arXiv:1705.07368v2 [cs.CL] 25 May 2017

Mixed Membership Word Embeddings for Computational Social Science

James Foulds
California Institute for Telecommunications and Information Technology
and Department of Computer Science & Engineering
University of California, San Diego
jfoulds@ucsd.edu

Abstract

Word embeddings improve the performance of NLP systems by revealing the hidden structural relationships between words. These models have recently risen in popularity due to the performance of scalable algorithms trained in the big data setting. Despite their success, word embeddings have seen very little use in computational social science NLP tasks, presumably due to their reliance on big data, and to a lack of interpretability. I propose a probabilistic model-based word embedding method which can recover *interpretable embeddings, without big data*. The key insight is to leverage the notion of *mixed membership* modeling, in which global representations are shared, but individual entities (i.e. dictionary words) are free to use these representations to uniquely differing degrees. Leveraging connections to topic models, I show how to train these models in high dimensions using a combination of state-of-the-art techniques for word embeddings and topic modeling. Experimental results show an improvement in predictive performance of up to 63% in MRR over the skip-gram on small datasets. The models are interpretable, as embeddings of topics are used to encode embeddings for words (and hence, documents) in a model-based way. I illustrate this with two computational social science case studies, on NIPS articles and State of the Union addresses.

1 Introduction

Word embedding models, which learn to encode dictionary words with vector space representations, have been shown to be valuable for a variety of NLP tasks such as statistical machine translation (Vaswani et al., 2013), part-of-speech tagging, chunking, and named entity recogition (Collobert et al., 2011), as they provide a more nuanced representation of words than a simple indicator vector into a dictionary. These models have risen in popularity for NLP applications due to the success of models designed specifically for the big data setting. In particular, Mikolov et al. (2013a,b) showed that very simple word embedding models with high-dimensional representations can scale up to massive datasets, allowing them to outperform more sophisticated neural network language models which can process fewer documents given a fixed computational budget.

In this work, I offer a somewhat contrarian perspective to the currently prevailing trend of big data optimism, as exemplified by the work of Collobert et al. (2011), Mikolov et al. (2013a,b), and others, who argue that massive datasets are sufficient to allow language models to automatically resolve many challenging NLP tasks. It should be noted that “big” datasets are not always available, particularly in computational social science NLP applications, where the data of interest are often not obtained from large scale sources such as the internet and social media, but from sources such as press releases (Grimmer, 2010), academic journals (Mimno, 2012), books (Zhu et al., 2015), and transcripts of recorded speech (Brent, 1999; Nguyen et al., 2014; Guo et al., 2015). A very standard practice in the literature is to train word embedding models on a generic large corpus such as Wikipedia, and use the

NIPS	reinforcement belief learning policy algorithms Singh robot machine MDP planning
Google News	teaching learn learning reteaching learner_centered emergent_literacy kinesthetic_learning

Table 1: Most similar words to "*learning*," based on word embeddings trained on NIPS articles, and on the large generic Google News corpus (Mikolov et al., 2013a,b).

embeddings for NLP tasks on the target dataset, cf. (Collobert et al., 2011; Mikolov et al., 2013a; Pennington et al., 2014; Kiros et al., 2015). However, as we shall see here, this standard practice might not always be effective, as the size of a dataset does not necessarily correspond to its degree of relevance for a particular analysis. Even very large corpora have idiosyncrasies that can make their embeddings invalid for other domains. For instance, suppose we would like to use word embeddings to analyze scientific articles on machine learning. In Table 1, I report the most similar words to the word "*learning*" based on word embedding models trained on two corpora. For embeddings trained on articles from the NIPS conference, the most similar words are related to *machine learning*, as desired, while for Tomas Mikolov's embeddings trained on the massive, generic Google News corpus, the most similar words relate to *learning and teaching in the classroom*.

Even more concerningly, Bolukbasi et al. (2016) have recently shown that word embeddings from standard corpora can encode sexist assumptions implicit in a dataset, such as the analogy "man is to computer programmer as woman is to homemaker." Based on these results, it is reasonable to expect that they will also typically encode the worldview of the dominant culture, which in the U.S.A. is white, male, and Eurocentric, and is therefore inappropriate for studying, e.g., black female hip-hop artists' lyrics, or poetry by Syrian refugees. This emphasis on the language norms of the dominant culture could potentially lead to systematic bias against minorities, women, and people of color in NLP applications with real-world consequences, such as automatic essay grading and college admissions. In order to proactively combat these kinds of biases in large generic datasets, there is a need for effective word embedding models for small datasets, so that the most relevant datasets can be used for training, even when they are small.

In this paper, I propose a word embedding model that can be trained directly on a small to medium-sized corpus of interest, without the need for a separate big data training set. The primary insight is to use a data-efficient parameter sharing scheme via *mixed membership* modeling, with inspiration from topic models. Mixed membership models provide a flexible yet efficient latent representation, in which entities are associated with shared, global representations, but to uniquely varying degrees. We can identify the skip-gram word2vec model of Mikolov et al. (2013a,b) as corresponding to a certain naive Bayes topic model, which leads to mixed membership extensions, allowing the use of *fewer vectors than words*. I show that this leads to better modeling performance without big data, as measured by predictive performance (when the context is leveraged for prediction), as well as to interpretable latent representations that are valuable for computational social science applications.

2 Background

In this section, I provide the necessary background on word embeddings, as well as on topic models and mixed membership models. A more detailed discussion of related work is given in the Appendix. Traditional language models aim to predict words given the contexts that they are found in, thereby forming a joint probabilistic model for sequences of words in a language. Bengio et al. (2003) developed improved language models by using *distributed representations* (Hinton et al., 1986), in which words are represented by neural network synapse weights, or equivalently, vector space embeddings.

Later authors have noted that these *word embeddings* are useful for semantic representations of words, independently of whether a full joint probabilistic language model is learned, and that alternative training schemes can be beneficial for learning the embeddings. In particular, Mikolov et al. (2013a,b) proposed the *skip-gram* model, which inverts the language model prediction task and aims to *predict the context* given an input word. The skip-gram model is a log-bilinear discriminative probabilistic classifier parameterized by "input" word embedding vectors v_{w_i} for the input words w_i, and "output" word embedding vectors v'_{w_c} for context words $w_c \in \text{context}(i)$, as shown in Table 2, top-left.

Topic models such as *latent Dirichlet allocation* (LDA) (Blei et al., 2003) are another class of probabilistic language models that have been used for semantic representation (Griffiths et al., 2007).

	Skip-gram	Skip-gram topic model
Naive Bayes	For each word in the corpus w_i For each word $w_c \in context(i)$ Draw $w_c \vert w_i$ via $p(w_c \vert w_i) \propto exp({v'_{w_c}}^\intercal v_{w_i} + b_{w_c})$	For each word in the corpus w_i For each word $w_c \in context(i)$ Draw $w_c \vert w_i \sim$ Discrete($\phi^{(w_i)}$)
Mixed membership	For each word in the corpus w_i Draw a topic $z_i \sim$ Discrete($\theta^{(w_i)}$) For each word $w_c \in context(i)$ Draw $w_c \vert w_i$ via $p(w_c \vert w_i) \propto exp({v'_{w_c}}^\intercal v_{z_i} + b_{w_c})$	For each word in the corpus w_i Draw a topic $z_i \sim$ Discrete($\theta^{(w_i)}$) For each word $w_c \in context(i)$ Draw $w_c \vert w_i \sim$ Discrete($\phi^{(z_i)}$)

Table 2: "Generative" models. Identifying the skip-gram (top-left)'s word distributions with topics yields analogous topic models (right), and mixed membership modeling extensions (bottom).

A straightforward way to model text corpora is via unsupervised multinomial naive Bayes, in which a latent cluster assignment for each document selects a multinomial distribution over words, referred to as a *topic*, with which the documents' words are assumed to be generated. LDA topic models improve over naive Bayes by using a *mixed membership* model, in which the assumption that all words in a document d belong to the same topic is relaxed, and replaced with a *distribution* over topics $\theta^{(d)}$. In the model's assumed generative process, for each word i in document d, a topic assignment z_i is drawn via $\theta^{(d)}$, then the word is drawn from the chosen topic $\phi^{(z_i)}$. The mixed membership formalism provides a useful compromise between model flexibility and statistical efficiency: the K topics $\phi^{(k)}$ are shared across all documents, thereby sharing statistical strength, but each document is free to use the topics to its own unique degree. Bayesian inference further aids data efficiency, as uncertainty over $\theta^{(d)}$ can be managed for shorter documents. Some recent papers have aimed to combine topic models and word embeddings (Das et al., 2015; Liu et al., 2015), but they do not aim to address the small data problem for computational social science, which I focus on here.

3 The Mixed Membership Skip-Gram

To design our word embedding model for small corpora, we identify connections between word embeddings and topic models, and adapt advances from the topic modeling literature. Although the skip-gram is discriminative, in the sense that it does not jointly model the input words w_i, we can equivalently interpret it as encoding a "conditionally generative" process for the context given the words, in order to develop probabilistic models that extend the skip-gram. Following the *distributional hypothesis* (Harris, 1954), the skip-gram's word embeddings parameterize discrete probability distributions over words $p(w_c|w_i)$ which tend to co-occur, and tend to be semantically coherent – a property leveraged by the Gaussian LDA model of Das et al. (2015). By identifying these discrete distributions with *topics* $\phi^{(w_i)}$, we see that the skip-gram can be reinterpreted as a parameterization of a certain supervised naive Bayes topic model (Table 2, top-right). In this topic model, input words w_i are fully observed "cluster assignments," and the words in w_i's contexts are a "document." The skip-gram differs from this supervised topic model only in the parameterization of the "topics" via word vectors which encode the distribution over words via a log-bilinear model.

As in LDA, we can improve on this model by replacing the naive Bayes assumption with a mixed membership assumption. By applying the mixed membership representation to this topic model version of the skip-gram, we obtain the model in the bottom-right of Table 2.[1] After once again parameterizing this model with word embeddings, we obtain our final model, the *mixed membership skip-gram (MMSG)* (Table 2, bottom-left). In the model, each input word has a distribution over topics $\theta^{(w)}$. Each topic has a vector-space embedding v_k and each output word has an embedding v'_w. A topic $z_i \in \{1, \ldots, K\}$ is drawn for each context, and the words in the context are drawn from the log-bilinear model, using the vector for this topic v_{z_i}. We expect that the resulting *mixed membership word embeddings* are beneficial in the small-to-medium data regime for the following reasons:

1. By using **fewer input vectors than words**, we can reduce the size of the semantic representation to be learned (output vectors are generally discarded for the purposes of embedding).
2. The topic vectors are shared across all words, allowing more data to be used per vector.

[1]The model retains a naive Bayes assumption at the context level, for latent variable count parsimony.

Algorithm 1 Training the mixed membership skip-gram via annealed MHW and NCE

for $j = 1$: maxAnnealingIter **do**
  $T_j := T_0 + \lambda\kappa^j$
  for $i = 1$: N **do**
    $c \sim$ Uniform($|$context$(w_i)|$); $z_i^{(new)} \sim q_{w_c}$ //using cached samples from alias tables
    accept or reject $z_i^{(new)}$ via Equation 4; If accept, $z_i := z_i^{(new)}$
  end for
end for
$\hat{\theta}_k^{(w_i)} :\propto n_k^{(w_i)\neg i} + \alpha_k$
$[\mathbf{V}, \mathbf{V}', b] :=$ NCE$(inputWords = \mathbf{z}, contextWords = \mathbf{w})$

3. Polysemy is automatically addressed by clustering the words into topics, allowing for more topically focused and semantically coherent vector representations.

Of course, the model also requires more parameters to be learned, namely the mixed membership proportions $\theta^{(w)}$. Based on topic modeling, I hypothesized that with care, these added parameters need not adversely affect performance in the small-medium data regime, for two reasons: 1) we can use a Bayesian approach to effectively manage uncertainty in them, and to marginalize them out, which prevents them being a bottleneck during training; and 2) at test time, using the posterior for z_i given the context, instead of the "prior" $Pr(z_i|w_i, \theta)$, mitigates the impact of uncertainty in $\theta^{(w_i)}$:

$$Pr(z_i = k|w_i, \text{context}(i), \mathbf{V}, \mathbf{V}', \theta) \propto \theta_k^{(w_i)} \prod_{c \in \text{context}(i)} \frac{exp(v'^{\intercal}_{w_c} v_k + b_{w_c})}{\sum_{j'=1}^{V} exp(v'^{\intercal}_{j'} v_k + b_{j'})} . \quad (1)$$

If a single vector $\hat{v}_{w_i}$ for a word token is desired, we can obtain one from the above, leveraging the context, via the posterior mean, $\hat{v}_{w_i} \triangleq \sum_k v_k Pr(z_i = k|w_i, \text{context}(i), \mathbf{V}, \mathbf{V}', \theta)$. With fewer vectors than words, some model capacity is lost, but the flexibility of the mixed membership representation allows the model to compensate for this. When the number of shared vectors equals the number of words, the mixed membership skip-gram is strictly more representationally powerful than the skip-gram. With more vectors than words, we expect that the increased representational power would be beneficial in the big data regime. As this is not my goal, I leave this for future work.

4 Learning Mixed Membership Word Embeddings

To train the mixed membership skip-gram, an online EM algorithm with stochastic gradient M-step updates can readily be derived, similar to that of Tian et al. (2014)'s multi-prototype embedding model, which has multiple vectors per word. However, this is impractical due to a $O(KD)$ complexity for the E-step, and a $O(D)$ complexity for the M-step, where K and D are the number of topics/dictionary words, respectively. Instead, I propose a principled approximate algorithm that is sublinear time in both K and D. The overall strategy is to first impute the $\mathbf{z}$ assignments by simulated annealing, using an efficient implementation of a collapsed Gibbs sampler, thereby reducing the learning problem to standard word embedding. Given the imputed $\mathbf{z}$'s, we then learn the topic and word embeddings $\mathbf{V}$, $\mathbf{V}'$ via noise-contrastive estimation, which approximately performs maximum likelihood estimation (or equivalently, MAP estimation with an improper uniform prior). The overall algorithm can be understood as approximately finding a posterior mode, $\arg\max Pr(\mathbf{V}, \mathbf{V}', \theta, \mathbf{z}|\mathbf{w}, \mathbf{w}_c)$, in the spirit of iterated conditional models (Besag, 1986), but with approximate updates and with iteration avoided, since $\mathbf{z}$ can be estimated without the vectors. The overall algorithm is summarized in Algorithm 1.

4.1 Imputing the z's: Topic Model Pre-Clustering via Annealed Metropolis-Hastings-Walker

To engineer such an algorithm, the key insight is that our mixed membership skip-gram model (Table 2, bottom left) is equivalent to the topic model version (Table 2, bottom right), up to the parameterization, and the prior (if any). With sufficiently high dimensional embeddings, the log-bilinear model can capture any distribution $p(w_c|z_i)$, and so the maximum likelihood embeddings would encode the exact same word distributions as the MLE topics for the topic model, $\phi^{(z_i)}$. However, the topic model admits a collapsed Gibbs sampler (CGS) that efficiently resolves the cluster

	Input word = "Bayesian"
Model	Top words in topic for input word. Top 3 topics for word shown for mixed membership models.
SGTM	model networks learning neural bayesian data models approach network framework
SG	belief learning framework models methods markov function bayesian based inference
MMSGTM	neural bayesian networks mackay computation framework practical learning weigend backpropagation
	model models bayesian prior data parameters likelihood priors structure graphical
	monte carlo chain markov sampling mcmc method methods model bayesian
MMSG	neural networks weigend bayesian data mackay learning computation practical vol
	probability model data models priors algorithm bayesian likelihood set parameters
	carlo monte mcmc chain reversible sampling model posterior neural data

Table 3: SG = skip-gram, TM = topic model, MM = mixed membership.

assignments, which cause the bottleneck during the E-step. I therefore propose to reparameterize the mixed membership skipgram as its corresponding topic model for the purposes of imputing the $\mathbf{z}$'s, and run simulated annealing based on the collapsed Gibbs sampler for the topic model to impute the topic assignments. Then, with the $\mathbf{z}$'s fixed, learning the word and topic vectors corresponds to finding the optimal vectors for encoding these fixed ϕ distributions via the log-bilinear model.

This topic model pre-clustering step is reminiscent of Reisinger and Mooney (2010); Huang et al. (2012); Liu et al. (2015), who apply an off-the-shelf clustering algorithm (or LDA) to initially identify different clusters of contexts, and then apply word embedding algorithms on the cluster assignments. However, our clustering is learned based on the word embedding model itself, and clustering at test time is performed via Bayesian reasoning, in Equation 1, rather than via an ad-hoc method. With Dirichlet priors on the parameters, the collapsed Gibbs update is (derivation in the Appendix):

$$p(z_i = k|\cdot) \propto \left(n_k^{(w_i)\neg i} + \alpha_k\right) \prod_{c=1}^{|\text{context}(i)|} \frac{n_{w_c}^{(k)\neg i} + \beta_{w_c} + n_{w_c}^{(i,c)}}{n^{(k)\neg i} + \sum_{w'} \beta_{w'} + c - 1} , \quad (2)$$

where α and β are parameter vectors for Dirichlet priors over the topic and word distributions, $n_k^{(w_i)}$ and $n_{w_c}^{(k)\neg i}$ are input and output word/topic counts (excluding the current word), and $n_{w_c}^{(i,c)}$ is the number of occurrences of word w_c before the cth word in the ith context. We scale this algorithm up to thousands of topics using an adapted version of the recently proposed Metropolis-Hastings-Walker algorithm for high-dimensional topic models, which scales sublinearly in K (Li et al., 2014). The method uses a data structure called an *alias table*, which allows for amortized O(1) time sampling from discrete distributions. A Metropolis-Hastings update is used to correct for approximating the CGS update with a proposal distribution based on these samples. We can interpret the product over the context, which dominates the collapsed Gibbs update, as a *product of experts* (Hinton, 2002), where each word in the context is an "expert" which weighs in multiplicatively on the update. In order to approximate this via alias tables, we use proposals which approximate the product of experts with a "mixture of experts." We select a word w_c uniformly from the context, and the proposal q_{w_c} draws a candidate topic proportionally to the chosen context word's contribution to the update:

$$c \sim \text{Uniform}(|\text{context}(w_i)|) , \qquad q_{w_c}(k) \propto \frac{n_{w_c}^{(k)} + \beta_{w_c}}{n^{(k)} + \sum_{w'} \beta_{w'}} . \quad (3)$$

We expect these proposals to have some resemblance to the target distribution, but to be flatter, which is a property we'd generally like in a proposal distribution. The proposal is implemented efficiently by sampling from the experts via the alias table data structure, in amortized O(1) time, rather than in time linear in the sparsity pattern, as in (Li et al., 2014), since the proposal does not involve the sparse term (which is less important in our case). We perform simulated annealing to optimize over the posterior, which is very natural in a Metropolis-Hastings setting. Interpreting the negative log posterior as the energy function for a Boltzmann distribution at temperature T_j for iteration j, this is achieved by raising the model part of the Metropolis-Hastings acceptance ratio to the power of $\frac{1}{T_j}$:

$$z_i^{(new)} \sim q_{w_c} , \quad Pr(\text{accept } z_i^{(new)}|\cdot) = \min\left(1, \left(\frac{p(z_i = z_i^{(new)}|\cdot)}{p(z_i = z_i^{(old)}|\cdot)}\right)^{\frac{1}{T_j}} \frac{q_{w_c}(z_i^{(old)})}{q_{w_c}(z_i^{(new)})}\right) . \quad (4)$$

Annealing also helps with mixing, as the standard Gibbs updates can become nearly deterministic. We use a temperature schedule $T_j = T_0 + \lambda\kappa^j$, where T_0 is the target final temperature, $\kappa < 1$, and

Dataset	Frequency baseline	SG	SG +context	Google +context	MMSG prior	MMSG posterior	SGTM	SGTM +context	MMSGTM prior	MMSGTM posterior
NIPS	0.029	0.038	0.031	0.027	0.037	**0.062**	**0.055**	**0.064**	**0.046**	**0.074**
S.o.t. Union	0.021	0.025	0.023	0.021	0.022	**0.034**	**0.036**	**0.046**	**0.032**	**0.045**
Shakespeare	0.015	0.020	0.010	**0.032**	0.015	0.019	**0.025**	**0.043**	0.020	**0.025**
Du Bois	0.028	0.045	0.037	0.033	0.041	**0.053**	**0.052**	**0.081**	**0.050**	**0.066**

Table 4: Mean reciprocal rank of held-out context words. SG = skip-gram, TM = topic model, MM = mixed membership. Bold indicates statistically significant improvement versus SG.

λ controls the initial temperature, and therefore mixing in the early iterations. In my experiments, I use $T_0 = 0.0001$, $\kappa = 0.99$, and $\lambda = |\text{context}|$. The acceptance probability can be computed in time constant in K, and sampling is amortized constant time in K, so each iteration is in amortized constant time in K. Rao-Blackwellized estimates of the mixed membership proportions are obtained from the final sample as $\hat{\theta}_k^{(w_i)} \propto n_k^{(w_i)\neg i} + \alpha_k$.

4.2 Learning the Embeddings: Noise-Contrastive Estimation

Finally, with the topic assignments imputed and θ estimated via the topic model, we must learn the embeddings, which is still an expensive $O(D)$ per context via SGD for maximum likelihood estimation. This same complexity is also an issue for the standard skip-gram, which Mnih and Teh (2012); Mnih and Kavukcuoglu (2013) have addressed using the noise-contrastive estimation (NCE) algorithm of Gutmann and Hyvärinen (2010, 2012). NCE avoids the expensive normalization step, making the algorithm scale sublinearly in the vocabulary size D. The algorithm solves unsupervised learning tasks by transforming them into the supervised learning task of distinguishing the data from randomly sampled noise. As the number of noise samples tends to infinity, the method increasingly well approximates maximum likelihood estimation, while avoiding explicitly computing the normalization constant or its derivative, as required by a direct optimization of the log-likelihood. We use NCE as a principled approximation to MLE, and hence MAP estimation with a uniform prior, with the overall Algorithm 1 approximately finding a posterior mode. NCE takes as input the log-likelihood of a data point, i.e. a context word given an input word and its topic assignment

$$\log Pr(w_c|\vec{\mathbf{V}}, w_i, z_i, \mathbf{b}) \;=\; \log Pr^0(w_c|\vec{\mathbf{V}}, w_i, z_i, \mathbf{b}) + a \;=\; v'^{\intercal}_{w_c} v_{z_i} + b_{w_c} + a \, , \quad (5)$$

where Pr^0 refers to an unnormalized distribution, $\vec{\mathbf{V}}$ is the vector of all word and topic embeddings, and a is a parameter encoding the corresponding log normalization constant in the current context i. Following Mnih and Teh (2012), we fix $a = 0$, under the supposition that the NCE procedure will compensate for this by encouraging the distributions to "self-normalize" in order to optimize the NCE objective. NCE performs logistic regression to distinguish between the data samples and the noise samples. Supposing that there are k samples from the noise distribution per word-pair example, the NCE objective function for context i is

$$J^{(i)}(\vec{\mathbf{V}}, \mathbf{b}) \triangleq E_{P_d^{(i)}}[\log \sigma(G(w_c; \vec{\mathbf{V}}, w_i, z_i, \mathbf{b}))] - kE_{P_n}[\log(1 - \sigma(G(w_c; \vec{\mathbf{V}}, w_i, z_i, \mathbf{b})))] \quad (6)$$

where $P_d^{(i)}$ is the data distribution for words w_c context i, and $G(w_c; \vec{\mathbf{V}}, w_i, z_i, \mathbf{b}) \triangleq \log Pr(w_c|\vec{\mathbf{V}}, w_i, z_i, \mathbf{b}) - \log P_n(w_c)$ is the difference in log-likelihood between the model and the noise distributions. We learn the embeddings by stochastic gradient ascent on the NCE objective.

5 Experiments

The goals of my experiments were to validate the proposed methods, to study their applicability for computational social science research, and to substantiate my claims vis-à-vis small versus big data.

I measured the intrinsic quality of the learned embeddings via their predictive performance, at the *skip-gram's training task*, predicting context words w_c given input words w_i. For each dataset, I held out 10,000 (w_c, w_i) pairs uniformly at random, where $w_c \in \text{context}(i)$, $|\text{context}(i)| = 10$, and aimed to predict w_c given w_i (and optionally, $\text{context}(i) \setminus w_c$). Since there are a large number of classes, I treat this as a ranking problem, and report the mean reciprocal rank. The experiments were repeated and averaged over 5 train/test splits. I used four datasets of sociopolitical, scientific, and literary

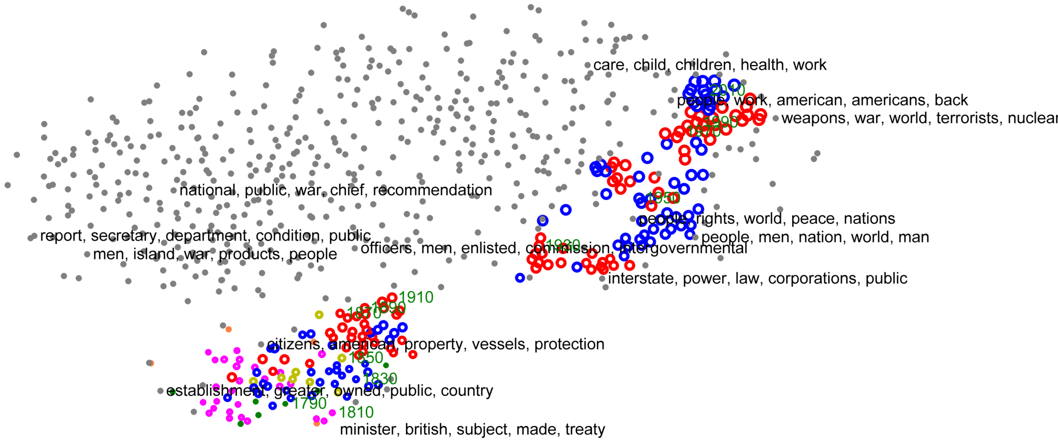

Figure 1: State of the Union, t-SNE. Color = party (red = GOP, blue = Democrats, ...), size = recency (year).

Nearest topic after composition of mean vectors for words	
object + recognition	objects visual object recognition model
character + recognition	recognition segmentation character
speech + recognition	speech recognition hmm system hybrid
computer + vision	computer vision ieee image pattern
computer + science	university science colorado department
bias + variance	error training set data performance
covariance + variance	gaussian distribution model matrix

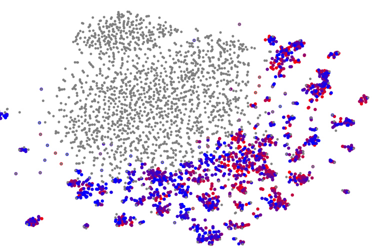

Figure 2: **Left:** Vector compositionality examples, NIPS. **Right:** NIPS documents/ topics, t-SNE.

interest: the corpus of NIPS articles from 1987 – 1999 ($N \approx 2.3$ million), the U.S. presidential state of the Union addresses from 1790 – 2015 ($N \approx 700,000$), the complete works of Shakespeare ($N \approx 240,000$; this version did not contain the Sonnets), and the writings of black scholar and activist W.E.B. Du Bois, as digitized by Project Gutenberg ($N \approx 170,000$).

The results are shown in Table 4. I compared to a word frequency baseline, the skip-gram (SG), and Tomas Mikolov/Google's vectors trained on Google News, $N \approx 100$ billion, via CBOW. Simulated annealing was performed for 1,000 iterations, NCE was performed for 1 million minibatches of size 128, and 128-dimensional embeddings were used (300 for Google). I used $K = 2,000$ for NIPS, $K = 500$ for state of the Union, and $K = 100$ for the two smaller datasets. Methods were able to leverage the remainder of the context, either by adding the context's vectors, or via the posterior (Equation 1), which helped for all methods except the naive skip-gram. We can identify several noteworthy findings. First, the generic big data vectors (Google+context) were outperformed by the skip-gram on 3 out of 4 datasets (and by the skip-gram topic model on the other), by a large margin, indicating that corpus-specific embeddings are often important. Second, the mixed membership models, using posterior inference, beat or matched their naïve Bayes counterparts, for both the word embedding models and the topic models. As hypothesized, posterior inference on z_i at test time was important for good performance. Finally, the topic models beat their corresponding word embedding models at prediction. We therefore consider word embeddings (and topic embeddings) to be primarily valuable for dimensionality reduction, rather than for prediction, at least in the small data regime.

I also performed several case studies. I obtained document embeddings, in the same latent space as the topic embeddings, by summing the posterior mean vectors $\hat{v}_{w_i}$ for each token (and similarly for author embeddings), and visualized them in two dimensions using t-SNE (Maaten and Hinton, 2008) (all vectors were normalized to unit length). The state of the Union addresses (Figure 1) are embedded almost linearly by year, with a major jump around the New Deal (1930s), and are well separated by party at any given time period. The embedded topics (in gray) allow us to interpret the latent space. The George W. Bush addresses are embedded near a "war on terror" topic ("weapons, war..."), and the Barack Obama addresses are embedded near a "stimulus" topic ("people, work...").

On the NIPS corpus, for the input word "Bayesian" (Table 3), the naive Bayes and skip-gram models learned a topic with words that refer to Bayesian networks, probabilistic models, and neural networks. The mixed membership models are able to separate this into more coherent and specific topics

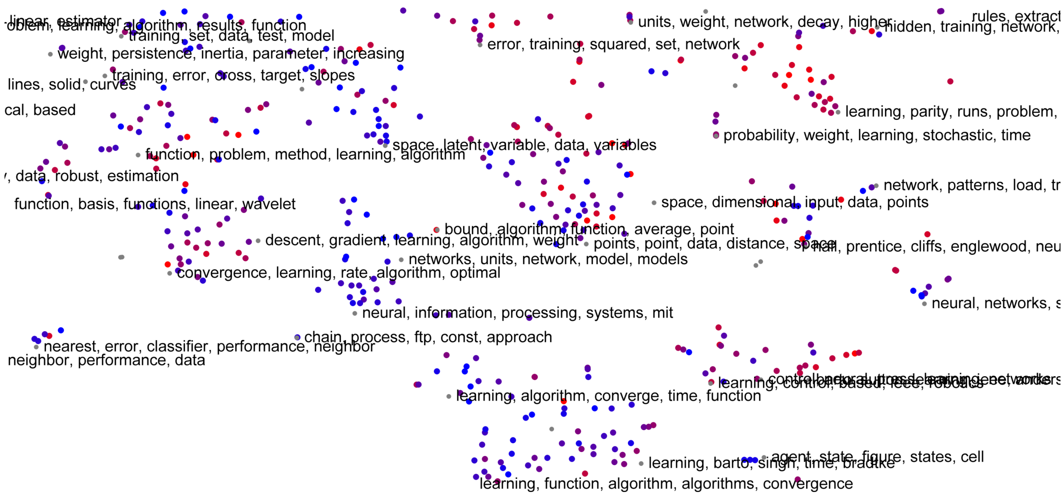

Figure 3: NIPS documents/topics, t-SNE, zoomed in. Blue/red = more recent/older, gray = topics.

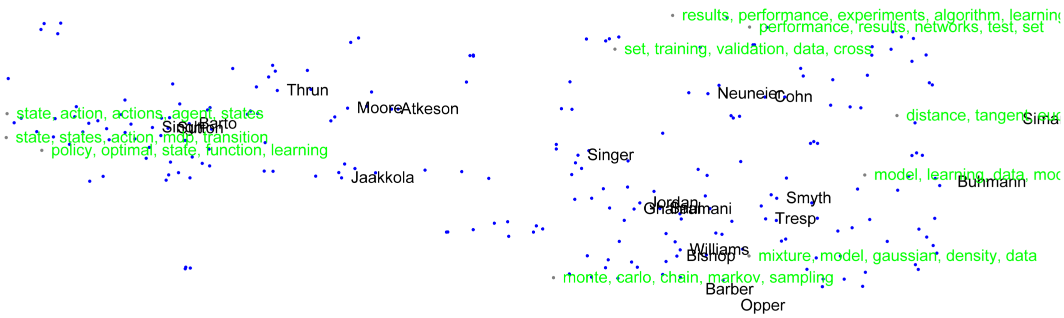

Figure 4: NIPS authors and topics, t-SNE, zoomed in. Blue = authors, gray = topics.

including Bayesian modeling, Bayesian training of neural networks (for which Sir David MacKay was a strong proponent, and Andreas Weigend wrote an influential early paper), and Monte Carlo methods. By performing the additive composition of word vectors, which we obtain by finding the prior mean vector for each word type w, $\bar{v}_w \triangleq \sum_k v_k \theta_k^{(w)}$ (and then normalizing), we obtain relevant topics v_k as nearest neighbors (Figure 2). Similarly, we find that $v_{objectRecognition} - \bar{v}_{object} + \bar{v}_{speech} \approx v_{speechRecognition}$, and $v_{speechRecognition} - \bar{v}_{speech} + \bar{v}_{character} \approx v_{characterRecognition}$.

The t-SNE visualization of NIPS documents (Figures 2, 3) shows some temporal clustering patterns (blue documents are more recent, red documents are older, and gray points are topics). Zooming in, we see regions of the space corresponding to learning algorithms (bottom), data space and latent space (center), training neural networks (top), and nearest neighbors (bottom-left). I also visualized the authors' embeddings via t-SNE (Figure 4). We find regions of latent space for reinforcement learning authors (left: "state, action,...," Singh, Barto, Sutton), probabilistic methods (right: "mixture, model," "monte, carlo," Bishop, Williams, Barber, Opper, Jordan, Ghahramani, Tresp, Smyth), and evaluation (top-right: "results, performance, experiments,...").

6 Conclusion

I have proposed a model-based method for training interpretable corpus-specific word embeddings, without big data, for computational social science. My approach leverages mixed membership representations, the Metropolis-Hastings-Walker algorithm and noise-contrastive estimation. Experimental results for prediction, and real-world case studies on NIPS and state of the Union addresses, indicate that high-quality embeddings and topics can be obtained using the algorithm. I plan to use this approach for substantive social science applications, and to address algorithmic bias/fairness issues. I also plan to extend the methods to leverage big data sets together with a target small data set.

Acknowledgments

I thank Eric Nalisnick and Padhraic Smyth for many helpful discussions.

A Related Work

In this appendix, I discuss related work in the literature and its relation to the proposed methods.

A.1 Topic Modeling and Word Embeddings

The *Gaussian LDA* model of Das et al. (2015) improves the performance of topic modeling by leveraging the semantic information encoded in word embeddings. Gaussian LDA modifies the generative process of LDA such that each topic is assumed to generate the vectors via its own Gaussian distribution. Similarly to our MMSG model, in Gaussian LDA each topic is encoded with a vector, in this case the mean of the Gaussian. It takes pre-trained word embeddings as input, rather than learning the embeddings from data within the same model, and does not aim to perform word embedding.

The topical word embedding (TWE) models of Liu et al. (2015) reverse this, as they take LDA topic assignments of words as input, and aim to use them to improve the resultant word embeddings. The authors propose three variants, each of which modifies the skip-gram training objective to use LDA topic assignments together with words. In the best performing variant, called *TWE-1*, a standard skip-gram word embedding model is trained independently with another skip-gram variant, which tries to predict context words given the input word's topic assignment. The skip-gram embedding and the topic embeddings are concatenated to form the final embedding.

At test time, a distribution over topics for the word given the context, $p(z_i|\text{context}(i))$ is estimated according to the topic counts over the other context words. Using this as a prior, a posterior over topics given both the input word and the context is calculated, and similarities between pairs of words (with their contexts) are averaged over this posterior, in a procedure inspired by those used by Reisinger and Mooney (2010); Huang et al. (2012). The primary similarity to our MMSG approach is the use of a training algorithm involving the prediction of context words, given a topic. Our method does this as part of an overall model-based inference procedure, and we learn mixed membership proportions $\theta^{(w)}$ rather than using empirical counts as the prior over topics for a word token. In accordance with the skip-gram's prediction model, we are thus able to model the context words in the data likelihood term when computing the posterior probability of the topic assignment. TWE-1 requires that topic assignments are available at test time. It provides a mechanism to predict contextual similarity, but not to predict held-out context words, so I was unable to compare to it in my experiments.

A.2 Multi-Prototype Embedding Models

Multi-prototype embeddings models are another relevant line of work. These models address lexical ambiguity by assigning multiple vectors to each word type, each corresponding to a different meaning of that word. Reisinger and Mooney (2010) propose to cluster the occurrences of each word type, based on features extracted from its context. Embeddings are then learned for each cluster. Huang et al. (2012) apply a similar approach, but they use initial single-prototype word embeddings to provide the features used for clustering. These clustering methods have some resemblance to our topic model pre-clustering step, although their clustering is applied within instances of a given word type, rather than globally across all word types, as in our methods. This results in models with more vectors than words, while we aim to find fewer vectors than words, to reduce the model's complexity for small datasets. Rather than employing an off-the-shelf clustering algorithm and then applying an unrelated embedding model to its output, our approach aims to perform model-based clustering within an overall joint model of topic/cluster assignments and word vectors.

Perhaps the most similar model to ours in the literature is the probabilistic multi-prototype embedding model of Tian et al. (2014), who treat the prototype assignment of a word as a latent variable, assumed drawn from a mixture over prototypes for each word. The embeddings are then trained using EM. Our MMSG model can be understood as the mixed membership version of this model, in which the prototypes (vectors) are shared across all word types, and each word type has its own mixed membership proportions across the shared prototypes. While a similar EM algorithm can be applied to the MMSG, the E-step is much more expensive, as we typically desire many more shared vectors (often in the thousands) than we would prototypes per a single word type (Tian et al. use ten in their experiments). We use the Metropolis-Hastings-Walker algorithm with the topic model reparameterization of our model in order to address this by efficiently pre-solving the E-step.

A.3 Mixed Membership Modeling

Mixed membership modeling is a flexible alternative to traditional clustering, in which each data point is assigned to a single cluster. Instead, mixed membership models posit that individual entities are associated with multiple underlying clusters, to differing degrees, as encoded by a mixed membership vector that sums to one across the clusters (Erosheva et al., 2004; Airoldi et al., 2014). These mixed membership proportions are generally used to model lower-level grouped data, such as the words inside a document. Each lower-level data point inside a group is assumed to be assigned to one of the shared, global clusters according to the group-level membership proportions. Thus, a mixed membership model consists of a mixture model for each group, which share common mixture component parameters, but with differing mixture proportions.

This formalism has lead to probabilistic models for a variety of applications, including medical diagnosis (Manton et al., 1994), population genetics (Pritchard et al., 2000), survey analysis (Erosheva, 2003), computer vision (Barnard et al., 2003; Fei-Fei and Perona, 2005), text documents (Hofmann, 1999; Blei et al., 2003), and social network analysis (Airoldi et al., 2008). Nonparametric Bayesian extensions, in which the number of underlying clusters is learned from data via Bayesian inference, have also been proposed (Teh et al., 2006). In this work, dictionary words are assigned a mixed membership distribution over a set of shared latent vector space embeddings. Each instantiation of a dictionary word (an "input" word) is assigned to one of the shared embeddings based on its dictionary word's membership vector. The words in its context ("output" words) are assumed to be drawn based on the chosen embedding.

B Derivation of the Collapsed Gibbs Update

Let $C_i = |\text{context}(i)|$ be the number of output words in the ith context, let $w_1^{(i)}, \dots, w_{C_i}^{(i)}$ be those output words, and let $\mathbf{w}_{\neg i}$ be the input words other that w_i (similarly, topic assignments $\mathbf{z}_{\neg i}$ and output words $\mathbf{w}^{(\neg i)}$). Then the collapsed Gibbs update samples from the conditional distribution

$$
\begin{aligned}
&p(z_i = k | \mathbf{z}_{\neg i}, w_i, w_1^{(i)}, \dots, w_{C_i}^{(i)}, \mathbf{w}_{\neg i}, \mathbf{w}^{(\neg i)}, \alpha, \beta) \\
&\quad \propto p(z_i = k, w_1^{(i)}, \dots, w_{C_i}^{(i)} | \mathbf{z}_{\neg i}, w_i, \mathbf{w}_{\neg i}, \mathbf{w}^{(\neg i)}, \alpha, \beta) \\
&\quad\quad = \int_{\phi^{(k)}} \int_{\theta^{(w_i)}} p(z_i = k, w_1^{(i)}, \dots, w_{C_i}^{(i)}, \phi^{(k)}, \theta^{(w_i)} | \mathbf{z}_{\neg i}, w_i, \mathbf{w}_{\neg i}, \mathbf{w}^{(\neg i)}, \alpha, \beta) \\
&\quad\quad = \int_{\phi^{(k)}} \int_{\theta^{(w_i)}} p(z_i = k, w_1^{(i)}, \dots, w_{C_i}^{(i)} | \phi^{(k)}, \theta^{(w_i)}, w_i) p(\phi^{(k)}, \theta^{(w_i)} | \mathbf{z}_{\neg i}, w_i, \mathbf{w}_{\neg i}, \mathbf{w}^{(\neg i)}, \alpha, \beta) \\
&\quad\quad = \int_{\phi^{(k)}} \int_{\theta^{(w_i)}} \theta_k^{(w_i)} \prod_{c=1}^{C_i} \phi_{w_c^{(i)}}^{(k)} \times p(\theta^{(w_i)} | \mathbf{z}_{\neg i : w_j = w_i}, \alpha) p(\phi^{(k)} | \mathbf{z}_{\neg i}, \mathbf{w}^{(\neg i)}, \beta) \\
&\quad\quad = \int_{\theta^{(w_i)}} \theta_k^{(w_i)} p(\theta^{(w_i)} | \mathbf{z}_{\neg i : w_j = w_i}, \alpha) \times \int_{\phi^{(k)}} \prod_{c=1}^{C_i} \phi_{w_c^{(i)}}^{(k)} p(\phi^{(k)} | \mathbf{z}_{\neg i}, \mathbf{w}^{(\neg i)}, \beta) \, .
\end{aligned}
$$

We recognize the first integral as the mean of a Dirichlet distribution which we obtain via conjugacy:

$$
p(\theta^{(w_i)} | \mathbf{z}_{\neg i : w_j = w_i}, \alpha) = \text{Dirichlet}(\mathbf{n}_{:}^{(w_i)\neg i} + \alpha) \, ,
$$

$$
\int_{\theta^{(w_i)}} \theta_k^{(w_i)} p(\theta^{(w_i)} | \mathbf{z}_{\neg i : w_j = w_i}, \alpha) = \frac{n_k^{(w_i)\neg i} + \alpha_k}{\sum_{k'} n_{k'}^{(w_i)\neg i} + \alpha_{k'}} \propto n_k^{(w_i)\neg i} + \alpha_k \, .
$$

The above can also be understood as the probability of the next ball drawn from a multivariate Polya urn model, also known as the Dirichlet-compound multinomial distribution, arising from the posterior predictive distribution of a discrete likelihood with a Dirichlet prior. We will need the full form of such a distribution to analyze the second integral. Once again leveraging conjugacy, we have:

$$\begin{aligned}
&\int_{\phi^{(k)}} \prod_{c=1}^{C_i} \phi^{(k)}_{w_c^{(i)}} p(\phi^{(k)}|\mathbf{z}_{\neg i}, \mathbf{w}^{(\neg i)}, \beta) \\
&= \int_{\phi^{(k)}} \prod_{c=1}^{C_i} \phi^{(k)}_{w_c^{(i)}} \frac{\Gamma(\sum_{v=1}^{D}(n_v^{(k)\neg i} + \beta_v))}{\prod_{v=1}^{D}\Gamma(n_v^{(k)\neg i} + \beta_v)} \prod_{v=1}^{D} {\phi_v^{(k)}}^{n_v^{(k)\neg i} + \beta_v - 1} \\
&= \int_{\phi^{(k)}} \frac{\Gamma(\sum_{v=1}^{D}(n_v^{(k)\neg i} + \beta_v))}{\prod_{v=1}^{D}\Gamma(n_v^{(k)\neg i} + \beta_v)} \prod_{v=1}^{D} {\phi_v^{(k)}}^{n_v^{(k)\neg i} + \beta_v + n_v^{(i)} - 1} \\
&= \frac{\Gamma(\sum_{v=1}^{D}(n_v^{(k)\neg i} + \beta_v))}{\prod_{v=1}^{D}\Gamma(n_v^{(k)\neg i} + \beta_v)} \frac{\prod_{v=1}^{D}\Gamma(n_v^{(k)\neg i} + \beta_v + n_v^{(i)})}{\Gamma(\sum_{v=1}^{D}(n_v^{(k)\neg i} + \beta_v + n_v^{(i)}))} \\
&\times \int_{\phi^{(k)}} \frac{\Gamma(\sum_{v=1}^{D}(n_v^{(k)\neg i} + \beta_v + n_v^{(i)}))}{\prod_{v=1}^{D}\Gamma(n_v^{(k)\neg i} + \beta_v + n_v^{(i)})} \prod_{v=1}^{D} {\phi_v^{(k)}}^{n_v^{(k)\neg i} + \beta_v + n_v^{(i)} - 1} \\
&= \frac{\Gamma(\sum_{v=1}^{D}(n_v^{(k)\neg i} + \beta_v))}{\prod_{v=1}^{D}\Gamma(n_v^{(k)\neg i} + \beta_v)} \frac{\prod_{v=1}^{D}\Gamma(n_v^{(k)\neg i} + \beta_v + n_v^{(i)})}{\Gamma(\sum_{v=1}^{D}(n_v^{(k)\neg i} + \beta_v + n_v^{(i)}))} \,,
\end{aligned}$$

where $n_v^{(i)}$ is the number of times that output word v occurs in the ith context, since the final integral is over the full support of a Dirichlet distribution, which integrates to one. Eliminating terms that aren't affected by the z_i assignment, the above is

$$\begin{aligned}
&\propto \frac{\prod_{v=1}^{D}\Gamma(n_v^{(k)\neg i} + \beta_v + n_v^{(i)})}{\Gamma(\sum_{v=1}^{D}(n_v^{(k)\neg i} + \beta_v + n_v^{(i)}))} \\
&= \frac{\prod_{v=1}^{D}\Big(\Gamma(n_v^{(k)\neg i} + \beta_v)\prod_{j=0}^{n_v^{(i)}-1}(n_v^{(k)\neg i} + \beta_v + j)\Big)}{\Gamma(\sum_{v=1}^{D}(n_v^{(k)\neg i} + \beta_v))\prod_{j=0}^{C_i-1}(\sum_{v=1}^{D}(n_v^{(k)\neg i} + \beta_v) + j)} \\
&\propto \frac{\prod_{v=1}^{D}\prod_{j=0}^{n_v^{(i)}-1}(n_v^{(k)\neg i} + \beta_v + j)}{\prod_{j=0}^{C_i-1}(\sum_{v=1}^{D}(n_v^{(k)\neg i} + \beta_v) + j)} \\
&= \prod_{c=1}^{C_i} \frac{n_{w_c}^{(k)\neg i} + \beta_{w_c} + n_{w_c^{(i,c)}}}{n^{(k)\neg i} + \sum_v \beta_v + c - 1}
\end{aligned}$$

where we have used the fact that $\Gamma(x+n) = (x+n-1)(x+n-2)...(x+1)x\Gamma(x)$ for any $x > 0$, and integer $n \geq 1$. We can interpret this as the probability of drawing the context words under the multivariate Polya urn model, in which the number of "colored balls" (word counts plus prior counts) is increased by one each time a certain color (word) is selected. In other words, in each step, corresponding to the selection of each context word, we draw a ball from the urn, then put it back, *along with another ball of the same color*. The $n_{w_c^{(i,c)}}$ and $c-1$ terms reflect that the counts have been changed by adding these extra balls into the urn in each step. The second to last equation shows that this process is exchangeable: it does not matter which order the balls were drawn in when determining the probability of the sequence. Multiplying this with the term from the first integral, calculated earlier, gives us the final form of the update equation,

$$p(z_i = k|\cdot) \propto (n_k^{(w_i)\neg i} + \alpha_k) \prod_{c=1}^{C_i} \frac{n_{w_c}^{(k)\neg i} + \beta_{w_c} + n_{w_j^{(i,c)}}}{n^{(k)\neg i} + \sum_v \beta_v + c - 1} \,.$$

References

Airoldi, E., Blei, D., Feinberg, S., and Xing, E. (2008). Mixed membership stochastic blockmodels. *Journal of Machine Learning Research*, 9:1981–2014.

Airoldi, E. M., Blei, D. M., Erosheva, E. A., and Fienberg, S. E. (2014). Introduction to mixed membership models and methods. In *Handbook of Mixed Membership Models and Their Applications*. Chapman and Hall/CRC.

Barnard, K., Duygulu, P., Forsyth, D., Freitas, N. d., Blei, D. M., and Jordan, M. I. (2003). Matching words and pictures. *Journal of Machine Learning Research*, 3(Feb):1107–1135.

Bengio, Y., Ducharme, R., Vincent, P., and Jauvin, C. (2003). A neural probabilistic language model. *Journal of Machine Learning Research*, 3(Feb):1137–1155.

Besag, J. (1986). On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 259–302.

Blei, D., Ng, A., and Jordan, M. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Bolukbasi, T., Chang, K.-W., Zou, J. Y., Saligrama, V., and Kalai, A. T. (2016). Man is to computer programmer as woman is to homemaker? Debiasing word embeddings. In *Advances in Neural Information Processing Systems 29*, pages 4349–4357.

Brent, M. R. (1999). An efficient, probabilistically sound algorithm for segmentation and word discovery. *Machine Learning*, 34(1):71–105.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537.

Das, R., Zaheer, M., and Dyer, C. (2015). Gaussian LDA for topic models with word embeddings. In *ACL*, pages 795–804.

Erosheva, E., Fienberg, S., and Lafferty, J. (2004). Mixed-membership models of scientific publications. *Proceedings of the National Academy of Sciences of the United States of America*, 101(Suppl 1):5220–5227.

Erosheva, E. A. (2003). Bayesian estimation of the grade of membership model. *Bayesian Statistics*, 7:501–510.

Fei-Fei, L. and Perona, P. (2005). A Bayesian hierarchical model for learning natural scene categories. In *Proceedings of the 2005 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 524–531. IEEE.

Griffiths, T. L., Steyvers, M., and Tenenbaum, J. B. (2007). Topics in semantic representation. *Psychological review*, 114(2):211.

Grimmer, J. (2010). A Bayesian hierarchical topic model for political texts: Measuring expressed agendas in senate press releases. *Political Analysis*, pages 1–35.

Guo, F., Blundell, C., Wallach, H., and Heller, K. (2015). The Bayesian echo chamber: Modeling social influence via linguistic accommodation. In *Artificial Intelligence and Statistics*, pages 315–323.

Gutmann, M. and Hyvärinen, A. (2010). Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *AISTATS*.

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

Harris, Z. S. (1954). Distributional structure. *Word*, 10(2-3):146–162.

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

Hinton, G. E., Mcclelland, J. L., and Rumelhart, D. E. (1986). Distributed representations. In *Parallel distributed processing: Explorations in the microstructure of cognition. Volume 1: Foundations*, chapter 3, pages 77–109. MIT Press, Cambridge, MA.

Hofmann, T. (1999). Probabilistic latent semantic indexing. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 50–57. ACM.

Huang, E. H., Socher, R., Manning, C. D., and Ng, A. Y. (2012). Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 873–882. Association for Computational Linguistics.

Kiros, R., Zhu, Y., Salakhutdinov, R. R., Zemel, R., Urtasun, R., Torralba, A., and Fidler, S. (2015). Skip-thought vectors. In *Advances in neural information processing systems*, pages 3294–3302.

Li, A. Q., Ahmed, A., Ravi, S., and Smola, A. J. (2014). Reducing the sampling complexity of topic models. In *KDD*, pages 891–900. ACM.

Liu, Y., Liu, Z., Chua, T.-S., and Sun, M. (2015). Topical word embeddings. In *AAAI*, pages 2418–2424.

Maaten, L. v. d. and Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(Nov):2579–2605.

Manton, K. G., Tolley, H. D., and Woodbury, M. A. (1994). *Statistical applications using fuzzy sets*. Wiley-Interscience.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. *ICLR*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119.

Mimno, D. (2012). Computational historiography: Data mining in a century of classics journals. *Journal on Computing and Cultural Heritage (JOCCH)*, 5(1):3.

Mnih, A. and Kavukcuoglu, K. (2013). Learning word embeddings efficiently with noise-contrastive estimation. In *Advances in Neural Information Processing Systems*, pages 2265–2273.

Mnih, A. and Teh, Y. W. (2012). A fast and simple algorithm for training neural probabilistic language models. In *Proceedings of the 29th International Conference on Machine Learning*.

Nguyen, V.-A., Boyd-Graber, J., Resnik, P., Cai, D. A., Midberry, J. E., and Wang, Y. (2014). Modeling topic control to detect influence in conversations using nonparametric topic models. *Machine Learning*, 95(3):381–421.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543.

Pritchard, J. K., Stephens, M., and Donnelly, P. (2000). Inference of population structure using multilocus genotype data. *Genetics*, 155(2):945–959.

Reisinger, J. and Mooney, R. J. (2010). Multi-prototype vector-space models of word meaning. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 109–117. Association for Computational Linguistics.

Teh, Y. W., Jordan, M. I., Beal, M. J., and Blei, D. M. (2006). Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101(476).

Tian, F., Dai, H., Bian, J., Gao, B., Zhang, R., Chen, E., and Liu, T.-Y. (2014). A probabilistic model for learning multi-prototype word embeddings. In *COLING*, pages 151–160.

Vaswani, A., Zhao, Y., Fossum, V., and Chiang, D. (2013). Decoding with large-scale neural language models improves translation. In *EMNLP*, pages 1387–1392.

Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., and Fidler, S. (2015). Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 19–27.