Seed-Guided Topic Discovery with Out-of-Vocabulary Seeds

Yu Zhang[1], **Yu Meng**[1], **Xuan Wang**[1], **Sheng Wang**[2], **Jiawei Han**[1]
[1]University of Illinois at Urbana-Champaign, IL, USA
[2]University of Washington, Seattle, WA, USA
{yuz9,yumeng5,xwang174,hanj}@illinois.edu swang@cs.washington.edu

Abstract

Discovering latent topics from text corpora has been studied for decades. Many existing topic models adopt a fully unsupervised setting, and their discovered topics may not cater to users' particular interests due to their inability of leveraging user guidance. Although there exist seed-guided topic discovery approaches that leverage user-provided seeds to discover topic-representative terms, they are less concerned with two factors: (1) the existence of out-of-vocabulary seeds and (2) the power of pre-trained language models (PLMs). In this paper, we generalize the task of seed-guided topic discovery to allow out-of-vocabulary seeds. We propose a novel framework, named SEETOPIC, wherein the general knowledge of PLMs and the local semantics learned from the input corpus can mutually benefit each other. Experiments on three real datasets from different domains demonstrate the effectiveness of SEETOPIC in terms of topic coherence, accuracy, and diversity.[1]

1 Introduction

Automatically discovering informative and coherent topics from massive text corpora is central to text analysis through helping users efficiently digest a large collection of documents (Griffiths and Steyvers, 2004) and advancing downstream applications such as summarization (Wang et al., 2009, 2022), classification (Chen et al., 2015; Meng et al., 2020b), and generation (Liu et al., 2021).

Unsupervised topic models have been the mainstream approach to topic discovery since the proposal of pLSA (Hofmann, 1999) and LDA (Blei et al., 2003). Despite their encouraging performance in finding informative latent topics, these topics may not reflect user preferences well, mainly due to their unsupervised nature. For example, given a collection of product reviews, a user may be specifically interested in product categories (e.g., "*books*", "*electronics*"), but unsupervised topic models may generate topics containing different sentiments (e.g., "*good*", "*bad*"). To consider users' interests and needs, seed-guided topic discovery approaches (Jagarlamudi et al., 2012; Gallagher et al., 2017; Meng et al., 2020a) have been proposed to find representative terms for each category based on user-provided seeds or category names.[2] However, there are still two less concerned factors in these approaches.

Table 1: Three datasets (Cohan et al., 2020; McAuley and Leskovec, 2013; Zhang et al., 2017) from different domains and their topic categories (i.e., seeds). **Red**: Seeds never seen in the corpus (i.e., out-of-vocabulary). In all three datasets, a large proportion of seeds are out-of-vocabulary.

Dataset	Category Names (Seeds)	
SciDocs (Scientific Papers)	cardiovascular diseases chronic kidney disease **chronic respiratory diseases** diabetes mellitus **digestive diseases** **hiv/aids**	**hepatitis a/b/c/e** mental disorders musculoskeletal disorders **neoplasms (cancer)** neurological disorders
Amazon (Product Reviews)	**apps for android** books **cds and vinyl** **clothing, shoes and jewelry** electronics	**health and personal care** **home and kitchen** movies and tv **sports and outdoors** video games
Twitter (Social Media Posts)	food **shop and service** **travel and transport** **college and university** **nightlife spot**	residence **outdoors and recreation** **arts and entertainment** **professional and other places**

The Existence of Out-of-Vocabulary Seeds. Previous studies (Jagarlamudi et al., 2012; Gallagher et al., 2017; Meng et al., 2020a) assume that all user-provided seeds must be **in-vocabulary** (i.e., appear at least once in the input corpus), so that they can utilize the occurrence statistics or Skip-Gram embedding methods (Mikolov et al., 2013) to model seed semantics. However, user-interested categories can have specific or composite descriptions, which may never appear in the corpus. Table 1 shows three datasets from different domains: sci-

[1]The code and datasets are available at https://github.com/yuzhimanhua/SeeTopic.

[2]In this paper, we use "seeds" and "category names" interchangeably.

entific papers, product reviews, and social media posts. In each dataset, documents can belong to one or more categories, and we list the category names provided by the dataset collectors. These seeds should reflect their particular interests. In all three datasets, we have a large proportion of seeds (45% in SciDocs, 60% in Amazon, and 78% in Twitter) that never appear in the corpus. Some category names are too specific (e.g., "*chronic respiratory diseases*", "*nightlife spot*") to be exactly matched, others are the composition of multiple entities (e.g., "*hepatitis a/b/c/e*", "*neoplasms (cancer)*", "*clothing, shoes and jewelry*").[3]

The Power of Pre-trained Language Models. Techniques used in previous studies are mainly based on LDA variants (Jagarlamudi et al., 2012) or context-free embeddings (Meng et al., 2020a). Recently, pre-trained language models (PLMs) such as BERT (Devlin et al., 2019) have achieved significant improvement in a wide range of text mining tasks. In topic discovery, the generic representation power of PLMs learned from web-scale corpora (e.g., Wikipedia or PubMed) may complement the information a model can obtain from the input corpus. Moreover, out-of-vocabulary seeds usually have meaningful in-vocabulary components (e.g., "*night*" and "*life*" in "*nightlife spot*", "*health*" and "*care*" in "*health and personal care*"). The optimized tokenization strategy of PLMs (Sennrich et al., 2016; Wu et al., 2016) can help segment the seeds into such meaningful components (e.g., "*nightlife*" $\rightarrow$ "*night*" and "*##life*"), and the contextualization power of PLMs can help infer the correct meaning of each component (e.g., "*##life*" and "*care*") in the category name. Therefore, PLMs are much needed in handling out-of-vocabulary seeds and effectively learning their semantics.

Contributions. Being aware of these two factors, in this paper, we study seed-guided topic discovery in the presence of out-of-vocabulary seeds. Our proposed SEETOPIC framework consists of two modules: (1) The *general* representation module uses a PLM to derive the representation of each term (including out-of-vocabulary seeds) based on the general linguistic knowledge acquired through pre-training. (2) The *seed-guided local* representation module learns in-vocabulary term embeddings specific to the input corpus and the given seeds. In order to optimize the learned representations for topic coherence, which is commonly reflected by pointwise mutual information (PMI) (Newman et al., 2010), our objective implicitly maximizes the PMI between each word and its context, the documents it appears, as well as the category it belongs to. The learning of the two modules is connected through an iterative ensemble ranking process, in which the general knowledge of PLMs and the term representations specifically learned from the target corpus conditioned on the seeds can complement each other.

To summarize, this study makes three contributions. (1) *Task*: we propose to study seed-guided topic discovery in the presence of out-of-vocabulary seeds. (2) *Framework*: we design a unified framework that jointly models general knowledge through PLMs and local corpus statistics through embedding learning. (3) *Experiment*: extensive experiments on three datasets demonstrate the effectiveness of SEETOPIC in terms of topic coherence, accuracy, and diversity.

2 Problem Definition

As shown in Table 1, we assume a seed can be either a single word or a phrase. Given a corpus $\mathcal{D}$, we use $\mathcal{V}_\mathcal{D}$ to denote the set of **terms** appearing in $\mathcal{D}$. In accordance with the assumption of category names, each term can also be a single word or a phrase. In practice, given a raw corpus, one can use existing phrase chunking tools (Manning et al., 2014; Shang et al., 2018) to detect phrases in it. *After* phrase chunking, if a category name is still not in $\mathcal{V}_\mathcal{D}$, we define it as **out-of-vocabulary**.

Problem Definition. *Given a corpus $\mathcal{D} = \{d_1, ..., d_{|\mathcal{D}|}\}$ and a set of category names $\mathcal{C} = \{c_1, ..., c_{|\mathcal{C}|}\}$ where some category names are out-of-vocabulary, the task is to find a set of in-vocabulary terms $\mathcal{S}_i = \{w_1, ..., w_S\} \subseteq \mathcal{V}_\mathcal{D}$ for each category c_i such that each term in $\mathcal{S}_i$ is semantically close to c_i and far from other categories c_j $(\forall j \neq i)$.*

3 The SEETOPIC Framework

In this section, we first introduce how we model general and local text semantics using a PLM mod-

[3] One possible idea to deal with composite seeds is to split them into multiple seeds. However, there are many possible ways to express the conjunctions (e.g., "*/*", "*()*", ",", and "*and*" in Table 1), which may require manual tuning. Besides, simple chunking rules will induce splits that break the semantics of the original composition (e.g., "*professional and other places*" may be split into "*professional*" and "*other places*"). Moreover, even after the split, some seeds are still out-of-vocabulary. Therefore, we propose to use PLMs to tackle out-of-vocabulary seeds in a unified way. In experiments, we will show that our model is able to tackle composite seeds. For example, given the seed "*hepatitis a/b/c/e*", we can find terms relevant to "*hepatitis b*" and "*hepatitis c*" (see Table 4).

ule and a seed-guided embedding learning module, respectively. Then, we present the iterative ensemble ranking process and our overall framework.

3.1 Modeling General Text Semantics using a PLM

PLMs such as BERT (Devlin et al., 2019) aim to learn generic language representations from web-scale corpora (e.g., Wikipedia or PubMed) that can be applied to a wide variety of text-related applications. To transfer such general knowledge to our topic discovery task, we employ a PLM to encode each category name and each in-vocabulary term to a vector. To be specific, given a term $w \in \mathcal{C} \cup \mathcal{V}_{\mathcal{D}}$, we input the sequence "`[CLS]` w `[SEP]`" into the PLM. Here, w can be a phrase containing multiple words, and each word can be out of the PLM's vocabulary. To deal with this, most PLMs use a pre-trained tokenizer (Sennrich et al., 2016; Wu et al., 2016) to segment each unseen word into frequent subwords. Then, the contextualization power of PLMs will help infer the correct meaning of each word/subword, so as to provide a more precise representation of the whole category.

After LM encoding, following (Sia et al., 2020; Thompson and Mimno, 2020; Li et al., 2020), we take the output of all tokens from the last layer and average them to get the term embedding $\boldsymbol{e}_w$. In this way, even if a seed c_i is **out-of-vocabulary**, we can still obtain its representation $\boldsymbol{e}_{c_i}$.

3.2 Modeling Local Text Semantics in the Input Corpus

The motivation of topic discovery is to discover latent topic structures from the input corpus. Therefore, purely relying on general knowledge in the PLM is insufficient because topic discovery results should adapt to the input corpus $\mathcal{D}$. Now, we introduce how we learn another set of embeddings $\{\boldsymbol{u}_w | w \in \mathcal{V}_{\mathcal{D}}\}$ from $\mathcal{D}$.

Previous studies on embedding learning assume that the semantic of a term is similar to its local context (Mikolov et al., 2013), the document it appears (Tang et al., 2015; Xun et al., 2017a), and the category it belongs to (Meng et al., 2020a). Inspired by these studies, we propose the following embedding learning objective.

$$\mathcal{J} = \underbrace{\sum_{d \in \mathcal{D}} \sum_{w_i \in d} \sum_{w_j \in \mathcal{C}(w_i,h)} p(w_j|w_i)}_{\text{context}} + \underbrace{\sum_{d \in \mathcal{D}} \sum_{w \in d} p(d|w)}_{\text{document}} + \underbrace{\sum_{c_i \in \mathcal{C}} \sum_{w \in \mathcal{S}_i} p(c_i|w)}_{\text{category}}, \tag{1}$$

where

$$p(z|w) = \frac{\exp(\boldsymbol{u}_w^T \boldsymbol{v}_z)}{\sum_{z'} \exp(\boldsymbol{u}_w^T \boldsymbol{v}_{z'})}, \quad (z \text{ can be } w_j, d, \text{ or } c_i). \tag{2}$$

In this objective, $\boldsymbol{u}_{w_i}$ (and $\boldsymbol{v}_{w_j}$), $\boldsymbol{v}_d$, $\boldsymbol{v}_{c_i}$ are the embedding vectors of terms, documents, and categories, respectively. $\mathcal{C}(w_i, h)$ is the set of context terms of w_i in d. Specifically, if $d = w_1 w_2 ... w_L$, then $\mathcal{C}(w_i, h) = \{w_j | i - h \leq j \leq i + h, j \neq i\}$, where h is the context window size.

Note that the last term in Eq. (1) encourages the similarity between each category c_i and its representative terms $\mathcal{S}_i$. Here, we adopt an iterative process to gradually update category-representative terms. Initially, $\mathcal{S}_i$ consists of just a few in-vocabulary terms similar to c_i according to the PLM. At each iteration, the size of $\mathcal{S}_i$ will increase to contain more category-discriminative terms (the selection criterion of these terms will be introduced in the next section), and we need to encourage their proximity with c_i in the next iteration.

Directly optimizing the full softmax in Eq. (2) is costly. Therefore, we adopt the negative sampling strategy (Mikolov et al., 2013) for efficient approximation.

Interpreting the Objective. In topic modeling studies, pointwise mutual information (PMI) (Newman et al., 2010) is a standard evaluation metric for topic coherence (Lau et al., 2014; Röder et al., 2015). Levy and Goldberg (2014) prove that the Skip-Gram embedding model is implicitly factorizing the PMI matrix. Following their proof, we can show that maximizing Eq. (1) is implicitly doing the following factorization:

$$\mathbf{U}_w^T [\mathbf{V}_w; \mathbf{V}_d; \mathbf{V}_c] = [\mathbf{X}_{ww}; \mathbf{X}_{wd}; \mathbf{X}_{wc}], \tag{3}$$

where the columns of $\mathbf{U}_w$, $\mathbf{V}_w$, $\mathbf{V}_d$, $\mathbf{V}_c$ are $\boldsymbol{u}_{w_i}$, $\boldsymbol{v}_{w_j}$, $\boldsymbol{v}_d$, $\boldsymbol{v}_{c_i}$, respectively ($w_i, w_j \in \mathcal{V}_{\mathcal{D}}$, $d \in \mathcal{D}$, $c_i \in \mathcal{C}$); $\mathbf{X}_{ww}$, $\mathbf{X}_{wd}$, and $\mathbf{X}_{wc}$ are PMI matrices.

$$\begin{aligned}
\mathbf{X}_{ww} &= \left[\log \left(\frac{\#_{\mathcal{D}}(w_i, w_j) \cdot \lambda_{\mathcal{D}}}{\#_{\mathcal{D}}(w_i) \cdot \#_{\mathcal{D}}(w_j) \cdot b} \right) \right]_{w_i, w_j \in \mathcal{V}_{\mathcal{D}}}, \\
\mathbf{X}_{wd} &= \left[\log \left(\frac{\#_d(w) \cdot \lambda_{\mathcal{D}}}{\#_{\mathcal{D}}(w) \cdot \lambda_d \cdot b} \right) \right]_{w \in \mathcal{V}_{\mathcal{D}},\, d \in \mathcal{D}}, \\
\mathbf{X}_{wc} &= \left[x_{w,c_i} \right]_{w \in \mathcal{V}_{\mathcal{D}},\, c_i \in \mathcal{C}}, \quad \text{where} \\
x_{w,c_i} &= \begin{cases} \log \frac{|\mathcal{C}|}{b}, & \text{if } w \in \mathcal{S}_i, \\ -\infty, & \text{if } w \in \mathcal{S}_j \ (\forall j \neq i). \end{cases}
\end{aligned} \tag{4}$$

Here, $\#_{\mathcal{D}}(w_i, w_j)$ denotes the number of co-occurrences of w_i and w_j in a context window in $\mathcal{D}$; $\#_{\mathcal{D}}(w)$ denotes the number of occurrences of w

in $\mathcal{D}$; $\lambda_{\mathcal{D}}$ is the total number of terms in $\mathcal{D}$; $\#_d(w)$ denotes the number of times w occurs in d; λ_d is the total number of terms in d; b is the number of negative samples. (For the derivation of Eq. (3), please refer to Appendix A.)

To summarize, the learned local representations $\boldsymbol{u}_w$ are implicitly optimized for topic coherence, where term co-occurrences are measured in context, document, and category levels.

3.3 Ensemble Ranking

We have obtained two sets of term embeddings that model text semantics from different angles: $\{\boldsymbol{e}_w | w \in \mathcal{C} \cup \mathcal{V}_{\mathcal{D}}\}$ carries the PLM's knowledge, while $\{\boldsymbol{u}_w | w \in \mathcal{V}_{\mathcal{D}}\}$ models the input corpus as well as user-provided seeds. We now propose an ensemble ranking method to leverage information from both sides to grab more discriminative terms for each category.

Given a category c_i and its current term set $\mathcal{S}_i$, we first calculate the scores of each term $w \in \mathcal{V}_{\mathcal{D}}$.

$$\begin{aligned} \text{score}_G(w|\mathcal{S}_i) &= \frac{1}{|\mathcal{S}_i|} \sum_{w' \in \mathcal{S}_i} \cos(\boldsymbol{e}_w, \boldsymbol{e}_{w'}), \\ \text{score}_L(w|\mathcal{S}_i) &= \frac{1}{|\mathcal{S}_i|} \sum_{w' \in \mathcal{S}_i} \cos(\boldsymbol{u}_w, \boldsymbol{u}_{w'}). \end{aligned} \tag{5}$$

The subscript "G" here means "general", while "L" means "local". Then, we sort all terms by these two scores, respectively. Each term w will hence get two rank positions $\text{rank}_G(w)$ and $\text{rank}_L(w)$. We propose the following ensemble score based on the reciprocal rank:

$$\text{score}(w|\mathcal{S}_i) = \left(\frac{1}{2} \left(\frac{1}{\text{rank}_G(w)} \right)^{\rho} + \frac{1}{2} \left(\frac{1}{\text{rank}_L(w)} \right)^{\rho} \right)^{1/\rho}. \tag{6}$$

Here, $0 < \rho \leq 1$ is a constant. In practice, instead of ranking all terms in the vocabulary, we only check the top-M results in the two ranking lists. If a term w is not among the top-M according to $\text{score}_G(w)$ (resp., $\text{score}_L(w)$), we set $\text{rank}_G(w) = +\infty$ (resp., $\text{rank}_L(w) = +\infty$). In fact, when $\rho = 1$, Eq. (6) becomes the *arithmetic* mean of the two reciprocal ranks $\frac{1}{\text{rank}_G(w)}$ and $\frac{1}{\text{rank}_L(w)}$. This is essentially the mean reciprocal rank (MRR) commonly used in ensemble ranking, where a high position in one ranking list can largely compensate a low position in the other. In contrast, when $\rho \to 0$, Eq. (6) becomes the *geometric* mean of the two reciprocal ranks (see Appendix B), where two ranking lists both have the "veto power" (i.e., a term needs to be ranked as top-M in both ranking lists to obtain a non-zero ensemble score). In experiment, we set $\rho = 0.1$ and show it outperforms MRR (i.e., $\rho = 1$) in our topic discovery task.

Algorithm 1: SEETOPIC

Input: A text corpus $\mathcal{D} = \{d_1, ..., d_{|\mathcal{D}|}\}$, a set of seeds $\mathcal{C} = \{c_1, ..., c_{|\mathcal{C}|}\}$, and a PLM.
Output: $(\mathcal{S}_1, ..., \mathcal{S}_{|\mathcal{C}|})$, where each $\mathcal{S}_i$ is a set of category-discriminative terms for c_i.

1 Compute $\{\boldsymbol{e}_w | w \in \mathcal{C} \cup \mathcal{V}_{\mathcal{D}}\}$ using the PLM;
2 // Initialize $\mathcal{S}_i$;
3 $\mathcal{S}_1, ..., \mathcal{S}_{|\mathcal{C}|} \leftarrow \emptyset$;
4 **for** $n \leftarrow 1$ *to* N **do**
5 **for** $i \leftarrow 1$ *to* $|\mathcal{C}|$ **do**
6 $w_n \leftarrow \underset{w \in \mathcal{V}_{\mathcal{D}} \setminus (\mathcal{S}_1 \cup ... \cup \mathcal{S}_{|\mathcal{C}|})}{\arg\max} \cos(\boldsymbol{e}_w, \boldsymbol{e}_{c_i})$;
7 $\mathcal{S}_i \leftarrow \mathcal{S}_i \cup \{w_n\}$;
8 // Update $\mathcal{S}_i$ for T iterations;
9 **for** $t \leftarrow 1$ *to* T **do**
10 Learn $\{\boldsymbol{u}_w | w \in \mathcal{V}_{\mathcal{D}}\}$ from the input corpus $\mathcal{D}$ and the up-to-date representative terms $\mathcal{S}_1, ..., \mathcal{S}_{|\mathcal{C}|}$ according to Eq. (1);
11 $\text{score}_G(w|\mathcal{S}_i)$ and $\text{score}_L(w|\mathcal{S}_i) \leftarrow$ Eq. (5);
12 $\text{score}(w|\mathcal{S}_i) \leftarrow$ Eq. (6);
13 $\mathcal{S}_1, ..., \mathcal{S}_{|\mathcal{C}|} \leftarrow \emptyset$;
14 **for** $n \leftarrow 1$ *to* $(t+1)N$ **do**
15 **for** $i \leftarrow 1$ *to* $|\mathcal{C}|$ **do**
16 $\mathcal{S}_i \leftarrow$ Eq. (7);
17 Return $(\mathcal{S}_1, ..., \mathcal{S}_{|\mathcal{C}|})$;

After computing the ensemble score $\text{score}(w|\mathcal{S}_i)$ for each w, we update $\mathcal{S}_i$. To guarantee that each $\mathcal{S}_i$ is category-discriminative, we do not allow any term to belong to more than one category. Therefore, we gradually expand each $\mathcal{S}_i$ by turns. At the beginning, we reset $\mathcal{S}_1 = ... = \mathcal{S}_{|\mathcal{C}|} = \emptyset$. When it is $\mathcal{S}_i$'s turn, we add one term $\mathcal{S}_i$ according to the following criterion:

$$\mathcal{S}_i \leftarrow \mathcal{S}_i \cup \{ \underset{w \in \mathcal{V}_{\mathcal{D}} \setminus (\mathcal{S}_1 \cup ... \cup \mathcal{S}_{|\mathcal{C}|})}{\arg\max} \text{score}(w|\mathcal{S}_i) \}. \tag{7}$$

3.4 Overall Framework

We summarize the entire SEETOPIC framework in Algorithm 1. To deal with **out-of-vocabulary** category names, we first utilize a PLM to find their nearest in-vocabulary terms as the initial category-discriminative term set $\mathcal{S}_i$ (Lines 1-7). After initialization, $|\mathcal{S}_i| = N$ ($\forall 1 \leq i \leq |\mathcal{C}|$). Note that for an in-vocabulary category name $c_i \in \mathcal{V}_{\mathcal{D}}$, itself will be added to the initial $\mathcal{S}_i$ as the top-1 similar in-vocabulary term.

After getting the initial $\mathcal{S}_i$, we update it by T iterations (Lines 8-16). At each iteration, according to the up-to-date $\mathcal{S}_1, \mathcal{S}_2, ..., \mathcal{S}_{|\mathcal{C}|}$, we relearn embeddings $\boldsymbol{u}_w$, $\boldsymbol{v}_w$, $\boldsymbol{v}_d$, and $\boldsymbol{v}_{c_i}$ using Eq. (1) (Line 10). The two set of embeddings, $\{\boldsymbol{e}_w | w \in \mathcal{C} \cup \mathcal{V}_{\mathcal{D}}\}$ (computed at Line 1) and $\{\boldsymbol{u}_w | w \in \mathcal{V}_{\mathcal{D}}\}$ (updated at Line 10), are then leveraged to perform ensemble ranking (Lines 11-12). Based on the

ensemble score $\text{score}(w|\mathcal{S}_i)$, we update $\mathcal{S}_i$ using Eq. (7) (Lines 13-16). After the t-th iteration, $|\mathcal{S}_i| = (t+1)N$ $(\forall 1 \leq i \leq |\mathcal{C}|)$.

Complexity Analysis. The time complexity of using the PLM is $\mathcal{O}((|\mathcal{C}| + |\mathcal{V}_\mathcal{D}|)\alpha_{\text{PLM}})$, where α_{PLM} is the complexity of encoding one term via the PLM. The total complexity of local embedding is $\mathcal{O}(T\lambda_\mathcal{D}(h+|\mathcal{C}|)b)$ because in each iteration $1 \leq t \leq T$, every $w \in \mathcal{D}$ interacts with every other term in the context window of size h, its belonging document, and each category $c_i \in \mathcal{C}$, and each update involves b negative samples. The total complexity of ensemble ranking is $\mathcal{O}(T|\mathcal{V}_\mathcal{D}||\mathcal{C}||\mathcal{S}_i|)$ as in each iteration $1 \leq t \leq T$, we compute scores between each $w \in \mathcal{V}_\mathcal{D}$ and each $w' \in \mathcal{S}_i$ $(\forall c_i \in \mathcal{C})$.

4 Experiments

4.1 Experimental Setup

Datasets. We conduct experiments on three public datasets from different domains: (1) **SciDocs** (Cohan et al., 2020)[4] is a large collection of scientific papers supporting diverse evaluation tasks. For the MeSH classification task (Coletti and Bleich, 2001), about 23K medical papers are collected, each of which is assigned to one of the 11 common disease categories derived from the MeSH vocabulary. We use the title and abstract of each paper as documents and the 11 category names as seeds. (2) **Amazon** (McAuley and Leskovec, 2013)[5] contains product reviews from May 1996 to July 2014. Each Amazon review belongs to one or more product categories. We use the subset sampled by Zhang et al. (2020, 2022), which contains 10 categories and 100K reviews. (3) **Twitter** (Zhang et al., 2017)[6] is a crawl of geo-tagged tweets in New York City from August 2014 to November 2014. The dataset collectors link these tweets with Foursquare's POI database and assign them to 9 POI categories. We take these category names as input seeds.

Seeds used in the three datasets are shown in Table 1. Dataset statistics are summarized in Table 2. For all three datasets, we use AutoPhrase (Shang et al., 2018)[7] to perform phrase chunking in the corpus, and we remove words and phrases occurring less than 3 times.

Previous studies (Jagarlamudi et al., 2012; Meng et al., 2020a) have tried some other datasets (e.g., RCV1, 20 Newsgroups, NYT, and Yelp). However, the category names they use in these datasets are all picked from **in-vocabulary** terms. Therefore, we do not consider these datasets for evaluation in our task settings.

Table 2: Dataset Statistics.

Dataset	SciDocs	Amazon	Twitter
#Documents	23,473	100,000	135,529
#In-vocabulary Terms (After Phrase Chunking)	55,897	56,942	17,577
Avg Doc Length	239.8	119.0	6.7
#Seeds	11	10	9
#Out-of-vocabulary Seeds (After Phrase Chunking)	5	6	7

Following (Sia et al., 2020), we adopt a 60-40 train-test split for all three datasets. The training set is used as the input corpus $\mathcal{D}$, and the testing set is used for calculating topic coherence metrics (see evaluation metrics for details).

Compared Methods. We compare our SEETOPIC framework with the following methods, including seed-guided topic modeling methods, seed-guided embedding learning methods, and PLMs. (1) **SeededLDA** (Jagarlamudi et al., 2012)[8] is a seed-guided topic modeling method. It improves LDA by biasing topics to produce input seeds and by biasing documents to select topics relevant to the seeds they contain. (2) **Anchored CorEx** (Gallagher et al., 2017)[9] is a seed-guided topic modeling method. It incorporates user-provided seeds by balancing between compressing the input corpus and preserving seed-related information. (3) **Labeled ETM** (Dieng et al., 2020)[10] is an embedding-based topic modeling method. It incorporates distributed representation of each term. Following (Meng et al., 2020a), we retrieve representative terms according to their embedding similarity with the category name. (4) **CatE** (Meng et al., 2020a)[11] is a seed-guided embedding learning method for discriminative topic discovery. It takes category names as input and jointly learns term embedding and specificity from the input corpus. Category-discriminative terms are then selected based on both embedding similarity with the category and specificity. (5) **BERT** (Devlin et al., 2019)[12] is a PLM. Following Lines 1-7 in Algorithm 1, we use BERT to encode each input category name and each term to a vector, and then perform similarity search to directly find all repre-

[4] https://github.com/allenai/scidocs
[5] http://jmcauley.ucsd.edu/data/amazon/index.html
[6] https://github.com/franticnerd/geoburst
[7] https://github.com/shangjingbo1226/AutoPhrase
[8] https://github.com/vi3k6i5/GuidedLDA
[9] https://github.com/gregversteeg/corex_topic
[10] https://github.com/adjidieng/ETM
[11] https://github.com/yumeng5/CatE
[12] https://huggingface.co/bert-base-uncased

Table 3: NPMI, LCP, MACC, and Diversity of compared algorithms on three datasets. NPMI and LCP measure topic coherence; MACC measures term accuracy; Diversity (abbreviated to Div.) measures topic diversity. **Bold**: the highest score. Underline: the second highest score. *: significantly worse than SEETOPIC (p-value < 0.05). **: significantly worse than SEETOPIC (p-value < 0.01).

Methods	SciDocs				Amazon				Twitter			
	NPMI	LCP	MACC	Div.	NPMI	LCP	MACC	Div.	NPMI	LCP	MACC	Div.
SeededLDA	0.056**	**-0.616**	0.156**	0.451**	0.070**	-0.753	0.147**	0.393**	0.013**	-2.254**	0.195**	0.696**
Anchored CorEx	0.106**	-1.090**	0.264**	**1.000**	0.134**	-0.982*	0.333**	**1.000**	0.090**	-2.192**	0.233**	**1.000**
Labeled ETM	0.334*	-0.775**	0.458**	0.961*	0.308**	-1.051**	0.585**	**1.000**	0.305*	-1.098**	0.268**	0.989
CatE	0.345*	-0.725**	0.633**	**1.000**	0.317**	-0.844**	0.856*	**1.000**	**0.356**	**-0.827**	0.483**	**1.000**
BERT	0.313**	-0.841**	0.740**	0.891**	0.294**	-1.093**	0.832**	**1.000**	0.313**	-1.044**	0.627	0.944**
BioBERT	0.309**	-0.852**	**0.938**	0.982**	–	–	–	–	–	–	–	–
SEETOPIC-NoIter	0.341**	-0.768**	0.887	**1.000**	0.322**	-0.986**	0.892	**1.000**	0.318	-1.004**	0.618	**1.000**
SEETOPIC	**0.358**	-0.634	0.909	**1.000**	**0.342**	**-0.696**	**0.904**	**1.000**	0.320	-0.907	**0.633**	**1.000**

sentative terms. (6) **BioBERT** (Lee et al., 2020)[13] is a PLM. It is used in the same way as BERT. Since BioBERT is specifically trained for biomedical text mining tasks, we report its performance on the SciDocs dataset only. (7) **SEETOPIC-NoIter** is a variant of our SEETOPIC framework. In Algorithm 1, after initialization (Lines 1-7), it executes Lines 9-16 only once (i.e., $T = 1$) to find all representative terms.

Here, all seed-guided topic modeling and embedding baselines (i.e., SeededLDA, Anchored CorEx, CatE, and Labeled ETM) can only take **in-vocabulary** seeds as input. For a fair comparison, we run Lines 1-7 in Algorithm 1 to get the initial representative in-vocabulary terms for each category, and input these terms as seeds into the baselines. In other words, all compared methods use BERT/BioBERT to initialize their term sets.

Evaluation Metrics. We evaluate topic discovery results from three different angles: topic coherence, term accuracy, and topic diversity.

(1) **NPMI** (Lau et al., 2014) is a standard metric in topic modeling to measure *topic coherence*. Within each topic, it calculates the normalized pointwise mutual information for each pair of terms in $\mathcal{S}_i$.

$$\text{NPMI} = \frac{1}{|\mathcal{C}|}\sum_{i=1}^{|\mathcal{C}|}\frac{1}{\binom{|\mathcal{S}_i|}{2}}\sum_{w_j,w_k\in\mathcal{S}_i}\frac{\log\frac{P(w_j,w_k)}{P(w_j)P(w_k)}}{-\log P(w_j,w_k)}, \quad (8)$$

where $P(w_j, w_k)$ is the probability that w_j and w_k co-occur in a document; $P(w_j)$ is the marginal probability of w_j.[14]

(2) **LCP** (Mimno et al., 2011) is another standard metric to measure *topic coherence*. It calculates the pairwise log conditional probability of top-ranked terms.

$$\text{LCP} = \frac{1}{|\mathcal{C}|}\sum_{i=1}^{|\mathcal{C}|}\frac{1}{\binom{|\mathcal{S}_i|}{2}}\sum_{\substack{w_j,w_k\in\mathcal{S}_i\\ j<k}}\log\frac{P(w_j,w_k)}{P(w_j)}. \quad (9)$$

Note that **PMI** (Newman et al., 2010) is also a standard metric for topic coherence. We do observe that SEETOPIC outperforms baselines in terms of PMI in most cases. However, since our local embedding step is implicitly optimizing a PMI-like objective, we no longer use it as our evaluation metric.

(3) **MACC** (Meng et al., 2020a) measures *term accuracy*. It is defined as the proportion of retrieved terms that actually belong to the corresponding category according to the category name.

$$\text{MACC} = \frac{1}{|\mathcal{C}|}\sum_{i=1}^{|\mathcal{C}|}\frac{1}{|\mathcal{S}_i|}\sum_{w_j\in\mathcal{S}_i}\mathbf{1}(w_j \in c_i), \quad (10)$$

where $\mathbf{1}(w_j \in c_i)$ is the indicator function of whether w_j is relevant to category c_i. MACC requires human evaluation, so we invite five annotators to perform independent annotation. The reported MACC score is the average MACC of the five annotators. A high inter-annotator agreement is observed, with Fleiss' kappa (Fleiss, 1971) being 0.856, 0.844, and 0.771 on SciDocs, Amazon, and Twitter, respectively.

(4) **Diversity** (Dieng et al., 2020) measures the *mutual exclusivity* of discovered topics. It is the percentage of unique terms in all topics, which corresponds to our task requirement that each retrieved term is discriminatively close to one category and far from the others.

$$\text{Diversity} = \frac{|\bigcup_{i=1}^{|\mathcal{C}|}\mathcal{S}_i|}{\sum_{i=1}^{|\mathcal{C}|}|\mathcal{S}_i|}. \quad (11)$$

Experiment Settings. We use BioBERT as the

[13] https://huggingface.co/dmis-lab/biobert-v1.1

[14] When calculating Eqs. (8) and (9), to avoid $\log 0$, we use $P(w_j, w_k) + \epsilon$ and $P(w) + \epsilon$ to replace $P(w_j, w_k)$ and $P(w)$, respectively, where $\epsilon = 1/|\mathcal{D}|$.

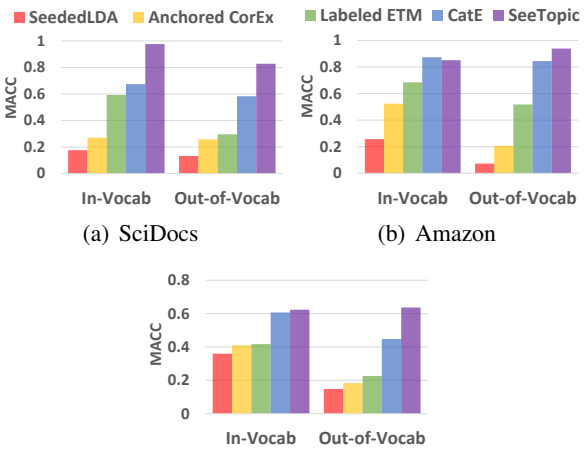
PLM on SciDocs, and BERT-base-uncased as the PLM on Amazon and Twitter. The embedding dimension of $\boldsymbol{u}_w$ is 768 (the same as $\boldsymbol{e}_w$); the number of negative samples $b = 5$. In ensemble ranking, the length of the general/local ranking list $M = 100$; the hyperparameter ρ in Eq. (6) is set as 0.1; the number of iterations $T = 4$; after each iteration, we increase the size of $\mathcal{S}_i$ by $N = 3$. We use the top-10 ranked terms in each topic for final evaluation (i.e., $|\mathcal{S}_i| = 10$ in Eqs. (8)-(11)). Experiments are run on Intel Xeon E5-2680 v2 @ 2.80GHz and one NVIDIA GeForce GTX 1080.

4.2 Performance Comparison

Table 3 shows the performance of all methods. We run each experiment 3 times with the average score reported. To show statistical significance, we conduct a two-tailed unpaired t-test to compare SEETOPIC and each baseline. (The performance of BERT and BioBERT is deterministic according to our usage. When comparing SEETOPIC with them, we conduct a two-tailed Z-test instead.) The significance level is also marked in Table 3.

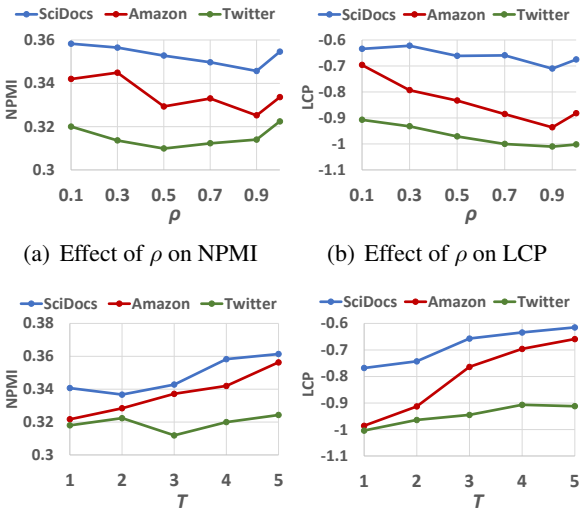
We have the following observations from Table 3. (1) Our SEETOPIC model performs consistently well. In fact, it achieves the highest score in 8 columns and the second highest in the remaining 4 columns. (2) Classical seed-guided topic modeling baselines (i.e., SeededLDA and Anchored CorEx) perform not well in respect of NPMI (topic coherence) and MACC (term accuracy). Embedding-based topic discovery approaches (i.e., Labeled ETM and CatE) make some progress, but they still significantly underperform the PLM-empowered SEETOPIC model on SciDocs and Amazon. (3) SEETOPIC consistently performs better than SEETOPIC-NoIter on all three datasets, indicating the positive contribution of the proposed iterative process. (4) SEETOPIC guarantees the mutual exclusivity of $\mathcal{S}_1, ..., \mathcal{S}_{|\mathcal{C}|}$. In comparison, SeededLDA, Labeled ETM, and BERT cannot guarantee such mutual exclusivity.

In-vocabulary vs. Out-of-vocabulary. Figure 1 compares the MACC scores of different seed-guided topic discovery methods on in-vocabulary categories and out-of-vocabulary categories. We find that the performance improvement of SEETOPIC upon baselines on out-of-vocabulary categories is larger than that on in-vocabulary ones. For example, on Amazon, SEETOPIC underperforms CatE on in-vocabulary categories but outperforms CatE on out-of-vocabulary ones; on Twitter, the gap between SEETOPIC and baselines becomes much more evident on out-of-vocabulary categories. Note that all baselines in Figure 1 do not utilize the power of PLMs, so this observation validates our claim that PLMs are helpful in tackling out-of-vocabulary seeds.



(a) SciDocs (b) Amazon (c) Twitter

Figure 1: MACC of seed-guided topic discovery methods on in-vocabulary categories and out-of-vocabulary categories.



(a) Effect of ρ on NPMI (b) Effect of ρ on LCP (c) Effect of T on NPMI (d) Effect of T on LCP

Figure 2: Parameter study of SEETOPIC measured by topic coherence.

4.3 Parameter Study

We study the effect of two important hyperparameters: ρ (the hyperparameter in ensemble ranking) and T (the number of iterations). We vary the value of ρ in $\{0.1, 0.3, 0.5, 0.7, 0.9, 1\}$ (SEETOPIC uses $\rho = 0.1$ by default) and the value of T in $\{1, 2, 3, 4, 5\}$ (SEETOPIC uses $T = 4$ by default, and SEETOPIC-NoIter is the case when $T = 1$). Figure 2 shows the change of model performance measured by NPMI and LCP.

Table 4: Top-5 representative terms retrieved by different algorithms for three out-of-vocabulary categories from SciDocs, Amazon, and Twitter. ✓: at least 3 of the 5 annotators judge the term as relevant to the seed. ✗: at most 2 of the 5 annotators judge the term as relevant to the seed.

Method	Top-5 Representative Terms
Dataset: SciDocs, Category Name: hepatitis a/b/c/e	
SeededLDA	patients (✗), treatment (✗), placebo (✗), study (✗), group (✗)
Anchored CorEx	expression (✗), gene (✗), cells (✗), genes (✗), genetic (✗)
Labeled ETM	hepatitis b virus hbv dna (✓), serum hbv dna (✓), serum alanine aminotransferase (✗), alanine aminotransferase alt (✗), below detection limit (✗)
CatE	chronic hepatitis b virus hbv infection (✓), hepatitis b e antigen hbeag (✓), hepatitis b virus hbv dna (✓), normal alanine aminotransferase (✗), hbeag-negative chronic hepatitis b (✓)
BioBERT	hepatitis b virus hbv dna (✓), chronic hepatitis b virus hbv infection (✓), hepatitis b e antigen hbeag (✓), hepatitis b virus hbv infection (✓), chronic hepatitis c virus hcv (✓)
SeeTopic-NoIter	hepatitis b virus hbv dna (✓), hepatitis b e antigen hbeag (✓), chronic hepatitis b virus hbv infection (✓), hepatitis b surface antigen hbsag (✓), hbeag-negative chronic hepatitis b (✓)
SeeTopic	chronic hepatitis b virus hbv infection (✓), hbeag-negative chronic hepatitis b (✓), hepatitis c virus hcv-infected (✓), hepatitis b virus hbv dna (✓), chronic hepatitis c virus hcv (✓)
Dataset: Amazon, Category Name: sports and outdoors	
SeededLDA	use (✗), good (✗), one (✗), product (✗), like (✗)
Anchored CorEx	sports (✓), use (✗), size (✗), wear (✗), fit (✓)
Labeled ETM	cars and tracks (✓), tracks and cars (✓), search options (✗), championships (✗), cool bosses (✗)
CatE	outdoorsmen (✓), outdoor activities (✓), cars and tracks (✓), foot support (✓), offers plenty (✗)
BERT	cars and tracks (✓), outdoor activities (✓), outdoorsmen (✓), sports (✓), sporting events (✓)
SeeTopic-NoIter	outdoorsmen (✓), outdoor activities (✓), cars and tracks (✓), indoor soccer (✓), bike riding (✓)
SeeTopic	canoeing (✓), picnics (✓), bike rides (✓), bike riding (✓), rafting (✓)
Dataset: Twitter, Category Name: travel and transport	
SeededLDA	nyc (✗), new york (✗), line (✓), high (✗), time square (✓)
Anchored CorEx	new york (✗), post photo (✓), new (✗), day (✗), today (✗)
Labeled ETM	tourism (✓), theview (✓), file (✗), morning view (✓), gma (✗)
CatE	maritime (✓), tourism (✓), natural history (✗), scenery (✓), elevate (✗)
BERT	maritime (✓), tourism (✓), natural history (✗), olive oil (✗), baggage claim (✓)
SeeTopic-NoIter	maritime (✓), tourism (✓), natural history (✗), scenery (✓), navy (✗)
SeeTopic	wildlife (✓), scenery (✓), maritime (✓), highlinepark (✗), aquarium (✓)

According to Figures 2(a) and 2(b), in most cases, the performance of SeeTopic deteriorates as ρ increases from 0.1 to 0.9. Thus, setting $\rho = 0.1$ always leads to competitive NPMI and LCP scores on the three datasets. Although $\rho = 1$ is better than $\rho = 0.9$, its performance is still suboptimal in comparison with $\rho = 0.1$. This finding indicates that replacing the mean reciprocal rank (i.e., $\rho = 1$) with our proposed Eq. (6) is reasonable. According to Figures 2(c) and 2(d), SeeTopic usually performs better when there are more iterations. On SciDocs and Twitter, the scores start to converge after $T = 4$. Besides, more iterations will result in longer running time. Overall, we believe setting $T = 4$ strikes a good balance.

4.4 Case Study

Finally, we show the terms retrieved by different methods as a case study. From each of the three datasets, we select an out-of-vocabulary category and show its topic discovery results in Table 4. We mark a retrieved term as correct (✓) if at least 3 of the 5 annotators judge the term as relevant to the seed. Otherwise, we mark the term as incorrect (✗).

For the category "*hepatitis a/b/c/e*" from SciDocs, SeededLDA and Anchored CorEx can only find very general medical terms, which are relevant to all seeds in SciDocs and thus inaccurate; Labeled ETM and CatE find terms about "*alanine aminotransferase*", whose elevation suggest not only hepatitis but also other diseases like diabetes and heart failure, thus not discriminative either; BioBERT and SeeTopic, with the power of a PLM, can accurately pick terms relevant to "*hepatitis b*" and "*hepatitis c*". For the category "*sports and outdoors*" from Amazon, SeededLDA and Anchored CorEx again find very general terms, most of which are not category-discriminative; Labeled ETM and CatE are able to pick more specific terms such as "*cars and tracks*", but they still make mistakes; BERT, as a PLM, can accurately find terms that have lexical overlap with the category name (e.g., "*outdoorsmen*", "*sporting events*"), meanwhile such terms are less diverse; SeeTopic-NoIter starts to discover more concrete terms than BERT (e.g., "*indoor soccer*", "*bike riding*") by leveraging local text semantics; the full SeeTopic model, with an iterative updating process, can find more specific and informative terms (e.g., "*canoeing*", "*picnics*", and "*rafting*"). For the category "*travel and trans-*

port" from Twitter, both BERT and CatE make mistakes by including the term "*natural history*"; SEETOPIC-NoIter, without an iterative update process, also includes this error; the full SEETOPIC model finally excludes this error and achieves the highest accuracy in the retrieved top-5 terms among all compared methods.

5 Related Work

Seed-Guided Topic Discovery. Seed-guided topic models aim to leverage user-provided seeds to discover underlying topics according to users' interests. Early studies take LDA (Blei et al., 2003) as the backbone and incorporate seeds into model learning. For example, Andrzejewski et al. (2009) consider must-link and cannot-link constraints among seeds as priors. SeededLDA (Jagarlamudi et al., 2012) encourages topics to contain more seeds and encourages documents to select topics relevant to the seeds they contain. Anchored CorEx (Gallagher et al., 2017) extracts maximally informative topics by jointly compressing the corpus and preserving seed relevant information. Recent studies start to utilize embedding techniques to learn better word semantics. For example, CatE (Meng et al., 2020a) explicitly encourages distinction among retrieved topics via category-name guided embedding learning. However, all these models require the provided seeds to be in-vocabulary, mainly because they focus on the input corpus only and are not equipped with general knowledge of PLMs.

Embedding-Based Topic Discovery. A number of studies extend LDA to involve word embedding. The common strategy is to adapt distributions in LDA to generate real-valued data (e.g., Gaussian LDA (Das et al., 2015), LFTM (Nguyen et al., 2015), Spherical HDP (Batmanghelich et al., 2016), and CGTM (Xun et al., 2017b)). Some other studies think out of the LDA backbone. For example, TWE (Liu et al., 2015) uses topic structures to jointly learn topic embeddings and improve word embeddings. CLM (Xun et al., 2017a) collaboratively improves topic modeling and word embedding by coordinating global and local contexts. ETM (Dieng et al., 2020) models word-topic correlations via word embeddings to improve the expressiveness of topic models. More recently, Sia et al. (2020) show that directly clustering word embeddings (e.g., word2vec or BERT) also generates good topics; Thompson and Mimno (2020) further find that BERT and GPT-2 discover high-quality topics, but RoBERTa does not. These models are unsupervised and hard to be applied to seed-guided settings. In contrast, our SEETOPIC framework joint leverages PLMs, word embeddings, and seed information.

6 Conclusions and Future Work

In this paper, we study seed-guided topic discovery in the presence of out-of-vocabulary seeds. To understand and make use of in-vocabulary components in each seed, we utilize the tokenization and contextualization power of PLMs. We propose a seed-guided embedding learning framework inspired by the goal of maximizing PMI in topic modeling, and an iterative ensemble ranking process to jointly leverage general knowledge of the PLM and local signals learned from the input corpus. Experimental results show that SEETOPIC outperforms seed-guided topic discovery baselines and PLMs in terms of topic coherence, term accuracy, and topic diversity. A parameter study and a case study further validate some design choices in SEETOPIC.

In the future, it would be interesting to extend SEETOPIC to seed-guided hierarchical topic discovery, where parent and child information in the input category hierarchy may help infer the meaning of out-of-vocabulary nodes.

Acknowledgments

We thank anonymous reviewers for their valuable and insightful feedback. Research was supported in part by US DARPA KAIROS Program No. FA8750-19-2-1004, SocialSim Program No. W911NF-17-C-0099, and INCAS Program No. HR001121C0165, National Science Foundation IIS-19-56151, IIS-17-41317, and IIS 17-04532, and the Molecule Maker Lab Institute: An AI Research Institutes program supported by NSF under Award No. 2019897, and the Institute for Geospatial Understanding through an Integrative Discovery Environment (I-GUIDE) by NSF under Award No. 2118329. Any opinions, findings, and conclusions or recommendations expressed herein are those of the authors and do not necessarily represent the views, either expressed or implied, of DARPA or the U.S. Government.

References

David Andrzejewski, Xiaojin Zhu, and Mark Craven. 2009. Incorporating domain knowledge into topic modeling via dirichlet forest priors. In *ICML'09*, pages 25–32.

Kayhan Batmanghelich, Ardavan Saeedi, Karthik Narasimhan, and Sam Gershman. 2016. Nonparametric spherical topic modeling with word embeddings. In *ACL'16*, pages 537–542.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *JMLR*, 3:993–1022.

Xingyuan Chen, Yunqing Xia, Peng Jin, and John Carroll. 2015. Dataless text classification with descriptive lda. In *AAAI'15*, pages 2224–2231.

Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel S Weld. 2020. Specter: Document-level representation learning using citation-informed transformers. In *ACL'20*, pages 2270–2282.

Margaret H Coletti and Howard L Bleich. 2001. Medical subject headings used to search the biomedical literature. *JAMIA*, 8(4):317–323.

Rajarshi Das, Manzil Zaheer, and Chris Dyer. 2015. Gaussian lda for topic models with word embeddings. In *ACL'15*, pages 795–804.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT'19*, pages 4171–4186.

Adji B Dieng, Francisco JR Ruiz, and David M Blei. 2020. Topic modeling in embedding spaces. *TACL*, 8:439–453.

Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378.

Ryan J Gallagher, Kyle Reing, David Kale, and Greg Ver Steeg. 2017. Anchored correlation explanation: Topic modeling with minimal domain knowledge. *TACL*, 5:529–542.

Thomas L Griffiths and Mark Steyvers. 2004. Finding scientific topics. *PNAS*, 101(suppl 1):5228–5235.

Thomas Hofmann. 1999. Probabilistic latent semantic indexing. In *SIGIR'99*, pages 50–57.

Jagadeesh Jagarlamudi, Hal Daumé, and Raghavendra Udupa. 2012. Incorporating lexical priors into topic models. In *EACL'12*, pages 204–213.

Jey Han Lau, David Newman, and Timothy Baldwin. 2014. Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality. In *EACL'14*, pages 530–539.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Omer Levy and Yoav Goldberg. 2014. Neural word embedding as implicit matrix factorization. *NIPS'14*, pages 2177–2185.

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020. On the sentence embeddings from pre-trained language models. In *EMNLP'20*, pages 9119–9130.

Yang Liu, Zhiyuan Liu, Tat-Seng Chua, and Maosong Sun. 2015. Topical word embeddings. In *AAAI'15*, pages 2418–2424.

Zequn Liu, Shukai Wang, Yiyang Gu, Ruiyi Zhang, Ming Zhang, and Sheng Wang. 2021. Graphine: A dataset for graph-aware terminology definition generation. In *EMNLP'21*, pages 3453–3463.

Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *ACL'14, System Demonstrations*, pages 55–60.

Julian McAuley and Jure Leskovec. 2013. Hidden factors and hidden topics: understanding rating dimensions with review text. In *RecSys'13*, pages 165–172.

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 *WWW'20*, pages 2121–2132.

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 *EMNLP'20*, pages 9006–9017.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS'13*, pages 3111–3119.

David Mimno, Hanna Wallach, Edmund Talley, Miriam Leenders, and Andrew McCallum. 2011. Optimizing semantic coherence in topic models. In *EMNLP'11*, pages 262–272.

David Newman, Jey Han Lau, Karl Grieser, and Timothy Baldwin. 2010. Automatic evaluation of topic coherence. In *NAACL-HLT'10*, pages 100–108.

Dat Quoc Nguyen, Richard Billingsley, Lan Du, and Mark Johnson. 2015. Improving topic models with latent feature word representations. *TACL*, 3:299–313.

Jiezhong Qiu, Yuxiao Dong, Hao Ma, Jian Li, Kuansan Wang, and Jie Tang. 2018. Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec. In *WSDM'18*, pages 459–467.

Michael Röder, Andreas Both, and Alexander Hinneburg. 2015. Exploring the space of topic coherence measures. In *WSDM'15*, pages 399–408.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *ACL'16*, pages 1715–1725.

Jingbo Shang, Jialu Liu, Meng Jiang, Xiang Ren, Clare R Voss, and Jiawei Han. 2018. Automated phrase mining from massive text corpora. *IEEE TKDE*, 30(10):1825–1837.

Suzanna Sia, Ayush Dalmia, and Sabrina J Mielke. 2020. Tired of topic models? clusters of pretrained word embeddings make for fast and good topics too! In *EMNLP'20*, pages 1728–1736.

Jian Tang, Meng Qu, and Qiaozhu Mei. 2015. Pte: Predictive text embedding through large-scale heterogeneous text networks. In *KDD'15*, pages 1165–1174.

Laure Thompson and David Mimno. 2020. Topic modeling with contextualized word representation clusters. *arXiv preprint arXiv:2010.12626*.

Dingding Wang, Shenghuo Zhu, Tao Li, and Yihong Gong. 2009. Multi-document summarization using sentence-based topic models. In *ACL'09*, pages 297–300.

Mu-Chun Wang, Zixuan Liu, and Sheng Wang. 2022. Textomics: A dataset for genomics data summary generation. In *ACL'22*.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Guangxu Xun, Yaliang Li, Jing Gao, and Aidong Zhang. 2017a. Collaboratively improving topic discovery and word embeddings by coordinating global and local contexts. In *KDD'17*, pages 535–543.

Guangxu Xun, Yaliang Li, Wayne Xin Zhao, Jing Gao, and Aidong Zhang. 2017b. A correlated topic model using word embeddings. In *IJCAI'17*, pages 4207–4213.

Chao Zhang, Keyang Zhang, Quan Yuan, Fangbo Tao, Luming Zhang, Tim Hanratty, and Jiawei Han. 2017. React: Online multimodal embedding for recency-aware spatiotemporal activity modeling. In *SIGIR'17*, pages 245–254.

Yu Zhang, Shweta Garg, Yu Meng, Xiusi Chen, and Jiawei Han. 2022. Motifclass: Weakly supervised text classification with higher-order metadata information. In *WSDM'22*, pages 1357–1367.

Yu Zhang, Yu Meng, Jiaxin Huang, Frank F Xu, Xuan Wang, and Jiawei Han. 2020. Minimally supervised categorization of text with metadata. In *SIGIR'20*, pages 1231–1240.

A The Embedding Learning Objective

In Section 3.2, we propose the following embedding learning objective:

$$\mathcal{J} = \underbrace{\sum_{d\in\mathcal{D}}\sum_{w_i\in d}\sum_{w_j\in\mathcal{C}(w_i,h)}\frac{\exp(\boldsymbol{u}_{w_i}^T\boldsymbol{v}_{w_j})}{\sum_{w'\in\mathcal{V}_\mathcal{D}}\exp(\boldsymbol{u}_{w_i}^T\boldsymbol{v}_{w'})}}_{\mathcal{J}_{\text{context}}} + \underbrace{\sum_{d\in\mathcal{D}}\sum_{w\in d}\frac{\exp(\boldsymbol{u}_w^T\boldsymbol{v}_d)}{\sum_{d'\in\mathcal{D}}\exp(\boldsymbol{u}_w^T\boldsymbol{v}_{d'})}}_{\mathcal{J}_{\text{document}}} + \underbrace{\sum_{c_i\in\mathcal{C}}\sum_{w\in\mathcal{S}_i}\frac{\exp(\boldsymbol{u}_w^T\boldsymbol{v}_{c_i})}{\sum_{c'\in\mathcal{C}}\exp(\boldsymbol{u}_w^T\boldsymbol{v}_{c'})}}_{\mathcal{J}_{\text{category}}}. \tag{12}$$

Now we prove that maximizing $\mathcal{J}$ is implicitly performing the factorization in Eq. (3).

Levy and Goldberg (2014) have proved that maximizing $\mathcal{J}_{\text{context}}$ is implicitly doing the following factorization.

$$\boldsymbol{u}_{w_i}^T\boldsymbol{v}_{w_j} = \log\Big(\frac{\#_\mathcal{D}(w_i,w_j)\cdot\lambda_\mathcal{D}}{\#_\mathcal{D}(w_i)\cdot\#_\mathcal{D}(w_j)\cdot b}\Big), \quad \text{i.e.,}\ \ \mathbf{U}_w^T\mathbf{V}_w = \mathbf{X}_{ww}. \tag{13}$$

We follow their approach to consider the other two terms $\mathcal{J}_{\text{document}}$ and $\mathcal{J}_{\text{category}}$ in Eq. (12). Using the negative sampling strategy to rewrite $\mathcal{J}_{\text{document}}$, we get

$$\sum_{w\in\mathcal{V}_\mathcal{D}}\sum_{d\in\mathcal{D}}\#_d(w)\Big(\log\sigma(\boldsymbol{u}_w^T\boldsymbol{v}_d)+b\mathbb{E}_{d'}\big[\log\sigma(-\boldsymbol{u}_w^T\boldsymbol{v}_{d'})\big]\Big), \tag{14}$$

where $\sigma(\cdot)$ is the sigmoid function. Following (Levy and Goldberg, 2014; Qiu et al., 2018), we assume the negative sampling distribution $\propto \lambda_d$.[15] Then, the objective becomes

$$\sum_{w\in\mathcal{V}_\mathcal{D}}\sum_{d\in\mathcal{D}}\#_d(w)\log\sigma(\boldsymbol{u}_w^T\boldsymbol{v}_d) + \sum_{w\in\mathcal{V}_\mathcal{D}}\#_\mathcal{D}(w)\sum_{d'\in\mathcal{D}}\frac{b\cdot\lambda_{d'}}{\lambda_\mathcal{D}}\log\sigma(-\boldsymbol{u}_w^T\boldsymbol{v}_{d'}). \tag{15}$$

For a specific term-document pair (w,d), we consider its effect in the objective:

$$\mathcal{J}_{w,d} = \#_d(w)\log\sigma(\boldsymbol{u}_w^T\boldsymbol{v}_d)+\#_\mathcal{D}(w)\frac{b\cdot\lambda_d}{\lambda_\mathcal{D}}\log\sigma(-\boldsymbol{u}_w^T\boldsymbol{v}_d). \tag{16}$$

Let $x_{w,d} = \boldsymbol{u}_w^T\boldsymbol{v}_d$. To maximize $\mathcal{J}_{w,d}$, we should have

$$0 = \frac{\partial\mathcal{J}_{w,d}}{\partial x_{w,d}} = \#_d(w)\sigma(-x_{w,d}) - \frac{\#_\mathcal{D}(w)\cdot b\cdot\lambda_d}{\lambda_\mathcal{D}}\sigma(x_{w,d}). \tag{17}$$

[15] In practice, the negative sampling distribution $\propto \lambda_d^{3/4}$, but related studies (Levy and Goldberg, 2014; Qiu et al., 2018) usually assume a linear relationship in their derivation.

That is,

$$e^{2x_{w,d}} - \Big(\frac{\#_d(w)\cdot\lambda_{\mathcal{D}}}{\#_{\mathcal{D}}(w)\cdot b\cdot\lambda_d} - 1\Big)e^{x_{w,d}} - \frac{\#_d(w)\cdot\lambda_{\mathcal{D}}}{\#_{\mathcal{D}}(w)\cdot b\cdot\lambda_d} = 0. \tag{18}$$

Therefore, $e^{x_{w,d}} = -1$ (which is invalid) or $e^{x_{w,d}} = \frac{\#_d(w)\cdot\lambda_{\mathcal{D}}}{\#_{\mathcal{D}}(w)\cdot b\cdot\lambda_d}$. In other words,

$$\begin{aligned} \boldsymbol{u}_w^T\boldsymbol{v}_d = x_{w,d} = \log\Big(\frac{\#_d(w)\cdot\lambda_{\mathcal{D}}}{\#_{\mathcal{D}}(w)\cdot b\cdot\lambda_d}\Big), \\ \text{i.e., } \mathbf{U}_w^T\mathbf{V}_d = \mathbf{X}_{wd}. \end{aligned} \tag{19}$$

Similarly, for $\mathcal{J}_{\text{category}}$, the objective can be rewritten as

$$\begin{aligned} &\sum_{w\in\mathcal{V}_{\mathcal{D}}}\sum_{c_i\in\mathcal{C}} \mathbf{1}_{w\in\mathcal{S}_i}\log\sigma(\boldsymbol{u}_w^T\boldsymbol{v}_{c_i}) + \\ &\sum_{w\in\mathcal{V}_{\mathcal{D}}} \mathbf{1}_{w\in\mathcal{S}_1\cup\ldots\cup\mathcal{S}_{|\mathcal{C}|}}\sum_{c'\in\mathcal{C}}\frac{b}{|\mathcal{C}|}\log\sigma(-\boldsymbol{u}_w^T\boldsymbol{v}_{c'}). \end{aligned} \tag{20}$$

Following the derivation of $\mathcal{J}_{\text{document}}$, we get

$$\begin{aligned} \boldsymbol{u}_w^T\boldsymbol{v}_{c_i} = x_{w,c_i} = \log\Big(\frac{\mathbf{1}_{w\in\mathcal{S}_i}|\mathcal{C}|}{\mathbf{1}_{w\in\mathcal{S}_1\cup\ldots\cup\mathcal{S}_{|\mathcal{C}|}}\cdot b}\Big), \\ \text{i.e., } \mathbf{U}_w^T\mathbf{V}_{c_i} = \mathbf{X}_{wc}. \end{aligned} \tag{21}$$

Putting Eqs. (13), (19), and (21) together gives us Eq. (3).

B The Ensemble Ranking Function

In Section 3.3, we propose the following ensemble ranking function:

$$\text{score}(w|\mathcal{S}_i) = \Big(\frac{1}{2}\Big(\frac{1}{\text{rank}_G(w)}\Big)^{\rho} + \frac{1}{2}\Big(\frac{1}{\text{rank}_L(w)}\Big)^{\rho}\Big)^{1/\rho}. \tag{22}$$

Now we prove this ranking function is a generalization of the *arithmetic* mean reciprocal rank (i.e., MRR) and the *geometric* mean reciprocal rank:

$$\begin{aligned} \lim_{\rho\to 1}\text{score}(w|\mathcal{S}_i) &= \frac{1}{2}\Big(\frac{1}{\text{rank}_G(w)} + \frac{1}{\text{rank}_L(w)}\Big); \\ \lim_{\rho\to 0}\text{score}(w|\mathcal{S}_i) &= \Big(\frac{1}{\text{rank}_G(w)}\cdot\frac{1}{\text{rank}_L(w)}\Big)^{1/2}. \end{aligned} \tag{23}$$

The case of $\rho\to 1$ is trivial. When $\rho\to 0$, we aim to show that

$$\lim_{\rho\to 0}\log\text{score}(w|\mathcal{S}_i) = \log\Big(\frac{1}{\text{rank}_G(w)}\cdot\frac{1}{\text{rank}_L(w)}\Big)^{1/2}. \tag{24}$$

In fact, let $r_G = \frac{1}{\text{rank}_G(w)}$ and $r_L = \frac{1}{\text{rank}_L(w)}$.

$$\begin{aligned} \lim_{\rho\to 0}\log\text{score}(w|\mathcal{S}_i) &= \lim_{\rho\to 0}\log\Big(\frac{1}{2}r_G^{\rho} + \frac{1}{2}r_L^{\rho}\Big)^{1/\rho} \\ &= \lim_{\rho\to 0}\frac{\log\big(\frac{1}{2}r_G^{\rho} + \frac{1}{2}r_L^{\rho}\big)}{\rho} \\ &= \lim_{\rho\to 0}\frac{\frac{\frac{1}{2}r_G^{\rho}\log r_G + \frac{1}{2}r_L^{\rho}\log r_L}{\frac{1}{2}r_G^{\rho} + \frac{1}{2}r_L^{\rho}}}{1} \\ &= \frac{\lim_{\rho\to 0}\Big(r_G^{\rho}\log r_G + r_L^{\rho}\log r_L\Big)}{\lim_{\rho\to 0}\Big(r_G^{\rho} + r_L^{\rho}\Big)} \\ &= \frac{\log r_G + \log r_L}{2} \\ &= \log(r_G\cdot r_L)^{1/2}. \end{aligned} \tag{25}$$

The third line is obtained by applying L'Hopital's rule.