# Enhancing Neural Topic Model with Multi-Level Supervisions from Seed Words

Anonymous ACL submission


## Abstract

Efforts have been made to apply topic seed words to improve the topic interpretability of topic models. However, due to the semantic diversity of natural language, supervisions from seed words could be ambiguous, making it hard to be incorporated into the current neural topic models. In this paper, we propose *SeededNTM*, a neural topic model enhanced with supervisions from seed words on both word and document levels. We introduce a context-dependency assumption to alleviate the ambiguities with context document information, and an auto-adaptation mechanism to automatically balance between multi-level information. Moreover, an intra-sample consistency regularizer is proposed to deal with noisy supervisions via encouraging perturbation and semantic consistency. Extensive experiments on multiple datasets show that SeededNTM can derive semantically meaningful topics and outperforms the state-of-the-art seeded topic models in terms of topic quality and classification accuracy.


## 1 Introduction

Unsupervised topic models, despite their efficiency in uncovering the underlying latent topics in text corpora (Blei et al., 2003), may suffer from poor topic interpretability as the semantic interpretability of latent space is poorly explored (Chang et al., 2009; Newman et al., 2011; Eshima et al., 2020) and the generated topics may not match users' desires (Jagarlamudi et al., 2012; Gallagher et al., 2017; Harandizadeh et al., 2022). To address this problem, topic seed words are incorporated as additional prior knowledge to provide richer semantic information and indicate users' preferences. Compared to sample-wise information like document labels, seed words can be easier to access, more widely applicable, and with a milder level of human bias.

Many works in conventional topic models incorporate seed words as guidance. Some works extend Latent Dirichlet Allocation (LDA) into seeded models (Andrzejewski and Zhu, 2009; Jagarlamudi et al., 2012; Li et al., 2016; Eshima et al., 2020), and some draw inspiration from information theory (Gallagher et al., 2017) or word embeddings (Meng et al., 2020a). While most of the conventional topic models struggle with the growing number of topics and documents, with the recent development of neural topic models (NTM), keyETM (Harandizadeh et al., 2022) is proposed to incorporate seed words into NTM to combine the advantages of NTM of scalability on large datasets.

However, keyETM only focuses on regularizing word-topic relations with seed words and fails to combine document-level topic information, which is essential as the semantics of words may vary under different context documents. As shown in Figure 1(a), under different contexts, the word 'apple' has different semantic meanings and may belong to different topics, even if it co-occurs with the seed word 'company'. This inspires us to incorporate supervisions from seed words into NTM on both word and document level and balance information from both levels for better inference of topics, thus achieving better topic interpretability.

There still remain challenges to effectively combining multi-level supervisions from seed words into the current framework of NTM. Firstly, the **mean-field assumption** made in current NTMs prevents the model from combining topic preferences of words and documents because they are assumed to be conditionally independent. Secondly, as shown in Figure 1(b), document level supervisions from seed words can be noisy due to the semantic ambiguity of natural languages. Previous work (Li et al., 2018) tried to tackle the problem via a neighbor consistency regularization. However, the neighbor-based method can be time-consuming, limiting the scalability on large datasets, and noisy

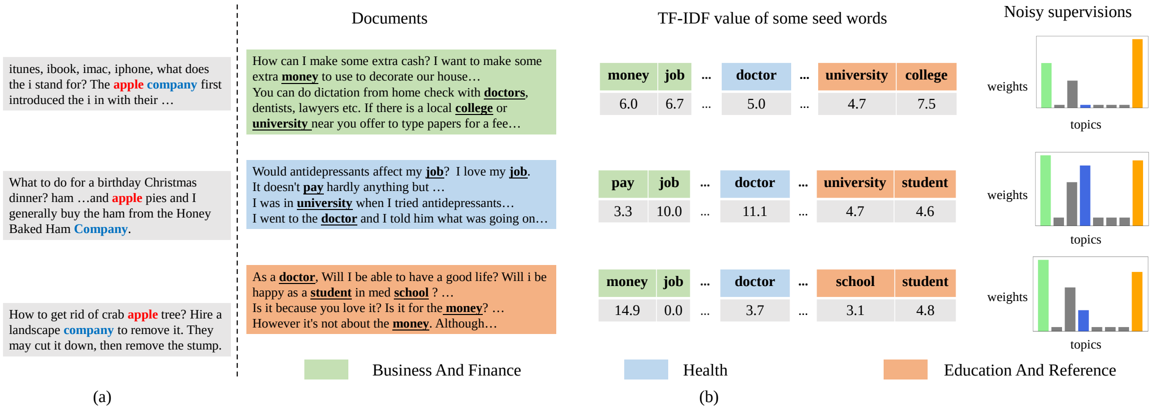


Figure 1: Examples from UIUC Yahoo Answers dataset. (a) Multiple semantic meanings of the word 'apple' under different contexts. (b) Seed words from three different topics bring noises to each other when estimating document topic preferences.

neighbors may cause cumulative errors.

To address these challenges, we propose a novel neural topic model *SeededNTM*, which incorporates seed words as supervisions and auto-adaptively balances information from both word and document level. During variational inference, we drop the mean-field assumption and make a context-dependency assumption to assist the inference of per-word topic assignment with context document information. Based on this assumption, we implement an auto-adaptation mechanism between multi-level information inspired by the idea of *product of experts* (Hinton, 2002). Moreover, to deal with the noisy document supervisions, we propose a novel regularizer that encourages **intra-sample** consistency to avoid time-consuming neighbor finding and cumulative errors. The regularizer encourages consistency between perturbed samples to preserve local structures and consistency between the semantics of outputs from different encoders to improve robustness.

Our contributions are summarized as follows:

- We propose SeededNTM, a novel neural topic model that leverages supervisions from seed words on both word and document level.
- We propose a reasonable context-dependency assumption and develop an auto-adaptation mechanism to automatically balance between word level and document level information.
- We propose an intra-sample consistency regularizer to deal with noises from document level supervisions by encouraging both perturbation and semantic consistency,.
- Extensive experiments on three public datasets show that SeededNTM can derive semantically meaningful topics and outperforms the state-of-the-art seeded topic models in terms of NPMI and classification accuracy.

## 2 Related Works

### 2.1 Neural Topic Model

The recent developments of neural variational inference (Kingma and Welling, 2014; Rezende et al., 2014) enable the application of neural networks on topic models to deal with scalability issues. NVDM (Miao et al., 2016) and ProdLDA (Srivastava and Sutton, 2017) are two representative works. Gaussian and logistic normal distribution are leveraged as approximations of the Dirichlet prior in the original LDA. Subsequently, various works have been proposed (Nan et al., 2019; Dieng et al., 2020; Nguyen and Luu, 2021), aiming for better inference of topics. Among these works, the most relevant to our work is VRTM (Rezaee and Ferraro, 2020). It explicitly models each word's the topic assignments $z_n$ while other works collapse them for simplicity. However, the mean-field assumption in VRTM prevents the model from combining context document information when inferring words' topic preferences, limiting its performance.

### 2.2 Topic Model with Prior Knowledge

Introducing prior knowledge into topic models has been a widely adopted way to improve topic interpretability. Some works (Bianchi et al., 2021a,b) incorporate pre-trained embeddings of words and documents to convey prior knowledge from additional datasets. Though effective, topic models with pre-trained embeddings remain unsupervised, and cannot mine information based on users' interests. Sample-wise knowledge, like labels (Blei and Mcauliffe, 2008; Wang and Yang, 2020) and covariates (Eisenstein et al., 2011; Card et al., 2018) can reflect the semantic structure information of the corpus but can be difficult to acquire and may introduce strong biases. On the other hand, topic seed words, as a kind of topic-wise knowledge, can be easier to access and more applicable. SeededLDA (Jagarlamudi et al., 2012) paired each topic with a seed topic and biased documents to topics if they have corresponding seed words. And keyATM (Eshima et al., 2020) improved upon SeededLDA by allowing topics with no seed word and better empirical hyperparameters. Anchored CorEx (Gallagher et al., 2017) proposed an information-theoretic framework and incorporates seed words by anchoring them to topics. CatE (Meng et al., 2020a) took category names as seed words and learned a discriminative embedding space for topics and words. And SEETOPIC (Zhang et al., 2022) improved upon CatE by using BERT to handle out-of vocabulary seed words. Recently, to combine the advantages of NTMs on scalability, keyETM (Harandizadeh et al., 2022) is proposed to incorporate seed words into NTM by regularizing word-topic relations with seed words and pre-trained word embeddings.

### 2.3 Weakly-Supervised Text Classification

Weakly-supervised text classification is a branch of classification task to build a text classifier with a few relevant words or descriptions for each category and no sample-wise labels. Because of the similar settings with seeded topic modeling, a few topic model-based methods are proposed (Chen et al., 2015; Li et al., 2016, 2018), and some recent works (Meng et al., 2020b; Wang et al., 2021; Zhang et al., 2021) attempt to bootstrap the seed word list to obtain stronger supervisions.

Despite similar settings, weakly-supervised text classification and seeded topic modeling differ in many aspects. While seeded topic modeling aims at discovering latent semantic structures of current corpus and focuses on the interpretability of learned topics, weakly-supervised text classification aims to build classifiers that generalize well on unseen data and focuses on the validity of the document-category partitions. Unsupervised topics are allowed in seeded topic modeling, and documents are interpreted as mixtures of multiple topics, while in weakly-supervised text classification, every category is assumed to be known in advance, and a document may be assumed to belong to a single category.

## 3 Background

### 3.1 Problem Formulation

Consider a corpus with $D$ documents, where each document $d$ contains $N_d$ words $\boldsymbol{w}_d = \{w_{d1}, w_{d2}, \dots, w_{dN_d}\}$, each belonging to a vocabulary of size $V$. And suppose that we have $K$ topics, each provided with a set of $L_k$ seed words denoted by $S_k = \{s_{k1}, s_{k2}, \dots, s_{kL_k}\}$. Our goal is to derive topics from the corpus that are semantically coherent with corresponding seed word sets.

### 3.2 Generative Story and Variational Inference

Our model builds on the generative story in (Srivastava and Sutton, 2017), where the Dirichlet prior is approximated via a logistic normal distribution. The generative story is summarized as follows, where $\alpha$ is the parameter for prior distribution and $\beta_k$ denotes the word distribution for the $k$-th topic:

For document $d$, draw topic distribution $\theta \sim \mathcal{LN}(\mu_0(\alpha), \sigma_0^2(\alpha))$;

For $w_{dn}$ in this document:

Draw topic $z_{dn} \sim Cat(\theta)$;

Draw word $w_{dn} \sim Cat(\beta_{z_{dn}})$;

Based on the generative story, variational inference is used to approximate posterior distribution of latent variables $\theta_d$ and $\boldsymbol{z}_d = \{z_{d1}, z_{d2}, \dots, z_{dN_d}\}$ to maximize the likelihood of observed data. And the evidence lower bound (ELBO) can be derived as

$$\begin{aligned}\mathcal{L}(\boldsymbol{w}) =& E_{q(\theta,\boldsymbol{z}|\boldsymbol{w})} \log \left(p(\boldsymbol{w}|\theta, \boldsymbol{z}; \beta)\right) \\ &- E_{q(\theta,\boldsymbol{z}|\boldsymbol{w})} \log \left(\frac{q(\theta, \boldsymbol{z}|\boldsymbol{w})}{p(\theta, \boldsymbol{z})}\right) \\ =& -(\mathcal{L}_{rec} + \mathcal{L}_{kl}),\end{aligned} \tag{1}$$

where $q(\theta, \boldsymbol{z}|\boldsymbol{w})$ is the joint variational distribution.

## 4 Methodology

In this section, we introduce our proposed *SeededNTM*. We start by introducing the model architecture and the designs of multi-level pseudo supervisions. Then we focus on our proposed auto-adaptation mechanism based on context-dependency assumption and our noise-reduction consistency regularizer. Finally, we introduce our training objective and summarize the training procedure with Algorithm 1.

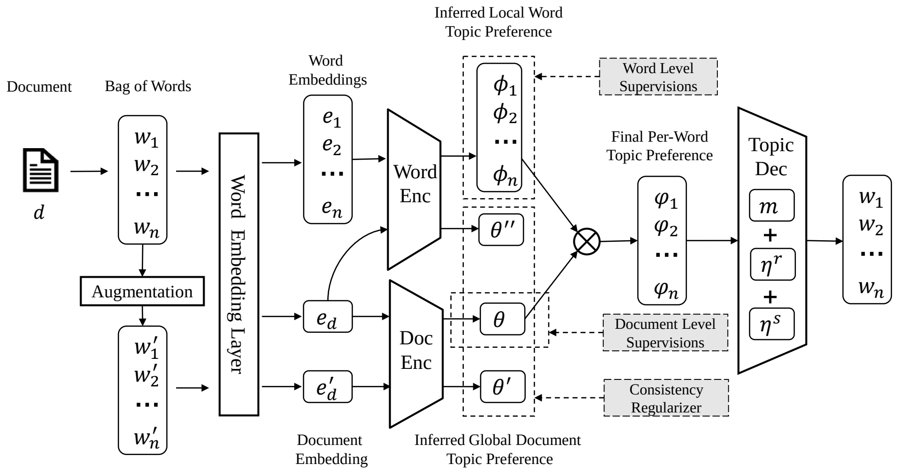


Figure 2: The overall structure of SeededNTM. The grey boxes indicate the training losses in SeededNTM, and the dashed boxes indicate the variables used in loss computations.

## 4.1 Model Architecture

### 4.1.1 Document Encoder

A multi-layer network is used as document encoder to infer the document-topic distributions $\theta$ for document $d$ with a word set $\boldsymbol{w}$. The words are first encoded into word embedding vectors $E_d = \{e_1, e_2, \ldots, e_{N_d}\}$ and then averaged to obtain the document embedding $e_d$. Then the mean vector $\mu$ and the diagonal of the covariance matrix $\sigma^2$ are further encoded with two sub-networks $\mu = f_\mu(e_d)$ and $\sigma^2 = f_\sigma(e_d)$, and the document-topic distribution is sampled via the reparameterization trick with $\epsilon \sim \mathcal{N}(0, \boldsymbol{I})$ and $\theta = softmax(\mu + \sigma \cdot \epsilon)$. The above procedure is donoted as $\theta = F_d(d)$.

### 4.1.2 Word Encoder

Word encoder encodes words to local word-topic preferences $\phi$. For a word $w_n$, it is first encoded to the embedding vector $e_n$, followed by a feed-forward network activated with a softmax function. The above procedure is donoted as $\phi_n = F_w(w_n)$.

### 4.1.3 Topic Decoder

The decoder contains topic-word distribution and reconstructs documents with topic mixtures. Inspired by (Eisenstein et al., 2011), we disassemble topics in log-space into three parts, background $m$, regular topic $\eta^r$, and seed topic $\eta^s$. The background term is estimated with the overall log frequencies of words from the corpus, and both regular and seed topics act as additional deviations on $m$. The possibility $\beta_{kv}$ for word $w_v$ in topic $k$ is

$$\beta_{kv} = \frac{\exp(m_v + \eta^r_{kv} + \eta^s_{kv})}{\sum_v \exp(m_v + \eta^r_{kv} + \eta^s_{kv})}, \qquad (2)$$

where $\eta^r_k$ is a $V$-dimensional parameter vector whose elements at positions corresponding to $S_k$ are fixed to zero. And $\eta^s_k$ is defined as

$$\eta^s_{kv} = \begin{cases} \kappa, & w_v \in S_k, \\ 0, & \text{otherwise}, \end{cases} v \in \{1, \cdots, V\}, \qquad (3)$$

where $\kappa$ is a hyperparameter of seeding strength.

## 4.2 Multi-Level Supervisions

### 4.2.1 Document Level Supervision

With seed words, we can regularize the inferred document-topic distribution $\theta$ with the pseudo distribution $\hat{\theta}$ which is estimated via the *tf-idf* scores of seed words appearing in the document. Formally, for a document $d$, its corresponding $\hat{\theta}$ is

$$\hat{\theta}_k = \frac{\frac{1}{L_k}\sum_{s \in S_k} tfidf(s,d)}{\sum_k \left(\frac{1}{L_k}\sum_{s \in S_k} tfidf(s,d)\right)}, k \in \{1, \ldots, K\}. \qquad (4)$$

And we regularize $\theta$ by minimizing the KL divergence between $\theta$ and $\hat{\theta}$,

$$\mathcal{L}_d(\theta, \hat{\theta}) = KL(\hat{\theta} \| \theta) = \sum_k \hat{\theta}_k \log(\frac{\hat{\theta}_k}{\theta_k}). \qquad (5)$$

### 4.2.2 Word Level Supervision

Local word-topic preferences $\phi$ can also be regularized by seed words. We estimate the pseudo word-topic distribution $\hat{\phi}$ with co-occurrence measured by the conditional possibility $p(w|s) = df(w,s)/df(s)$ of word $w$ and seed word $s$, where $df(\cdot)$ is the number of documents containing $s$ or both $s$ and $w$. And the pseudo possibility for word $w_n$ belonging to topic $k$ is

$$\hat{\phi}_{nk} = \frac{\frac{\tau}{L_k}\sum_{s\in S_k} p(w_n|s)}{\sum_k \left(\frac{\tau}{L_k}\sum_{s\in S_k} p(w_n|s)\right)}, \quad (6)$$

where $\tau$ is a temperature factor to sharpen the distribution. And we also use KL divergence to minimize the distance between $\hat{\phi}_n$ and $\phi_n$,

$$\mathcal{L}_w(\phi_n, \hat{\phi}_n) = KL(\hat{\phi}_n \| \phi_n) = \sum_k \hat{\phi}_{nk} \log(\frac{\hat{\phi}_{nk}}{\phi_{nk}}). \quad (7)$$

### 4.3 Auto-Adaptation of Multi-Level Information

In previous work (Rezaee and Ferraro, 2020), the inferred posterior distribution $q(\theta, \boldsymbol{z}|\boldsymbol{w})$ is decomposed with a mean-field assumption as

$$q(\theta, \boldsymbol{z}|\boldsymbol{w}) = q(\theta|\boldsymbol{w})\prod_n q(z_n|w_n), \quad (8)$$

but as we mentioned before, per-word topic preferences can be ambiguous without context document information. Therefore, instead of mean-field assumption, we introduce a context-dependency assumption by taking document topic distribution $\theta$ into consideration,

$$q(\theta, \boldsymbol{z}|\boldsymbol{w}) = q(\theta|\boldsymbol{w})\prod_n q(z_n|w_n, \theta). \quad (9)$$

As $z_n$ is now conditioned on both $w_n$ and $\theta$, how to properly balance information from word and document remains unsolved. Inspired by the idea of *product of experts* (Hinton, 2002), we propose an auto-adaptation mechanism to automatically combine local word-topic preference $\phi_n$ and the global document-topic preference $\theta$ and implement the combination as products of two distributions,

$$\varphi_{nk} = q(z_n = k|\theta, w_n) = \frac{\phi_{nk}\theta_k}{\sum_k(\phi_{nk}\theta_k)}. \quad (10)$$

In this way, we avoid manually weighting the global and local topic preferences and achieve auto-adaptation between multi-level information. Potential ambiguities in per-word topic preferences get re-weighted by the global document-topic distributions, and topics with higher probabilities in both distributions are further encouraged.

### 4.4 Noise-Reduction Consistency Regularizer

Document level supervisions can be biased by seed words' semantic diversity and ambiguity. To avoid time-consuming nearest neighbor method (Li et al., 2018), inspired by recent works in noisy label learning (Li et al., 2020; Englesson and Azizpour, 2021), we propose a consistency regularizer that encourages intra-sample consistency.

In this regularizer, we encourage outputs from the document encoder to be consistent with perturbed samples, $d^{'} \sim \mathcal{A}(d)$, where $\mathcal{A}$ is an data augmentation function. Each perturbed sample can be viewed as a neighbor with the original sample in feature space, and by encouraging perturbation consistency, we can preserve local structures without finding nearest neighbors.

Moreover, we encourage consistency with the outputs from the word encoder. The word encoder takes supervisions from the word-word co-occurrences and contains more fine-grained information than the document level. By encouraging consistency with the predictions of the word encoder on document embeddings, we incorporate semantic information from the word level to help correct the predictions from the document encoder and improve its robustness to noises.

We use the symmetric KL Divergence to measure the distance between two distributions, and our consistency regularizer is summarized as follows.

$$\begin{aligned} SKL(a,b) =& KL(a\|b) + KL(b\|a), \\ \mathcal{L}_c(d) =& SKL(\theta, F_d(d^{'})) + SKL(\theta, F_w(d)). \end{aligned} \quad (11)$$

### 4.5 Training Objectives

With the new assumption in Eq.9, $\mathcal{L}_{rec}$ and $\mathcal{L}_{kl}$ in Eq.1 can be further derived as

$$\begin{aligned} \mathcal{L}_{rec} =& -\sum_{n,k} \varphi_{nk} \log \beta_{kw_n}, \\ \mathcal{L}_{kl} =& KL\left(\mathcal{N}(\mu,\sigma^2)\|\mathcal{N}(\mu_0,\sigma_0^2)\right) + \sum_n KL\left(\varphi_n\|\theta\right). \end{aligned} \quad (12)$$

Detailed derivations can be found in Appendix A. Our final training objectives is

$$\mathcal{L}_{tr} = \mathcal{L}_{rec} + \lambda_0\mathcal{L}_{kl} + \lambda_1\mathcal{L}_d + \lambda_2\mathcal{L}_w + \lambda_3\mathcal{L}_c, \quad (13)$$

where $\lambda_0$ is KL annealing factor and gradually increases to 1 during training and $\lambda_1, \lambda_2, \lambda_3$ are hyperparameters. The overall structure of Seeded-NTM is shown in Figure 2, and the training procedure is described in Algorithm 1.

**Algorithm 1** The SeededNTM training procedure.

**Input:** corpus $\mathcal{D}$, topic number $K$, seed word sets $S = \{S_1, S_1, \ldots, S_K\}$, initial KL annealing factor $\lambda_0$, hyperparameters $\lambda_1$, $\lambda_2$, $\lambda_3$, max iteration number $T$.

**for** $t$ from 1 to $T$ **do**
  randomly sample a batch of $B$ documents;
  $\mathcal{L}_{batch} \leftarrow 0$;
  $\lambda_0 \leftarrow \min(\lambda_0 + \frac{1}{T}, 1.0)$;
  compute $\beta_k$ for each topic $k$ by Eq.3;
  **for** each document $d$ in the batch **do**
    compute $\theta$ with encoder $F_d$;
    compute $\phi_n$ for each $w_n$ with encoder $F_w$;
    compute $\varphi_d = \{\varphi_1, ..., \varphi_n\}$ by Eq.10;
    $\mathcal{L}_{batch} \leftarrow \mathcal{L}_{batch} + \mathcal{L}_{tr}$ by Eq.13
  **end for**
  update model parameters with $\nabla\mathcal{L}_{batch}$
**end for**

## 5 Experiments

### 5.1 Datasets

We conduct our experiments on three datasets: **20 Newsgroups**, **UIUC Yahoo Answers**, and **DBPedia**. 20 Newsgroups (Lang, 1995) is a dataset that contains around 20,000 newsgroup documents and is commonly used in the topic modeling field. And to verify our model's scalability, we adopt two other larger datasets, the UIUC Yahoo Answers dataset (Chang et al., 2008) and DBPedia (Zhang et al., 2015), which contain 150,000 and 630,000 samples, respectively. We preprocess each dataset and split them for training and testing. The detailed procedure of preprocessing and the statistical summaries for each dataset can be viewed in Appendix B.

### 5.2 Seed Words Extraction

To avoid human biases, we follow (Jagarlamudi et al., 2012; Gallagher et al., 2017) and adopt an automatic approach to extract seed words. For each dataset, we set the topic number $K$ the same as its class number, and use Information Gain (IG) to identify the words having the highest mutual information with the class. Specifically, IG of a word $w$ in class $c$ is

$$IG(w, c) = H(c) - H(c|w), \tag{14}$$

where $H(c)$ is the entropy of class $c$ and $H(c|w)$ denotes the conditional entropy of $c$ given $w$. For each class, we choose the top $L$ words with the highest IG scores as seed words.

### 5.3 Evaluation of Topic Coherence

**Evaluation Metrics.** We use Normalized Pointwise Mutual Information (NPMI), to evaluate the coherence of learned topics. NPMI between words $w_i$ and $w_j$ is defined as:

$$NPMI(w_i, w_j) = \frac{\log \frac{p(w_i, w_j)}{p(w_i)p(w_j)}}{-\log p(w_i, w_j)}. \tag{15}$$

As we are dealing with topic models with seed words, we take the top $N$ non-seed words and predefined $L$ seed words for each topic and measure NPMI among the $N + L$ words. For unsupervised methods, we pick the top $N + L$ words. By considering both seed and non-seed words, the NPMI scores can measure how well the learned topics fit the predefined aspects of interests. Also, the score implicitly reflects topic diversity, as topics with a high coherence score with seed words are more likely to be diverse as long as their seed words are distinct. We report NPMI with $N = 10, L = 5$ on both train and test sets. Results with different seed word numbers can be viewed in Appendix C.

**Baselines.** We compare SeededNTM with the following baselines. For unsupervised topic models, we compare with LDA (Blei et al., 2003), which is a representative conventional neural topic models, and CombinedTM (Bianchi et al., 2021a), which enhances prodLDA (Srivastava and Sutton, 2017) with contextualized embeddings from BERT. For seed-guided topic models, we compare with SeededLDA (Jagarlamudi et al., 2012), STM (Li et al., 2016), Anchored CorEx (Gallagher et al., 2017), CatE (Meng et al., 2020a), keyATM (Eshima et al., 2020) and keyETM (Harandizadeh et al., 2022), which we have introduced in related works.

**Results.** The performances of topic coherence are reported in Table 1. As we can see, most seeded topic models achieve better topic coherence than unsupervised ones as the seed words provide additional semantic information. SeededNTM outperforms the baselines in most settings, demonstrating the effectiveness of our approach. Note that the advantages become more significant on the largest datasets, DBPedia, indicating its scalability when facing datasets of huge scale. We can find that keyETM sometimes performs worse performances than conventional methods like STM and

| Methods | 20 Newsgroups | | | | Yahoo Answer | | | | DBPedia | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | NPMI | | F1 | | NPMI | | F1 | | NPMI | | F1 | |
| | train | test | Macro | Micro | train | test | Macro | Micro | train | test | Macro | Micro |
| LDA | 0.279 | 0.248 | - | - | 0.183 | 0.160 | - | - | 0.064 | -0.023 | - | - |
| CombinedTM | 0.288 | 0.237 | - | - | 0.251 | 0.129 | - | - | 0.233 | 0.142 | - | - |
| Seeded LDA | 0.273 | 0.244 | 0.347 | 0.332 | 0.213 | 0.206 | 0.580 | 0.564 | 0.266 | 0.263 | 0.836 | 0.838 |
| STM | 0.345 | 0.307 | 0.494 | 0.520 | 0.272 | 0.254 | 0.582 | 0.588 | 0.311 | 0.298 | 0.862 | 0.865 |
| Anchor CorEx | 0.360 | 0.313 | 0.387 | 0.358 | 0.295 | 0.282 | 0.494 | 0.487 | 0.312 | 0.295 | 0.773 | 0.767 |
| CatE | 0.360 | 0.331 | 0.313 | 0.313 | 0.316 | 0.234 | 0.456 | 0.457 | 0.307 | 0.278 | 0.725 | 0.723 |
| keyATM | 0.303 | 0.282 | 0.328 | 0.307 | 0.177 | 0.175 | 0.609 | 0.593 | 0.279 | 0.274 | 0.842 | 0.843 |
| keyETM | 0.362 | 0.332 | 0.316 | 0.334 | 0.250 | 0.234 | 0.468 | 0.461 | 0.261 | 0.252 | 0.730 | 0.730 |
| SeededNTM | **0.368** | **0.335** | **0.567** | **0.575** | **0.332** | **0.298** | **0.628** | **0.626** | **0.328** | **0.314** | **0.913** | **0.915** |

Table 1: The NPMI and F1 scores on three datasets. Results are averaged over multiple runs with different random seeds. Standard deviations can be viewed in Appendix.

| Methods | NPMI | | F1 | |
|---|---|---|---|---|
| | train | test | Macro | Micro |
| SeededNTM | **0.368** | **0.335** | **0.567** | **0.575** |
| SeededNTM-noise | 0.360 | 0.324 | 0.561 | 0.567 |
| SeededNTM-NN | 0.359 | 0.326 | 0.566 | 0.571 |
| SeededNTM-w.o.doc | 0.275 | 0.208 | 0.486 | 0.507 |
| SeededNTM-w.o.word | 0.364 | 0.327 | 0.563 | 0.571 |
| SeededNTM-mean | 0.284 | 0.221 | 0.537 | 0.542 |

Table 2: Results of different variants of SeededNTM on 20 Newsgroups dataset.

keyATM, indicating the necessity to incorporate document level information. Anchor CorEx and CatE are strong baselines on some occasions, as Anchor CorEx has an information-theory-based objective similar to NPMI, and CatE takes the order of words as additional information when learning embeddings.

### 5.4 Evaluation of Classification

**Evaluation Metrics.** Except for evaluating coherence of learned topics, we evaluate how well the document-topic distribution is learned with a classification task. Here we take the maximum probability in the document topic distribution as the predicted label to test topic models' ability to extract semantic information from documents. We use Macro and Micro F1 scores as the evaluation metrics. As most baselines cannot predict on new data, we report the results on the train set and take the test set for validation.

**Baselines.** We compare SeededNTM on classification with the aforementioned baselines except for the unsupervised ones. Specifically, we follow CatE's original paper and use WeSTClass model (Meng et al., 2018) to classify its outputs.

**Results.** Table 1 summarizes the F1 scores on three datasets. SeededNTM outperforms other baseline models on most occasions, indicating our model can understand the semantics of the documents and learn more reliable and helpful topic distributions for each document. Among the baselines methods, seededNTM, STM, and keyATM achieve better performances on three datasets, as they incorporate information from seed words on both levels.

### 5.5 Ablation Studies

We analyze the effects of different modules of SeededNTM by comparing among the following variants: 1) SeededNTM-noise: SeededNTM without the consistency regularizer, 2) SeededNTM-NN: SeededNTM without the consistency regularizer and with a neighbor-based noise-reduction method as in (Li et al., 2018). 3) SeededNTM-w.o.doc: SeededNTM without document encoder, 4) SeededNTM-w.o.word: SeededNTM without word encoder, 5) SeededNTM-mean: SeededNTM with the mean-field assumption as in (Rezaee and Ferraro, 2020).

Results are provided in Table 2, from which we can draw the following conclusions. The effectiveness of the noise-reduction method can be proved by the comparisons between variants with and without noise regularizer. Both SeededNTM-NN and original SeededNTM outperform SeededNTM-noise. And the effectiveness of our intra-sample consistency regularizer can be further demonstrated by the improvements of SeededNTM over SeededNTM-NN. The decreases in SeededNTM-w.o.doc and SeededNTM-w.o.word indicate the importance of information on both document and word levels. Moreover, the decay on SeededNTM-mean proves the effectiveness of our proposed as-

| | Seeded LDA | STM | Anchor CorEx | CatE | keyATM | keyETM | SeededNTM |
|---|---|---|---|---|---|---|---|
| Intrusion | 0.381 | 0.348 | 0.719 | 0.695 | 0.143 | 0.576 | **0.762** |
| MACC | 0.469 | 0.475 | 0.361 | 0.423 | 0.473 | 0.472 | **0.504** |

Table 3: Human evaluation results on word intrusion task and MACC of different models on UIUC Yahoo Dataset.

| | Topic 1: Game&Recreation | Topic2: Arts | Topic3: Pregnancy&Parenting |
|---|---|---|---|
| Seed words | pokemon, game, diamond, games, trade | book, harry, potter, books, poem | pregnancy, baby, weeks, child, pregnant |
| Seeded LDA | play, ps, wii, level, code | read, know, names, love, movie | just period time days day |
| STM | ps, wii, level, code, xbox | read, story, write, series, movie | period, doctor, sex, months, normal |
| Anchor CorEx | play, pearl, playing, fc, ps | read, write, reading, writing, author | months, period, days, week, birth |
| CatE | gba, ds, nintendo, replay, mew | rowling, hallows, novel, author, deathly | trimester, babies, conception, expecting, womb |
| KeyATM | play, ps, just, need, wii | read, know, just, good, think | just, know, time, period, day |
| KeyETM | know, think, good, really, want | question, answer, read, come, called | year, years, old, months,feel |
| SeededNTM | fc, wii, nintendo, ds, pearl | hallows, deathly, author, rowling, novel | ovulation, period, ttc, ovulating, pill |

Table 4: Top five words of part of the topics and corresponding seed words learned by different models on UIUC Yahoo Answers dataset.

sumption and the necessity to balance context document information when modeling per-word topic assignments.

### 5.6 Human Evaluation

Apart from automated evaluation metrics, we hope to further demonstrate our model's ability to discover semantically meaningful topics through the judgements from human, as automated metrics can be sometimes biased (Hoyle et al., 2021).

**Metrics**: We adopt two human evaluation metrics: accuracy in the word intrusion task (Chang et al., 2009) and MACC score (Meng et al., 2020a). In the word intrusion task, participants are given the top words of a certain topic and an intruding word from another topic, and are asked to identify the intruding term. For MACC, participants need to classify whether the top words are consistent with their corresponding seed word set, i.e.

$$\mathrm{MACC} = \frac{1}{K}\sum_{k=1}^{K}\frac{1}{N}\sum_{i=1}^{N}\mathbb{1}(w_{ki} \in S_k) \quad (16)$$

where $\mathbb{1}(w_{ki} \in S_k)$ indicates whether word $w_{ki}$ belongs to the topic with seed word set $S_k$. We conduct human evaluation on UIUC Yahoo Answer dataset, and take the top 5 words from each topic for evaluation. For each metric, we invite 10 graduate students to independently fulfil the task, and the participants in two groups do not overlap to avoid information leak. More details can be viewed in Appendix C. The results are shown in Table 3. We can find that SeededNTM achieves best results on both metrics, which further demonstrates the quality of our learned topics.

**Case Study:** Here we present part of topics learned by SeededNTM on Yahoo Answer dataset along with topics learned by baselines methods using the same seed words in the aforementioned experiments in Table 4. We can find that some baselines like Anchor CorEx, keyATM, and KeyETM, tend to put high weights on several commonly used words like 'play', 'great', 'good', while SeededNTM tends to pay attention to words that are more specific such as 'nintendo', a Japanese multinational video game company who releases the game 'Pokemon', and 'rowling', the author of Harry Potter, and 'ttc', meaning 'trying to conceive'.

Besides presentations of topics, we conduct more other qualitative experiments under different settings to verify the generalization ability of our model. Please refer to Appendix C.3 for more information.

## 6 Conclusions

In this paper, we propose *SeededNTM* to improve topic interpretability together with scalability. We leverage supervisions from seed words on both word and document levels and propose a context-dependency assumption. An auto-adaptation mechanism is designed to balance word and context document information. Moreover, we propose an intra-sample consistency regularizer to deal with noisy document level supervisions. Perturbation consistency and semantic consistency are encouraged to improve the model's robustness to noises. Through quantitative and qualitative experiments on three datasets, we demonstrate that SeededNTM can derive semantically meaningful topics and outperforms state-of-the-art baselines.

## References


David Andrzejewski and Xiaojin Zhu. 2009. Latent dirichlet allocation with topic-in-set knowledge. In *Proceedings of the NAACL HLT 2009 Workshop on Semi-Supervised Learning for Natural Language Processing*, pages 43–48.

Federico Bianchi, Silvia Terragni, and Dirk Hovy. 2021a. Pre-training is a hot topic: Contextualized document embeddings improve topic coherence. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 759–766.

Federico Bianchi, Silvia Terragni, Dirk Hovy, Debora Nozza, and Elisabetta Fersini. 2021b. Cross-lingual contextualized topic models with zero-shot learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1676–1683.

David Blei and Jon Mcauliffe. 2008. Supervised topic models. In *Advances in Neural Information Processing Systems*, volume 20.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022.

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*, pages 2031–2040.

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.

Ming-Wei Chang, Lev-Arie Ratinov, Dan Roth, and Vivek Srikumar. 2008. Importance of semantic representation: Dataless classification. In *AAAI*, volume 2, pages 830–835.

Xingyuan Chen, Yunqing Xia, Peng Jin, and John Carroll. 2015. Dataless text classification with descriptive lda. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29.

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.

Jacob Eisenstein, Amr Ahmed, and Eric P Xing. 2011. Sparse additive generative models of text. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 1041–1048. Citeseer.

Erik Englesson and Hossein Azizpour. 2021. Generalized jensen-shannon divergence loss for learning with noisy labels. *Advances in Neural Information Processing Systems*, 34.

Shusei Eshima, Kosuke Imai, and Tomoya Sasaki. 2020. Keyword assisted topic models. *arXiv preprint arXiv:2004.05964*.

Ryan J Gallagher, Kyle Reing, David Kale, and Greg Ver Steeg. 2017. Anchored correlation explanation: Topic modeling with minimal domain knowledge. *Transactions of the Association for Computational Linguistics*, 5:529–542.

Bahareh Harandizadeh, J. Hunter Priniski, and Fred Morstatter. 2022. Keyword assisted embedded topic model. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, WSDM '22, page 372–380.

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

Alexander Hoyle, Pranav Goel, Andrew Hian-Cheong, Denis Peskov, Jordan Boyd-Graber, and Philip Resnik. 2021. Is automated topic model evaluation broken? the incoherence of coherence. *Advances in Neural Information Processing Systems*, 34:2018–2033.

Jagadeesh Jagarlamudi, Hal Daumé III, and Raghavendra Udupa. 2012. Incorporating lexical priors into topic models. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 204–213.

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 *Machine Learning Proceedings 1995*, pages 331–339. Elsevier.

Chenliang Li, Jian Xing, Aixin Sun, and Zongyang Ma. 2016. Effective document labeling with very few seed words: A topic model approach. In *Proceedings of the 25th ACM international on conference on information and knowledge management*, pages 85–94.

Junnan Li, Richard Socher, and Steven C.H. Hoi. 2020. Dividemix: Learning with noisy labels as semi-supervised learning. In *International Conference on Learning Representations*.

Ximing Li, Changchun Li, Jinjin Chi, Jihong Ouyang, and Chenliang Li. 2018. Dataless text classification: A topic modeling approach with document manifold. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 973–982.

Yu Meng, Jiaxin Huang, Guangyuan Wang, Zihan Wang, Chao Zhang, Yu Zhang, and Jiawei Han. 2020a. Discriminative topic mining via category-name guided text embedding. In *Proceedings of The Web Conference 2020*, pages 2121–2132.

Yu Meng, Jiaming Shen, Chao Zhang, and Jiawei Han. 2018. Weakly-supervised neural text classification. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 983–992.

Yu Meng, Yunyi Zhang, Jiaxin Huang, Chenyan Xiong, Heng Ji, Chao Zhang, and Jiawei Han. 2020b. Text classification using label names only: A language model self-training approach. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9006–9017.

Yishu Miao, Lei Yu, and Phil Blunsom. 2016. Neural variational inference for text processing. In *International conference on machine learning*, pages 1727–1736. PMLR.

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.

David Newman, Edwin V Bonilla, and Wray Buntine. 2011. Improving topic coherence with regularized topic models. *Advances in neural information processing systems*, 24.

Thong Nguyen and Anh Tuan Luu. 2021. Contrastive learning for neural topic model. *Advances in Neural Information Processing Systems*, 34:11974–11986.

Mehdi Rezaee and Francis Ferraro. 2020. A discrete variational recurrent topic model without the reparametrization trick. In *Advances in Neural Information Processing Systems*, volume 33, pages 13831–13843. Curran Associates, Inc.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. 2014. Stochastic backpropagation and approximate inference in deep generative models. In *International conference on machine learning*, pages 1278–1286. PMLR.

Akash Srivastava and Charles Sutton. 2017. Autoencoding variational inference for topic models. In *5th International Conference on Learning Representations*.

Xinyi Wang and Yi Yang. 2020. Neural topic model with attention for supervised learning. In *International Conference on Artificial Intelligence and Statistics*, pages 1147–1156. PMLR.

Zihan Wang, Dheeraj Mekala, and Jingbo Shang. 2021. X-class: Text classification with extremely weak supervision. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3043–3053.

Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. 2020. Unsupervised data augmentation for consistency training. *Advances in Neural Information Processing Systems*, 33:6256–6268.

Lu Zhang, Jiandong Ding, Yi Xu, Yingyao Liu, and Shuigeng Zhou. 2021. Weakly-supervised text classification based on keyword graph. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2803–2813.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28:649–657.

Yu Zhang, Yu Meng, Xuan Wang, Sheng Wang, and Jiawei Han. 2022. Seed-guided topic discovery with out-of-vocabulary seeds. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 279–290.

## A Derivation of ELBO-based Loss

The Evidence Lower Bound (ELBO) for our model is

$$ELBO(\boldsymbol{w}) = E_{q(\theta,\boldsymbol{z}|\boldsymbol{w})} \log p(\boldsymbol{w}|\theta,\boldsymbol{z};\beta) - E_{q(\theta,\boldsymbol{z}|\boldsymbol{w})} \log \left( \frac{q(\theta,\boldsymbol{z}|\boldsymbol{w})}{p(\theta,\boldsymbol{z})} \right). \tag{A.1}$$

To maxmize the ELBO, we minimize its opposite number as training loss, which is

$$\mathcal{L}_{elbo} = -E_{q(\theta,\boldsymbol{z}|\boldsymbol{w})} \log p(\boldsymbol{w}|\theta,\boldsymbol{z};\beta) + E_{q(\theta,\boldsymbol{z}|\boldsymbol{w})} \log \left( \frac{q(\theta,\boldsymbol{z}|\boldsymbol{w})}{p(\theta,\boldsymbol{z})} \right). \tag{A.2}$$

And we denote

$$\begin{aligned} \mathcal{L}_{rec} &= -E_{q(\theta,\boldsymbol{z}|\boldsymbol{w})} \log p(\boldsymbol{w}|\theta,\boldsymbol{z};\beta), \\ \mathcal{L}_{kl} &= E_{q(\theta,\boldsymbol{z}|\boldsymbol{w})} \log \left( \frac{q(\theta,\boldsymbol{z}|\boldsymbol{w})}{p(\theta,\boldsymbol{z})} \right), \\ \mathcal{L}_{elbo} &= \mathcal{L}_{rec} + \mathcal{L}_{kl}. \end{aligned} \tag{A.3}$$

For the posterior $q(\theta,\boldsymbol{z}|\boldsymbol{w})$, we have

$$q(\theta,\boldsymbol{z}|\boldsymbol{w}) = q(\theta|\boldsymbol{w}) \prod_n q(z_n|\theta,w_n). \tag{A.4}$$

For $p(\boldsymbol{w}|\theta,\boldsymbol{z};\beta)$, we have

$$p(\boldsymbol{w}|\theta,\boldsymbol{z};\beta) = \prod_n p(w_n|z_n;\beta). \tag{A.5}$$

So for $\mathcal{L}_{rec}$ we have

$$\begin{aligned} \mathcal{L}_{rec} &= -E_{q(\theta,\boldsymbol{z}|\boldsymbol{w})} \log p(\boldsymbol{w}|\theta,\boldsymbol{z};\beta) \\ &= -E_{q(\theta|\boldsymbol{w})} E_{q(z_1|\theta,w_1)} \dots E_{q(z_N|\theta,w_N)} \log p(\boldsymbol{w}|\theta,\boldsymbol{z};\beta) \\ &= -E_{q(\theta|\boldsymbol{w})} \sum_n E_{q(z_n|\theta,w_n)} \log p(w_n|z_n;\beta). \end{aligned} \tag{A.6}$$

The expectation $E_{q(\theta|\boldsymbol{w})}$ can be estimated using a sample-based method by sampling $\theta \sim q(\theta|\boldsymbol{w})$, and given $\theta$, $\varphi_{nk} = q(z_n = k|\theta, w_n)$ can be computed with Eq.10. So we have

$$\mathcal{L}_{rec} \approx -\sum_{n,k} \varphi_{nk} \log \beta_{k w_n}. \tag{A.7}$$

For $\mathcal{L}_{kl}$ we have

$$\begin{aligned} \mathcal{L}_{kl} &= E_{q(\theta,\boldsymbol{z}|\boldsymbol{w})} \log \left( \frac{q(\theta,\boldsymbol{z}|\boldsymbol{w})}{p(\theta,\boldsymbol{z})} \right) \\ &= E_{q(\theta|\boldsymbol{w})} \log \left( \frac{q(\theta|\boldsymbol{w})}{p(\theta)} \right) + E_{q(\theta|\boldsymbol{w})} \sum_n E_{q(z_n|\theta,w_n)} \log \left( \frac{q(z_n|\theta,w_n)}{p(z_n|\theta)} \right) \\ &= KL\left(q(\theta|\boldsymbol{w}) \| p(\theta)\right) + E_{q(\theta|\boldsymbol{w})} \sum_n KL\left(q(z_n|\theta,w_n) \| p(z_n|\theta)\right). \end{aligned} \tag{A.8}$$

The former term can be approximated using Laplace approximation to the Dirichlet prior, and can be calculated in closed form as $KL\left(\mathcal{N}(\mu,\sigma^2) \| \mathcal{N}(\mu_0,\sigma_0^2)\right)$ (Srivastava and Sutton, 2017). And the latter term can be estimated by Monte Carlo sampling with $\theta \sim q(\theta|\boldsymbol{w})$:

$$E_{q(\theta|\boldsymbol{w})} \sum_n KL\left(q(z_n|\theta,w_n) \| p(z_n|\theta)\right) \approx \sum_n KL(\varphi_n \| \theta). \tag{A.9}$$

# B More Details of Datasets

## B.1 Dataset Descriptions

Three datasets are used in out experiments: **20 Newsgroups**, **UIUC Yahoo Answers**, and **DBPedia**. 20 Newsgroups (Lang, 1995) is a collection of newsgroup documents containing 11,000 train samples and 7,000 test samples in 20 classes. It is a common dataset that is widely used in topic modeling field. The UIUC Yahoo Answers dataset (Chang et al., 2008) contains 150,000 question-answer pairs belonging to 15 categories. It is a classification dataset and is used in topic models in (Card et al., 2018). DBPedia (Zhang et al., 2015) is extracted from Wikipedia and contains 560,000 train samples and 70,000 test samples belonging to 14 ontology classes. DBPedia is a classification dataset, and to the best of our knowledge, it is the first time that DBPedia has been used for topic modeling, but similar datasets (though much smaller) from Wikipedia have been adopted to test topic models (Nguyen and Luu, 2021).

## B.2 Preprocess Procedures for Datasets

We preprocess documents in each dataset by tokenizing, filtering out stop words, words with document frequency above 70%, and words appearing in less than around 100 documents (depending on the dataset). The final vocabulary sizes for each dataset after preprocessing vary from 2,000 to 20,000. Then we remove the documents shorter than two words.

Specifically, for the UIUC Yahoo Answer dataset, we follow the approach used in (Card et al., 2018), and drop the *Cars and Transportation* and *Social Science* classes and merge *Arts* and *Arts and Humanities* into one class, producing 15 categories, each with 10,000 documents.

As for the augmentation functions $\mathcal{A}$, we use the word level augmentation method proposed in (Xie et al., 2020) by randomly replacing words with lower tf-idf scores. Around of 10% words are replaced in our experiments.

## B.3 Statistics of Datasets

We summarize the statistics for the three datasets after preporcessing in Table.B.1

Table B.1: Summary of the statistics of three datasets

| | 20 Newsgroups | Yahoo Answer | DBPedia |
|---|---|---|---|
| Class Number | 20 | 15 | 14 |
| Vocabulary Size | 2,004 | 7,468 | 19,975 |
| Train Set Size | 10,732 | 119,747 | 559,710 |
| Test Set Size | 7,105 | 29,937 | 69,962 |
| Avg Doc Length | 44.308 | 46.089 | 22.730 |
| Token Number | 790,324 | 6,898,796 | 13,682,938 |

# C More Experimental Details

## C.1 Implementation Datails

As for the training environment, we implement our method based on **PyTorch** 1.6.0 with Python 3.7.9 and perform our experiments on 4 GeForce RTX 2080Ti. For model structure, the dimension for our word embedding layer is 300, and the dimension for the hidden layer in the document encoder is 256. We use a 0.2 dropout rate in our encoder during training. We present our choices for hyperparameters in Table.C.1. Hyperparameters are determined by grid search on the smallest dataset, 20 Newsgroups, and fine-tuned on other two large datasets. The final hyperparameters are shown in Table C.1.

## C.2 Baselines

We give detailed descriptions of our baselines here.

| | LR | batch size | $\lambda_1$ | $\lambda_2$ | $\lambda_3$ | $\tau$ | $\kappa$ |
|---|---|---|---|---|---|---|---|
| 20 Newsgroups | 0.001 | 64 | 2.0 | 10.0 | 10.0 | 4.0 | 3.0 |
| Yahoo Answer | 0.001 | 128 | 2.0 | 10.0 | 5.0 | 4.0 | 3.0 |
| DBPedia | 0.0005 | 256 | 2.0 | 10.0 | 1.0 | 4.0 | 3.0 |

Table C.1: The choices of hyperparameters for each dataset.

- **LDA** (Blei et al., 2003): LDA is one of the most popular unsupervised conventional topic models that deduce posterior distribution via Gibbs sampling or variational inference.
- **CombinedTM**: CombinedTM enhances topic model with contextualized embeddings from pre-trained language model to improve model's semantic expression ability and leads to more coherent topics. CombinedTM is an extension of prodLDA (Srivastava and Sutton, 2017), one of the most representative neural topic models. It uses black-box neural variational inference and optimizes the model with stochastic gradient descent, increasing the model's scalability.
- **SeededLDA** (Jagarlamudi et al., 2012): SeededLDA pairs each regular topic with a topic containing only seed words and biases documents' topic preferences in Gibbs sampling if they contain seed words.
- **STM** (Li et al., 2016): STM is a topic model-based weakly-supervised text classification method that incorporates both document and word level supervisions to improve classification accuracies.
- **Anchored CorEx** (Gallagher et al., 2017): Anchored CorEx is based on an information-theoretic framework and tries to derive maximally informative topics based on seed words.
- **CatE** (Meng et al., 2020a): CatE aims at deriving topics with a single seed word for each topic. It uses a word embedding method and tries to learn a discriminative embedding space for both topics and words.
- **keyATM** (Eshima et al., 2020): keyATM improves upon SeededLDA by allowing some seed-word-free topics.
- **keyETM** (Harandizadeh et al., 2022): keyETM incorporates seed words into NTM by regularizing word-topic and topic-word distributions on word level with seed words and pre-trained word embeddings.

### C.3 More Results on NPMI and F1 score

| Methods | 20 Newsgroups | | | | Yahoo Answer | | | | DBPedia | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | NPMI | | F1 | | NPMI | | F1 | | NPMI | | F1 | |
| | train | test | Macro | Micro | train | test | Macro | Micro | train | test | Macro | Micro |
| LDA | 0.006 | 0.008 | - | - | 0.003 | 0.011 | - | - | 0.018 | 0.032 | - | - |
| CombinedTM | 0.012 | 0.011 | - | - | 0.002 | 0.003 | - | - | 0.014 | 0.019 | - | - |
| Seeded LDA | 0.000 | 0.000 | 0.001 | 0.003 | 0.002 | 0.002 | 0.001 | 0.001 | 0.000 | 0.001 | 0.001 | 0.001 |
| STM | 0.002 | 0.004 | 0.006 | 0.006 | 0.020 | 0.022 | 0.028 | 0.032 | 0.003 | 0.004 | 0.003 | 0.014 |
| Anchor CorEx | 0.000 | 0.000 | 0.000 | 0.001 | 0.000 | 0.000 | 0.001 | 0.001 | 0.001 | 0.002 | 0.003 | 0.004 |
| CatE | 0.003 | 0.002 | 0.006 | 0.005 | 0.002 | 0.002 | 0.003 | 0.004 | 0.001 | 0.001 | 0.007 | 0.006 |
| keyATM | 0.002 | 0.002 | 0.011 | 0.013 | 0.001 | 0.005 | 0.002 | 0.002 | 0.001 | 0.001 | 0.001 | 0.001 |
| keyETM | 0.002 | 0.003 | 0.006 | 0.005 | 0.001 | 0.003 | 0.016 | 0.027 | 0.010 | 0.009 | 0.026 | 0.024 |
| SeededNTM | 0.002 | 0.003 | 0.002 | 0.002 | 0.001 | 0.007 | 0.002 | 0.002 | 0.005 | 0.007 | 0.010 | 0.010 |

Table C.2: Standard deviations of the results in Table 1.

| Methods | 20 Newsgroups | | | | Yahoo Answer | | | | DBPedia | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | NPMI | | F1 | | NPMI | | F1 | | NPMI | | F1 | |
| | train | test | Macro | Micro | train | test | Macro | Micro | train | test | Macro | Micro |
| LDA | 0.287 | 0.258 | - | - | 0.192 | 0.173 | - | - | 0.069 | -0.007 | - | - |
| CombinedTM | 0.289 | 0.242 | - | - | 0.257 | 0.228 | - | - | 0.237 | 0.145 | - | - |
| Seeded LDA | 0.295 | 0.276 | 0.335 | 0.334 | 0.197 | 0.192 | 0.583 | 0.561 | 0.276 | 0.267 | 0.822 | 0.825 |
| STM | 0.355 | 0.331 | 0.493 | 0.511 | 0.299 | 0.290 | 0.594 | 0.605 | 0.311 | 0.302 | 0.869 | 0.874 |
| Anchor CorEx | 0.353 | 0.315 | 0.383 | 0.344 | 0.309 | 0.300 | 0.454 | 0.446 | 0.313 | 0.296 | 0.773 | 0.767 |
| CatE | 0.375 | **0.353** | 0.339 | 0.330 | 0.351 | 0.296 | 0.423 | 0.428 | 0.333 | 0.316 | 0.703 | 0.700 |
| keyATM | 0.297 | 0.270 | 0.313 | 0.308 | 0.172 | 0.170 | 0.604 | 0.586 | 0.275 | 0.267 | 0.815 | 0.814 |
| keyETM | 0.368 | 0.338 | 0.325 | 0.339 | 0.243 | 0.232 | 0.415 | 0.428 | 0.250 | 0.242 | 0.607 | 0.609 |
| SeededNTM | **0.381** | 0.335 | **0.558** | **0.565** | **0.357** | **0.309** | **0.611** | **0.610** | **0.338** | **0.330** | **0.901** | **0.901** |

Table C.3: The NPMI and F1 scores on three datasets when $N$=10,$L$=3

### C.4 Comparisons with Weakly-Supervised Classification Methods

We compare SeededNTM with two recent weakly-supervised classification methods, **XClass** (Wang et al., 2021) and **ClassKG** (Zhang et al., 2021). Both models take seed words/class names as label information and bootstrap seed word lists in some ranked orders, which could be viewed as topics. For fair comparison, we replace the BERT-based document classifier in both methods with a multi-layer MLP classifier same as SeededNTM. The results are shown in Table C.4. We can find that SeededNTM outperforms both methods on NPMI and F1 score, which may indicates that the unsupervised losses in NTM might help the model make better use of the unlabeled or noisy-labeled documents.

| | 20Newsgroups | | | | Yahoo Answer | | | | DBpedia | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | NPMI | | F1 | | NPMI | | F1 | | NPMI | | F1 | |
| | train | test | macro | micro | train | test | macro | micro | train | test | macro | micro |
| XClass | 0.297 | 0.279 | 0.434 | 0.402 | 0.264 | 0.204 | 0.511 | 0.456 | 0.248 | 0.139 | 0.810 | 0.791 |
| ClassKG | 0.334 | 0.281 | 0.509 | 0.512 | 0.233 | 0.147 | 0.613 | 0.611 | 0.244 | 0.188 | 0.843 | 0.844 |
| Seeded NTM | **0.368** | **0.335** | **0.567** | **0.575** | **0.332** | **0.296** | **0.629** | **0.626** | **0.328** | **0.314** | **0.913** | **0.915** |

Table C.4: Comparisons of the NPMI and F1 scores on three datasets between weak-supervised classification methods and SeededNTM.

### C.5 Class Names as Seed Words

Following the setting of weakly-supervised classification, we take the class names (one or two words for each class) as seed wordsand run the experiments on Yahoo Answer dataset. Results are shown in Table C.5. We can draw similar conclusions from these results as in the main paper.

| | NPMI | | F1 | |
|---|---|---|---|---|
| | train | test | macro | micro |
| Seeded LDA | 0.176 | 0.173 | 0.415 | 0.400 |
| STM | 0.263 | 0.250 | 0.574 | 0.584 |
| Anchor CorEx | 0.267 | 0.246 | 0.403 | 0.401 |
| CatE | 0.324 | 0.217 | 0.260 | 0.277 |
| keyATM | 0.156 | 0.156 | 0.406 | 0.406 |
| keyETM | 0.160 | 0.152 | 0.195 | 0.280 |
| XClass | 0.270 | 0.184 | 0.480 | 0.446 |
| ClassKG | 0.255 | 0.163 | 0.546 | 0.545 |
| Seeded NTM | **0.332** | **0.298** | **0.599** | **0.603** |

Table C.5: NPMI and F1 scores on Yahoo Answer dataset with seed words derived from class names.

### C.6 Details of Human Evaluation

To perform human evaluations on topic quality, we invite 20 graduate students majoring in computer science, who participate the experiments for course credits. We divide the participants into two non-overlapping groups for word intrusion and MACC respectively to avoid information leak. All participants are told that they are performing human evaluations for an automatic method and none of their privacy information are collected during the experiments. The surveys used in human evaluation experiments are shown in Figure C.1.

### C.7 More Qualitative Results

Here we present some more qualitative results under different settings which are not included in the main paper due to the space limit.

#### C.7.1 Topic with Incomplete Seed Words

In the experiments in the main paper, seed words are assumed to be complete and accurately represent latent topics in the corpus. However, in practical situations, users may only be interested in part of the corpus or have little prior knowledge, leading to incomplete seed words. To simulate such situations, we preserve seed words for only three topics and leave other topics unsupervised. We present the results of SeededNTM along with the two latest baselines, keyATM and keyETM in Table C.6.

For three supervised topics, SeededNTM can discover words related to the seed words as it does under complete seed words, while KeyATM and keyETM produce semantically incoherent topics, such as irrelevant words "god" and "world" appearing in the topic 'Education&Reference' from keyETM. SeededNTM can also discover meaningful unsupervised topics similar to the seeded topics in former experiments, such as 'Pets' and 'Computer&Internet', while keyATM and keyETM find incoherent or unrelated topics. Moreover, new topics which are not included in the original seed word sets can also be discovered by SeededNTM, such as 'Craigslist', a famous American classified advertisements website.

| Topics | keyATM | KeyETM | SeededNTM |
|---|---|---|---|
| Business&Finance | need, want, work, time, business | phone, card, business, download, video | loan, bank, tax, payment, income |
| Health | just, know, day, time, good | water, hair, product, cup, add | pregnancy, pregnant, pill, ovulation, period |
| Education | school, college, know, just, work | god, book, books, world, classes | colleges, classes, degree, gpa, schools |
| Pets | dog, just, dogs, know, cat | old, wear, house, clean, big | puppy, kitten, puppies, breed, litter |
| Computer&Internet | just, need, want, download, know | - | wireless, router, vista, phones, cable |
| New Topic | - | time, long, way, probably, usually | craigslist, ebay, google, shops, sites |

Table C.6: Top five words learned on UIUC Yahoo Answers dataset while only 3 topics are with seed words.

#### C.7.2 Noisy Seed Words

The seed word set may contain irrelevant words in real-world practice due to users' mistakes or unfamiliarity with the corpus. To simulate such situations, we manually intrude irrelevant words from other topics into the seed words. The results are shown in Table C.7, from which SeededNTM can still find meaningful topics when there are noisy intrusions in the seed words, while keyATM and keyETM provide topics that are less explicit and coherent.

| Topics | noisy word | keyATM | KeyETM | SeededNTM |
|---|---|---|---|---|
| Society&Culture | company | people, just, think, life, believe | life, believe, world, man, word | christian, religious, beliefs, faith,christianity |
| Sports | phones | think, good, year, game, best | game, pokemon, play, points, level | baseball, league, win, fans, nfl |
| Beauty&Style | cat | product, look, color, just, want, | product, cute, black, color, clothes | jpg, shoes, hollister, shirt, curly |

Table C.7: The top five words of topics learned on UIUC Yahoo Answers dataset with noisy seed words.

#### C.7.3 Transferred Seed Words

One way to explore an unfamiliar dataset is to start with topics from another known corpus. In this experiment, we transfer the topical seed words from 20 Newsgroups and DBPedia and use them for training SeededNTM on UIUC Yahoo Answers dataset. Topics learned with the transferred seed words are presented in Table C.8, along with the topics learned in the original topics. We can find that though these datasets are collected from entirely different sources, some semantically meaningful topics can still be discovered with transferred seed words, and some lead to slightly different concepts from the originals. Moreover, the results indicates that topic-wise supervisions are flexible and bear less bias than sample-wise supervisions.

| Seed Words | 20News | Yahoo |
|---|---|---|
| god, atheists, religion | belief, religions, existence | belief, religious, christians |
| graphics, format, image | files, ftp, screen | picture, jpg, albums |
| space, launch, orbit | moon, solar,flight | paint, walls, room |
| | DBPedia | Yahoo |
| football, league, played | player, professional, team | qb, wr, rb |
| high, school, students | schools, secondary, grades | degree, college, university |
| species, family, flowering | endemic, native, habitat | plant, soil, flowers |

Table C.8: The top words of topics learned with transferred seed words from 20 Newsgroups and DBPedia.

### C.7.4 Exploration on the various aspects of single concept

Due to the ambiguity of natural language, a single word or concept may relate to various topics with different meanings, especially for some common words such as 'apple', 'doctor' or 'card'. In this case, we assume that the users aim at using topic models to understand different topics in the corpus related to a single word. We start with a single word, 'card'. We set only one topic with a single seed word 'card' and leave other topics unsupervised. Then we use the topic model to generate one supervised topic about 'card' and several unsupervised topics. Iteratively, we treat the most related word in the topic 'card' as the seed word for a new topic and train another topic model under new settings. The results are shown in Table C.9. Due to space limitations, we only list the topic 'card' in round 4 and round 5. From the results, SeededNTM shows its ability to distinguish different semantic topics related to the same word, which can be used to assist users with understanding complex concepts.

| Round | seed words | SeededNTM |
|---|---|---|
| 1 | card | phone, phones, cell, cards, sim, mobile |
| 2 | card | itunes, ipods, vista, router, dvd, xp |
| | phone | phones, cell, verizon, mobile, cingular,motorola |
| 3 | card | credit, money, pay, loan, bank, cards |
| | phone | phones, know, cell, cards, mobile, verizon |
| | itunes | ipod, download, windows, songs, music, files |
| 4 | card | camera, cards, digital, memory, laptop, graphics |
| 5 | card | wii, grphics, cards, memory, dell, ram |

Table C.9: The top five words of topics learned on UIUC Yahoo Answer dataset with iteratively-given seed words.

## D Limitations and Potential Risks of SeededNTM

Though SeededNTM achieves good performances in our experiments, there are still some limitations. Firstly, supervisions from seed words, though flexible, are also very weak and vulnerable to noises. Though we introduce some ways to improve the model's robustness, it is still possible that the model may crash under intentional attacks. Secondly, seed words in our model are used as pseudo supervisions. A more elegant way is to incorporate it into the generative story. As for potential risks, seeded topic models can be used to trace a specific topic, so it is possible that it's used to track someone's information from texts collected from the internet, violating personal privacy.

## Human Evaluation————Word Intrusion

This survey asks you to look at lists of six words produced by an automatic computer program. For each list, you'll be answering the question: "Which word does not belong?" For each question, click the words whose meaning or usage is most unlike that of the other words. If you feel that multiple words do not belong, choose the one that you feel is most out of place. This study should take approximately 10–15 minutes to complete. Your response will be completely anonymous.

* **01** Which word does not belong to the current list of words?

- ○ chihuahua
- ○ pup
- ○ old
- ○ bunny
- ○ puppies
- ○ superstar

(a) Survey for word intrusion task

## Human Evaluation————MACC Score

This survey asks you to look at lists of five words produced by an automatic computer program. For each list, you'll be answering the question: "Which of the following words belong to the topic of the current keywords?" For each question, click the words whose meaning or usage are similar with that of the keywords. If you feel that all words are not similiar with the keywords, click the “none” button at the bottom. This study should take approximately 10–15 minutes to complete. Your response will be completely anonymous.

* **01** Which of the following words belong to the topic of the current keywords (pay, money, credit, job, company) ?

- ☐ business
- ☐ bills
- ☐ paying
- ☐ bank
- ☐ card
- ☐ none

(b) Survey for MACC score

Figure C.1: Examples of the survey we used in human evaluation.