Variational Autoencoders for Topic Modeling

Kelly Geyer
klgeyer@bu.edu

1. Task

Topic models are widely used for unsupervised learning of groupings, or topics, for a given corpus of documents. They are used in a variety of applications, which include language processing and bioinformatics. While there are several approaches to topic modeling, in this project we focus on improving the probabilistic topic models that assume the generative process of Latent Dirichlet Analysis (LDA), introduced by [2]. In particular, this class of topic models uses Baysian modeling to estimate both the word-topic and document-topic probabilities. In recent works, probabilistic topic modeling has been improved in both performance and computational efficiency using VAEs [12, 15, 4].

Objective. We want to explore potential improvements for topic modeling, using Variational Autoencoders (VAEs) [10]. The authors of [15] introduce two methods of VAE topic models, and compare them to traditional LDA. Ultimately, we will replicate the analysis from [15].

2. Related Work

In this project we compare the performance of three different types of topic models. A topic model is an unsupervised method of clustering words into clusters, otherwise called *topics*, as shown by Figure 1. They take in a collection of documents, and return two types of summaries: (i) word-by-topic scores, and (ii) distributions over topics for each document.

Latent Dirichlet Analysis (LDA). We consider generative topic models which assume that words, and respective their topics, are generated from a mixture probability distributions. That is, the components of the mixture come from different *topics*. The original generative topic model, called Latent Dirichlet Analysis (LDA), was introduced by [2]. It assumes a generative process described by Algorithm 1. Ultimately, LDA seeks to use Bayes' Theorem to estimate the parameters θ (document-topic scores) and β_{z_n} (topic-document-word scores). Overall, the marginal likelihood of a document d is

Algorithm 1: Generative process of LDA.

```
for each document $d = 1, 2, \ldots, D$ do
    Draw topic from distribution $\theta \sim$ Dirichlet($\alpha$)
    for each word in d, $w = 1, 2, \ldots, V$ do
        Sample topic $z_n \sim$ Multinomial($1, \theta$)
        Sample word $w_n \sim$ Multinomial($1, \beta_{z_n}$)
    end
end
```

$$p(d|\alpha, \beta) = \int_{\theta} \left(\prod_{n=1}^{N} \sum_{z_n=1}^{K} p(w_n|z_n, \beta) p(z_n|\theta) \right) p(\theta|\alpha) d\theta \tag{1}$$

where K is the number of topics.

Autoencoded Variational Inference for Topic Models (AVITM). Applying a traditional Variational Autoencoder (VAE) to the LDA topic model formulation (1) poses some some issues. First, is that it is difficult to apply the reparameterization trick to find the Evidence Lower Bound (ELBO) since $\theta \sim$ Dirichlet(α) [15]. The AVITM counters this issue two ways. First, it integrates out the z's from Equation (1),

$$p(d|\alpha, \beta) = \int_{\theta} \left(\prod_{n=1}^{N} p(w_n|\beta, \theta) \right) p(\theta|\alpha) d\theta \tag{2}$$

Next, it uses a Laplace approximation from [7] for $p(\theta|\alpha)$ in Equation 2,

$$p(\theta(h)|\alpha) = \frac{\Gamma(\Sigma_{k=1}^{K} \alpha_k)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \prod_{k=1}^{K} \theta_k^{\alpha_k} \cdot g(1^T h), \tag{3}$$

where $\theta = \sigma(h)$ and $\sigma(\cdot)$ is the softmax function. Using these tricks together gives rise to an variational objective that is much easier to optimize [15].

ProdLDA. In LDA, we assume that the probability $p(d|\theta, \beta)$ is a mixture of multinomials, as illustrated in Algorithm 1. However this can often result in poor quality

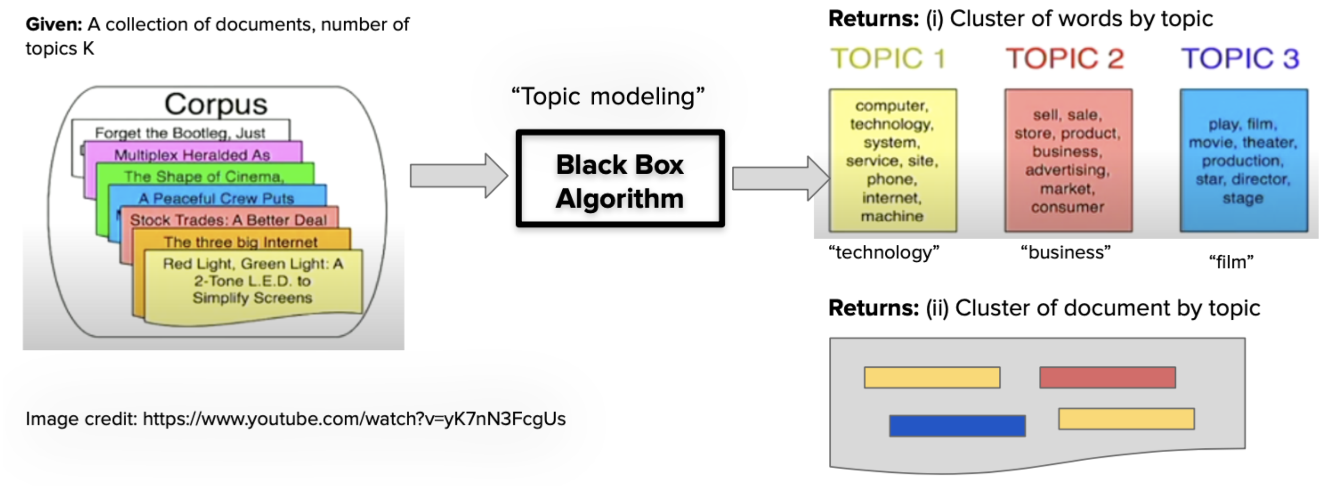

Given: A collection of documents, number of topics K
Corpus
Forget the Bootleg, Just
Multiplex Heralded As
The Shape of Cinema,
A Peaceful Crew Puts
Stock Trades: A Better Deal
The three big Internet
Red Light, Green Light: A 2-Tone L.E.D. to Simplify Screens
"Topic modeling"
Black Box Algorithm
Returns: (i) Cluster of words by topic
TOPIC 1
TOPIC 2
TOPIC 3
computer, technology, system, service, site, phone, internet, machine
sell, sale, store, product, business, advertising, market, consumer
play, film, movie, theater, production, star, director, stage
"technology"
"business"
"film"
Returns: (ii) Cluster of document by topic
Image credit: https://www.youtube.com/watch?v=yK7nN3FcgUs

Figure 1. Concept of a topic model: sort the words of a collection of documents into topics. They return (i) topic-by-word scores, and (ii) distributions for each topic over each document.

topics that do not have qualitative value, even when they might have low perplexity scores [5, 8]. ProdLDA uses a weighted product of scores to counter this issue, instead of the multinomial prior from Algorithm 1. The resulting generative process of ProdLDA remains similar to LDA, except that β is unnormalized, and the conditional of w_n is $w_n|\beta,\theta \sim \text{Multinomial}(1, \sigma(\beta\theta))$ [15]. The approach for ProdLDA is explained further in Section 3.

Motivation. In summary, there are two main motivations associated with using VAEs as topic models.

1. *Computational efficiency* - Probabilistic topic models such as LDA are estimated using Gibbs sampling. The draw-back to this approach is that it is very slow in practice. VAEs use techniques that frame estimation as an optimization problem, and can be solved much more quickly than Gibbs sampling. However, one can use variational inference to find θ and β, which make the estimation of LDA faster. In the work [9], the authors improve the scalability and speed of LDA by using stochastic variational inference to estimate posterior distributions.
2. *Interpretability* - The document-topic and word-topic probabilities are intuitive variables for summarizing a corpus of documents. However, in practice the resulting topic keywords may appear as nonsensical, and the output may be prone to subjectivity in interpretation [5]. Previously, VAEs for topic models have proposed enforcing sparsity upon the topic probability matrices. To counter this, there is an emerging body on extending VAEs to sparse data and regularization: [1, 6, 17]. In the case of AVITM and ProdLDA, the authors atempt to improve topic quality using a Laplace approximation of the Dirichlet prior on θ.

3. Approach

In this section, I describe the approach I took to implement ProdLDA. My code for this method is largely based upon the author's open-source code [1] However, I carefully looked at the code and made changes since it is several years old. For instance, it was developed in Python version 2.7, and uses depreciated versions of TensorFlow (v1.13). The following summarizes the implementation of ProdLDA:

Data. I used the preprocessed data set provided by [15]. It is the 20 Newsgroup data set split into training and testing sets, of 11,258 and 7,487 articles respectively.

Optimizer. ADAM with a learning rate of 0.002, and $\beta_1 = 0.99$.

Variational Objective (loss).

$$\begin{aligned} L(\theta) = \sum_{d=1}^{D} \Bigg[-\frac{1}{2} \Big(tr\left(\Sigma_1^{-1}\Sigma_0\right) + (\mu_1 - \mu_0)^T \Sigma_1^{-1} (\mu_1 - \mu_0) \\ - K + \log\big(\frac{|\Sigma_1|}{|\Sigma_0|}\big) \\ + E_{\epsilon\sim\mathcal{N}(0,1)} \left[w_d^T \log\left(\sigma(\beta)\sigma(\mu_0 + \Sigma_0^{1/2}\epsilon)\right)\right] \end{aligned} \tag{4}$$

[1] The open-source code for the paper [15] can be found at https://github.com/akashgit/autoencoding_vi_for_topic_models.

There are two feed forward neural networks f_μ and f_Σ. For a document d, we define $q(\theta)$ to be logistic normal with $\mu_0 = f)\mu(d, \delta)$ and diagonal covariance $\Sigma_0 = \text{diag}(f_\Sigma(d, \delta))$. We can generate samples from $q(\theta)$ by $\epsilon \sim \mathcal{N}(0, I)$. The parameter $\theta = \sigma(\mu_0 + \Sigma_0^{1/2}\epsilon)$.

Forward Steps. Steps 1-3 are the encoding process, steps 4-6 deal with the latent layer, and the remaining steps decode and reconstruction.

1. Softplus of Linear transformation $f_1 : (D \times V) \to (D \times 100)$
2. Softplus of Linear transformation $f_2 : (D \times 100) \to (D \times 100)$
3. Perform dropout of the encoder output by removing scores of 0.2 or less
4. Estimate the posterior mean μ_θ and variance $^2_\theta$ of θ by batch normalization
5. Take a sample $\epsilon \sim \mathcal{N}(0, I)$
6. Estimate the latent layer, $z = \mu_\theta +^2_\theta \epsilon$
7. Take softmax of z to obtain probabilities
8. Perform dropout, remove the probabilities of 0.2 or less
9. Obtain the reconstructed distribution over vocabulary, mirroring the steps of the encoder.

Baseline methods. In all I compare two methods with ProdLDA: (i) AVITM and (ii) Variational LDA. I used an implementation of AVITM from the authors' open-source code. Similarly as for ProdLDA, I had to carefully review this code line-by-line and make changes so it would be compatible for Python 3.7. I used the implementation of variational LDA from the Python module `sklearn` [3, 14].

Evaluation methods. I performed the evaluation methods myself, and implemented the necessary code.

4. Dataset and Metrics

The data set used for evaluation is a collection of news articles written in English, covering various topics. In particular, this project uses the 20 Newsgroups data set [11]. This is a collection of nearly 20,000 articles, which cover 20 different topics such as sports, politics, technology, and several others.

Data processing. The authors of [15] divided the 20 Newsgroup data set into a training and test set, containing 11,258 and 7,487 articles respectively. Their preprocessing included tokenization, removal of non UTF-8 characters, and removing English stop-words. The vocabulary of the processed data is approximately 2,000 words. In all, the resulting features are a document-vocabulary count matrix where rows represent independent documents, and there is a column for each vocabulary term.

Metrics. Recall that our analysis is unsupervised in that we are not using any truth labels. Therefore, we cannot use traditional error metrics. Instead, we want to evaluate the interpretability of the model output with various measures. We consider the following metrics for evaluation:

1. *Perplexity* - This metric is based upon the log-likelihood $\mathcal{L}$. Ideally, a high value for log-likelihood would imply that the model is representative of the data. The perplexity is inversely related to the log-likelihood so lower scores are desired, as shown in Equation (5). To calculate perplexity, the articles are divided into training and testing sets. A topic model is fitted using the training data, and perplexity is calculated using the held-out test set.

$$\text{perplexity(test docs)} = \exp\left(-\frac{\mathcal{L}(\text{test docs})}{\text{\# of words}}\right) \quad (5)$$

2. *Topic coherence* - Topic coherence is a measure of how similar the semantic meaning of words are within a topic. Ideally, we want words within a topic to be highly associated in meaning. There are several measures of topic coherence, and in this report we use the UMass measure from [13]. The measure uses a pairwise score function between vocabulary terms,

$$\text{Coherence}_{\text{UMass}} = \sum_{i<j} \log\left(\frac{D(w_i, w_j) + 1}{D(w_i)}\right), \quad (6)$$

   where $D(w_i)$ is the count of documents containing the word w_i, and $D(w_i, w_j)$ is the count of documents containing both words w_i and w_j. We calculate the UMass measure for each topic, using the top 10 words.

3. *Qualitative Analysis* - We will evaluate the top ten words assigned to each topic to see if there is a common theme. An undesirable result would be to have topic keywords that appear to be unrelated. We hope for the resulting keywords to ultimately have a human perscribable theme.

5. Results

In our experiment we fit the models ProdLDA, AVITM, and LDA VI to the training data set. We calculated the evaluation metrics *perplexity* and *coherence* using the held-out test set. These evaluation methods are explained in Section 4.

Hyperparameter	ProdLDA	AVITM	LDA VI
Dimension of layer 1	100	100	-
Dimension of layer 2	100	100	-
Number of topics	50	50	50
Batch size	200	200	200
Learning Rate	0.002	0.01	0.7
Training epochs	100	100	100

Table 1. This table contains the hyperparameters used in topic modeling experiment.

Hyperparameters. The models ProdLDA and AVITM have the same hyperparameters, and we used the values from the open-source code for [15]. The default hyperparameters were used for LDA VI too, defined in [3], which also match the open-source code for [15]. The hyperparameters in our experiment are explained in Table 1.

Evaluation. We consider three different types of evaluation, which are introduced in Section 4.

1. *Perplexity* - First, we look at the perplexity scores for each model that are featured in Table 1. Recall that perplexity measures model fit as described by Equation (5), and that lower values are favorable. This result suggests that LDA VI is the best model. In spirit, this result matches the original results in that the models are ranked $prodLDA < AVITM < LDAVI$. However, my perplexity values are different from the original paper [15].

2. *Topic Coherence* - It is demonstrated by [5] that perplexity scores may not correspond well with human judgement. Therefore, we proceed with calculating an additional measure of quality. We calculate a coherence score for each of the resulting 50 topics for each model, described by Equation (6). Recall that this value measures intrinsic score, and therefore may be more indicative of the quality of topic keywords. Figure 2 contains frequency histograms of coherence scores for each topic (50 topics total), for all three models. The vertical lines represent the average coherence scores for each model. Note that higher values of coherence are desirable. In spirit, this result matches the original results in that the models are ranked $LDAVI < AVITM < ProdLDA$. However, it is not clear that I am using the same coherence calculation as [15]. There are several well-known methods of calculating topic coherence, and [15] neither mentions a specific one or include their calculation in their open-source code.

3. *Qualitative Analysis* - In this analysis, we look at the top 10 keywords of a handful of topics (out of 50 topics total), for each model. These results are featured in Figure 3. In general, the models ProdLDA and AVITM had higher quality topics than LDA VI. That is, the words in these topics appeared to be more similar to one another and it was easy to apply a human label. Additionally, they seemed to identify the same topics, even on a detailed level (E.g., sports/baseball). The quality of the topics from LDA VI is noticeably lower. The words just don't match as well as they do for the topics of the previous models. Plus, sometimes LDA VI includes a keyword that seems inappropriate for the topic (E.g., 'come' and 'say' in the Religion/Christianity topic.

Model	Perplexity Score
prodLDA	1153.7977
AVITM	1130.1957
LDA VI	777.7551

Table 2. Perplexity scores for each model. This score evaluates model fit, and lower value of perplexity are desirable.

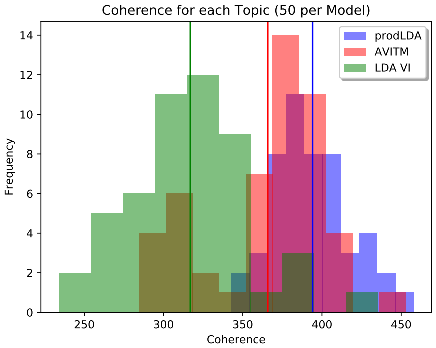


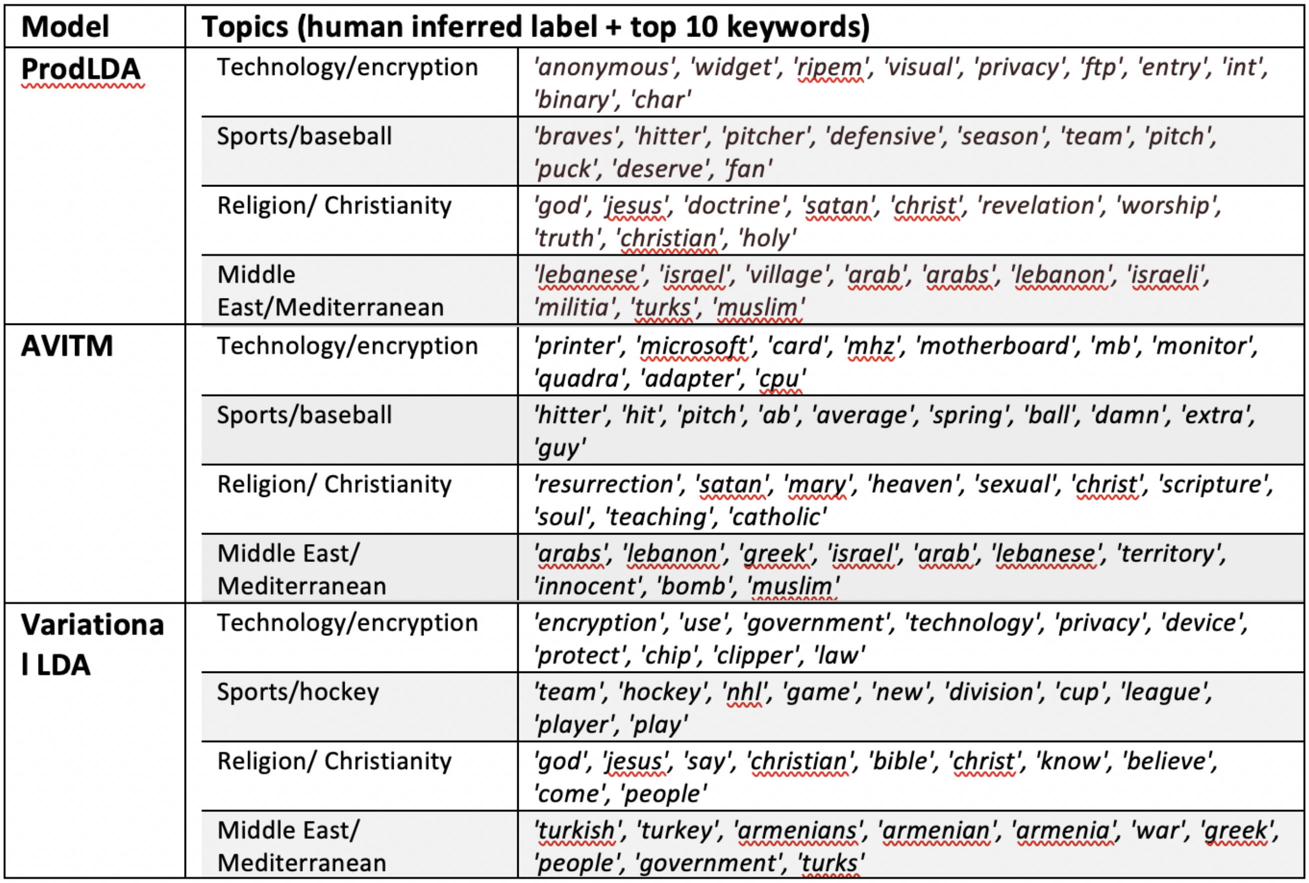
Figure 2. Frequency histogram of topic coherence scores (50 topics total) for each model. The vertical lines represent the mean score for a model's topics. Higher coherence scores are more desirable, and indicates intrinsic value of a topic.

Task	File names
Run the experiment	run.py
Summarize the results of experiment	make_pls.py
Format score code as a module	source/__init__.py
Preprocess & format data	source/data.py
Implement AVITM model	source/nvlda.py
Implement ProdLDA model	source/prodlda.py
Calculate evaluation metrics	source/results.py

Table 3. List of tasks and their respective files.

6. Detailed Roles

Table 6 contains the description of each file that I either edited, or wrote from scratch.



Model	Topics (human inferred label + top 10 keywords)	
ProdLDA	Technology/encryption	'anonymous', 'widget', 'ripem', 'visual', 'privacy', 'ftp', 'entry', 'int', 'binary', 'char'
	Sports/baseball	'braves', 'hitter', 'pitcher', 'defensive', 'season', 'team', 'pitch', 'puck', 'deserve', 'fan'
	Religion/ Christianity	'god', 'jesus', 'doctrine', 'satan', 'christ', 'revelation', 'worship', 'truth', 'christian', 'holy'
	Middle East/Mediterranean	'lebanese', 'israel', 'village', 'arab', 'arabs', 'lebanon', 'israeli', 'militia', 'turks', 'muslim'
AVITM	Technology/encryption	'printer', 'microsoft', 'card', 'mhz', 'motherboard', 'mb', 'monitor', 'quadra', 'adapter', 'cpu'
	Sports/baseball	'hitter', 'hit', 'pitch', 'ab', 'average', 'spring', 'ball', 'damn', 'extra', 'guy'
	Religion/ Christianity	'resurrection', 'satan', 'mary', 'heaven', 'sexual', 'christ', 'scripture', 'soul', 'teaching', 'catholic'
	Middle East/ Mediterranean	'arabs', 'lebanon', 'greek', 'israel', 'arab', 'lebanese', 'territory', 'innocent', 'bomb', 'muslim'
Variational LDA	Technology/encryption	'encryption', 'use', 'government', 'technology', 'privacy', 'device', 'protect', 'chip', 'clipper', 'law'
	Sports/hockey	'team', 'hockey', 'nhl', 'game', 'new', 'division', 'cup', 'league', 'player', 'play'
	Religion/ Christianity	'god', 'jesus', 'say', 'christian', 'bible', 'christ', 'know', 'believe', 'come', 'people'
	Middle East/ Mediterranean	'turkish', 'turkey', 'armenians', 'armenian', 'armenia', 'war', 'greek', 'people', 'government', 'turks'

Figure 3. Display of the top 10 keywords for a handful of topics (out of 50 topics total) for each model.

7. Conclusion

The objective of this project was to study and replicate the results of the paper [15]. I was able to achieve the same conclusions as the original paper, as discussed in Section 5. That is, the models ProdLDA and AVITM find more meaningful topics than LDA VI due to their reparameterization strategy. This experiment also suggested that topic coherence is a more meaningful quantitative score than perplexity (I.e., model fit) in this setting. However, my estimated perplexity and topic coherence scores were very different from the original paper.

Challenges. The primary challenge in this work was learning the details of implementation for the experiments in [15]. This information is critical for reproducible experiments.

The hyperparameters are not defined in the paper, along with the approach of the model architecture, described in Section 3. I learned about them from reading the open-source code of the paper. I believe that it is very likely that the hyperparameters defined in code are the same values used the paper's experiments. Yet, I cannot be certain this is true.

Another challenge is that it is not clear *how* or why many the hyperparamters are selected. Varying the values of the hyperparameters used in the experiment may produce different qualitative results for ProdLDA and AVITM. For instance, it has been shown by [16] that generative topic models can be sensitive to the prior of the Dirichlet parameter θ. In addition to prior values, the models may be sensitive to varying the number of predefined topics.

8. Code Repository

The GitHub repository for this project is `https://github.com/kgeyer/cs591-project`.

References

[1] Luigi Antelmi, Nicholas Ayache, Philippe Robert, and Marco Lorenzi. Sparse multi-channel variational autoencoder for the joint analysis of heterogeneous data. 2019.

[2] David M Blei. Probabilistic topic models. *Communications of the ACM*, 55(4):77–84, 2012.

[3] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122, 2013.

[4] Sophie Burkhardt and Stefan Kramer. Decoupling sparsity and smoothness in the dirichlet variational autoencoder topic model. *Journal of Machine Learning Research*, 20(131):1–27, 2019.

[5] Jonathan Chang, Sean Gerrish, Chong Wang, Jordan L Boyd-Graber, and David M Blei. Reading tea leaves: How humans interpret topic models. In *Advances in neural information processing systems*, pages 288–296, 2009.

[6] Bin Dai, Yu Wang, John Aston, Gang Hua, and David Wipf. Connections with robust pca and the role of emergent sparsity in variational autoencoder models. *The Journal of Machine Learning Research*, 19(1):1573–1614, 2018.

[7] Philipp Hennig, David Stern, Ralf Herbrich, and Thore Graepel. Kernel topic models. In *Artificial Intelligence and Statistics*, pages 511–519, 2012.

[8] Geoffrey E Hinton and Russ R Salakhutdinov. Replicated softmax: an undirected topic model. In *Advances in neural information processing systems*, pages 1607–1614, 2009.

[9] Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

[10] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[11] Ken Lang. Newsweeder: Learning to filter netnews. In *Machine Learning Proceedings 1995*, pages 331–339. Elsevier, 1995.

[12] Yishu Miao, Edward Grefenstette, and Phil Blunsom. Discovering discrete latent topics with neural variational inference. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2410–2419. JMLR. org, 2017.

[13] David Mimno, Hanna M Wallach, Edmund Talley, Miriam Leenders, and Andrew McCallum. 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, 2011.

[14] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[15] Akash Srivastava and Charles Sutton. Autoencoding variational inference for topic models. *arXiv preprint arXiv:1703.01488*, 2017.

[16] Hanna M Wallach, David M Mimno, and Andrew McCallum. Rethinking lda: Why priors matter. In *Advances in neural information processing systems*, pages 1973–1981, 2009.

[17] He Zhao, Piyush Rai, Lan Du, Wray Buntine, and Mingyuan Zhou. Variational autoencoders for sparse and overdispersed discrete data. *arXiv preprint arXiv:1905.00616*, 2019.