Self-Supervised Neural Topic Modeling

Seyed Ali Bahrainian[1,2], Martin Jaggi[2], and Carsten Eickhoff[1]

bahrainian@brown.edu, martin.jaggi@epfl.ch, carsten_eickhoff@brown.edu
[1]Department of Computer Science, Brown University, USA
[2]MLO, EPFL, Switzerland

Abstract

Topic models are useful tools for analyzing and interpreting the main underlying themes of large corpora of text. Most topic models rely on word co-occurrence for computing a topic, *i.e.*, a weighted set of words that together represent a high-level semantic concept. In this paper, we propose a new light-weight Self-Supervised Neural Topic Model (SNTM) that learns a rich context by learning a topic representation jointly from three co-occurring words and a document that the triplet originates from. Our experimental results indicate that our proposed neural topic model, SNTM, outperforms previously existing topic models in coherence metrics as well as document clustering accuracy. Moreover, apart from the topic coherence and clustering performance, the proposed neural topic model has a number of advantages, namely, being computationally efficient and easy to train.

1 Introduction

Topic models are a means of exploratory document analysis which aim at discovering the underlying themes and narratives within a corpus of text. These models have been extensively used for discovering the latent topical structure of texts in various applications, such as social media analysis (Nguyen and Shirai, 2015), news analysis (Mele et al., 2019), understanding scientific articles (Wang and Blei, 2011; Bahrainian et al., 2018) and more.

The most well-known topic model is the Latent Dirichlet Allocation (LDA) (Blei et al., 2003b), a generative probabilistic model that relies on co-occurrence patterns between observed words to compute latent topics. The inference step of LDA is commonly based on approximation methods such as variational inference or collapsed Gibbs sampling, due to the intractability of exact inference at scale (Neal, 1993).

On the other hand, the success of neural word embedding models such as Variational Auto Encoders (Kingma and Welling, 2014) or Word2Vec (Mikolov et al., 2013) that also rely on capturing word co-occurrence patterns while using neural network black-box inference opened a new path to neural topic modeling.

Subsequently, several neural topic models emerged. However, some of these models came with limitations such as: (1) not being able to compute per-document topic distributions, e.g., NVDM (Miao et al., 2016) , (2) difficulty of training with respect to computational cost, e.g., topic models based on Generative Adversarial Networks (GANs) (Wang et al., 2020). (3) Priors based on distributions such as a logistic normal distribution (Miao et al., 2016) which may not always model the co-occurrence behaviour among words realistically, as they expect topic proportions of a corpus to follow such patterns.

Previous research by (Levy and Goldberg, 2014) on word embeddings has shown that a variant of Point-wise Mutual Information (PMI) (Church and Hanks, 1990) computes association patterns among words from a text corpus very similar to that of the Word2Vec model (Mikolov et al., 2013) even outperforming Word2Vec on word similarity tasks. This is an indication that PMI is a simple yet effective method for computing word associations and word embeddings.

In this paper, we propose the Self-supervised Neural Topic Model (SNTM) that firstly utilizes the Normalized Point-wise Mutual Information (NPMI) measure to construct a graph of word connections in order to identify the most prominent words that have the strongest co-occurrence with other words. We show that this method can compute probability scores for words such that the top most probable words are highly similar to those computed using LDA. Secondly, we design a self-supervised neural network architecture that is

trained jointly with top triple word co-occurrences as well as documents that contain them. This approach exposes the neural network to a rich context, i.e., co-occurrence of the triplets with all words from the documents containing the triplet in order to learn topics. This method allows for learning topics from a much richer context information as compared to most other topic models such as LDA or even neural variants such as NVLDA (Srivastava and Sutton, 2017) which learn from a single word co-occurrence at a time. Here we strive for proposing a different light-weight approach that takes advantage of leveraging multiple word co-occurrences in each training step.

Therefore, the main contributions of this paper are as follows:

1. We propose a self-supervised neural topic model that can learn topics from rich context information in an efficient way.[1]

2. We show that this model computes highly coherent topics while setting the state of the art in terms of document clustering accuracy among topic models.

The remainder of this paper is organized as follows: In Section 2 we present an overview of related work on neural topic modeling. Section 3 presents SNTM, our novel topic model. We evaluate the topic model in terms of topic coherence and document clustering accuracy on a public dataset in Section 4. Finally, we conclude the paper in Section 5 and present directions of future work.

2 Related Work

In this section we review the related work with a focus on neural topic models.

One early work on neural topic modeling is the Neural Variational Document Model (NVDM) by (Miao et al., 2016) which is based on the concept of variational auto encoders. As an unsupervised generative model NVDM is a variational auto encoder consisting of a Multi Layer Perceptron (MLP) encoder that compresses document representations into continuous hidden vectors and a softmax decoder that reconstructs the documents by independently generating the words. A limitation of this model is that it does not explicitly model topic assignments, meaning that per-document topic proportions cannot be computed. The model is also based on a Gaussian prior over a hidden state, modeling topics of a document.

Later, inspired by NVDM, the Gaussian Softmax Model (GSM) (Miao et al., 2017) was introduced. As an improvement over NVDM, the GSM modeled topics by providing parameterizable distributions over topics in the framework of neural variational inference.

Subsequently, another neural topic model, the NVLDA (Srivastava and Sutton, 2017), was proposed. This model is based on an approximation of the Dirichlet prior using a Logistic-Normal distribution.

In the past few years, a number of neural topic models based on the Generative Adversarial Networks (GANs) (Goodfellow et al., 2014) have been introduced. The Adversarial-neural Topic Model (ATM) (Wang et al., 2019) is one such model based on the notion of adversarial training, which however comes at significant additional computational cost. ATM models topics using a Dirichlet prior which is able to capture topics with multiple distinctive focuses as compared with the logistic-normal prior.

Another model based on bi-directional GANs is the Gaussian-BAT (Wang et al., 2020) which uses a Drichlet prior and can infer topic distributions of input documents. Additionally, Gaussian-BAT models a topic using a multivariate Gaussian and incorporates the word relatedness into the modeling process. Previous work on modeling contexts of words have also used multivariate Gaussians (Vilnis and McCallum, 2015) or Gaussian Mixture Models (Bahrainian and Crestani, 2018).

Finally, another work closely related to ours is the topic modeling method proposed by (Arora et al., 2013). They use the idea of anchor words to model topics. Their approach named FastAnchorWords performs a distance-based search of farthest words from previously found anchor words.

In this work we propose the first self-supervised neural topic modeling method that first uses a novel method for ranking words in terms of importance and association with other words to compute the main seed words upon which topics are formed. Second, our proposed model, SNTM, goes beyond the basic word co-occurrence methods used in most neural topic models and incorporates a framework for joint learning of co-occurrence of three seed words along with the documents that they originate from in a self-supervised setting. This

[1] https://github.com/ali-bahrainian/SNTM

method allows learning co-occurrence representations from hundreds of words at each step. Our empirical results corroborate that our topic model sets a new state-of-the-art performance in terms of document clustering accuracy as well as topic coherence among existing topic models.

3 A Self-Supervised Neural Topic Model

In this section we introduce SNTM, our proposed neural topic model. In the following subsections we first present a background related to our model, then elaborate on the model architecture.

3.1 Background

As stated in Section 1, previous research by (Levy and Goldberg, 2014) on word embeddings has shown the effectiveness of PMI in computing word similarity as compared with the Word2Vec model. To elaborate further, the authors investigate the objective function of the Skip-Gram with Negative Sampling (SGNS) variant of Word2Vec which is based on the Noise-Contrastive Estimation (NCE). The objective function of SGNS for the word w and its context c is:

$$\sum_{w \in V_w} \sum_{c \in V_c} logP(D=1|c,w) + q \cdot E_{c_N \sim P_D} logP(D=0|c,w) \tag{1}$$

where q is the number of negative samples and c_N is the sampled context, drawn according to an empirical unigram distribution.

They formally prove that the above objective function is equal to the PMI function shifted by a global constant:

$$\text{PMI}(w_i, c_j) - log(q) \tag{2}$$

The PMI of the word w and its context c is defined as:

$$\text{PMI}(w,c) = log \frac{count(w,c)}{count(w).count(c)} \tag{3}$$

The above function can return positive values for observed correlated occurrences of w and c but it can also return negative values for uncorrelated outcomes or even worse for unobserved examples. Therefore, it is common practice in NLP research to use positive PMI which is defined as $max(PMI(w,c), 0)$.

Now, going back to Equation 2, we can observe that a shifted PMI simply discards positive PMI values less than $log(q)$ to further filter out (w,c) pair outcomes with low correlations.

The paper (Levy and Goldberg, 2014) then concludes that the shifted PMI does far better in optimizing the objective function of SGNS and outperforms Word2Vec word vectors in word similarity tasks. Thus, given the effectiveness of the shifted PMI in word similarity tasks, we propose a method based on a normalized version of the shifted PMI for grouping similar words. In the following subsection we discuss the model in detail.

From this point on, whenever we discuss PMI or Normalized PMI (NPMI), we refer to the positive values of these functions.

3.2 Model Architecture

SNTM performs the following steps to model topics: First, it identifies top seed words from the vocabulary V that are most important and representative of a given document collection. Then it splits them into k clusters for computing k topics. Finally, triplets of seed words are paired with documents where they originate from and used to train a feed-forward neural network with a single hidden layer in a self-supervised setting to learn the final topics.

In order to compute the seed words we propose to compute the shifted NPMI (SNPMI) table for all words in V as follows:

$$\text{SNPMI}(w,c) = \frac{log \frac{P(w,c)}{P(w).P(c)} - log(q)}{-log(P(w,c))} \tag{4}$$

where P(*) denotes:

$$P(*) = \frac{count(*)}{corpus - word - count} \tag{5}$$

The intuition behind this equation is firstly that, as mentioned in the background, the shifted PMI computes word similarity with a high accuracy outperforming Word2Vec. Secondly, since our goal is to design a topic model, it is important to take the normalized word frequencies into account so that the modeling of topics follows the word occurrence proportions from the entire dataset.

In order to find the most important words of a corpus of documents in terms of relatedness with other words in the corpus we propose the following equation for computing each element of vector M of size n for V of size n where :

$$\mathbf{m}_{vi} = \sum_{c \in V} \frac{\text{SNPMI}(w,c)}{n} \tag{6}$$

In order to show the effectiveness of this approach we first show two examples and later in

the same section visualize the effect of the above equation using a graph of words:

Example 1. Let us say we would like to identify the outlier word among a set of four words 'secure', 'encryption', 'system' and 'space'. Using SNPMI values with $q = 1$ trained on the 20 News Group dataset (i.e. see Section 4.1 for details) for the four words we compute the scores 0.22, 0.17, 0.13 and 0.0 respectively. The scores clearly indicate that the outlier word is 'space'. This method assigns a relatively low score to a word which has a weaker association with other words.

Example 2. In this example we make two empirical experiments. In the first experiment we compare the top words as scored by Equation 6 against top words as computed by the LDA model. In order to do so we train an LDA model with 20 topics and default parameters on the 20 News Group dataset and take the top 10 words from each topic amounting to 200 not necessarily unique words. By comparing the top 200 words using each method, we find out that there is a 67% overlap between the top words using each method. Furthermore, in the second experiment we take all the words in a single LDA topic for all 20 topics and re-rank the words with non-zero probabilities using Equation 6. We find a 92% overlap between the top 10 words computed using the two methods.

Both above examples show that this method of ranking words is highly effective in scoring higher the most coherent and connected words in a set. Moreover, the scoring easily points out those words that are considered outliers. The two examples show two different use cases. We are more interested in the second example where the scoring could identify most top words as computed by the LDA topic model.

Our goal is to take the top i words (i.e., those with the highest scores) and use them as seed words to form topic clusters. For this purpose we draw the top i words according to the empirical unigram distribution $i = k * 20$ as seed words.

Let us consider a graph structure where the nodes are words and the edges represent the positive SNPMI scores. The edges that are most connected and with higher scores are the words that Equation 6 assigns the highest scores to and are selected in the unigram distribution of seed words. Figure 1 shows a part of this graph based on real data from the 20 News Group dataset and for $k = 20$. The circled words in the graph are among the top words

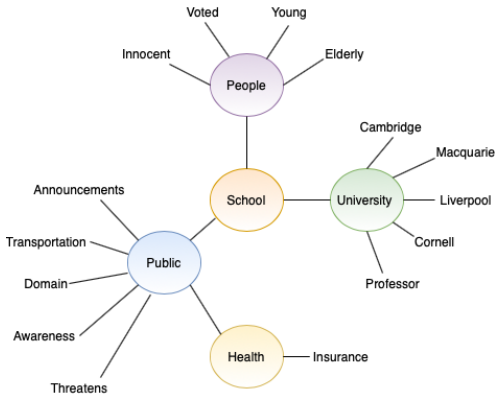

Figure 1: An example showing the seed words selection process

returned by Equation 6 (i.e. the seed words) while the other non-circled words are ranked lower. The edges in the graph connect word pairs where one word is among the top 10 similar words to the other according to the NPMI score. While each circled word is connected with other highly co-occurring but non-circled words those non-circled words are often too specific and with few connections to other words that they may not be representative of the main themes of a given dataset for the level of granularity specified by, k, the given number of topics. For instance, while the word 'university' is among the top words as computed by Equation 6, the non-circled connected words to it such as 'Cambridge' and 'Cornell' are too specific that may not be so representative of the entire corpus.

As the next step, we represent each word from the i words in terms of its SNPMI with every other word in the set of seed words as feature vectors. That is, each word is presented with a feature vector of size i having the value of 1 at its own designated index and the respective SNPMI at every other index. Subsequently, we train a K-Means clustering with k equals to the desired number of topics. The result is k clusters of seed words that serve as a basis for computing topics.

Subsequently, we present a self-supervised neural network method for computing the topics. We propose to learn a joint representation of triplets of the seed words computed in the previous step alongside a document where the triplet appears in. For this purpose, we draw random combinations of the seed words from each topic and pair each of them with a document where the triplet appears in. Furthermore, we use the class label of the triplet computed by the K-Means algorithm

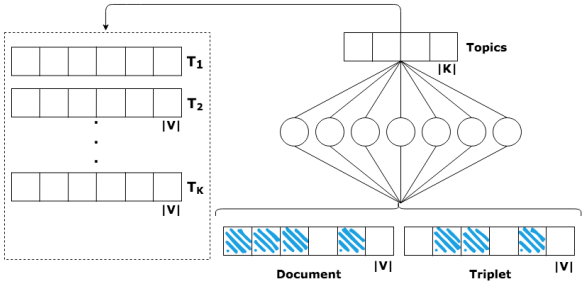

Figure 2: The architecture of SNTM, our proposed self-supervised topic model

as its training target. In other words, the target is a k-dimensional vector where for the triplet coming from class $cl \in [1 \dots k]$, the cl index is set to 1 while others are set to 0. Figure 2 shows a schematic presenting the SNTM. As can be seen in the figure, the input to the network is two vectors both of the size of the vocabulary $|V|$. One is designated as an input document while the other is used to represent a triplet drawn from the seed words. In other words, the document is presented as a binary vector in the size of the vocabulary. The triplet is also presented as another binary vector in the size of the vocabulary. Therefore, the input vector to the network is of size $2 * |V|$. Such joint representation enables the model to observe co-occurrence patterns between hundreds of words (i.e. present in the selected input document as well as the triplet) at a time and facilitates learning a rich context.

To further elaborate on the details, we also present the training algorithm of SNTM in Algorithm 1. Moreover, we present a diagram in Figure 3 showing the flow of the training steps of SNTM visually.

Algorithm 1 Training Algorithm

```
procedure TRAIN
    Input: SNPMI table, words w_1 to w_V, number of topics K, seed_words = []
    for i = 1 to V:
        compute m_vi using Equation 6 by inputting the SNPMI table
        seed_words ← seed_words.union(m_vi)
    sorted_seed_words ← seed_words.sort(by=descending)
    top_seed_words ← sorted_seed_words.select_top(count=20*K)
    seed_word_clusters ← k-means(top_seed_words , K)
    docs = []
    labels = []
    for each cluster in seed_word_clusters:
        for each word-triplet combination in cluster:
            docs ← docs.union(training documents containing the word-triplet)
            labels ← labels.union(cluster label of the word-triplet)
    training_features ← binary vector representation of word-triplets and docs (size 2*V)
    training_labels ← one-hot encoding of labels (size K)
    train the Feed Forward Neural Networks as shown in Figure 2
```

In order to train the network we optimize the cross entropy loss using the stochastic gradient descent optimizer. We define the explicit loss function of the neural network architecture as a multi-class classifier. Formally, the cross entropy loss function for multi-class 1 to K for K topics and N training samples is:

$$CrossEntropyLoss = -\frac{1}{N}\sum_{i}^{N} log(\frac{exp(x_{label-k})}{\sum_{k}^{K} exp(x_k)}) \quad (7)$$

At inference time in order to compute each topic we feed every word $w \in V$ to the model and get a k dimensional vector showing the probability of each word in each topic. Transposing this vector and aggregating the results of the same inference step over all other words results in all k topic-word distributions.

Analogously, in order to compute the per-document topic proportions we feed a document as a bag-of-words representation to the network and obtain the probability of each topic for the given document.

3.3 Discussion on Model Advantages

The main advantages of the proposed model are:

1. The SNTM is light-weight and although training it is slower than Bayesian models such as LDA, it is still trainable on commodity hardware such as a laptop and does not even approach the computational demand of other neural topic models such as the GAN-based ones discussed in Section 2.

2. Bayesian models such as LDA are very effective at dealing with sparse data with missing values. On the other hand it has been shown in various models such as Word2Vec that neural networks are best at handling dense vectors. SNTM is designed to take advantage of this feature to learn joint representations of hundreds of word co-occurrences coming from a document paired with a triplet at once.

3. The self-supervised training uses input data with very limited noise as opposed to the common trend that topic models aim at learning every word co-occurrence, although the correlation between the two words might be very slight, thus introducing noise into the model.

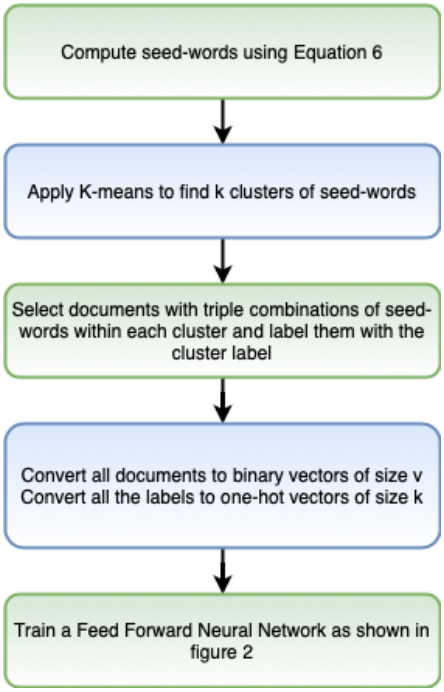

Figure 3: Flow of steps of Training SNTM: Seed words selection, converting documents and triplets to feature vectors, and training the neural network.

4 Evaluation

In this section we evaluate SNTM by comparing it against several baseline models in terms of topic coherence as well as document clustering accuracy.

4.1 Datasets

In order to evaluate the different topic models, we use two public datasets that are commonly encountered in the topic modeling literature.

20 News Group Dataset. This dataset (Lang, 1995) is one of the most frequently used datasets in topic modeling and document clustering. It contains a total of approximately $20,000$ news articles, divided in 20 different classes. The dataset contains $11,259$ training samples and $7,488$ test samples. The most important advantage of the 20 news group dataset is the availability of class labels, making it feasible to be used for evaluating document clustering models.

The New York Times News Dataset. This dataset contains a large number of news articles covering a wide range of subjects published between 2007 and 2015. However, following the common method of sub-sampling this dataset for topic modeling research and to compare as closely as possible to (Wang et al., 2020), the top baseline topic model, we randomly sample $100,000$ news articles and use them for topic coherence evaluation. Due to the lack of class labels for the articles, this dataset cannot be used for evaluating clustering.

The Grolier Dataset. This dataset [2] contains encyclopedia articles covering a range of different labels. It contains $30,991$ documents. This dataset cannot be used for document clustering evaluation due to a lack of class labels.

The AGNews Dataset.

This dataset [3] contains news articles with class labels. The dataset consists of $96,000$ training samples and $7,600$ test samples with class labels. This dataset contains documents from four different classes. Since, two of the above datasets already cover news articles evaluated for topic coherence, we use this third news dataset only for evaluating document clustering.

4.2 Baseline Models

Here we list all the baselines used in the experiments. We use the default settings for all models, as indicated in their respective original papers:

LDA (Blei et al., 2003b) is a probabilistic topic model based on hierarchical Bayesian networks.

NVDM (Miao et al., 2016) is based on variational auto encoders. For further details we refer to Section 2.

GSM (Miao et al., 2017) is a topic model designed based on the NVDM.

NVLDA (Srivastava and Sutton, 2017) is another topic model based on the variational auto encoder with the logistic-normal prior.

ProdLDA (Srivastava and Sutton, 2017) is a topic model which enhances LDA in terms of topic coherence in which the distribution over individual words is a product of experts.

ATM (Wang et al., 2019) is another neural topic model based on adversarial training.

Gaussian-BAT (Wang et al., 2020) is a topic model based on GANs which uses a Dirichlet prior and can model a topic using a multivariate Gaussian.

W-LDA (Nan et al., 2019) is based on Wasserstein autoencoders.

4.3 Experimental Results

In this section we evaluate and compare our topic model against other models in terms of topic coherence as well as document clustering accuracy.

[2] https://cs.nyu.edu/ roweis/data.html

[3] http://groups.di.unipi.it/gulli/AG_corpus_of_news_articles.html

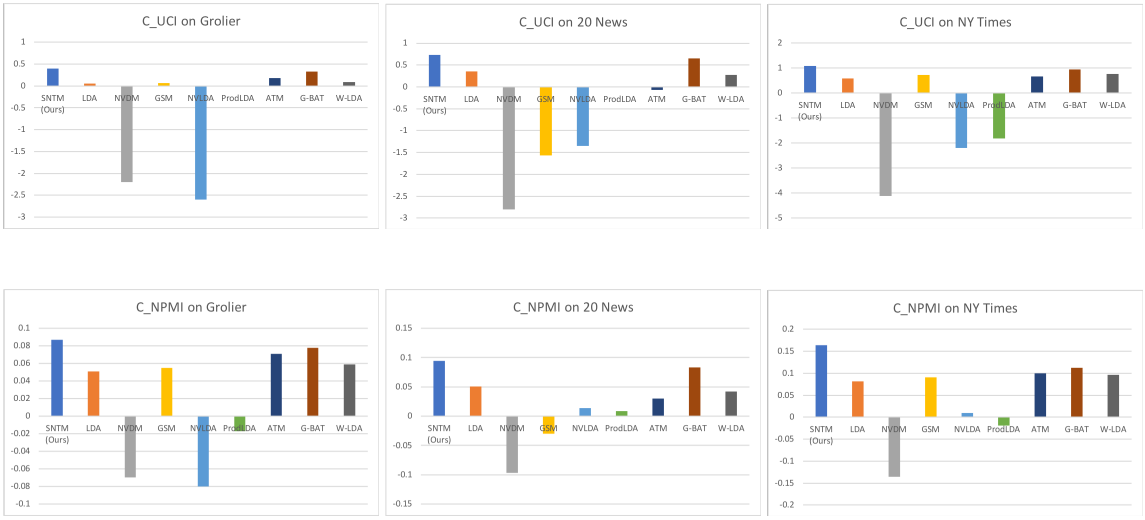

Figure 4: C_NPMI scores (bottom) and C_UCI scores (upper) for all computed topics by each model on the 20 news group, the NY Times, and the Grolier datasets for topic number settings 20, 30, 50, 75 and 100.

Moreover, we assess the output topics qualitatively. Finally, we show the impact of a parameter of SNTM.

In order to train our proposed topic model, following common practice, we lower-case all words, remove stop words and punctuation marks and additionally remove any words which occur less than three times in the corpus. We train the model for $2,000$ epochs and use a hidden layer size of 300 with a learning rate of 0.01 and the variable q set to 1 as the model parameters for each corpus.

Topic Coherence. In this experiment, we compare SNTM in terms of topic coherence against all other baseline models. Topic models can be evaluated using sequence likelihood on held-out data and this was traditionally a common evaluation method. However, (Chang et al., 2009) experimentally observed that likelihood is a measure contrary to human judgment. As such since one important goal of topic models is to be used as tools for humans to make sense of and explore document collections, the recent trend is to evaluate topics based on coherence metrics. The work of (Röder et al., 2015) and (Wang et al., 2020) are examples of topic coherence evaluation. Here, we take the same approach to evaluation.

First, we conduct an experiment comparing all models in terms of the coherence scores C_UCI and C_NPMI with five topic number settings (i.e., 20, 30, 50, 75 and 100) by considering all computed topics from all models. This experiment examines the coherence quality of topics produced by each model.

Figure 4 presents the results of this experiment on both the 20 news group dataset as well as the New York Times dataset. As we show in the figure, our model achieves higher overall coherence scores (C_NPMI as well as C_UCI) on both datasets for all topics when compared against any of the baseline models including the Gaussian-BAT. This is while our model requires significantly less computational resources to be trained as compared with the GAN-based model.

We conclude from this experiment that, overall, SNTM computes topics that are more coherent than the other models.

As a second experiment on coherence, we follow the approach of (Wang et al., 2020) in computing the C_NPMI and the C_UCI coherence scores. That is, we take five topic number settings 20, 30, 50, 75 and 100 (i.e similar to our previous experiment) for each of our datasets. However, we then calculate the average topic coherence values among topics whose coherence values are ranked at the top 50%, 70%, 90%, and 100% positions. As an example, for calculating the average C_NPMI value of SNTM@70%, we first compute the average C_NPMI coherence with the selected topics whose C_NPMI values are ranked at the top 70% for each topic number setting, and then average the five coherence scores with each corresponding to a particular topic number setting.

This experiment is designed to examine the av-

	Our model	LDA	NVDM	GSM	NVLDA	ProdLDA	ATM	Gaussian_BAT	W-LDA
C_UCI	0.6528	0.3399	-2.9496	-1.6083	-1.3466	-1.5044	-0.3871	0.5925	0.3271
C_NPMI	0.0924	0.0523	-0.0984	-0.0400	-0.0207	-0.0083	0.0207	0.0819	0.0486

Table 1: A comparison between all models in terms of two main coherence scores C_UCI and C_NPMI on the 20 News Group dataset. Higher numbers are better. The coherence scores are computed by averaging topic number settings 20, 30, 50, 75 and 100 and averaging over topics whose coherence values are ranked at the top 50%, 70%, 90%, and 100% positions.

Topic 1	**Topic 2**	**Topic 3**	**Topic 4**	**Topic 5**
encryption	team	jesus	space	control
clipper	game	bible	solar	laws
chip	play	christian	nasa	guns
secure	teams	people	satellite	firearms
escrow	played	belief	launch	death
digital	season	faith	lunar	citizens
private	players	life	shuttle	weapons
keys	playoff	christ	planetary	people
encrypted	goal	truth	astronomy	debate
communications	league	religious	orbital	deaths

Table 2: Sample topics computed by SNTM, our topic model, with $K = 20$ on the 20 News group dataset. Top ten words based on their computed probability are used to present each topic. We deduce that the topics from left to right are about cryptography, sports, Christianity, space and gun control laws.

erage coherence of top topics as well as the less coherent ones, although analyzing the coherence of all computed topics (as done in the previous experiment) is more comprehensive.

In this way we compute the C_NPMI and the C_UCI coherence metrics. Table 1 shows the results of this experiment on the 20 News Group dataset. We can observe that our model generates topics that are far more coherent than those of the baseline models in terms of the both coherence metrics.

Sample Output Topics. We present five example topics from the topics computed by our model on the 20 News Group dataset when the number of topics k was set to 20. Table 2 shows these topics. For each topic the top 10 words based on their computed probability are shown. We can observe in the table that the topics are very coherent such that one can easily deduce a higher semantic meaning as to what each topic is inferring to.

Document Clustering. In this experiment, we evaluate SNTM in terms of document clustering accuracy and compare it with other baseline models. The 20 News group dataset comes with class labels for 20 different news classes. We use this dataset for this experiment and set the number of topics to 20 to resonate with the number of ground-truth classes. In order to compute document clustering accuracy, similar to (Wang et al., 2020), we use the following equation:

$$ACC = max \frac{\sum_{i=1}^{N_t} ind(l_i = map(c_i))}{N_t} \quad (8)$$

where N_t is the number of documents in the test set, ind$(\cdot)$ is the indicator function, l_i is the ground-truth label of $i-th$ document, c_i is the category assignment, and map ranges over all possible one-to-one mappings between labels and clusters.

Table 3 reports the results of the document clustering experiment. We can conclude from this experiment that our topic model, SNTM, outperforms all other baseline models in the clustering task including the top GAN-based baseline. We conclude from this experiment that SNTM is highly effective at distinguishing texts of different topic categories from one another.

The Effect of the q Parameter. In this experiment, we analyze the effect of parameter q presented in Section 3. We recall that $log(q)$ is the threshold below which NPMI correlations are discarded and set to 0. While all other experiments in the paper were carried out with $q = 1$, we would like to analyze other values of q. In (Levy and Goldberg, 2014) values of 1, 5, and 15 were used. Here we also explore the effect of these values.

We repeat the same topic coherence experiment on the 20 News Group dataset.

In Table 4 we report the results of this experiment. We can see that higher values of q yield higher topic coherence. We also observe a bigger leap from $q = 1$ to $q = 5$ than from $q = 5$ to $q = 15$. This may mean that correlations with a lower score cause more noise in the data and affect the seed words selection process more severely.

Following (Wang et al., 2020), we compute up to 100 topics. Using higher values for q such as 50

Acc. (%)	SNTM(ours)	LDA	NVLDA	ProdLDA	G_BAT	W-LDA
20News	**42.16**	35.36	33.31	33.82	41.25	34.21
AGNews	**86.37**	79.74	76.53	77.43	75.81	83.87

Table 3: Comparing SNTM against the baseline models in terms of document clustering accuracy on the 20 News Group and the AGNews datasets.

	q=1	q=5	q=15
C_UCI	0.6528	0.6853	0.6971
C_NPMI	0.0924	0.0942	0.0948

Table 4: Analyzing different values of parameter q of SNTM.

causes a lack of availability of sufficient numbers of words to model topics due to discarding.

Despite this, we summarize our empirical findings here: Setting a higher value of q faces three main challenges: 1) As we move to higher NPMI values, there might not even be sufficient numbers of words to create topics with. 2) Using a very high value of q (e.g. 50) also means discarding information from the dataset which may lead to computing topics that are not representative of the entire corpus. 3) In a few experiments that we carried out by using $q = 50$ and $k = 10$, we observe that the computed topics have a few top words which are highly coherent with one another in terms of C_NPMI but then joined with other words that make the coherence score drop.

It is noteworthy, to mention that hierarchical topic models such as the work of (Blei et al., 2003a) also show fewer but more specific words as we move down a hierarchy.

Given these findings and the association with hierarchical topic models, we conclude that perhaps topics that are computed with a higher value of q can be expected to contain fewer words, making this a path to designing a hierarchical topic model variant. We leave this extension for future work.

Our final conclusion in this experiment is that small numbers of q such as $q = 5$ can provide slightly more coherent topics with reduced noise.

5 Conclusions and Future Work

In this paper, we introduce a novel neural topic model that can learn from co-occurrence patterns between many words at the same time and thus learn coherent topics very efficiently using self-supervised learning. The model can be trained on commodity hardware and does not require specific architectures such as GPUs.

We empirically show that our proposed topic model, SNTM, sets a new state of the art for topic coherence as well as document clustering accuracy.

Future work can include further analysis of training data selection methods that may result in improved topic model performance. Additionally, we believe that the variable q can be adjusted in order to obtain more fine-grained topics. The same functionality can be exploited to model topic hierarchies in terms of generating topics ranging from generic to highly fine-grained and contextual topics. Finally, in this paper we investigated the setting where three co-occurring seed words were used for training the model to expose the model to richer context information. In the future we will further explore other number of word combinations. Designing an altogether hierarchical topic model is another potential direction of future work.

Acknowledgements

This research is supported in part by the NSF (IIS-1956221), and SNSF (P2TIP2_187932). The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of NSF, SNSF, or the U.S. Government.

References

Sanjeev Arora, Rong Ge, Yonatan Halpern, David Mimno, Ankur Moitra, David Sontag, Yichen Wu, and Michael Zhu. 2013. A practical algorithm for topic modeling with provable guarantees. In *International Conference on Machine Learning*, pages 280–288. PMLR.

Seyed Ali Bahrainian and Fabio Crestani. 2018. Augmentation of human memory: Anticipating topics that continue in the next meeting. In *Proceedings of the 2018 Conference on Human Information Interaction & Retrieval*, pages 150–159.

Seyed Ali Bahrainian, Ida Mele, and Fabio Crestani. 2018. Predicting topics in scholarly papers. In *European Conference on Information Retrieval*, pages 16–28.

David M. Blei, Thomas L. Griffiths, Michael I. Jordan, and Joshua B. Tenenbaum. 2003a. Hierarchical topic models and the nested chinese restaurant process. In *Advances in Neural Information Processing Systems 16 [Neural Information Processing Systems, NIPS 2003*, pages 17–24.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003b. Latent dirichlet allocation. volume 3, pages 993–1022.

Jonathan Chang, Sean Gerrish, Chong Wang, Jordan Boyd-Graber, and David Blei. 2009. Reading tea leaves: How humans interpret topic models. *Advances in neural information processing systems*, 22:288–296.

Kenneth Ward Church and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *Comput. Linguist.*, 16(1):22–29.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, volume 27, pages 2672–2680.

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*.

Ken Lang. 1995. Newsweeder: Learning to filter netnews. In *in Proceedings of the 12th International Machine Learning Conference (ML95*.

Omer Levy and Yoav Goldberg. 2014. Neural word embedding as implicit matrix factorization. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, Montreal, Quebec, Canada*, pages 2177–2185.

Ida Mele, Seyed Ali Bahrainian, and Fabio Crestani. 2019. Event mining and timeliness analysis from heterogeneous news streams. *Information Processing Management*, 56(3):969 – 993.

Yishu Miao, Edward Grefenstette, and Phil Blunsom. 2017. Discovering discrete latent topics with neural variational inference. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70, pages 2410–2419.

Yishu Miao, Lei Yu, and Phil Blunsom. 2016. Neural variational inference for text processing. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, page 1727–1736. JMLR.org.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space.

Feng Nan, Ran Ding, Ramesh Nallapati, and Bing Xiang. 2019. Topic modeling with wasserstein autoencoders. *arXiv preprint arXiv:1907.12374*.

Radford M Neal. 1993. *Probabilistic inference using Markov chain Monte Carlo methods*.

Thien Hai Nguyen and Kiyoaki Shirai. 2015. Topic modeling based sentiment analysis on social media for stock market prediction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1354–1364.

Michael Röder, Andreas Both, and Alexander Hinneburg. 2015. Exploring the space of topic coherence measures. In *Proceedings of the eighth ACM international conference on Web search and data mining*, pages 399–408.

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

Luke Vilnis and Andrew McCallum. 2015. Word representations via gaussian embedding. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Chong Wang and David M Blei. 2011. Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 448–456.

Rui Wang, Xuemeng Hu, Deyu Zhou, Yulan He, Yuxuan Xiong, Chenchen Ye, and Haiyang Xu. 2020. Neural topic modeling with bidirectional adversarial training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 340–350. Association for Computational Linguistics.

Rui Wang, Deyu Zhou, and Yulan He. 2019. ATM: adversarial-neural topic model. *Inf. Process. Manag.*, 56(6).