Improving Neural Topic Models using Knowledge Distillation

Alexander Hoyle*
Computer Science
University of Maryland
College Park, MD
hoyle@umd.edu

Pranav Goel*
Computer Science
University of Maryland
College Park, MD
pgoel1@umd.edu

Philip Resnik
Linguistics / UMIACS
University of Maryland
College Park, MD
resnik@umd.edu

Abstract

Topic models are often used to identify human-interpretable topics to help make sense of large document collections. We use knowledge distillation to combine the best attributes of probabilistic topic models and pretrained transformers. Our modular method can be straightforwardly applied with any neural topic model to improve topic quality, which we demonstrate using two models having disparate architectures, obtaining state-of-the-art topic coherence. We show that our adaptable framework not only improves performance in the aggregate over all estimated topics, as is commonly reported, but also in head-to-head comparisons of aligned topics.

1 Introduction

The core idea behind the predominant *pretrain and fine-tune* paradigm for transfer learning in NLP is that general language knowledge, gleaned from large quantities of data using unsupervised objectives, can serve as a foundation for more specialized endeavors. Current practice involves taking the full model that has amassed such general knowledge and fine-tuning it with a second objective appropriate to the new task (see Raffel et al., 2019, for an overview). Using these methods, pre-trained transformer-based language models (e.g., BERT, Devlin et al., 2019) have been employed to great effect on a wide variety of NLP problems, thanks, in part, to a fine-grained ability to capture aspects of linguistic context (Clark et al., 2019; Liu et al., 2019; Rogers et al., 2020).

However, this paradigm introduces a subtle but insidious limitation that becomes evident when the downstream application is a topic model. A topic model may be cast as a (stochastic) autoencoder (Miao et al., 2016), and we could fine-tune a pre-trained transformer with an identical document reconstruction objective. But in replacing the original topic model, we lose the property that makes it desirable: its interpretability. The transformer gains its contextual power from its ability to exploit a huge number of parameters, while the interpretability of a topic model comes from a dramatic dimensionality reduction.

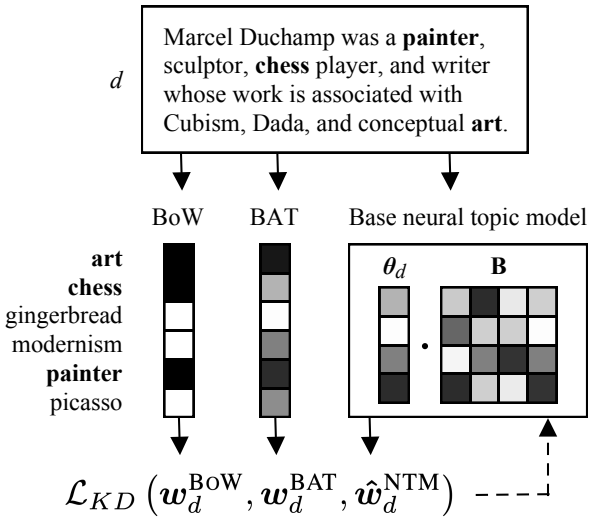

Figure 1: Improving a base neural topic model with knowledge distillation. A document is mapped through both a standard BoW representation and a **B**ERT-based **A**uto-encoder "**T**eacher" (BAT), yielding two distributions over words. These are used as the ground truth in the "student" topic model's document reconstruction loss $\mathcal{L}_{KD}$ (backpropagated along the dotted line). Crucially, the BAT distribution assigns mass to unobserved but related terms (unbolded).

We combine the advantages of these two approaches—the rich contextual language knowledge in pretrained transformers and the intelligibility of topic models—using *knowledge distillation* (Hinton et al., 2015). In the original formulation, knowledge distillation involves training a parameter-rich *teacher* classifier on large swaths

*Equal contribution.

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 1752–1771,
November 16–20, 2020. ©2020 Association for Computational Linguistics

of data, then using its high-quality probability estimates over outputs to guide a smaller *student* model. Since the information contained in these estimates is useful—a picture of an ox will yield higher label probabilities for BUFFALO than APRICOT—the student needs less data to train and can generalize better.

We show how this principle can apply equally well to improve unsupervised topic modeling, which to our knowledge has not previously been attempted. While distillation usually involves two models of the same type, it *can* also apply to models of differing architectures. Our method is conceptually quite straightforward: we fine-tune a pretrained transformer (Sanh et al., 2019) on a document reconstruction objective, where it acts in the capacity of an autoencoder. When a document is passed through this BERT autoencoder, it generates a distribution over words that includes unobserved but related terms. We then incorporate this distilled document representation into the loss function for topic model estimation. (See Figure 1.)

To connect this method to the more standard supervised knowledge distillation, observe that the unsupervised "task" for both an autoencoder and a topic model is the reconstruction of the original document, i.e. prediction of a distribution over the vocabulary. The BERT autoencoder, as "teacher", provides a dense prediction that is richly informed by training on a large corpus. The topic model, as "student", is generating its own prediction of that distribution. We use the former to guide the latter, essentially as if predicting word distributions were a multi-class labeling problem.[1] Our approach, which we call **B**ERT-based **A**utoencoder as **T**eacher **(BAT)**, obtains best-in-class results on the most commonly used measure of topic coherence, normalized pointwise mutual information (NPMI, Aletras and Stevenson, 2013) compared against recent

state-of-the-art-models that serve as our baselines.

In order to accomplish this, we adopt neural topic models (NTM, Miao et al., 2016; Srivastava and Sutton, 2017; Card et al., 2018; Burkhardt and Kramer, 2019; Nan et al., 2019, *inter alia*), which use various forms of black-box distribution-matching (Kingma and Welling, 2014; Tolstikhin et al., 2018).[2] These now surpass traditional methods (e.g. LDA, Blei, 2003, and variants) in topic coherence. In addition, it is easier to modify the generative model of a neural topic model than for a classic probabilistic latent-variable model, where changes generally require investing effort in new variational inference procedures or samplers. In fact, because we leave the base NTM unmodified, our approach is flexible enough to easily accommodate *any* neural topic model, so long as it includes a word-level document reconstruction objective. We support this claim by demonstrating improvements on models based on both Variational (Card et al., 2018) and Wasserstein (Nan et al., 2019) auto-encoders.

To summarize our contributions:

- We introduce a novel coupling of the knowledge distillation technique with generative graphical models.
- We construct knowledge-distilled neural topic models that achieve better topic coherence than their counterparts without distillation on three standard English-language topic-modeling datasets.
- We demonstrate that our method is not only effective but modular, by improving topic coherence in a base state-of-the-art model by modifying only a few lines of code.[3]
- In addition to showing overall improvement across topics, our method preserves the topic analysis of the base model and improves coherence on a topic-by-topic basis.

2 Methodology

2.1 Background on topic models

Topic modeling is a well-established probabilistic method that aims to summarize large document corpora using a much smaller number of latent *topics*. The most prominent instantiation, LDA (Blei, 2003), treats each document as a mixture over K latent topics, $\boldsymbol{\theta}_d$, where each topic is a distribution

[1] An interesting conceptual link here can be found in Latent Semantic Analysis (LSA, Landauer and Dumais, 1997), an early predecessor of today's topic models. The original discussion introducing LSA has a very autoencoder-like flavor, explicitly illustrating the deconstruction of a collection of sparsely represented documents and the reconstruction of a dense document-word matrix.

[2] As a standard example, Srivastava and Sutton (2017) encode a document's bag-of-words with a neural network to parameterize the latent topic distribution, then sample from the distribution to reconstruct the BoW.

[3] See Appendix F. Our full implementation, including dataset preprocessing, is available at `github.com/ahoho/kd-topic-models`.

over words $\boldsymbol{\beta}_k$. By presenting topics as ranked word lists and documents in terms of their probable topics, topic models can provide legible and concise representations of both the entire corpus and individual documents.

In classical topic models like LDA, distributions over the latent variables are estimated with approximate inference algorithms tailored to the generative process. Changes to the model specification—for instance, the inclusion of a supervised label—requires attendant changes in the inference method, which can prove onerous to derive. For some probabilistic models, this problem may be circumvented by the variational auto-encoder (VAE, Kingma and Welling, 2014), which introduces a *recognition model* that approximates the posterior with a neural network. As a result, *neural topic models* have capitalized on the VAE framework (Srivastava and Sutton, 2017; Card et al., 2018; Burkhardt and Kramer, 2019, *inter alia*) and other deep generative models (Wang et al., 2019; Nan et al., 2019). In addition to their flexibility, the best models now yield more coherent topics than LDA.

Although our method (Section 2.3) is agnostic as to the choice of neural topic model, we borrow from Card et al. (2018) for both formal exposition and our base implementation (Section 3). Card et al. (2018) develop SCHOLAR, a generalization of the first successful VAE-based neural topic model (PRODLDA, Srivastava and Sutton, 2017). The generative story is broadly similar to that of LDA, although the uniform Dirichlet prior is replaced with a logistic normal ($\mathcal{LN}$):[4]

For each document d:

- Draw topic distribution $\boldsymbol{\theta}_d \sim \mathcal{LN}(\boldsymbol{\alpha}_0)$
- For each word w_{id} in the document:

$$w_{id} \sim \text{Multinomial}\left(1, f(\boldsymbol{\theta}_d, \mathbf{B})\right)$$

Following PRODLDA, $\mathbf{B}$ is a $K \times V$ matrix where each row corresponds to the kth topic-word probabilities in log-frequency space. The multinomial distribution over a document's words is parameterized by

$$f(\boldsymbol{\theta}_d, \mathbf{B}) = \sigma\left(\boldsymbol{m} + \boldsymbol{\theta}_d^\top \mathbf{B}\right) \tag{1}$$

where $\boldsymbol{m}$ is a vector of fixed empirical background word frequencies and $\sigma(\cdot)$ is the softmax function. We highlight that each document is treated as a bag of words, $\boldsymbol{w}_d^{\text{BoW}}$.

To perform inference on the model, VAE-based models like SCHOLAR approximate the true intractable posterior $p(\boldsymbol{\theta}_d \mid \cdot)$ with a neural *encoder* network $g(\boldsymbol{w}_d)$ that parameterizes the variational distribution $q\left(\boldsymbol{\theta}_d \mid g(\cdot)\right)$ (here, a logistic normal with diagonal covariance). The Evidence Lower BOund (ELBO) is therefore

$$\begin{aligned}\text{ELBO} &= \mathbb{E}_{q(\boldsymbol{\theta}_d \mid \cdot)}\left[\mathcal{L}_R\right] \\ &\quad - \text{KL}\left[q\left(\boldsymbol{\theta}_d \mid \boldsymbol{w}_d^{\text{BoW}}, \boldsymbol{x}_d\right) \mid\mid p\left(\boldsymbol{\theta}_d\right)\right], \quad (2)\\ \mathcal{L}_R &= \left(\boldsymbol{w}_d^{\text{BoW}}\right)^\top \log f(\boldsymbol{\theta}_d, \mathbf{B}), \quad (3)\end{aligned}$$

which is optimized with stochastic gradient descent. The form of the reconstruction error $\mathcal{L}_R$ is a consequence of the independent multinomial draws.

2.2 Background on knowledge distillation

It is instructive to think of Eq. (1) as a latent logistic regression, intended to approximate the distribution over words in a document. Under this lens, the neural topic model outlined above can be cast as a multi-label classification problem. Indeed, it accords with the standard structure: there is a softmax over logits estimated by a neural network, coupled with a cross-entropy loss.

However, because $\boldsymbol{w}_d^{\text{BoW}}$ is a sparse bag of words, the model is limited in its ability to generalize. During backpropagation (Eq. (3)), the topic parameters will only update to account for observed terms, which can lead to overfitting and topics with suboptimal coherence.

In contrast, dense document representations can capture rich information that bag-of-words representations cannot.

These observations motivate our use of *knowledge distillation* (KD, Hinton et al., 2015). The authors argue that the knowledge learned by a large "cumbersome" classifier on extensive data—e.g., a deep neural network or an ensemble—is expressed in its probability estimates over classes, and not just contained in its parameters. Hence, these teacher estimates for an input may be repurposed as soft labels to train a smaller student model. In practice, the loss against the true labels is linearly interpolated with a loss against the teacher probabilities, Eq. (4). We discuss alternative ways to integrate outside information in Section 6.

[4] This choice is because the reparameterization trick behind VAEs used to be limited to location-scale distributions, but recent developments (e.g., Figurnov et al., 2018) have lifted that restriction, as Burkhardt and Kramer (2019) demonstrate with several Dirichlet-based NTMs using VAEs.

2.3 Combining neural topic modeling with knowledge distillation

The knowledge distillation objective. To apply KD to a "base" neural topic model, we replace the reconstruction term $\mathcal{L}_R$ in Eq. (3) with $\mathcal{L}_{KD}$, as follows:

$$\begin{aligned} \boldsymbol{w}_d^{\text{BAT}} &= \sigma\left(\boldsymbol{z}_d^{\text{BAT}}/T\right) N_d \\ \hat{\boldsymbol{w}} &= f(\boldsymbol{\theta}_d, \mathbf{B}; T) \\ \mathcal{L}_{KD} &= \lambda T^2 \left(\boldsymbol{w}_d^{\text{BAT}}\right)^\top \log \hat{\boldsymbol{w}} + (1-\lambda)\mathcal{L}_R \end{aligned} \quad (4)$$

Here, $\boldsymbol{z}_d^{\text{BAT}}$ are the logits produced by the teacher network for a given input document d, meaning that $\boldsymbol{w}_d^{\text{BAT}}$ acts as a smoothed pseudo-document. T is the softmax temperature, which controls how diffuse the estimated probability mass is over the words (hence $f(\cdot; T)$ is Eq. (1) with the corresponding scaling). This differs from the original KD in two ways: (a) it scales the estimated probabilities by the document length N_d, and (b) it uses a multi-label loss.

The teacher model. We generate the teacher logits $\boldsymbol{z}^{\text{BAT}}$ using the pretrained transformer DISTILBERT (Sanh et al., 2019), itself a distilled version of BERT (Devlin et al., 2019).[5] BERT-like models are generally pretrained on large domain-general corpora with a language-modeling like objective, yielding an ability to capture nuances of linguistic context more effectively than bag-of-words models (Clark et al., 2019; Liu et al., 2019; Rogers et al., 2020). Mirroring the NTM's formulation as a variational auto-encoder, we treat DISTILBERT as a *deterministic* auto-encoder, fine-tuning it with the document-reconstruction objective $\mathcal{L}_R$ on the same dataset. Thus, we use a **B**ERT-based **A**utoencoder as our **T**eacher model, hence **BAT**.[6]

Clipping the logit distribution. Depending on preprocessing, V may number in the tens of thousands of words. This leads to a long tail of probability mass assigned to unlikely terms, and breaks standard assumptions of sparsity. Tang et al. (2020), working in a classification setting, find that truncating the logits to the top-n classes and assigning uniform mass to the rest improves accuracy. We instead choose the top $c\,N_d, c \in \mathbb{R}^+$ logits and assign *zero* probability to the remaining elements to enforce sparsity.

[5] DISTILBERT's light weight accommodates longer documents, necessary for topic modeling. Even with this change, we divide very long documents into chunks, estimating logits for each chunk and taking the pointwise mean. More complex schemes (i.e., LSTMs, Hochreiter and Schmidhuber, 1997) yielded no benefit.

[6] A reader familiar with variational NTMs may notice that we haven't mentioned an obvious means of incorporating representations from a pretrained transformer: encoding the document representation from a BERT-like model. This yields unimpressive results; see Appendix D.1.

	D	V	Avg N_d	Preprocessing details
20NG	18k	2k	87.1	Srivastava and Sutton (2017)
Wiki	28.5k	20k	1395.4	Nan et al. (2019)
IMDb	50k	5k	95.0	Card et al. (2018)

Table 1: Corpus statistics, which vary in total number of documents (D), vocabulary size (V), and average document length (N_d).

3 Experimental Setup

3.1 Data and Metrics

We validate our approach using three readily available datasets that vary widely in domain, corpus and vocabulary size, and document length: 20 Newsgroups (20NG, Lang, 1995),[7] Wikitext-103 (Wiki, Merity et al., 2017),[8] and IMDb movie reviews (IMDb, Maas et al., 2011).[9] These are commonly used in neural topic modeling, with preprocessed versions provided by various authors; see references in Table 1 for details. For consistency with prior work, we use a train/dev/test split of 48/12/40 for 20NG, 70/15/15 for Wiki, and 50/25/25 for IMDb.[10]

We seek to discover a latent space of topics that is meaningful and useful to people (Chang et al., 2009). Accordingly, we evaluate topic coherence using normalized mutual pointwise information (NPMI), which is significantly correlated with human judgments of topic quality (Aletras and Stevenson, 2013; Lau et al., 2014) and widely used to evaluate topic models.[11] We follow precedent and calculate (internal) NPMI using the top ten words in each topic, taking the mean across the NPMI scores for individual topics. Internal NPMI is estimated with reference co-occurrence counts from a held-out dataset from the same corpus,

[7] qwone.com/~jason/20Newsgroups

[8] s3.amazonaws.com/research.metamind.io/wikitext/wikitext-103-v1.zip

[9] ai.stanford.edu/~amaas/data/sentiment

[10] The splits are used to estimate NPMI. Dev splits are used to select hyperparameters, and test splits are run after hyperparameters are selected and frozen.

[11] We also obtain competitive results for document perplexity, which has also been used widely but correlates negatively with human coherence evaluations (Chang et al., 2009).

i.e., the dev or test split. While internal NPMI is the metric of choice for most prior work, we also provide external NPMI results using Gigaword 5 (Parker et al., 2011), following Card et al. (2018).

3.2 Experimental Baselines

We select three experimental baseline models that represent diverse styles of neural topic modeling.[12] Each achieves the highest NPMI on the majority of its respective datasets, as well as a considerable improvement over previous neural and non-neural topic models (such as Srivastava and Sutton, 2017; Miao et al., 2016; Ding et al., 2018). All our baselines are roughly contemporaneous with one another, and had yet to be compared in a head-to-head fashion prior to our work.

SCHOLAR. Card et al. (2018) use a VAE-based (Kingma and Welling, 2014) neural topic modeling setup (as introduced in Srivastava and Sutton, 2017) with a logistic normal prior to approximate the Dirichlet, and provide an elegant way to incorporate document metadata.

DVAE. Burkhardt and Kramer (2019) use a Dirichlet prior, where its reparameterization is enabled by rejection sampling variational inference. This allows it to tap into the same generative story as the original LDA formulations of Blei (2003), and to enjoy the advantageous properties of the Dirichlet like multi-modality (Wallach et al., 2009; Wang et al., 2020).

W-LDA. Nan et al. (2019) forego the VAE in favor of a Wasserstein auto-encoder (Tolstikhin et al., 2018), using a Dirichlet prior that is matched by minimizing Maximum Mean Discrepancy. They find the method leads to state-of-the-art coherence on several datasets and encourages topics to exhibit greater word diversity.

We demonstrate the modularity of our core innovation by combining our method with both SCHOLAR and W-LDA (Section 4).

3.3 Our Models and Settings

As discussed in Section 2.3, our approach relies on a "base" neural topic model and unnormalized probabilities over words estimated by a transformer as "teacher". We discuss each in turn.

Neural topic models augmented with knowledge distillation. We experiment with both SCHOLAR and W-LDA as base models. The former constitutes our primary model and point of comparison with baselines, while the latter is a proof-of-concept that attests to our method's modularity; we added knowledge distillation to W-LDA with only a few lines of code (Appendix F). We evaluate both at $K = 50$ and $K = 200$ topics.

We tune using NPMI, with reference co-occurrence counts taken from a held-out development set from the relevant corpus. For our baselines, we use the publicly-released author implementations.[13] While we generally attempt to retain the original hyperparameter settings when available, we do perform an exhaustive grid search on the SCHOLAR baselines and SCHOLAR+BAT to ensure fairness in comparison (ranges, optimal values, and other details in Appendix E.1).

Our method also introduces additional hyperparameters: the weight for KD loss, λ (Eq. (4)); the softmax temperature T; and the proportion of the word-level teacher logits that we retain (relative to document length, see clipping in Section 2.3). For most dataset-K pairs, we find that we can improve topic quality under most settings, with a relatively small set of values for each hyperparameter leading to better results. In fact, following the extensive search on SCHOLAR+BAT, we found we could tune W-LDA within a few iterations.

Topic models rely on random sampling procedures, and to ensure that our results are robust, we report the average values across five runs (previously unreported by the authors of our baselines).

The DISTILBERT teacher. We fine-tune a modified version of DISTILBERT with the same document reconstruction objective as the NTM ($\mathcal{L}_R$, Eq. (3)) on the training data. Specifically, DISTILBERT maps a WordPiece-tokenized (Wu et al., 2016) document d to an l-dimensional hidden vector with a transformer (Vaswani et al., 2017), then back to logits over V words (tokenized with the same scheme as the topic model). For long documents, we split into blocks of 512 tokens and mean-pool the transformer outputs. We use the pretrained model made available by the authors (Wolf

[12] This use of "baseline" should not be confused with the "base" neural topic model augmented by knowledge distillation (Section 2.3).

[13] SCHOLAR: github.com/dallascard/scholar
W-LDA: github.com/awslabs/w-lda
DVAE: github.com/sophieburkhardt/dirichlet-vae-topic-models
For augmented models we start with our own reimplementations of the baseline approaches in a common codebase, validated by obtaining comparable results to the original authors on their datasets.

	$K = 50$			$K = 200$		
	20NG	**Wiki**	**IMDb**	**20NG**	**Wiki**	**IMDb**
DVAE	0.340	0.490	0.145	0.316	0.450	0.160
W-LDA	0.279	0.494	0.136	0.188	0.308	0.095
SCHOLAR	0.322 (0.007)	0.494 (0.005)	0.168 (0.002)	0.263 (0.002)	0.473 (0.005)	0.140 (0.001)
SCH. + BAT	**0.354** (0.004)	**0.521** (0.009)	**0.182** (0.002)	**0.332** (0.002)	**0.513** (0.001)	**0.175** (0.003)

Table 2: The NPMI for our baselines (Section 3.2) compared with BAT (explained in Section 2.3) using SCHOLAR as our base neural architecture. We achieve better NPMI than all baselines across three datasets and $K = 50$, $K = 200$ topics. We use 5 random restarts and report the standard deviation.

et al., 2019). We train until perplexity converges on the same held-out dev set used in the topic modeling setting. Unsurprisingly, DISTILBERT achieves dramatically lower perplexity than all topic model baselines. Note that we need only train the model once per corpus, and can experiment with different NTM variations using the same $\boldsymbol{z}^{\text{BAT}}$.

4 Results and Discussion

Using the VAE-based SCHOLAR as the base model, topics discovered using BAT are more coherent, as measured via NPMI, than previous state-of-the-art baseline NTMs (Table 2), improving on the DVAE and W-LDA baselines, and the baseline of SCHOLAR without the KD augmentation. We establish the robustness of our approach's improvement by taking the mean across multiple runs with different random seeds, yielding consistent improvement over all baselines for all the datasets. We validate the approach using a smaller and larger number of topics, $K = 50$ and 200, respectively.

In addition to its improved performance, BAT can apply straightforwardly to other models, because it makes very few assumptions about the base model—requiring only that it rely on a word-level reconstruction objective, which is true of the majority of neural topic models proposed to date. We illustrate this by using the Wasserstein auto-encoder (W-LDA) as a base NTM, showing in Table 3 that BAT improves on the unaugmented model.[14]

We report the dev set results (corresponding to the test set results in Tables 2 and 3) in Appendix A—the same pattern of results is obtained, for all the models.

[14]We note that the W-LDA baseline did not tune well on 200 topics, further complicated by the model's extensive run time. As such, we focus on augmenting that model for 50 topics, consistent with the number of topics on which Nan et al. (2019) report their results. We add preliminary results using BAT with DVAE in Appendix C.

Finally, we also compute NPMI using reference counts from an external corpus (Gigaword 5, Parker et al., 2011) for SCHOLAR and SCHOLAR+BAT (Table 4). We find the same patterns generally hold: in all but one setting (`Wiki`, $K = 50$), BAT improves topic coherence relative to SCHOLAR. These external NPMI results suggest that our model avails itself of the distilled general language knowledge from pretrained BERT, and moreover that our fine-tuning procedure does not overfit to the training data.

	20NG	**Wiki**	**IMDb**
W-LDA	0.279 (0.010)	**0.494** (0.012)	0.136 (0.008)
+BAT	**0.299** (0.010)	**0.505** (0.014)	**0.162** (0.003)

Table 3: Mean NPMI (s.d.) across 5 runs for W-LDA (Nan et al., 2019) and W-LDA+BAT for $K = 50$, showing improvement on two of three datasets. This demonstrates that our method is *modular* and can be used with base neural topic models that vary significantly in architecture.

K		**SCHOLAR**	**+BAT**
50	`20ng`	0.147 (0.002)	**0.170** (0.006)
	`Wiki`	**0.193** (0.006)	**0.187** (0.004)
	`IMDb`	0.149 (0.003)	**0.161** (0.003)
200	`20ng`	0.111 (0.001)	**0.171** (0.002)
	`Wiki`	0.177 (0.003)	**0.190** (0.008)
	`IMDb`	0.122 (0.002)	**0.159** (0.003)

Table 4: External NPMI (s.d.) for the base SCHOLAR and SCHOLAR+BAT. Models selected according to performance on the development set using internal NPMI.

5 Impact of BAT on Individual Topics

Following standard practice, we have established that our models discover more coherent topics *on average* when compared to others (Tables 2 and 3).

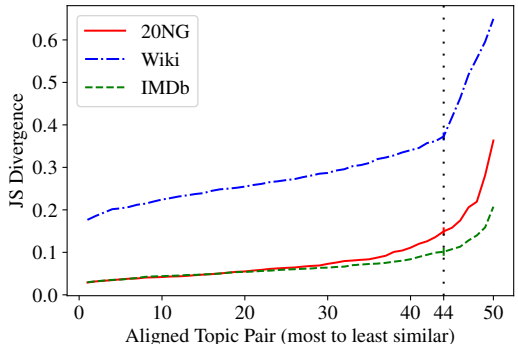

Figure 2: Jensen-Shannon divergence for aligned topic pairs in the SCHOLAR and SCHOLAR+BAT models.

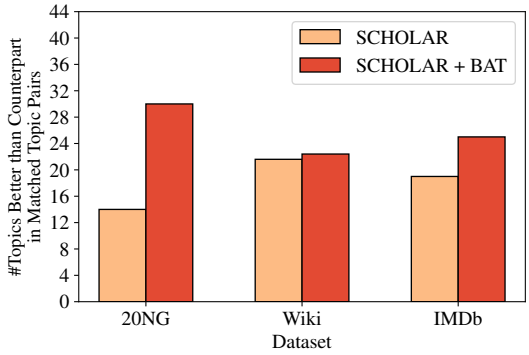

Figure 3: Number of matched topic pairs where SCHOLAR+BAT improves coherence, compared with the number of matched pairs where the baseline improves coherence.

Now, we look more closely at the extent to which those improvements are *meaningful* at the level of individual topics. To do so we directly compare topics discovered by the baseline neural topic model (SCHOLAR) with corresponding topics obtained when that model is augmented with BAT, looking at the NPMIs of the corresponding topics as well as considering them qualitatively.

We align the topics in the base and augmented SCHOLAR models using a variation of competitive linking, which produces a greedy approximation to optimal weighted bipartite graph matching (Melamed, 2000). A fully connected weighted bipartite graph is constructed by linking all topic pairs across (but not within) the two models, with the weight for a topic pair being the similarity between their word distributions as measured by Jenson-Shannon (JS) divergence (Wong and You, 1985; Lin, 1991). We pick the pair (t_i, t_j) with the lowest JS divergence and add it to the resulting alignment, then remove t_i and t_j from consideration and iterate until no pairs are left. The resulting aligned topic pairs can then be sorted by their JS divergences to directly compare corresponding topics.[15]

Fig. 2 shows the JS-divergences for aligned topic pairs, for our three corpora. Based on visual inspection, we choose the 44 most aligned topic pairs as being meaningful for comparison; beyond this point, the topics do not bear a conceptual relationship (using the same threshold for the three datasets for simplicity).

When we consider these conceptually related topic pairs, we see that the model augmented with BAT has the topic with the higher NPMI value more often across all three datasets (Fig. 3). This means that BAT is not just producing improvements in the aggregate (Section 4): its effect can be interpreted more specifically as identifying the same space of topics generated by an existing model and, in most cases, improving the coherence of individual topics. This highlights the modular value of our approach.

Table 5 provides qualitative discussion for one example from each corpus, which we have selected for illustration from a single randomly selected run of the baseline SCHOLAR and SCHOLAR+BAT models for $K = 50$. We find that, consistent with prior work on automatic evaluation of topic models, differences in NPMI do appear to correspond to recognizable subjective differences in topic quality. So that readers may form their own judgments, Appendix G presents 15 aligned pairs for each corpus, selected randomly by stratifying across levels of alignment quality to create a fair sample to review.

6 Related Work

Integrating embeddings into topic models. A key goal in our use of knowledge distillation is to incorporate relationships between words that may not be well supported by the topic model's input documents alone. Some previous topic models have sought to address this issue by incorporating external word information, including word senses (Ferrugento et al., 2016) and pretrained word embeddings (Hu and Tsujii, 2016; Yang et al., 2017; Xun et al., 2017; Ding et al., 2018). More recently, Bianchi et al. (2020) have incorporated BERT embeddings into the encoder to improve

[15] Note that more similar topics have lower JS-divergence, so we are seeking to minimize rather than maximize total weight. We use JS-divergence because it is conveniently symmetric and finite.

		NPMI	Topic
20ng	SCHOLAR	0.454	nhl hockey player coach ice playoff team league stanley european
	SCHOLAR+BAT	0.523	nhl hockey player team coach playoff cup wings stanley leafs
Wiki	SCHOLAR	0.547	jtwc jma typhoon monsoon luzon geophysical pagasa guam cyclone southwestward
	SCHOLAR+BAT	0.621	jtwc jma typhoon meteorological intensification monsoon dissipating shear outflow trough
IMDb	SCHOLAR	0.197	adaptation version novel bbc versions jane kenneth handsome adaptations faithful
	SCHOLAR+BAT	0.218	adaptation novel book read books faithful bbc version versions novels

Table 5: Selected examples of SCHOLAR+BAT improving on topics from SCHOLAR. We observe that the improved 20ng topic is more cleanly focused on the NHL (removing *european*, adding the Toronto Maple *Leafs*, evoking the Stanley *Cup* rather than the more generic *ice*); the improved wiki topic about typhoons is more clearly concentrated on meterological terms, rather than interspersing specific locations of typhoons (*luzon*, *guam*); and the improved IMDb topic more cleanly reflects what we would characterize as "video adaptations" by bringing in terms about that subject (*book*, *books*, *novels*, *read*) in place of predominant words relating to particular adaptations. Randomly selected examples can be found in Appendix G.

topic coherence. (See Appendix D.1 for our own related experiments, which yielded mixed results.) We refer the reader to Dieng et al. (2020) for an extensive and up-to-date overview.

A limitation of these approaches is that they simply import general, non-corpus-specific word-level information. In contrast, representations from a pretrained transformer can benefit from both general language knowledge and corpus-dependent information, by way of the pretraining and fine-tuning regime. By regularizing toward representations conditioned on the document, we remain coherent relative to the topic model data. An additional key advantage for our method is that it involves only a slight change to the underlying topic model, rather than the specialized designs by the above methods.

Knowledge distillation. While the focus was originally on single-label image classification, KD has also been extended to the multi-label setting (Liu et al., 2018b). In NLP, KD has usually been applied in supervised settings (Kim and Rush, 2016; Huang et al., 2018; Yang et al., 2020), but also in some unsupervised tasks (usually using an unsupervised teacher for a supervised student) (Hu et al., 2020; Sun et al., 2020). Xu et al. (2018) use word embeddings jointly learned with a topic model in a procedure they term distillation, but do not follow the method from Hinton et al. (2015) that we employ (instead opting for joint-learning). Recently, pretrained models like BERT have offered an attractive choice of teacher model, used successfully for a variety of tasks such as sentiment classification and paraphrasing (Tang et al., 2019a,b). Work in distillation often cites a reduction in computational cost as a goal (e.g., Sanh et al., 2019), although we are aware of at least one effort that is focused specifically on interpretability (Liu et al., 2018a).

Topic diversity. Coherence, commonly quantified automatically using NPMI, is the current standard for evaluating topic model quality. Recently several authors (Dieng et al., 2020; Burkhardt and Kramer, 2019; Nan et al., 2019) have proposed additional metrics focused on the diversity or uniqueness of topics (based on top words in topics). However, no one metric has yet achieved acceptance or consensus in the literature. Moreover, such measures fail to distinguish between the case where two topics share the same set of top n words, therefore coming across as essentially identical, versus when one topic's top n words are repeated individually across multiple other topics, indicating a weaker and more diffuse similarity to those topics. We discuss issues related to topic diversity in Appendix D.2.

7 Conclusions and Future Work

To our knowledge, we are the first to distill a "black-box" neural network teacher to guide a probabilistic graphical model. We do this in order to combine the expressivity of probabilistic topic models with the precision of pretrained transformers. Our modular method sits atop any neural topic model (NTM) to improve topic quality, which we demonstrate using two NTMs of highly disparate architectures (VAEs and WAEs), obtaining state-of-the-art topic coherence across three datasets from different domains. Our adaptable framework does not just produce improvements in the aggregate (as is commonly reported): its effect can be interpreted more specifically as identifying the same space of topics generated by an existing model and, in most cases,

improving the coherence of individual topics, thus highlighting the modular value of our approach.

In future work, we also hope to explore the effects of the pretraining corpus (Gururangan et al., 2020) and teachers (besides BERT) on the generated topics. Another intriguing direction is exploring the connection between our methods and neural network interpretability. The use of knowledge distillation to facilitate interpretability has also been previously explored, for example, in Liu et al. (2018a) to learn interpretable decision trees from neural networks. In our work, as the weight on the BERT autoencoder logits λ goes to one, the topic model begins to describe less the *corpus* and more the *teacher*. We believe mining this connection can open up further research avenues; for instance, by investigating the differences in such teacher-topics conditioned on the pre-training corpus. Finally, although we are motivated primarily by the widespread use of topic models for identifying interpretable topics (Boyd-Graber et al., 2017, Ch. 3), we plan to explore the ideas presented here further in the context of downstream applications like document classification.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grants 2031736 and 2008761 and by Amazon. We thank Pedro Rodriguez, Shi Feng, and our anonymous reviewers for their helpful comments. Appreciation to Adam Forbes for the design of Fig. 1. We also thank the authors of Card et al. (2018), Nan et al. (2019), and Burkhardt and Kramer (2019) for their publicly available implementations.

References

Nikolaos Aletras and Mark Stevenson. 2013. Evaluating topic coherence using distributional semantics. In *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013) – Long Papers*, pages 13–22, Potsdam, Germany. Association for Computational Linguistics.

Federico Bianchi, Silvia Terragni, and Dirk Hovy. 2020. Pre-training is a Hot Topic: Contextualized Document Embeddings Improve Topic Coherence. *arXiv:2004.03974 [cs]*.

David M Blei. 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, page 30.

Jordan Boyd-Graber, Yuening Hu, and David Mimno. 2017. *Applications of Topic Models*, volume 11 of *Foundations and Trends in Information Retrieval*. NOW Publishers.

Sophie Burkhardt and Stefan Kramer. 2019. Decoupling Sparsity and Smoothness in the Dirichlet Variational Autoencoder Topic Model. *Journal of Machine Learning Research*, 20(131):27.

Dallas Card, Chenhao Tan, and Noah A. Smith. 2018. Neural models for documents with metadata. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2031–2040, Melbourne, Australia. Association for Computational Linguistics.

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

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. Association for Computational Linguistics.

Jacob Devlin, 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, Minneapolis, Minnesota. Association for Computational Linguistics.

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

Ran Ding, Ramesh Nallapati, and Bing Xiang. 2018. Coherence-aware neural topic modeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 830–836, Brussels, Belgium. Association for Computational Linguistics.

Adriana Ferrugento, Hugo Gonçalo Oliveira, Ana Alves, and Filipe Rodrigues. 2016. Can topic modelling benefit from word sense information? In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 3387–3393, Portorož, Slovenia. European Language Resources Association (ELRA).

Mikhail Figurnov, S. Mohamed, and A. Mnih. 2018. Implicit reparameterization gradients. In *NeurIPS*.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 8342–8360, Online. Association for Computational Linguistics.

Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Hengtong Hu, Lingxi Xie, Richang Hong, and Qi Tian. 2020. Creating Something from Nothing: Unsupervised Knowledge Distillation for Cross-Modal Hashing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3123–3132.

Weihua Hu and Jun'ichi Tsujii. 2016. A latent concept topic model for robust topic inference using word embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 380–386.

Mingkun Huang, Yongbin You, Zhehuai Chen, Yanmin Qian, and Kai Yu. 2018. Knowledge Distillation for Sequence Model. In *Interspeech 2018*, pages 3703–3707. ISCA.

Martin Jankowiak and Fritz Obermeyer. 2018. Pathwise Derivatives Beyond the Reparameterization Trick. In *International Conference on Machine Learning*, pages 2235–2244.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, Austin, Texas. Association for Computational Linguistics.

Diederik P. Kingma and Max Welling. 2014. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Thomas K. Landauer and Susan T. Dumais. 1997. A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, 104(2):211–240.

Ken Lang. 1995. NewsWeeder: Learning to Filter Netnews. In Armand Prieditis and Stuart Russell, editors, *Machine Learning Proceedings 1995*, pages 331–339. Morgan Kaufmann, San Francisco (CA).

Jey Han Lau, David Newman, and Timothy Baldwin. 2014. Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 530–539, Gothenburg, Sweden. Association for Computational Linguistics.

Jianhua Lin. 1991. Divergence measures based on the shannon entropy. *IEEE Transactions on Information theory*, 37(1):145–151.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019. Linguistic knowledge and transferability of contextual representations. 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 1073–1094, Minneapolis, Minnesota. Association for Computational Linguistics.

Xuan Liu, Xiaoguang Wang, and Stan Matwin. 2018a. Improving the Interpretability of Deep Neural Networks with Knowledge Distillation. In *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 905–912. IEEE.

Yongcheng Liu, Lu Sheng, Jing Shao, Junjie Yan, Shiming Xiang, and Chunhong Pan. 2018b. Multi-Label Image Classification via Knowledge Distillation from Weakly-Supervised Detection. *2018 ACM Multimedia Conference on Multimedia Conference - MM '18*, pages 700–708.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

I Dan Melamed. 2000. Models of translational equivalence among words. *Computational Linguistics*, 26(2):221–249.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. Pointer Sentinel Mixture Models. *ICLR*.

Yishu Miao, Lei Yu, and Phil Blunsom. 2016. Neural Variational Inference for Text Processing. *ICML*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.

Feng Nan, Ran Ding, Ramesh Nallapati, and Bing Xiang. 2019. Topic modeling with Wasserstein autoencoders. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6345–6381, Florence, Italy. Association for Computational Linguistics.

Viet-An Nguyen, Jordan L Ying, and Philip Resnik. 2013. Lexical and hierarchical topic regression. In *Advances in neural information processing systems*, pages 1106–1114.

Robert Parker, David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2011. English Gigaword Fifth Edition.

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. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A Primer in BERTology: What we know about how BERT works. *arXiv:2002.12327 [cs]*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. *NeurIPS EMC$\hat{2}$ Workshop*.

Akash Srivastava and Charles Sutton. 2017. Autoencoding variational inference for topic models. In *ICLR*.

Haipeng Sun, Rui Wang, Kehai Chen, Masao Utiyama, Eiichiro Sumita, and Tiejun Zhao. 2020. Knowledge Distillation for Multilingual Unsupervised Neural Machine Translation. *arXiv:2004.10171 [cs]*.

Jiaxi Tang, Rakesh Shivanna, Zhe Zhao, Dong Lin, Anima Singh, Ed H. Chi, and Sagar Jain. 2020. Understanding and Improving Knowledge Distillation. *arXiv:2002.03532 [cs, stat]*.

Raphael Tang, Yao Lu, and Jimmy Lin. 2019a. Natural Language Generation for Effective Knowledge Distillation. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pages 202–208, Hong Kong, China. Association for Computational Linguistics.

Raphael Tang, Yao Lu, Linqing Liu, Lili Mou, Olga Vechtomova, and Jimmy Lin. 2019b. Distilling Task-Specific Knowledge from BERT into Simple Neural Networks. *arXiv:1903.12136 [cs]*.

Ilya Tolstikhin, Sylvain Gelly, Olivier Bousquet, and Bernhard Scholkopf. 2018. Wasserstein Auto-Encoders. *ICLR*, page 16.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Hanna M. Wallach, David M. Mimno, and Andrew McCallum. 2009. Rethinking LDA: Why Priors Matter. In Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 1973–1981. Curran Associates, Inc.

Rui Wang, Xuemeng Hu, Deyu Zhou, Yulan He, Yuxuan Xiong, Chenchen Ye, and Haiyang Xu. 2020. Neural Topic Modeling with Bidirectional Adversarial Training. *ACL*, page 11.

Rui Wang, Deyu Zhou, and Yulan He. 2019. ATM:Adversarial-neural Topic Model. *Information Processing & Management*, 56(6):102098.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. HuggingFace's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Andrew KC Wong and Manlai You. 1985. Entropy and distance of random graphs with application to structural pattern recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (5):599–609.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Ł ukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. *arXiv:1609.08144 [cs]*.

Hongteng Xu, Wenlin Wang, Wei Liu, and Lawrence Carin. 2018. Distilled Wasserstein Learning for Word Embedding and Topic Modeling. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 1716–1725. Curran Associates, Inc.

Guangxu Xun, Yaliang Li, Wayne Xin Zhao, Jing Gao, and Aidong Zhang. 2017. A correlated topic model using word embeddings. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, IJCAI'17, pages 4207–4213, Melbourne, Australia. AAAI Press.

Weiwei Yang, Jordan Boyd-Graber, and Philip Resnik. 2017. Adapting topic models using lexical associations with tree priors. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1901–1906.

Ziqing Yang, Yiming Cui, Zhipeng Chen, Wanxiang Che, Ting Liu, Shijin Wang, and Guoping Hu. 2020. TextBrewer: An Open-Source Knowledge Distillation Toolkit for Natural Language Processing. *ACL Demo Session*.

	$K = 50$			$K = 200$		
	20NG	**Wiki**	**IMDb**	**20NG**	**Wiki**	**IMDb**
DVAE	0.341	0.512	0.137	0.312	0.470	0.155
W-LDA	0.294	0.500	0.136	0.203	0.310	0.095
SCHOLAR	0.343 (0.003)	0.504 (0.007)	0.167 (0.002)	0.279 (0.002)	0.478 (0.005)	0.139 (0.002)
SCH. + BAT	0.377 (0.006)	0.526 (0.009)	0.180 (0.002)	0.343 (0.002)	0.518 (0.001)	0.174 (0.003)

Table 6: The development-set NPMI for our baselines (Section 3.2) compared with BAT (explained in Section 2.3) using SCHOLAR as our base neural architecture. We achieve better NPMI than all baselines across three datasets and $K = 50, K = 200$ topics. We use 5 random restarts report the standard deviation.

Appendix

A Dev Set Results

We optimized our models on the dev set, froze the optimal models, and showed the results on the test set in Tables 2 and 3. We show the corresponding dev set results for those models in Tables 6 and 7.

	20NG	**Wiki**	**IMDb**
W-LDA	0.294 (0.014)	0.500 (0.013)	0.136 (0.009)
+BAT	**0.316** (0.010)	**0.511** (0.016)	**0.162** (0.003)

Table 7: The mean development-set NPMI (std. dev.) across 5 runs for W-LDA and W-LDA+BAT for $K = 50$, showing improvement on all datasets. This demonstrates that our innovation is *modular* and can be used with base neural topic models that vary in architecture.

B Extrinsic Classification Results

The primary goal of our method is to improve the coherence of generated topics. It is natural, however, to ask about the impact of our method on downstream applications. We include here a preliminary exploration suggesting that the addition of BAT does not hurt performance in document classification.

In our setup, we seek to predict document labels y_d from the MAP estimate of a document's topic distribution, $\boldsymbol{\theta}_d$. Specifically, we classify the newsgroup to which a document was posted for the 20 newsgroups data (e.g., `talk.politics.misc`) and a binary sentiment label for the IMDb review data. We train a random forest classifier using default parameters from `scikit-learn` (Pedregosa et al., 2011) and report the accuracies in Table 8 (averaged across 5 runs).

Much like other work that is aimed at topic coherence rather than their downstream use in supervised models (Nan et al., 2019), we find that our method has little impact on predictive performance. While it is possible that improvements may be obtained by specifically tuning models for classification, or by integrating BAT into model variations that combine lexical and topic representations (e.g. Nguyen et al., 2013), we leave this to future work.

	K	**SCHOLAR**	**+BAT**
`20ng`	50	0.676 (0.003)	0.669 (0.005)
	200	0.683 (0.002)	0.679 (0.004)
`IMDb`	50	0.829 (0.003)	0.823 (0.011)
	200	0.805 (0.003)	0.814 (0.004)

Table 8: Random forest classification accuracy on `20ng` and `IMDb` datasets, using topic estimates from SCHOLAR and SCHOLAR + BAT .

C Using BAT with DVAE

We further illustrate our method's modularity by applying BAT to our own reimplementation of DVAE (Burkhardt and Kramer, 2019).[16] In contrast to the author's primary implementation, which estimates the model with rejection sampling variational inference (used in Section 4), we reimplemented DVAE, approximating the Dirichlet gradient via pathwise derivatives (Jankowiak and Obermeyer, 2018), similar to Burkhardt and Kramer (2019)'s alternative model variant using implicit gradients.

Our reimplementation shows baseline behavior substantially similar to the author's implementation. In the course of our experimentation, we noted a degeneracy in this model, in which high NPMI is achieved but at the cost of redundant topics. This failure mode is well-established, but as discussed in Appendix D.2, we find the measures proposed to diagnose topic diversity (including those proposed by Burkhardt and Kramer, 2019; Nan et al., 2019) to be problematic. Rather than use these metrics,

[16] We appreciate a reviewer's suggestion that we add a +BAT comparison for DVAE.

therefore, we took a coarse but simple approach and filtered out any models that yielded more than one pair of identical topics, averaged across five runs (defined as having two topics with the same set of top-10 words). This filtering eliminated many hyperparameter settings, leading us to believe that DVAE is not robust to this problem.

Ultimately, we find that applying BAT to DVAE does not hurt, and also does not help appreciably (Table 9). In addition, when applying the above filtering criterion to our main SCHOLAR and SCHOLAR + BAT models, we still obtain the positive results reported in Table 6. [17]

	20NG	Wiki	IMDb
DVAE	0.376 (0.004)	**0.517** (0.006)	**0.169** (0.007)
+BAT	**0.401** (0.005)	**0.515** (0.007)	**0.169** (0.006)

Table 9: Mean development set NPMI (s.d.) across 5 runs for DVAE (Burkhardt and Kramer, 2019) and DVAE+BAT for $K = 50$.

D Methodological Notes

D.1 Using BERT in the encoder

In SCHOLAR, the encoder takes the following form:

$$\boldsymbol{\pi}_d = g\left(\left[\boldsymbol{W}\boldsymbol{w}_d^{\text{BOW}}\right]\right) \quad (5)$$

$$\boldsymbol{\theta}_d \sim \mathcal{LN}\left(\mu_\nu(\boldsymbol{\pi}_d),\, \sigma_\nu(\boldsymbol{\pi}_d)\right) \quad (6)$$

Where the weight matrix $\boldsymbol{W}$, along with the parameters of nueral networks $\mu(\cdot)$ and $\sigma(\cdot)$, are our variational parameters.

Card et al. (2018) propose that pre-trained word2vec (Mikolov et al., 2013) embeddings can replace $\boldsymbol{W}$, meaning that the document representation made available to the encoder is an l-dimensional sum of word embeddings. Card et al. (2018) argue that fixed embeddings act as an inductive prior which improves topic coherence. Likewise, we might want to encode the document representation from a BERT-like model and, in fact, this has been attempted with some success (Bianchi et al., 2020). The hypothesis is that a structure-dependent representation of the document can better parameterize its corresponding topic distribution.

[17] For $K = 50$. The single-pair threshold proves too restrictive for the $K = 200$ case, where no hyperparameter settings pass the threshold. Increasing the tolerance to a maximum of 5 redundant pairs with $K = 200$ leads to a somewhat lower average NPMI overall, but the same directional improvement, i.e. SCHOLAR+BAT yields a significantly higher NPMI than SCHOLAR.

Setting	NPMI
Randomly updated embeds.	0.170 (0.007)
Fixed word2vec embeds.	0.172 (0.004)
Random 784-dim doc. rep. + w2v	0.175 (0.007)
Mean-pooled 784-dim BERT output + w2v	0.172 (0.002)
Random 5000-dim doc. rep. + w2v	0.178 (0.007)
5000-dim predicted probs. from BAT + w2v	0.180 (0.008)

Table 10: Effect on topic coherence of passing various document representations to the SCHOLAR encoder (using the IMDb data). Each setting describes the document representation provided to the encoder, which is transformed by one feed-forward layer of 300-dimensions followed by a second down to K dimensions. "+ w2v" indicates that we first concatenated with the sum of the 300-dimensional word2vec embeddings for the document. Note that these early findings are based on a different IMDb development set, a 20% split from the training data. They are thus not directly comparable to the results reported elsewhere in the text, which used a separate held-out development set.

We experimented with this method as well, using both the hidden BERT representation and the predicted probabilities, although we also include a fixed randomized baseline to maintain parameter parity. Results for IMDb are reported in Table 10, and we find at best a mild improvement over the baselines.[18] We suspect the reason for this tepid result is both that (a) in training, the effect of estimated local document-topic proportions on the global topic-word distributions is diffuse and indirect; and (b) the compression of the representation into k dimensions causes too much of the high-level linguistic information to be lost. Nonetheless, owing to the slight benefit, we do pass the logits to the encoder in our SCHOLAR-based model. We avoid this change for the model based on W-LDA to underscore the modularity of our method.

D.2 Topic Diversity

Burkhardt and Kramer (2019) have found a degeneracy in some topic models, wherein a single topic will be repeated more than once with slightly varying terms (e.g., several Dadaism topics). Burkhardt and Kramer (2019) and others (Nan et al., 2019; Dieng et al., 2020) have independently proposed related metrics to quantify the problem, but the literature has not converged on a solution. In contrast to NPMI, we are not aware of any work that as-

[18] We also fail to reproduce the findings of Card et al. (2018), showing no meaningful improvement in topic coherence with fixed word2vec embeddings. It appears that this is a consequence of their tuning for perplexity rather than NPMI.

sesses the validity of such metrics with respect to human judgements.

Moreover, all these proposals suffer from a common problem: because they are global measures of word overlap, they fail to account for *how* words are repeated across topics. For instance, Topic Uniqueness (Nan et al., 2019) is identical regardless of whether all of a topic's top words are all repeated in a single second topic, or individual top words from that topic are repeated in several other topics. In addition, the measures inappropriately penalize partially-related topics.

They also penalize polysemy—and, more generally, the contextual flexibility of word meanings. One of the key *advantages* of latent topics, compared to surface lexical summaries, is that the same word can contribute differently to an understanding of what different topics are about. As a real example from our experience, in modeling a set of documents related to paid family and medical leave, words like *parent*, *mother*, and *father* are prominent in one topic related to parental leave when a child is born (accompanying other terms like *newborn* and *maternity_leave*) and also in another topic related to taking leave to care for family members, including elderly parents (accompanying other terms like *elderly* and *aging*). The fact that topic models permit a word like *parent* to be prominent in both of these clearly distinct topics, emphasizing two different aspects of the word relative to the collection as a whole (being a parent taking care of children, being a child taking care of parents), is a feature, not a bug. We consider the question of topic diversity an important direction for future work.

E Experimental Procedures

In this section, we first provide details of our hyperparameters and tuning procedures, then turn to our computing infrastructure and the rough runtime of the SCHOLAR model.

E.1 Hyperparameter Tuning and Optimal Values

We used well-tuned baselines to establish thresholds for performance on NPMI (following the reported hyperparameters in Card et al., 2018; Burkhardt and Kramer, 2019; Nan et al., 2019). While developing our model, we performed a coarse-grained initial hyperparameter sweep to identify ranges that were not beating the threshold, and decided to exclude those ranges when performing a full grid search. We report the hyperparameter ranges used in this search, along with their optimal values (as determined by development set NPMI), in Tables 11 to 15. These produced the final set of results (Tables 2, 3, 6 and 7).

For the DISTILBERT training, we use the default hyperparameter settings for the `bert-base-uncased` model (Wolf et al., 2019). Our code is a modified version of the MMIMDB multimodal sequence classification code from the same codebase as DISTILBERT (`https://github.com/huggingface/transformers/tree/master/examples/contrib/mm-imdb`), and we use all default hyperparameter settings specified there. We train for 7500 steps for `20ng`, and 17000 steps for `Wiki` and `IMDb` (this corresponds to convergence on development-set perplexity).

E.2 Computing Infrastructure and Runtime

For the full hyperparameter sweep, we used an Amazon Web Services ParallelCluster `https://github.com/aws/aws-parallelcluster` with 40 nodes of `g4dn.xlarge` instances (consisting of Nvidia T4 GPUs with 16 GB RAM), which ran for about 5 days. For initial experimentation, we used a SLURM cluster with a mix of consumer-grade Nvidia GPUs (e.g., 1080, 2080).

In terms of runtime, SCHOLAR) and our own SCHOLAR+BAT are equal and this is true for any of our baseline model augmented with BAT. It is important to note that the overhead in terms of the overall runtime comes only from training the DISTILBERT encoder on the full dataset first and inference time for obtaining the logits after training. Thus, users should keep in mind the initial step of training and inferring teacher model logits and saving them; once that is done for the dataset, our model does not add to the runtime. We show the comparison between the full runtimes, including the initial step, in Fig. 4.

F Changes to W-LDA

In Fig. 5, we show the changes to the W-LDA model necessary to accommodate our method. Ignoring the code to load & clip the logits, also constituting a minor change, we introduce about a dozen lines.

Dataset: 20NG		k = 50		k = 200	
	Values Tried	**SCHOLAR (optimal values)**	**SCHOLAR+BAT (optimal values)**	**SCHOLAR (optimal values)**	**SCHOLAR+BAT (optimal values)**
lr	0.002*	0.002	0.002	0.002	0.002
α	1.0*	1.0	1.0	1.0	1.0
λ	$\{0.25, 0.5, 0.75, 0.95, 0.99, 0.999\}$	-	0.75	-	0.99
T	$\{1.0, 2.0, 3.0, 5.0\}$	-	2.0	-	5.0

Table 11: Hyperparameter ranges and optimal values (as determined by development set NPMI) for SCHOLAR and SCHOLAR+BAT , on the **20NG** dataset. **lr** is the learning rate, α is the hyperparameter for the logistic normal prior, λ is the weight on the teacher model logits from Eq. (4), and T is the softmax temperature from Eq. (4). Other hyperparamter values (which can be accessed in our code base) which were kept at their default values are not reported here. Values marked with the * are also kept at their default values per the base SCHOLAR model (https://github.com/dallascard/scholar). All different sweeps in the grid search were run for **500 epochs** with a **batch size = 200**.

Dataset: Wiki		k = 50		k = 200	
	Values Tried	**SCHOLAR (optimal values)**	**SCHOLAR+BAT (optimal values)**	**SCHOLAR (optimal values)**	**SCHOLAR+BAT (optimal values)**
lr	$\{0.001, 0.002, 0.005\}$	0.001	0.001	0.002	0.005
α	$\{0.0005, 0.00075, 0.001, 0.005, 0.01\}$	0.01	0.00075	0.0005	0.001
anneal	$\{0.25, 0.5, 0.75\}$	0.25	0.5	0.25	0.5
λ	$\{0.4, 0.5, 0.6, 0.7, 0.75, 0.8\}$	-	0.75	-	0.75
T	$\{1.0, 2.0\}$	-	1.0	-	1.0
clipping	$\{1.0, 1.5, 2.0\}$	-	2.0	-	1.5

Table 12: Hyperparameter ranges and optimal values (as determined by development set NPMI) for SCHOLAR and SCHOLAR+BAT , on the **Wiki** dataset. **lr** is the learning rate, α is the hyperparameter for the logistic normal prior, **anneal** controls the annealing (as explained in Appendix B in Card et al. (2018)), λ is the weight on the teacher model logits from Eq. (4), T is the softmax temperature from Eq. (4), and **clipping** controls how much of the logit distribution to clip (Section 2.3). Other hyperparamter values (which can be accessed in our code base) which were kept at their default values are not reported here. All different sweeps in the grid search were run for **500 epochs** with a **batch size = 500**.

Dataset: IMDb		k = 50		k = 200	
	Values Tried	**SCHOLAR (optimal values)**	**SCHOLAR+BAT (optimal values)**	**SCHOLAR (optimal values)**	**SCHOLAR+BAT (optimal values)**
lr	0.002*	0.002	0.002	0.002	0.002
α	$\{0.01, 0.1, 0.5, 1.0\}$	0.5	0.5	0.1	0.1
anneal	$\{0.25, 0.5, 0.75\}$	0.25	0.25	0.25	0.5
λ	$\{0.25, 0.5, 0.75, 0.99\}$	-	0.5	-	0.99
T	$\{1.0, 2.0\}$	-	1.0	-	1.0
clipping	$\{0.0, 1.0, 10.0\}$	-	10.0	-	0.0

Table 13: Hyperparameter ranges and optimal values (as determined by development set NPMI) for SCHOLAR and SCHOLAR+BAT , on the **IMDb** dataset. **lr** is the learning rate, α is the hyperparameter for the logistic normal prior, **anneal** controls the annealing (as explained in Appendix B in Card et al. (2018)), λ is the weight on the teacher model logits from Eq. (4), T is the softmax temperature from Eq. (4), and **clipping** controls how much of the logit distribution to clip (Section 2.3). Other hyperparamter values (which can be accessed in our code base) which were kept at their default values are not reported here. Values marked with the * are also kept at their default values per the base SCHOLAR model (https://github.com/dallascard/scholar). All different sweeps in the grid search were run for **500 epochs** with a **batch size = 200**.

	Values Tried	W-LDA (optimal values)	W-LDA+BAT (optimal values)
	(Dataset: 20NG)		
lr	$\{0.002\}$	0.002	0.002
$\boldsymbol{\alpha}$	$\{0.1, 1.0\}$	0.1	0.1
$\boldsymbol{\lambda}$	$\{0.75, 0.99\}$	-	0.75
$\boldsymbol{T}$	$\{1.0, 2.0\}$	-	1.0
	(Dataset: Wiki)		
lr	$\{0.001\}$	0.001	0.001
$\boldsymbol{\alpha}$	$\{0.01, 0.1\}$	0.1	0.1
$\boldsymbol{\lambda}$	$\{0.25, 0.75\}$	-	0.25
$\boldsymbol{T}$	$\{1.0, 2.0, 5.0\}$	-	2.0
clipping	$\{1.0, 2.0\}$	-	1.0
	(Dataset: IMDb)		
lr	$\{0.002\}$	0.002	0.002
$\boldsymbol{\alpha}$	$\{0.1\}$	0.1	0.1
$\boldsymbol{\lambda}$	$\{0.75\}$	-	0.75
$\boldsymbol{T}$	$\{1.0\}$	-	1.0

Table 14: Hyperparameter ranges and optimal values (as determined by development set NPMI) for W-LDA and W-LDA+BAT , on all three datasets. **lr** is the learning rate, $\boldsymbol{\alpha}$ is the hyperparameter for the dirichlet prior, $\boldsymbol{\lambda}$ is the weight on the teacher model logits from Eq. (4), $\boldsymbol{T}$ is the softmax temperature from Eq. (4), and **clipping** controls how much of the logit distribution to clip (Section 2.3). Other hyperparameter values (which can be accessed in our codebase) which were kept at their default values in the original baseline code are not reported here (also see Nan et al. (2019) and `https://github.com/awslabs/w-lda/`). Values marked with the * are also kept at their default values. All different sweeps in the grid search were run for **500 epochs** and noise parameter = 0.5 (see Nan et al. (2019)). For 20NG and IMDb, we used **batch size = 200**, and for Wiki, we used **batch size = 360**.

	k = 50			k = 200		
	20NG	**Wiki**	**IMDb**	**20NG**	**Wiki**	**IMDb**
Optimal Dirichlet Prior	0.6		0.2	0.6		0.2

Table 15: For DVAE, we tried four values for the Dirichlet Prior (as per the values tried by the authors in Burkhardt and Kramer (2019)) - $\{0.01, 0.1, 0.2, 0.6\}$ and report the optimal values corresponding to the dev set results (Table 2) and test set results (Table 6) in this table. Within the model variations available in the codebase for DVAE (`https://github.com/sophieburkhardt/dirichlet-vae-topic-models`) we use the Dirichlet VAE based on RSVI which is shown to give the highest NPMI scores in Burkhardt and Kramer (2019).

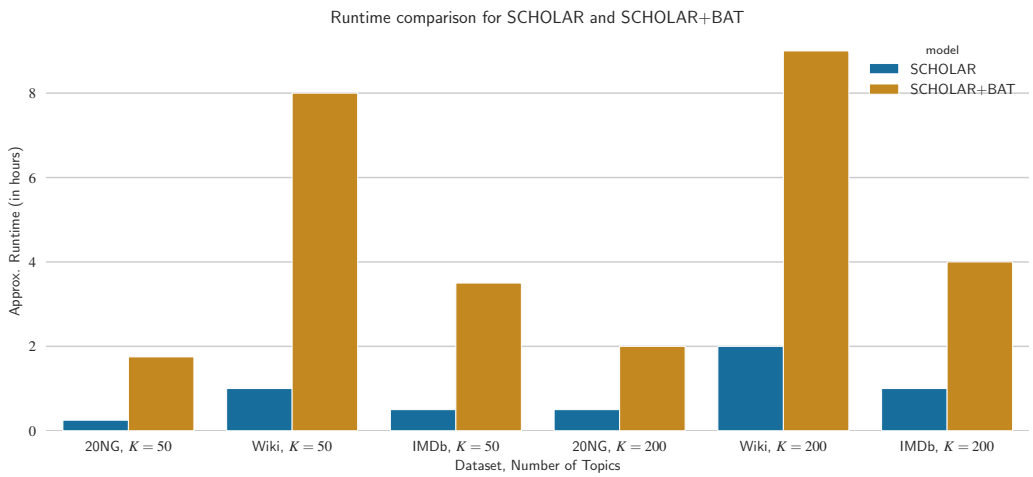

Figure 4: Runtime comparison for SCHOLAR) and our own SCHOLAR+BAT - Note that the overhead due to BAT is only a due to the training and inference time required to obtain the DISTILBERT encoder logits on the full dataset first, and once the teacher logits are available, the run time of both models is the same. We depict the full approximate time (in hours) including this initial overhead in case of BAT .

G Impact of BAT on Individual Topics: Aligned Topic Pair Examples

For each corpus (20NG, Wiki, and IMDb), a single comparison of base and BAT-augmented (SCHOLAR vs. SCHOLAR+BAT) 50-topic models was selected randomly, from the five runs used in computing average performance in Fig. 3.

For each of those pairs of models, we then randomly selected 15 aligned topic pairs from that set of 50 to include in the tables below. Specifically, a full set of 50 topic pairs was partitioned according to JS divergence into the 10 most similar pairs, the next 10 most similar, and so forth, for a total of five "brackets" of topic alignment quality. Three topic pairs were then selected at random from each bracket, hence 15 pairs in all, in order to yield a fair picture of what pairs look like at various qualities of topic alignment.

In the tables below (Tables 16 to 18), we present pairs sorted from best to worst alignment quality. Recall that for NPMI, higher is better, and for JS divergence, lower score indicates a higher quality match (or alignment) for the topic pair.

```
### In `compute_op.py`

## Retrieve BERT logits
docs = self.data.get_documents(key='train')
if self.args['use_kd']:
    split_on = docs.shape[1] // 2
    docs, bert_logits = docs[:,:split_on], docs[:,split_on:]
    t = self.args['kd_softmax_temp']
    kd_docs = nd.softmax(bert_logits / t) * nd.sum(docs, axis=1, keepdims=True)

# [... unchanged lines ...]

## Compute loss
with autograd.record():
    # [... unchanged lines ...]
    if self.args['use_kd']:
        kd_logits = nd.log_softmax(x_reconstruction_u / t)
        logits = nd.log_softmax(x_reconstruction_u)

        kd_loss_reconstruction = nd.mean(nd.sum(- kd_docs * kd_logits, axis=1))
        loss_reconstruction = nd.mean(nd.sum(- docs * logits, axis=1))

        loss_total = self.args['recon_alpha'] * (
            self.args['kd_loss_alpha'] * t * t * (kd_loss_reconstruction) +
            (1 - self.args['kd_loss_alpha']) * loss_reconstruction
        )
    else:
        # [... unchanged lines ...]
```

Figure 5: Modified portions of W-LDA model to accommodate BAT. We omit definitions of additional command-line arguments and data loading, but they are similarly brief.

Pair #	SCHOLAR vs SCHOLAR+BAT (NPMI, Top 10 Topic Words)	JS Divergence
1	SCHOLAR: (0.399, 'sin eternal lord heaven pray christ prayer jesus god hell') SCHOLAR+BAT: (0.394, 'eternal god hell sin heaven christ jesus christianity faith life')	0.0287
4	SCHOLAR: (0.3512, 'score goal puck penalty season shot tie pitch game defensive') SCHOLAR+BAT: (0.3838, 'score goal season game puck shot leafs penalty play playoff')	0.0345
8	SCHOLAR: (0.4307, 'doctrine church catholic scripture spirit biblical revelation bible resurrection christ') SCHOLAR+BAT: (0.4454, 'biblical church bible scripture doctrine catholic interpretation passage teaching jesus')	0.0417
11	SCHOLAR: (0.7109, 'turks armenian genocide jews mountain armenians turkish proceed nazi armenia') SCHOLAR+BAT: (0.7297, 'turks genocide turkish armenian armenia armenians massacre turkey proceed muslim')	0.0425
15	SCHOLAR: (0.2626, 'cryptography security network privacy mailing internet mail encrypt anonymous user') SCHOLAR+BAT: (0.289, 'anonymous mail network privacy internet security cryptography encrypt electronic ftp')	0.0479
17	SCHOLAR: (0.3501, 'rider bike ride helmet motorcycle dog bmw dod honda seat') SCHOLAR+BAT: (0.3843, 'helmet bike rider ride dog motorcycle dod rear honda bmw')	0.0498
22	SCHOLAR: (0.307, 'voltage circuit amp heat battery electronics frequency signal audio ac') SCHOLAR+BAT: (0.3236, 'circuit voltage amp wire audio wiring signal outlet input pin')	0.0641
27	SCHOLAR: (0.4018, 'passage verse jesus biblical resurrection scripture translation interpretation bible prophet') SCHOLAR+BAT: (0.5262, 'jesus christ lord sin heaven resurrection holy mary father son')	0.071
28	SCHOLAR: (0.2469, 'nt printer windows microsoft mac unix postscript pc os print') SCHOLAR+BAT: (0.2786, 'font color image format printer display pixel graphic postscript directory')	0.0729
31	SCHOLAR: (0.2109, 'crash backup gateway disk windows install memory boot floppy cache') SCHOLAR+BAT: (0.3141, 'disk floppy dos scsi ram cache controller isa swap windows')	0.0864
35	SCHOLAR: (0.2589, 'scientific science disease medicine treatment energy observe observation patient scientist') SCHOLAR+BAT: (0.2705, 'science morality objective scientific moral existence observation universe definition theory')	0.093
40	SCHOLAR: (0.1252, 'interested kit sale advance email address australia thanks april mail') SCHOLAR+BAT: (0.206, 'mail email mailing address list thanks interested fax please send')	0.1173
41	SCHOLAR: (0.2842, 'insurance tax hospital coverage health pay canadian kid care economy') SCHOLAR+BAT: (0.2354, 'dealer car price insurance buy pay sell money honda ford')	0.1319
45	SCHOLAR: (0.3165, 'waco clinton president bush senate batf tax fbi compound vote') SCHOLAR+BAT: (0.5144, 'nsa crypto clipper escrow wiretap secure encryption chip warrant scheme')	0.1791
48	SCHOLAR: (0.2329, 'screen mouse monitor printer inch resolution tube apple font print') SCHOLAR+BAT: (0.275, 'heat fuel tube cool detector radar gas nuclear hole cold')	0.2527

Table 16: Fifteen aligned topic pairs from the 20NG dataset.

Pair #	SCHOLAR vs SCHOLAR+BAT (NPMI, Top 10 Topic Words)	JS Divergence
1	SCHOLAR: (0.5804, 'prognosis protein symptom intravenous diagnosis syndrome medication abnormality infection dysfunction') SCHOLAR+BAT: (0.5464, 'abnormality prognosis intravenous receptor syndrome antibiotic inflammation diagnosis mutation dos')	0.163
4	SCHOLAR: (0.6036, 'parsec brightest orbiting astronomer planetary brightness luminosity jupiter constellation orbit') SCHOLAR+BAT: (0.586, 'orbiting habitable gliese planetary extrasolar parsec brightness luminosity orbital jupiter')	0.1787
8	SCHOLAR: (0.4432, 'lap peloton uci breakaway sprint ferrari bmc tyre podium sauber') SCHOLAR+BAT: (0.4902, 'lap sprint podium finisher quickest uci mclaren ferrari peloton rosberg')	0.1879
11	SCHOLAR: (0.5662, 'ny renumbering cr realigned intersects intersecting hamlet concurrency routing truncated') SCHOLAR+BAT: (0.5888, 'ny intersects renumbering intersecting realigned cr concurrency routing intersection hamlet')	0.1989
15	SCHOLAR: (0.4866, 'byzantine caliphate ibn caliph byzantium abbasid thrace constantinople vassal umayyad') SCHOLAR+BAT: (0.4686, 'byzantium thrace caliphate nikephoros antioch byzantine envoy umayyad principality constantinople')	0.2076
17	SCHOLAR: (0.4944, 'gubernatorial kentucky reelection republican democrat frankfort candidacy legislator congressman caucus') SCHOLAR+BAT: (0.494, 'gubernatorial reelection legislator congressman candidacy caucus whig democrat kentucky veto')	0.2211
22	SCHOLAR: (0.4069, 'electrification electrified locomotive train nok railway freight oslo commuter nsb') SCHOLAR+BAT: (0.3567, 'nok electrified electrification commuter oslo tramway freight livery bergen locomotive')	0.2391
27	SCHOLAR: (0.4187, 'gatehouse chancel nave stonework anglesey castle demography domesday storey vaulted') SCHOLAR+BAT: (0.4035, 'domesday demography cheshire gatehouse storey borough manor priory mersey avon')	0.2564
28	SCHOLAR: (0.4041, 'frigate brig convoy hm torpedoed rigging destroyer sailed sighted starboard') SCHOLAR+BAT: (0.4126, 'brig frigate privateer rigging schooner sloop corvette sighted indiaman brest')	0.2617
31	SCHOLAR: (0.2876, 'raaf battalion aircrew beachhead moresby amberley brigade usaaf dso jagdgeschwader') SCHOLAR+BAT: (0.5148, 'platoon counterattack bridgehead divisional battalion mortar perimeter brigade beachhead regimental')	0.2651
35	SCHOLAR: (0.3361, 'thanhouser filmfare bollywood filmography directorial kumar telugu starred biopic hindi') SCHOLAR+BAT: (0.5322, 'kumar bollywood directorial filmography telugu filmfare prasad malayalam bachchan hindi')	0.2888
40	SCHOLAR: (0.7394, 'batsman wicket bowled bowler bowling wisden cricketer selector inning crease') SCHOLAR+BAT: (0.761, 'bowled wisden selector batsman bowler wicket cricketer crease spinner mcc')	0.3045
41	SCHOLAR: (0.4571, 'statute constitutionality plaintiff unconstitutional defendant judicial appellate amendment jurisdiction judiciary') SCHOLAR+BAT: (0.4569, 'prosecutor prosecution investigator testified testimony conviction convicted verdict sentenced pleaded')	0.3137
45	SCHOLAR: (0.4178, 'edda mahabharata scripture purana goddess poem poetic shiva prose devotional') SCHOLAR+BAT: (0.3379, 'northumbria inscription kingship deity shrine annals worshipped attested buddha vassal')	0.3658
48	SCHOLAR: (0.5286, 'cavalry grenadier flank bridgehead infantry bayonet brigade artillery regiment repulsed') SCHOLAR+BAT: (0.3652, 'dso despatch raaf gallantry adjutant instructor aviator canberra airman citation')	0.544

Table 17: Fifteen aligned topic pairs from the Wiki dataset.

Pair #	SCHOLAR vs SCHOLAR+BAT (NPMI, Top 10 Topic Words)	JS Divergence
1	SCHOLAR: (0.2333, 'scientist monster cgi alien creature scientists attack bullets aliens sci') SCHOLAR+BAT: (0.2636, 'scientist alien creature monster aliens computer cgi space giant scientists')	0.0273
4	SCHOLAR: (0.165, 'vhs copy remember dvd ago tape saw video years loved') SCHOLAR+BAT: (0.1844, 'vhs copy tape remember dvd bought ago saw video available')	0.0327
8	SCHOLAR: (0.1146, 'kids kid dad parents mom christmas decides dies santa guy') SCHOLAR+BAT: (0.118, 'dad mom kids parents kid uncle decides christmas dies cat')	0.0379
11	SCHOLAR: (0.1968, 'adaptation version novel bbc versions jane kenneth handsome adaptations faithful') SCHOLAR+BAT: (0.2181, 'adaptation novel book read books faithful bbc version versions novels')	0.0383
15	SCHOLAR: (0.2758, 'show episodes episode shows abc season aired sitcom television seasons') SCHOLAR+BAT: (0.2678, 'seasons episodes show aired episode abc sitcom season television network')	0.0416
17	SCHOLAR: (0.0863, 'fails wooden lacks unconvincing shallow contrived wretched embarrassing thin embarrassment') SCHOLAR+BAT: (0.1047, 'lacks pacing fails contrived flat irritating lacking chemistry unconvincing uninteresting')	0.0424
22	SCHOLAR: (0.174, 'documentary footage interviews music documentaries disc dvd musicians extras insight') SCHOLAR+BAT: (0.1796, 'footage available documentary release dvd print interviews vhs subtitles audio')	0.0459
27	SCHOLAR: (0.1532, 'sheriff car town decides husband killer police investigate chase security') SCHOLAR+BAT: (0.2504, 'murder murdered detective killer murderer police murders suspects secretary serial')	0.0531
28	SCHOLAR: (0.3054, 'christian religious god religion christ faith church jesus beliefs truth') SCHOLAR+BAT: (0.1027, 'filmmaker intellectual filmmakers pretentious artistic subject content sake context claim')	0.0539
31	SCHOLAR: (0.1136, 'gags school rock band cartoons record boys principal radio metal') SCHOLAR+BAT: (0.3144, 'songs musical singing sing dancing singer concert song numbers dance')	0.0556
35	SCHOLAR: (0.0813, 'development seemed boring predictable weak explanation slow potential interesting suspense') SCHOLAR+BAT: (0.092, 'hour asleep minutes seemed sounded sat felt rented waste confusing')	0.0616
40	SCHOLAR: (0.1821, 'noir murder detective gritty crime cop thriller tough clint veteran') SCHOLAR+BAT: (0.1027, 'cop dennis sheriff gangster boss agent villain hopper action chases')	0.0679
41	SCHOLAR: (0.095, 'porn cops girls random camera amateurish tedious amateur screaming chick') SCHOLAR+BAT: (0.1548, 'kills killed killer screaming killing kill boyfriend woods walks dies')	0.0803
45	SCHOLAR: (0.1884, 'planet wars sci graphics space science game robot fiction weapons') SCHOLAR+BAT: (0.0959, 'action development fighting sequences visuals realistic epic fight battles cool')	0.0925
48	SCHOLAR: (0.1732, 'book read books novel adaptation author reading disappointed adapted translation') SCHOLAR+BAT: (0.1063, 'liked overall surprised disappointed enjoyed pleasantly pretty expectations seemed expecting')	0.1196

Table 18: Fifteen aligned topic pairs from the IMDB dataset.