw a topic, $z \sim \mathbf{\Theta}_{LDA}^{(d)}$
  - Draw a word, $w \sim \mathbf{\Phi}_{LDA}^{(z_d)}$
  - Update $[\bar{v}_z, v'_w]$:= NEG(in $= z$, out $= w$)
- For each document d in D:
  - For each token i in d:
    Update $v_d := v_d + \frac{\bar{v}_{z_{di}}}{|\bar{v}_{z_{di}}|}$
  - Normalize $v_d := \frac{v_d}{|v_d|}$

Smoothing steps: Calculate $\mathbf{\Phi}_{NEA} \propto exp(\mathbf{V}^{(W)\prime\intercal}\bar{\boldsymbol{V}}^{(K)})$

which further approximates the NCE objective as

$$\log \sigma({v'_w}^\intercal \bar{v}_z) + \sum_{i=1}^{k} E_{w_i \sim p_n(w)} \log \sigma(-{v'_{w_i}}^\intercal \bar{v}_z)) \ ,$$

where $p_n(w)$ is a "noise" distribution, and k is the number of "negative" samples from it per word. With the embeddings we recover NEA's "smoothed" encodings of the topics:

$$\mathbf{\Phi}_{NEA} \propto exp(\mathbf{V}^{(W)\prime\intercal}\bar{\boldsymbol{V}}^{(K)}) \ . \tag{7}$$

Finally, we construct document vectors by summing the corresponding (normalized) topic vectors according to the pre-trained LDA model's topic assignments Z, for each token of that document.[5] We normalize all document vectors to unit length to avoid any impact of the document length, producing the final document embeddings $\mathbf{V}^{(D)}$. The pseudocode for training NEA to mimic LDA is shown in Algorithm 1.

4.2 General NEA Algorithm

More generally, the NEA method can be extended to encode any topic model's parameters, which are typically conditional distributions given a single parent assignment, $P(a_i|parent(a_i))$, into *vector representations* $\mathbf{V}^{(i)}$, $\mathbf{V}^{(i)\prime}$ while also providing *smoothed versions* of the parameters $P_{NEA}(a_i|parent(a_i))$. We illustrate the model architecture to train NEA for general topic models in Figure 3. In the embedding steps, for each iteration, we draw samples a_i from the conditional discrete distributions for documents, authors, topics, words, etc., followed by updating the input and output vectors by

5 Note that other aggregation approaches like concatenation can also be used here, but for a large number of topics, the concatenation approach may encounter the curse of dimensionality issue.

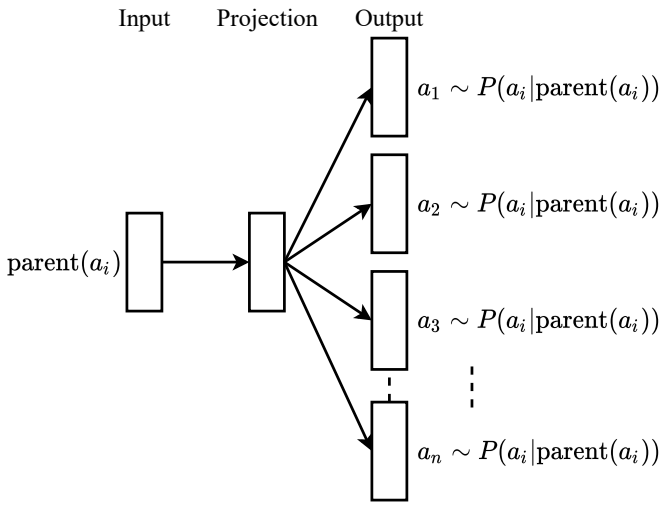

Figure 3
The model architecture of training NEA for general topic models of the form $P(a_0)\prod_{i=1}^{n} P(a_i|\text{parent}(a_i))$, where the a_i are discrete variables i.e. documents, authors, topics, words. The training objective is to learn vector representations for parent(a_i) that are good at predicting drawn samples a_i from the conditional discrete distribution $P(a_i|\text{parent}(a_i))$.

Algorithm 2 Training NEA for General Topic Models

Input: Trained topic model of the form $P(a_0)\prod_{i=1}^{n} P(a_i|\text{parent}(a_i))$, where the a_i are discrete variables i.e. documents, authors, topics, words
Output: Embeddings for each variable $\mathbf{V}^{(i)}$, $\mathbf{V}^{(i)\prime}$, smoothed distributions $P_{NEA}(a_i|\text{parent}(a_i))$

Embeddings steps:

- For each iteration t: //in practice, use mini-batches
  - sample $a_0 \sim P(a_0)$
  - For each random variable $a_i \in \{a_1 \dots a_n\}$:
    sample $a_i \sim P(a_i|\text{parent}(a_i))$
    update $[v^{(i)}_{\text{parent}(a_i)}, v^{(i)\prime}_{a_i}]$:= NEG(in=parent(a_i), out=a_i)

Smoothing steps:

- For each random variable $a_i \in \{a_1 \dots a_n\}$:
  - $P_{NEA}(a_i|\text{parent}(a_i)) \propto exp(v^{(i)\prime\intercal}_{a_i} v^{(i)}_{\text{parent}(a_i)})$

optimizing log-bilinear classification problems using negative sampling (discussed in Section 4.1). In the smoothing steps, we can recover the smoothed version of the parameters $P_{NEA}(a_i|parents(a_i))$ by the dot product of the corresponding input and output vectors learned in embeddings steps followed by a softmax projection onto the simplex. Our NEA algorithm for general topic models is shown in Algorithm 2.

4.3 Relationship Between NEA and the Embedded Topic Model

Now that we have described NEA, at this juncture it is worth comparing and contrasting it with another related approach, the embedded topic model (ETM) (Dieng, Ruiz, and Blei 2020). The ETM is a model which encodes topics via embedding vectors. Its assumed generative process is identical to the one shown in the bottom-left corner of Table 1. More concretely, the ETM parameterizes each topic with the inner product of vector representations for each word and its topic embedding, followed by a softmax function to produce word probabilities. The ETM further assumes a logistic normal prior on the document-topic proportions $\theta^{(d)}$ for each document d. Dieng, Ruiz, and Blei (2020) train the ETM via a variational inference algorithm.

The main similarity between NEA and the ETM is that when NEA is applied to standard LDA, the same underlying generative model is assumed, up to the prior (Table 1). However, there are a number of differences. First, while the ETM (Dieng, Ruiz, and Blei 2020) is a *model*, NEA is an *algorithm*. While the ETM has a dedicated inference algorithm which applies specifically to that model, NEA is a general-purpose algorithm for deconstructing any given topic model into an embedding representation. For example, in this paper we apply it to the author-topic model (ATM) (Rosen-Zvi et al. 2004) and to the mixed-membership skip-gram (Foulds 2018), in addition to LDA.

Furthermore, their learning algorithms differ significantly. To fit the ETM, we must solve a challenging inference problem over unobserved variables, which generally requires approximate inference algorithms, in this case variational inference. The learning problem is simpler for NEA since it fits to a pre-trained topic model. NEA optimizes its objective, which aims to reconstruct the target topic model, using a standard stochastic gradient descent method via negative sampling on simulated data from the pre-trained topic model. We speculate that its superior performance to the ETM in our experiments (cf. Section 5.1.1) is due in part to fewer issues with local optima (since the latent variables in the topic model are circumvented), and in part due to avoidance of the need to use a variational approximation.

5. Experiments

The goals of our experiments were to evaluate the NEA algorithm both for topic modeling and as a feature engineering method for classification tasks. We will release our source code upon acceptance.

We considered six datasets: the *NIPS* (a.k.a. *NeurIPS*) corpus with $1,740$ scientific articles from years 1987-1999 with 2.3M tokens and a dictionary size of 13,649 words, the *New York Times* corpus with 4,676 articles and a dictionary size of 12,042 words (another version of this corpus, denoted by *New York Times V2*, that contains 1.37M documents including all stop words is used for direct comparison to the ETM), *Bibtex*[6] containing 7,395 references as documents with a dictionary size of 1,643 words, the *Reuters−150* news wire articles corpus (15,500 articles with dictionary size of 8,349 words), *Ohsumed* medical abstracts (20,000 articles where document classes are 23 cardiovascular diseases), and a large *Wikipedia* corpus contained 4.6M articles with 811M tokens from the online encyclopedia with a dictionary of 7,700 words. Note that we removed stop words from all datasets as a standard pre-processing step except *New York Times V2* dataset.

6 http://mulan.sourceforge.net/datasets-mlc.html.

Table 2
Comparison to the results of Dieng, Ruiz, and Blei (2020) on *New York Times V2 with stop words*.

	NPMI	Diversity	Quality
LDA	0.13	0.14	0.0173
MHW LDA	0.15	0.10	0.0152
Δ-NVDM	0.17	0.11	0.0187
Labeled ETM	0.18	0.22	0.0405
NEA	**0.26**	**0.27**	**0.0693**

Table 3
Several example topics for LDA (trained via MHW (Li et al. 2014)) and their corresponding NEA reconstructions, *New York Times V2 with stop words* corpus, where models are trained for K=300. NEA improves the quality of the topics by clustering the stop words as separate topics (see last topic, italicized) instead of mixing them with the other topics, unlike LDA.

LDA	NEA	LDA	NEA	LDA	NEA	LDA	NEA	LDA	NEA
the	republicans	the	book	the	health	and	wine	*the*	*of*
to	democrats	of	books	health	patients	the	restaurant	*of*	*is*
republican	republican	and	read	to	doctors	with	restaurants	*and*	*in*
state	senator	to	author	of	medical	of	dishes	*in*	*the*
for	senate	book	write	for	hospitals	is	food	*to*	*be*
mr	democrat	is	authors	and	care	in	dinner	*for*	*and*
senate	democratic	in	reading	care	drug	at	bar	*is*	*on*
on	election	books	pages	in	drugs	to	menu	*on*	*for*
democrats	governor	that	magazine	medical	patient	wine	street	*as*	*at*
republicans	campaign	it	readers	drug	hospital	restaurant	wines	*at*	*as*

5.1 Performance for LDA

We start our analysis by evaluating how NEA performs at mimicking and improving LDA topic models. We fixed LDA's hyperparameters at α=0.1 and β=0.01 when K<500, otherwise we used α=0.01 and β=0.001. We trained LDA via the Metropolis-Hastings-Walker algorithm (MHW) (Li et al. 2014), due to its scalability in K. In NEA, negative sampling (NEG) was performed for 1 million minibatches of size 128 with 300-dimensional embeddings. We also considered an *ensemble* model where each topic is chosen between LDA and its corresponding NEA reconstruction, whichever has the highest coherence.

5.1.1 Comparison to Embedded Topic Model (ETM), LDA, and Other Baselines. For a direct comparison to the reported results for our strongest baseline, the embedded topic model (ETM) (Dieng, Ruiz, and Blei 2020) (a model which parameterizes a topic model similar to LDA with embeddings and is trained via variational inference), we first study the performance of NEA on another version of the New York Times corpus (*New York Times V2*) which contains 1.37M documents with a dictionary size of 10,283 including all stop words.[7] In Table 2 we directly report the results from Dieng, Ruiz, and Blei (2020) for LDA, the Δ-NVDM (a variant of the multinomial factor model of documents), and the labeled ETM (a variant of the ETM with pre-trained embeddings)

7 The *New York Times V2 with stop words* corpus was obtained from https://github.com/adjidieng/ETM.

Table 4
Randomly selected topic pairs from LDA and NEA, and their corresponding Topic Coherence (TC) score, with LDA trained on the *NIPS* corpus for K=2,000.

LDA	NEA	LDA	NEA	LDA	NEA	LDA	NEA
TC: -3.016	TC: **-1.270**	TC: -1.577	TC: **-1.272**	TC: -2.578	TC: **-1.376**	TC: -1.584	TC: **-1.316**
bayesian	bayesian	images	images	phrase	sentences	regression	regression
prior	bayes	image	image	sentences	phrase	linear	linear
bayes	posterior	recognition	visual	clause	structure	ridge	ridge
posterior	priors	vision	recognition	structure	sentence	quadratic	quadratic
framework	likelihood	pixel	pixels	sentence	clause	squared	variables
priors	prior	techniques	pixel	phrases	activation	nonparametric	nonparametric
likelihood	framework	pixels	illumination	syntactic	connectionist	dimensionality	squared
bars	note	visual	intensity	connectionist	phrases	variables	multivariate
note	probability	computed	pairs	tolerance	roles	smoothing	kernel
compute	bars	applied	matching	previous	agent	friedman	basis

with K =300.[8] Specifically, Dieng, Ruiz, and Blei (2020) report the normalized pointwise mutual information (NPMI) (Lau, Newman, and Baldwin 2014) coherence metric, topic diversity (the percentage of unique words in the top 25 words of all topics), and overall "topic quality" (simply the product of NPMI and diversity). NEA (trained to mimic MHW LDA – an approximation of LDA for scalability in number of topics) clearly outperformed the state-of-the-art ETM and the other baselines on all metrics on NYT in the presence of stop words. Its success was due in part to clustering the stop words as separate topics rather than mixing the stop words with the other topics. In Table 3, we show that NEA forms high quality topics while clustering the stop words as separate topics rather than mixing the stop words with the other topics (see last topic, italicized), as in LDA.

As clustering the stop words as separate topics is also one of the advantages for the ETM model, we found that NEA is better than the ETM on this task in terms of performance measurements in Table 2. Note that even though the model parameterization between "NEA mimicking MHW LDA" and the ETM is the same, the training algorithm is very different. Since NEA fits to a pre-trained topic model, the learning problem is easier, and is likely less vulnerable to local optima. The ETM also needs to make a variational approximation which may hurt its performance.

5.1.2 Quality of Topics. In the previous section, we demonstrated that NEA constructs high quality topics on *New York Times V2* corpus by disallowing the stop words to be mixed with other topics and by clustering stop words as separate topics. Here, we performed a comprehensive analysis on the other datasets, where stop words were removed, to better understand the performance of NEA. The analysis in this section demonstrates the advantage of NEA over LDA even if a dataset does not contain any stop words.

To get a quantitative comparison, we compared the topics' UMass coherence metric, which measures the semantic quality of a topic based on its T most probable words (we choose T=10 words), thereby quantifying the user's viewing experience (Mimno et al. 2011). Larger coherence values indicate greater co-occurrence of the words, hence higher quality topics. Coherence is very closely related to NPMI but is simpler, more

8 We directly compare our NEA model with the reported performance of Δ-NVDM and labeled ETM in Table 4 of the Dieng, Ruiz, and Blei (2020) paper on the exactly same dataset.

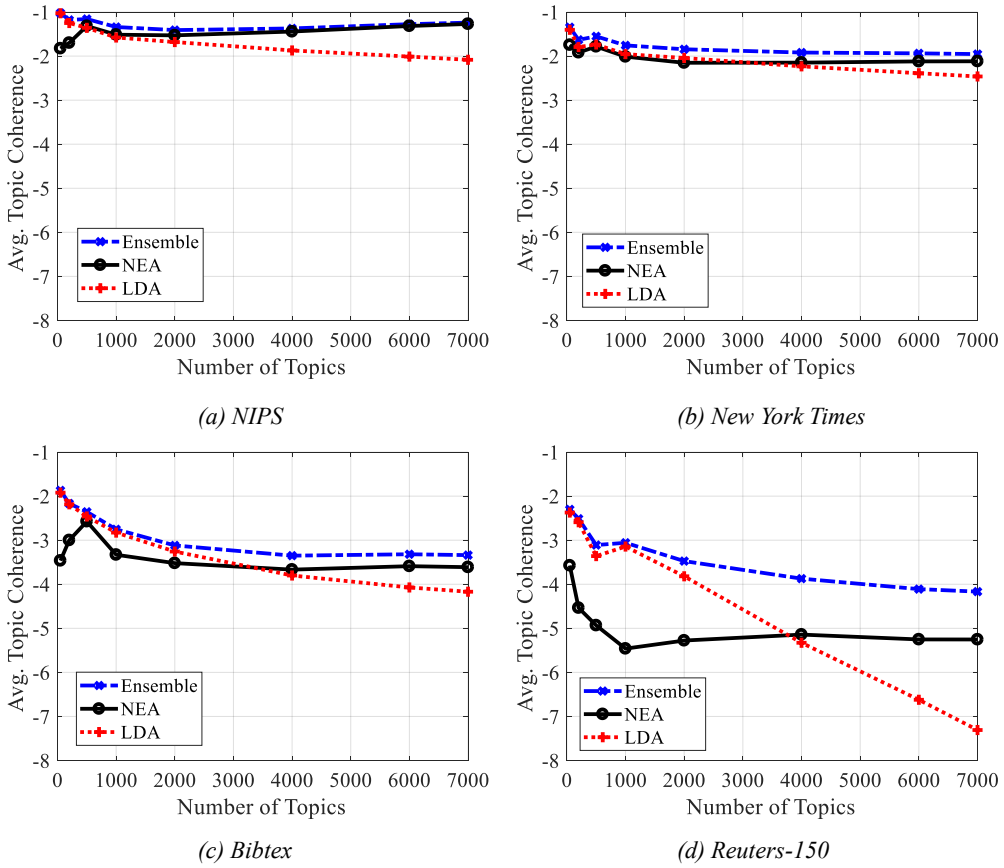

Figure 4
Comparison of average topic coherence $vs.$ number of topics K on: (a) *NIPS*, (b) *New York Times*, (c) *Bibtex*, and (d) *Reuters-150*. The ensemble model chooses the best topic between NEA and LDA.

widely used, and correlates similarly with human judgment. In Figure 4, the average topic coherence of LDA, NEA and their ensemble model[9] is shown with respect to the number of topics K. LDA works well with small K values, but when K becomes large, NEA outperforms LDA in average topic coherence scores on all datasets. Since the ensemble model chooses the best topics between NEA and LDA, it always performs the best.

We also conduct qualitative analysis which is complementary to our experiment in Figure 4. Instead of reporting results for the wide range of number of topics K, as in Figure 4, we pick some example K for the ease of demonstration in the qualitative analysis. First, we found that NEA generally recovers the same top words for LDA's "good topics." Most of the topics produced by both the LDA and NEA models are interpretable, and NEA was able to approximately recover the original LDA's topics. In Table 4, we show a few randomly selected example topics from LDA and NEA, while

9 Each topic is chosen between LDA and its corresponding NEA reconstruction, whichever has the highest coherence

Table 5
The worst four topics produced by LDA, in terms of per-topic coherence score, and their corresponding NEA topics, and their Topic Coherence (TC) score, with LDA trained on the *NIPS* corpus for K=7,000.

LDA	NEA	LDA	NEA	LDA	NEA	LDA	NEA
TC: -8.184	TC: **-1.204**	TC: -8.023	TC: **-1.062**	TC: -7.984	TC: **-1.390**	TC: -7.787	TC: **-0.798**
corresponds	parameters	symbolics	values	ryan	learning	paths	total
change	important	addressing	case	learning	methods	close	paths
cut	neural	choice	increase	bit	text	path	global
exact	change	perturbing	systems	inhibited	space	make	path
coincides	results	radii	rate	nice	combined	numbering	time
duplicates	report	centered	point	automatica	averaging	channels	fixed
volatility	cut	damping	feedback	tucson	area	rep	function
trapping	multiple	merits	input	infinitely	apply	scalars	yields
reading	experiments	vax	reduces	stacked	recognition	anism	close
ters	minimizing	unexplored	stage	exceeded	bit	viously	computation

Table 6
The worst four topics produced by NEA, in terms of per-topic coherence score, and their corresponding LDA topics , and their Topic Coherence (TC) score, with LDA trained on the *NIPS* corpus for K=7,000.

LDA	NEA	LDA	NEA	LDA	NEA	LDA	NEA
TC: **-2.027**	TC: -4.690	TC: **-2.615**	TC: -3.737	TC: **-2.433**	TC: -3.712	TC: **-3.183**	TC: -3.696
blake	models	strain	structure	insertion	space	learning	learning
condensation	exp	mars	length	hole	reinforcement	steps	steps
isard	blake	yield	variance	gullapalli	learning	computer	computer
models	similar	rolling	equal	reinforcement	fig	testing	testing
observations	condensation	mill	mars	smoothed	insertion	observation	people
entire	modified	cart	strain	reactive	hole	predetermined	bin
oxford	generally	tuning	weight	extreme	fit	cheng	observation
rabiner	cortical	material	intelligence	ram	gullapalli	utilizes	efficient
gelb	isard	friedman	cycle	gordon	maximum	efficient	utilizes
north	consisting	plot	friedman	consecutive	regions	updating	birth

LDA was trained on the NIPS corpus for K=2,000. In Table 5, we show the four worst topics from LDA, based on per-topic coherence score, and their corresponding NEA topics, on *NIPS* for K=7,000. In this case, NEA generated noticeably more meaningful topics than LDA. For fairness, we also show the four worst topics reconstructed by NEA, based on per-topic coherence score, and their corresponding LDA-generated topics for the same model in Table 6. In this case, LDA perhaps generates slightly more meaningful topics than NEA, although the relative performance is somewhat subjective. Note that in practice, we can always use the ensemble approach, choosing the best topic between LDA and NEA based on coherence.

In Table 7, we show the 4 topics with the largest improvement in coherence scores by NEA, for *Reuters−150* with 7,000 topics. We observe that these LDA topics were uninterpretable, and likely had very few words assigned to them. NEA tends to improve the quality of these "bad" topics, e.g. by replacing noisy or stop words with more semantically related ones. In particular, we found that NEA gave the most improvement for topics with few words assigned to them (see Figure 5 (left)) and when K becomes large, the majority of topics have few assigned words (see Figure 5 (right)). As a result, NEA improves the quality of most of the topics.

To further study this phenomenon, we showcase the improvement of "bad topics," those which have less than 200 words assigned to them, by the NEA model for all

Table 7
The four topics that were most improved by NEA over the original LDA topic, in terms of the difference between per-topic coherence score, with LDA trained on the *Reuters-150* corpus for K=7,000.

LDA	NEA	LDA	NEA	LDA	NEA	LDA	NEA
TC: -18.928	TC: **-2.601**	TC: -19.367	TC: **-3.120**	TC: -20.805	TC: **-4.844**	TC: -17.906	TC: **-2.035**
share	International	tonnes	announced	blah	blah	dlrs	debt
pittsburgh	common	yr	tonnes	aa	company	aa	canadian
aa	share	aa	addition	aaa	account	aaa	today
aaa	pittsburgh	aaa	asked	ab	advantage	ab	canada
ab	general	ab	accounts	abandon	acquisitions	abandon	decline
abandon	agreement	abandon	shares	abandoned	loss	abandoned	competitive
abandoned	tender	abandoned	surplus	abc	proposed	abc	conditions
abc	market	abc	secretary	abdul	considered	abdul	dlrs
abdul	june	abdul	heavy	aberrational	announced	aberrational	price
aberrational	dividend	aberrational	held	abide	base	abide	week

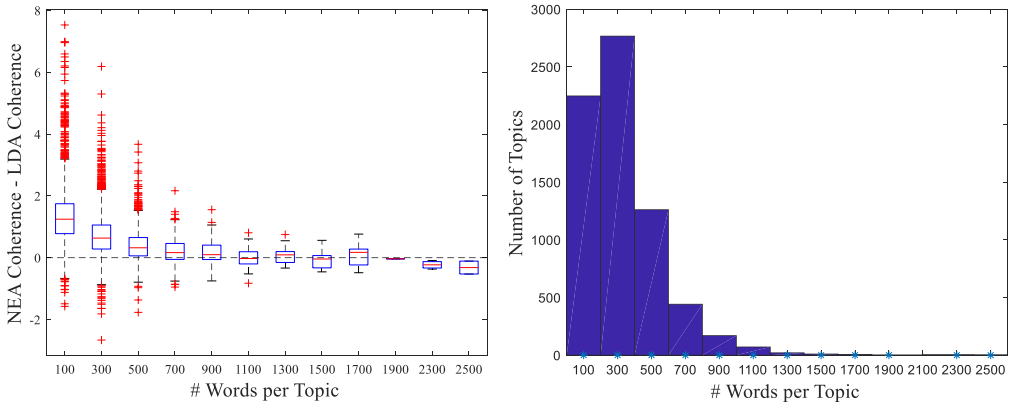

Figure 5
Improvement in coherence of NEA over LDA *vs.* number of words in the topic, K=7,000, *NIPS* dataset. The boxplot (left) shows the coherence improvement versus number of words per topic while the histogram (right) shows the number of topics in each bin.

datasets in Figure 6. For such topics (i.e. those with less than 200 words assigned to them), NEA leads to an improvement in coherence in almost all of the cases.

5.1.3 Performance for LDA Model on Big Data (Wikipedia). In this experiment, we evaluated NEA in a big data setting, using the *Wikipedia* corpus with K=10,000 topics. We scaled up LDA using a recent online inference algorithm for high-dimensional topic models, called SparseSCVB0 (Islam and Foulds 2019), which leverages both stochasticity and sparsity. The big data LDA model was trained on Wikipedia for 72 hours using SparseSCVB0 while NEA was trained on the SparseSCVB0 parameters for 24 hours with 128-dimensional embeddings. In Table 8, we see that NEA improves SparseSCVB0's average topic coherence and topic diversity on the *Wikipedia* dataset.

5.2 Performance for Author-Topic Model (ATM)

The author-topic model (ATM) (Rosen-Zvi et al. 2004) is useful for applications such as automated reviewer recommendations which could benefit from NEA smoothing. We

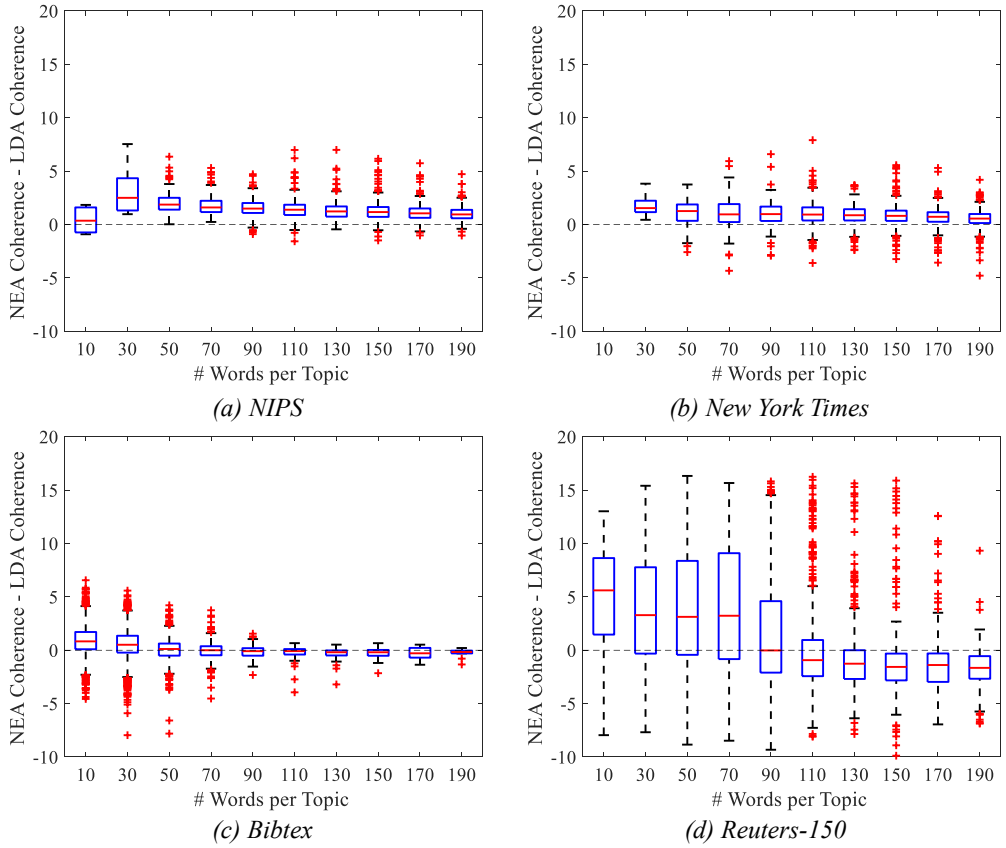

Figure 6
Improvement of bad topics (less than 200 assigned words) by NEA *vs.* number of words in the topic, K=7,000, for (a) *NIPS*, (b) *New York Times*, (c) *Bibtex*, and (d) *Reuters-150* corpus.

Table 8
Comparison of NEA with SparseSCVB0 on *Wikipedia* corpus for K=10,000 topics.

Models	Coherence	Diversity
SparseSCVB0	-2.264	0.029
NEA	**-2.204**	**0.031**

trained NEA for the author-topic model with the same hyperparameters used in Section 5.1.2. Since these applications are based on searching for similar authors, we can treat them as a ranking problem. Following Rosen-Zvi et al. (2004), we rank based on the *symmetric KL-divergence* between authors i and j:

$$sKL(i,j) = \sum_{t=1}^{K} [\theta_{it} \log \frac{\theta_{it}}{\theta_{jt}} + \theta_{jt} \log \frac{\theta_{jt}}{\theta_{it}}], \tag{8}$$

where θ_i is the ith author's distribution over topics. Using this distance metric, we searched over authors who wrote at least 5 papers in the full *NIPS* corpus – there are

Table 9
Mean reciprocal rank for co-author retrieval task on *NIPS* corpus.

Random Chance	ATM	NEA-Embed	Tf-idf	NEA-Smooth
0.043	0.083	0.086	0.091	**0.106**

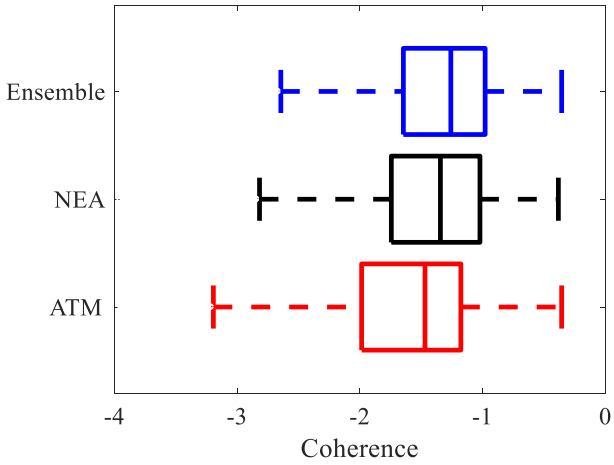

Figure 7
Per-topic coherence for the ATM, NEA-smoothed ATM, and ATM+NEA ensemble on *NIPS*, K=1,000 topics.

125 such authors out of the full set of 2037 authors. We reported the mean reciprocal rank (MRR) based on the rank of the most similar co-author from these 125 authors. To calibrate the results, we also construct a random chance baseline by simulating the ordering of the 125 authors using random permutation for 1 million runs. Table 9 shows the improvement in MRR using author vectors generated from NEA over the author-topic parameters of the ATM. Further improvement was achieved by the NEA-smoothed version of the ATM's parameters which also outperformed author vectors generated from a tf-idf baseline.

Similarly to LDA, Figure 7 shows that NEA improves the ATM's topics while the ensemble model outperforms NEA in terms of per-topic coherence, on the *NIPS* corpus with K=1,000.

5.3 Performance for Mixed Membership Skip-Gram Topic Model (MMSGTM)

We trained NEA for the mixed membership skip-gram topic model (MMSGTM) (Foulds 2018) using the same hyperparameter values as in previous experiments, while setting MMSGTM-specific hyperparameters to the values suggested by Foulds (2018). The original MMSG algorithm learns topic embeddings based on the MMSGTM's topic assignments Z, while NEA uses simulated data from the topic model. NEA is arguably more principled than the algorithm of Foulds (2018) due to its global variational objective. We found that NEA smooths and improves the speed of the training process (shown in Figure 8), while greatly reducing memory requirements as the topic assignments Z need not be stored for NEA.

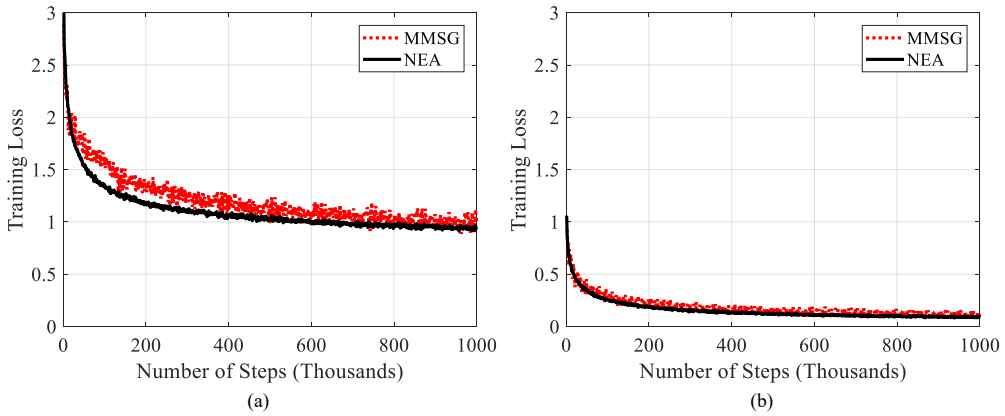

Figure 8
NEG loss of NEA and MMSG training for MMSGTM. K=1,000 topics on (a) *NIPS* and (b) *Reuters-150*.

Table 10
Comparison of NEA, when trained based on the MMSGTM, with MMSG in document categorization accuracy on *Reuters−150*, and *Ohsumed* datasets.

Datasets	MMSGTM	MMSG	NEA
Reuters-150	66.97	67.72	**68.59**
Ohsumed	32.41	33.63	**34.89**

Table 11
Comparison of NEA, when trained based on the LDA, with several other embedding methods in document categorization accuracy on *Reuters−150*, and *Ohsumed* datasets.

Datasets	#Classes	#Topics	Doc2Vec	LDA	NEA	CNN	SG	SG+LDA	SG+NEA
Reuters-150	116	500	55.89	64.26	67.15	69.43	70.80	69.13	**72.29**
Ohsumed	23	500	34.02	32.05	34.38	27.17	37.26	37.33	**38.88**

Table 12
Comparison of NEA, when trained based on the LDA, and combined with tf-idf features in document categorization accuracy on *Reuters−150*, and *Ohsumed* datasets.

Datasets	#Classes	#Topics	Tf-idf	Tf-idf+LDA	Tf-idf+SG	Tf-idf+NEA	Tf-idf+SG+NEA
Reuters-150	116	500	73.00	73.01	72.99	**73.14**	73.09
Ohsumed	23	500	43.07	43.05	43.04	**43.11**	43.08

5.4 Downstream Task: Document Categorization

In this set of experiments, we tested the performance of the learned vectors using NEA's document embeddings $\mathbf{V}^{(D)}$ as features for document categorization/classification. We used two standard document categorization benchmark datasets: *Reuters−150*, and

Ohsumed.[10] We used the standard train/test splits from the literature (e.g. for Ohsumed, 50% of documents were assigned to training and to test sets). Note that we also held-out 20% documents from training data as validation set to select hyper-parameters including the number of topics and embedding size via grid search in terms of accuracy on validation set. The documents in the validation set were merged again with the training data after completing hyper-parameter tuning. Logistic regression classifiers were trained on the features extracted on the training set for each method while classification accuracy was computed on the held-out test data. Continuing from the previous section, we first evaluated document categorization with NEA and MMSG, where both models were trained based on the MMSGTM (Table 10). Both NEA and MMSG improved document categorization accuracy compared to the MMSGTM on *Reuters−150* and *Ohsumed*, while NEA performed the best.

We studied NEA's performance in more detail in the context of LDA, which was trained with the same hyperparameters used in Section 5.1.2. We compared NEA with LDA and several popular models such as the skip-gram (SG) (Mikolov et al. 2013a,b), paragraph vector (Doc2Vec) (Le and Mikolov 2014), and a convolutional neural network (CNN) (Kim 2014). All the baseline models were trained using reported hyperparameters in the corresponding literatures. The results are given in Table 11.

We found that NEA had better classification accuracy than LDA and Doc2Vec. The CNN showed inconsistent results, outperforming Doc2Vec, LDA, and NEA on *Reuters−150*, but performing very poorly for *Ohsumed*. In NEA, the document vectors are encoded at the topic level rather than the word level, so it loses word-level information, which turned out to be beneficial for these specific classification tasks, at which SG features outperformed NEA's features. Interestingly, however, when both SG and NEA features were concatenated (SG + NEA), this improved the classification performance over each model's individual performance. This suggests that the combination of topic-level NEA and word-level SG vectors complement the qualities of each other and both are valuable for performance.

Note that tf-idf, which is notoriously effective for document categorization, outperformed all embeddings. In Table 12, we show the results when concatenating tf-idf with the other feature vectors from LDA, SG, and NEA, which in many cases improved performance over tf-idf alone. We observed the highest improvement over tf-idf for both document categorization tasks when we concatenated NEA vectors with tf-idf (Tf-idf + NEA), although the difference was not statistically significant. This may be because the *topical information* in NEA features is complementary to tf-idf, while SG's *word-based* features were redundant, and hence actually reduced performance. While the differences in accuracy between the NEA-concatenated features and the best baselines (SG, Tf-idf) were not statistically significant, the NEA-concatenated features X + *NEA* outperformed the unconcatenated features X in all 6 cases in Tables 11 and 12 ($X \in \{\text{SG}, \text{Tf-idf}, \text{TF-idf+SG}\}$, computed over both datasets). The improvement of X + *NEA* over X was statistically significant according to a Wilcoxon signed-rank test (2-sided, $p < 0.05$).

5.5 Case Study: Application to Mitigating Sociolinguistic Bias in Author Embeddings

We conducted a case study to demonstrate the practical use and benefits of the NEA method. The goal of the study was to investigate its use in identifying and mitigating

10 Document categorization datasets available at http://disi.unitn.it/moschitti/corpora.htm.

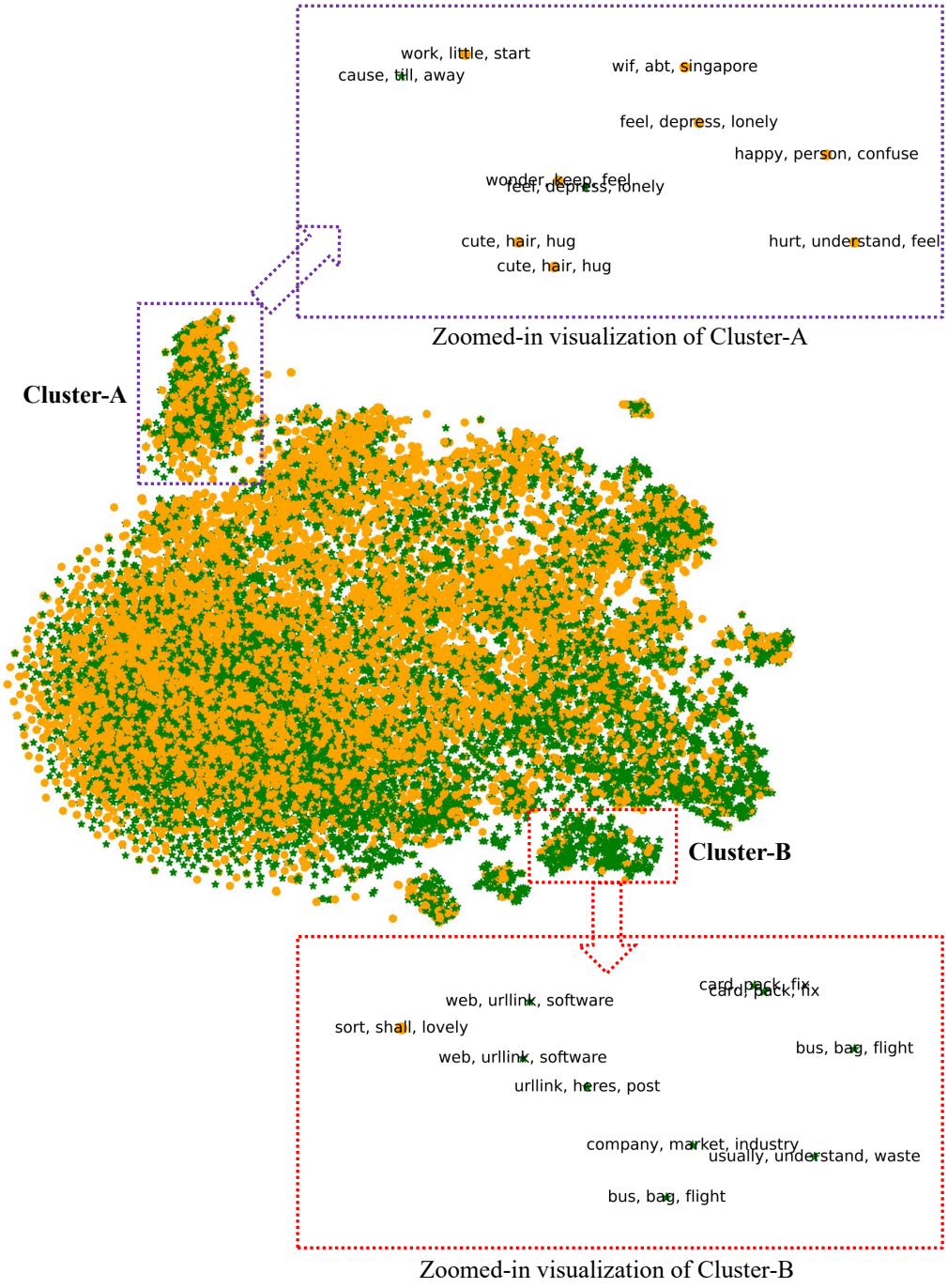

Figure 9
NEA embeddings of authors for the *Blog Authorship* corpus, projected via *t*-SNE, color-coded by the authors' gender: *orange dot* = female authors, and *green asterisk* = male authors. The zoomed-in visualizations of Cluster-A (predominately female authors) and Cluster-B (predominately male authors) show the 10 authors nearest to the cluster centers, annotated by the top words in each author's nearest topic to illustrate the distinct topical trends in these clusters.

gender bias in natural language processing models (Bolukbasi et al. 2016; Caliskan, Bryson, and Narayanan 2017; Gonen and Goldberg 2019). Differing patterns of language usage which are correlated with protected characteristics such as gender, race, age, and nationality can facilitate unwanted discrimination, called *sociolinguistic bias*, and this can be encoded by machine learning models (Deshpande, Pan, and Foulds 2020). To address this issue, our approach was to learn representations of the bloggers which encode their salient topical interests but not irrelevant gender information, a task known as *fair representation learning* (Zemel et al. 2013). Such debiased representations would potentially be valuable for fairness in recommendation systems, resume filtering for hiring purposes, information retrieval, etc. For this experiment, we used the Blog Authorship corpus (Schler et al. 2006) which consists of 681,288 posts of 19,320 bloggers (approximately 35 posts per person) with their self-provided gender information (*male* or *female*). Similarly to our experiments above, we trained an ATM on the dataset with K=1,000 topics, and then trained NEA to mimic the ATM. We then used the NEA model for both visualization and debiasing purposes, as discussed below.[11]

5.5.1 Visualizing NEA Embeddings for Male and Female Authors. We first used the NEA embeddings to visualize blog posts written by males and females, and thus expose any differences in their distributions which may potentially be a source of bias in downstream machine learning models. Since NEA allows us to learn latent vector-space embeddings for words, topics, documents, authors, etc., we can analyze the demographic differences with corresponding embeddings. In Figure 9, we show the t-SNE projected NEA embeddings for authors to explore the relationship between them in terms of gender: male (*green asterisks*) and female (*orange dots*). We found that there is some partial separation between the male and female author embeddings in the t-SNE space, indicating that there are indeed systematic differences in topics between male and female authors. For example, more female authors are located on the upper region and more male authors are located on the lower right region of the t-SNE space, though many authors regardless of their gender partially overlapped as well, particularly in the middle of the t-SNE space.

We also found several clusters of authors in the t-SNE space which were dominated by a particular gender. As shown in Figure 9, we investigated this phenomenon by selecting two clusters to inspect more closely: *Cluster-A*, which was dominated by female authors, and *Cluster-B*, which was dominated by male authors. For each cluster, to show the zoomed-in visualizations, we picked the top-10 nearest authors from the cluster centre and annotated them with the most similar topics of these authors in terms of the Euclidean distance between NEA-generated topics and corresponding author embeddings. These female- and male-author dominated clusters (Cluster-A and Cluster-B, respectively) showed a distinct topical trend on the t-SNE space. As seen in the figure, Cluster-A's authors are close to topics relating to emotions, while Cluster-B's authors are close to topics relating to the internet, business, and travel. Furthermore, the Cluster-A topics may not be salient to the authors' topical interests in a downstream task, while potentially acting as a proxy variable for gender which could encode sociolinguistic bias (Barocas and Selbst 2016). *The NEA embedding-based visualization was thus helpful for showing that debiasing interventions are likely to be impactful for this dataset.*

11 This analysis is an observational study on one particular dataset. Therefore, our results should not be used to support any claims about the nature of gender differences, their causes, or their implications.

Table 13
The top seven "most male" and "most female" topics on the *Blog Authorship* corpus, computed based on each topic's NEA embedding's dot product with the gender bias direction. Identifying the most gender-associated topics via NEA has potential future uses in mitigating gender bias in machine learning models.

Male	soldier	kerry	efforts	web	fox	company	america
	government	campaign	element	urllink	republican	market	attack
	iraq	election	useful	software	mission	industry	americans
	report	dean	particularly	service	draft	successful	nation
	unite	chief	conference	program	map	sales	political
	foreign	moore	merely	windows	goals	increase	source
	fund	political	effectively	page	pursue	benefit	conservative
	military	bush	simply	google	resolution	hire	politics
	weapons	vote	television	source	agent	manager	evidence
	countries	state	expression	network	expand	offer	generation
Female	care	hehe	shes	cuz	girls	bout	haha
	stop	wow	girlfriend	lil	friend	bday	den
	things	rain	shell	kinda	dance	hav	dun
	figure	ugh	girl	tho	boyfriend	luv	dunno
	mind	yep	flower	goin	guy	sum	sch
	days	hahahaha	shed	wanna	fun	talkin	lor
	happy	hehehe	theyll	lol	conversation	nite	wad
	dont	absolutely	hes	omg	night	jus	rite
	mean	whew	birthday	mad	wait	wit	wanna
	ill	geez	girls	gotta	talk	nothin	sia

5.5.2 Most Gendered Topics Per Bias Direction. We can analyze the demographic bias by computing a bias direction (Bolukbasi et al. 2016; Dev and Phillips 2019; Islam et al. 2019, 2021) with respect to the protected attribute, e.g. gender or race. Following Islam et al. (2021), we constructed overall male (v_m) and female (v_f) vectors by taking the average of NEA-generated author embeddings for male and female authors, respectively, and hence computing the overall gender bias direction, $v_B = \frac{v_m - v_f}{||v_m - v_f||}$. To get the "most male" and "most female" topics, we computed the dot products of NEA topic embeddings with v_B and sorted them accordingly. Note that the biggest and smallest dot products are associated with male and female vectors, respectively. Finally, we report the top male and female topics in Table 13. The results show that male and female authors tend to use very different topics with very distinct content, suggesting that there are subtle (and not-so-subtle) differences in the use of language between authors of different genders. For instance, the top male topics identified by the method were related to the military, politics, business, and computers, while the selected top female topics were related to moods, interpersonal relations, and informal language.

The NEA method allowed us to identify the gender-associated topics, information which potentially can help to mitigate gender bias in topic models, and other machine learning models trained on this dataset. For instance, the most gender-associated topics could be removed from the set of features used by a classifier or recommendation system. Potential application of this debiasing approach include fair recommendation of bloggers to followers, or more generally, combating discrimination in AI-based resume filtering for hiring (Deshpande, Pan, and Foulds 2020) and reducing gender bias in a model for recommending which articles should be cited by a new manuscript.

Table 14
Performance and fairness measures for downstream task using logistic regression (LR) models on original and debiased representations for ATM's author-topic distributions and NEA-generated author embeddings. In the case of bloggers categorization, higher is better (↑) for accuracy and $p\%$-Rule while lower is better (↓) for ϵ-DF and δ-DP. In the case of gender prediction, lower accuracy is better, as this corresponds to a debiased representation.

Models	Gender Prediction	Bloggers Categorization			
	Accuracy ↓	Accuracy ↑	ϵ-DF ↓	δ-DP ↓	$p\%$-Rule ↑
LR on original author-topic distributions	0.743	**0.387**	0.436	0.041	64.664
LR on debiased author-topic distributions (gendered topics removed)	0.738	0.384	0.384	0.039	68.119
LR on original author embeddings	0.684	0.370	0.459	0.033	63.185
LR on debiased author embeddings (gendered topics removed)	0.672	0.372	0.429	0.029	65.133
LR on debiased author embeddings (linear projection)	**0.457**	0.373	**0.183**	**0.013**	**83.270**

5.5.3 Mitigating Gender Bias in Blog Author Embeddings. In this experiment, we demonstrate a debiasing approach to mitigate gender bias in NEA-generated author embeddings. The *Blog Authorship* corpus contains the categories of bloggers with respect to the contents in their blog posts. There are 40 categories of bloggers such as advertising, arts, banking, education, engineering, fashion, law, religion, science, sports, technology, and so on. Our goal was to learn representations of blog authors which captured content information, i.e. by being predictive of the category labels, without encoding irrelevant gender information exposing sociolinguistic bias.

Our approach to debias author embeddings adapts recent work on attenuating bias in word vectors (Dev and Phillips 2019) to author-level debiasing. In Section 5.5.2, we showed how to compute the gender bias direction v_B, and hence identify the most gendered topics. The simplest approach we considered was to simply remove the most gendered topics from the topic model. From there, we can use the topics as features, or learn NEA embeddings from the modified topic model. Alternatively, a slightly more sophisticated approach, following (Dev and Phillips 2019), is to debias the original NEA-generated author embeddings $\mathbf{V}^{(A)}$ according to a linear projection of each author vector orthogonally onto v_B (computed in Section 5.5.2), which identifies the "bias component" of $\mathbf{V}^{(A)}$. We then achieve debiased author embeddings $\mathbf{V}^{(A)\prime}$ by subtracting its bias component as follows:

$$\mathbf{V}^{(A)\prime} = \mathbf{V}^{(A)} - (\mathbf{V}^{(A)} \cdot v_B) v_B \, . \tag{9}$$

For our fair representation learning task, we desire that the resulting blog author embeddings are not predictive of the authors' gender, while being predictive of the authors' category label. We first split dataset so that train and test sets contain 75% and 25% of authors, respectively. Logistic regression (LR) classifiers were trained on the original and debiased author embeddings for the training set, to predict category labels and to predict gender. Classification accuracy for category and gender were measured on the held-out test data. To evaluate the classifier models in terms of fairness regarding the category label, we additionally considered several fairness metrics including differential fairness ϵ-DF (Foulds et al. 2020) which handles multi-class classification, demographic parity δ-DP (Dwork et al. 2012; Zemel et al. 2013) which ensures similar outcome probabilities for each protected group, and the $p\%$-Rule (Zafar et al. 2017) which generalizes the 80% rule of the U.S. employment law (Biddle 2006). Note that δ-DP and the $p\%$-Rule were originally defined for binary outcome variables. In our multi-class problem,

we computed these metrics for each category, and inspired by the definition of ϵ-DF, reported the overall measures by taking the worst case among all categories of bloggers.

Table 14 shows accuracy and fairness metrics on held-out data for the LR models trained on the different feature sets. These features were: the ATM's original (i.e., not debiased) author-topic distribution features and their debiased version (via removing the top-10 most male and most female topics), the original NEA-generated author embeddings, and the two debiased NEA embedding approaches (the simple method which removes the most gendered topics prior to constructing author embeddings, and our gold standard version which uses the linear projection of authors in Equation 9). We found that the LR model trained on our linear projection-based debiased author embeddings was the fairest model. It substantially outperformed all other models in terms of all fairness metrics, and as desired, it had the lowest accuracy in predicting gender compared to the other models which indicates lower dependence between the authors' representations and their gender. Although the LR model with ATM's original author-topic distributions had the highest accuracy for blogger categorization, gender prediction accuracy was also highest, which is undesirable in this context. Moreover, this model performed worse in terms of the fairness metrics compared to all of the debiased models. Finally, we also found that LR models with our debiased author representations slightly improved the bloggers categorization accuracy compared to the LR model with the original author embeddings. It may seem counter-intuitive that debiasing improves accuracy, but this likely occurred because fairness interventions can reduce overfitting, which in some cases can result in improved accuracy due to improved generalization to unseen data (Keya et al. 2021; Islam, Pan, and Foulds 2021). In this case, removing gendered topics which are irrelevant to the categorization class labels was a win-win for both accuracy and fairness.

6. Related Work

Some prior research has aimed to combine aspects of topic models and word embeddings. The Gaussian LDA model (Das, Zaheer, and Dyer 2015) tries to improve the performance of topic modeling given the semantic information encoded in word embeddings, however the topics do not inform the embeddings. The reverse is true for the topical word embedding model (Liu et al. 2015) which uses LDA topic assignments of words to improve the resultant word embedding. The Skip-gram Topical Embedding (STE) model (Shi et al. 2017) aims to learn both embeddings and topics jointly by conditioning the embeddings on topics as well as words, and alternating between the updates for each model component in an EM-style algorithm. To benefit from neural networks there are other neurally-inspired ways of combining LDA and embeddings such as mixing the likelihood of LDA with the skip-gram model (Nguyen et al. 2015), learning word vectors jointly with document-level distributions of topic vectors (Moody 2016) or neural variational inference based continuous dense document representations (Miao, Yu, and Blunsom 2016). The main similarity between NEA and the above methods is that they each incorporate word embeddings and topics within a single model or algorithm. These approaches aim to achieve synergy between these models by either conditioning a topic model on word embeddings, conditioning a word embedding model on topics, or joint modeling and training of both word embedding and topic models together.

In contrast to these methods, instead of conditional or joint modeling of embeddings and topics, our NEA methodology views embeddings and topic distributions as alternate representations of the same model. The idea of parameterizing a topic model

via embeddings was previously used by the embedded topic model (ETM) (Dieng, Ruiz, and Blei 2020), a variant of neural network-based topic models that trains using vector representations of both words and topics, and by the mixed membership skip-gram (MMSG) (Foulds 2018). Our goals and methods are somewhat different. The ETM and MMSG are *models* with their own specific topic modeling architecture, while NEA is an *algorithm* which applicable to general LDA-style topic models. By learning to re-represent a topic model as an embedding model, our NEA method uses this representation to smooth the topics of a given topic model to improve coherence. It also produces topical embeddings which encode information which may be complementary to traditional neural network-based word embeddings (Mikolov et al. 2013b).

The utility of word embeddings can be further improved using feature-based approaches such as ELMo (Peters et al. 2018) or fine-tuning approaches such as BERT (Devlin et al. 2019) along with pre-trained language representations. These models can construct contextual representations, in which word representations are influenced by the context in which the words appear. Such models currently provide state-of-the-art performance at representation learning for many (perhaps even most) natural language processing tasks, particularly when using deep architectures based on the transformer (Vaswani et al. 2017), pre-trained on big data and fine-tuned on task-specific data. We view our work as complementary to that line of research, in that we focus on improving the coherence of topic models, which create interpretable representations designed for human consumption, rather than focusing on uninterpretable big data models designed for accurate prediction.

Work has recently begun on combining topic models with contextual embeddings and transformer-based language models, although much remains to be done. One method, called tBERT (Peinelt, Nguyen, and Liakata 2020), feeds both BERT embeddings and topics as features into a neural network which performs semantic similarity detection. This approach fuses information from both topics and BERT to solve a downstream task, but it does not aim to fundamentally unify the models in order to improve their representational abilities.

Another approach is to identify topics by performing clustering on the embeddings produced by models such as BERT, which can be done on vocabularly-level word embeddings (Sia, Dalmia, and Mielke 2020), document embeddings (Grootendorst 2022), or contextual word embeddings (Thompson and Mimno 2020). This strategy aims to learn a topic model based on a given (BERT-style) neural probabilistic language model, while NEA aims to learn a (word embedding-style) neural probabilistic language model based on a given topic model. An extension of NEA which uses contextual word embeddings is an exciting potential avenue for future research.

7. Conclusion and Future Work

We have proposed neural embedding allocation (NEA), a method for improving general LDA-style topic models by deconstructing them to reveal underlying semantic vector representations. Our experimental results show that NEA improves several diverse topic models' coherence and performs better than them at many tasks. We demonstrated the practical utility of the NEA algorithm by using it to address gender bias in NLP.

In future work, we plan to extend NEA to leverage transformer models such as BERT. For example, instead of learning fixed word embeddings based on a topic model, we could adapt NEA to learn BERT-style contextual word embeddings for each token while jointly learning topic and document embeddings. These embeddings could be seeded based on a pre-trained BERT model in order to capture knowledge from a big

data corpus. In this NEA extension, when simulating a word w from an LDA topic model in order to train the embedding model to mimic it, we would retrieve a context sentence from the corpus in which w occurs and use the BERT model to convert the fixed "input" word embedding to a contextual embedding. Thus, the topic embeddings are trained based on contextual rather than fixed word representations. The contextual word embeddings are further fine-tuned within the NEA process. A further possible extension is to make the topic embeddings be contextual as well.

Alternatively, instead of extending NEA to mimic a traditional topic model while leveraging BERT's contextual embeddings, we could adapt NEA to do the reverse task: learn a topic model based on a given BERT-style model. Just as NEA deconstructs a topic model to learn hidden vector representations, the NEA methodology could be leveraged to deconstruct a BERT-style model to recover hidden topic vectors. To accomplish this, consider a variation of BERT in which the contextual embeddings T_i for each token i are each mapped to one of a set of K topic embeddings v_{z_i}, corresponding to the token's topic assignment z_i, before performing BERT's training tasks such as the masked language model. We would then use a version of the NEA algorithm to teach this "BERT topic model" to mimic the original target BERT model's behavior at its pre-training and/or fine-tuning tasks. This approach would "compress" the BERT model into a smaller topic model which aims to encode its latent semantic knowledge.

Acknowledgments

This work was performed under the following financial assistance award: 60NANB18D227 from U.S. Department of Commerce, National Institute of Standards and Technology. This material is based upon work supported by the National Science Foundation under Grant No.'s IIS2046381; IIS1850023. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

Barocas, Solon and Andrew D Selbst. 2016. Big data's disparate impact. *Calif. L. Rev.*, 104:671.

Bender, Emily M, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 610–623.

Bengio, Yoshua, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of Machine Learning Research*, 3(Feb):1137–1155.

Biddle, Dan. 2006. *Adverse impact and test validation: A practitioner's guide to valid and defensible employment testing*. Gower Publishing, Ltd.

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

Bolukbasi, Tolga, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. 2016. Man is to computer programmer as woman is to homemaker? Debiasing word embeddings. *Advances in Neural Information Processing Systems*, 29:4349–4357.

Brown, Tom, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901.

Buciluǎ, Cristian, Rich Caruana, and Alexandru Niculescu-Mizil. 2006. Model compression. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 535–541, ACM.

Caliskan, Aylin, Joanna J Bryson, and Arvind Narayanan. 2017. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186.

Collobert, Ronan, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost)

from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537.

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

Deshpande, Ketki, Shimei Pan, and James Foulds. 2020. Mitigating demographic bias in AI-based resume filtering. *Fairness in User Modeling, Adaptation and Personalization Workshop*, pages 268–275.

Dev, Sunipa and Jeff Phillips. 2019. Attenuating bias in word vectors. In *International Conference on Artificial Intelligence and Statistics*, pages 879–887, PMLR.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

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

Dwork, Cynthia, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. 2012. Fairness through awareness. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, pages 214–226, ACM.

Dyer, Chris. 2014. Notes on noise contrastive estimation and negative sampling. *arXiv preprint arXiv:1410.8251*.

Foulds, James. 2018. Mixed membership word embeddings for computational social science. In *International Conference on Artificial Intelligence and Statistics*, pages 86–95, PMLR.

Foulds, James R, Rashidul Islam, Kamrun Naher Keya, and Shimei Pan. 2020. An intersectional definition of fairness. In *2020 IEEE 36th International Conference on Data Engineering*, pages 1918–1921, IEEE.

Gonen, Hila and Yoav Goldberg. 2019. Lipstick on a pig: Debiasing methods cover up systematic gender biases in word embeddings but do not remove them. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 609–614.

Griffiths, Thomas L, Mark Steyvers, and Joshua B Tenenbaum. 2007. Topics in semantic representation. *Psychological Review*, 114(2):211–244.

Grootendorst, Maarten. 2022. Bertopic: Neural topic modeling with a class-based tf-idf procedure. *arXiv preprint arXiv:2203.05794*.

Gutmann, Michael and Aapo Hyvärinen. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 297–304.

Gutmann, Michael U and Aapo Hyvärinen. 2012. Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *Journal of Machine Learning Research*, 13(Feb):307–361.

Hinton, Geoffrey, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *ArXiv preprint arXiv:1503.02531*.

Hinton, Geoffrey E. 2002. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800.

Hinton, Geoffrey E et al. 1986. Learning distributed representations of concepts. In *Proceedings of the Eighth Annual Conference of the Cognitive Science Society*, volume 1, pages 1–12, Amherst, MA.

Islam, Rashidul and James Foulds. 2019. Scalable collapsed inference for high-dimensional topic models. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2836–2845.

Islam, Rashidul, Kamrun Naher Keya, Shimei Pan, and James Foulds. 2019. Mitigating demographic biases in social media-based recommender systems. *The 25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (Social Impact Track)*.

Islam, Rashidul, Kamrun Naher Keya, Ziqian Zeng, Shimei Pan, and James Foulds. 2021. Debiasing career recommendations with neural fair collaborative filtering. In *Proceedings of the Web Conference 2021*, pages 3779–3790.

Islam, Rashidul, Shimei Pan, and James R Foulds. 2021. Can we obtain fairness for free? In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, pages 586–596.

Keya, Kamrun Naher and James Foulds. 2018. Neural embedding allocation: Distributed representations of words, topics, and documents. In *Mid-Atlantic Student Colloquium on Speech, Language and Learning*.

Keya, Kamrun Naher, Rashidul Islam, Shimei Pan, Ian Stockwell, and James Foulds. 2021. Equitable allocation of healthcare resources with fair survival models. In *Proceedings of the 2021 SIAM International Conference on Data Mining*, pages 190–198, SIAM.

Kim, Yoon. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1746–1751.

Lau, Jey Han, 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.

Le, Quoc and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International Conference on Machine Learning*, pages 1188–1196.

Levy, Omer and Yoav Goldberg. 2014. Neural word embedding as implicit matrix factorization. *Advances in Neural Information Processing Systems*, 27:2177–2185.

Li, Aaron Q, Amr Ahmed, Sujith Ravi, and Alexander J Smola. 2014. Reducing the sampling complexity of topic models. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 891–900, ACM.

Liu, Yang, Zhiyuan Liu, Tat-Seng Chua, and Maosong Sun. 2015. Topical word embeddings. In *Twenty-ninth AAAI Conference on Artificial Intelligence*, pages 2418–2424.

Liu, Yinhan, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Meng, Yu, Yunyi Zhang, Jiaxin Huang, Yu Zhang, Chao Zhang, and Jiawei Han. 2020. Hierarchical topic mining via joint spherical tree and text embedding. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1908–1917.

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

Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *ArXiv preprint arXiv:1301.3781*.

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*, 26:3111–3119.

Mimno, David, Hanna M Wallach, Edmund Talley, Miriam Leenders, and Andrew McCallum. 2011. Optimizing semantic coherence in topic models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 262–272, Association for Computational Linguistics.

Mnih, Andriy and Koray Kavukcuoglu. 2013. Learning word embeddings efficiently with noise-contrastive estimation. *Advances in Neural Information Processing Systems*, 26:2265–2273.

Moody, Christopher E. 2016. Mixing Dirichlet topic models and word embeddings to make LDA2vec. *arXiv preprint arXiv:1605.02019*.

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

Peinelt, Nicole, Dong Nguyen, and Maria Liakata. 2020. tBERT: Topic models and BERT joining forces for semantic similarity detection. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7047–7055, Association for Computational Linguistics, Online.

Pennington, Jeffrey, 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.

Peters, Matthew E., Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018.

Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, Association for Computational Linguistics, New Orleans, Louisiana.

Radford, Alec, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. http://openai-assets.s3.amazonaws.com/research-covers/language-unsupervised/language_understanding_paper.pdf.

Radford, Alec, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Rosen-Zvi, Michal, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth. 2004. The author-topic model for authors and documents. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, pages 487–494, AUAI Press.

Sahlgren, Magnus. 2008. The distributional hypothesis. *Italian Journal of Disability Studies*, 20:33–53.

Schler, Jonathan, Moshe Koppel, Shlomo Argamon, and James W Pennebaker. 2006. Effects of age and gender on blogging. In *AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs*, volume 6, pages 199–205.

Shi, Bei, Wai Lam, Shoaib Jameel, Steven Schockaert, and Kwun Ping Lai. 2017. Jointly learning word embeddings and latent topics. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 375–384.

Sia, Suzanna, Ayush Dalmia, and Sabrina J Mielke. 2020. Tired of topic models? Clusters of pretrained word embeddings make for fast and good topics too! In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 1728–1736.

Thompson, Laure and David Mimno. 2020. Topic modeling with contextualized word representation clusters. *arXiv preprint arXiv:2010.12626*.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*, 30:5998–6008.

Vaswani, Ashish, Yinggong Zhao, Victoria Fossum, and David Chiang. 2013. Decoding with large-scale neural language models improves translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1387–1392.

Zafar, Muhammad Bilal, Isabel Valera, Manuel Gomez Rogriguez, and Krishna P Gummadi. 2017. Fairness constraints: Mechanisms for fair classification. In *International Conference on Artificial Intelligence and Statistics*, pages 962–970, PMLR.

Zemel, Rich, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. 2013. Learning fair representations. In *International Conference on Machine Learning*, pages 325–333.

Zhu, Lixing, Yulan He, and Deyu Zhou. 2020. A neural generative model for joint learning topics and topic-specific word embeddings. *Transactions of the Association for Computational Linguistics*, 8:471–485.