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

Abstract

Mining latent topics from text corpora has been studied for decades. Many existing topic models adopt a fully unsupervised setting, and their retrieved topics may deviate from users' particular interests. Although some seed-guided topic mining approaches can leverage user-provided category words as guidance to discover topic representative terms, two factors are less concerned in those studies: (1) **the existence of out-of-vocabulary seeds**: user-interested categories may be specific and described by composite words/phrases that do not appear in the corpus, and (2) **the power of pre-trained language models (PLMs)**: the general knowledge learned by PLMs from web-scale corpora may complement the information a model can get from the input corpus. Being aware of these two factors, in this paper, we generalize the task of seed-guided topic mining to allow out-of-vocabulary seeds. We propose a novel framework, SEETOPIC (SEEd-guided TOPIC Mining), which jointly utilizes PLMs and seed-guided embeddings to model the semantics of in-vocabulary terms and out-of-vocabulary seeds. It features an iterative ensemble ranking process, wherein the general knowledge of PLMs and the local information learned from the input corpus mutually benefit each other. Experiments on three real datasets from different domains demonstrate the effectiveness of SEETOPIC in terms of topic coherence, accuracy, and distinctiveness. Code and datasets used in this paper are available in the Supplementary Material.

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 document summarization (Wang et al. 2009), question answering (Ji et al. 2012), text classification (Chen et al. 2015), and taxonomy construction (Zhang et al. 2018).

Unsupervised topic models have been the mainstream approach to topic discovery since the proposal of pLSA (Hofmann 1999) and LDA (Blei, Ng, and Jordan 2003). Despite their encouraging performance in finding informative latent topics, these topics are incapable of reflecting or considering user preferences, mainly due to the unsupervised nature of the methods. 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*") that may not be wanted by the user. To consider users' interests and needs, seed-guided topics mining approaches (Jagarlamudi, Daumé, and Udupa 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.[1] 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, Daumé, and Udupa 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 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: scientific papers, product reviews, and social media posts. In each dataset, a document belongs to one or more categories, and we list the category names

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

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) never seen in the corpus. Some category names are too specific (e.g., "*chronic respiratory diseases*", "*apps for android*", "*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*").[2]

The Power of Pre-trained Language Models (PLMs). Techniques used in previous studies are mainly based on LDA variants (Jagarlamudi, Daumé, and Udupa 2012) or context-free embeddings (Meng et al. 2020a). Recently, PLMs such as BERT (Devlin et al. 2019) and ELECTRA (Clark et al. 2020) have revolutionized the NLP field and achieved significant improvement in a wide range of text mining tasks. In topic mining, the generic representation power of PLMs learned from web-scale corpora (e.g., Wikipedia and 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 can help segment the seeds into such meaningful components (e.g., "*nightlife*" $\rightarrow$ "*night*" and "*life*"). 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.

Present Work. Being aware of these two factors, in this paper, we study seed-guided topic mining in the presence of out-of-vocabulary seeds. Our proposed SEETOPIC framework consists of two modules: (1) The *generic* representation module, which 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. Most PLMs have a pre-trained tokenizer (Sennrich, Haddow, and Birch 2016; Wu et al. 2016) that can segment new terms into subwords. Then, the contextualization power of PLMs can help us obtain accurate semantics of each word/subword. For example, PLMs can infer the meaning of "*care*" in "*health and personal care*" after seeing "*health*", yielding a more precise representation of the whole category. (2) The *seed-guided* representation module, which learns the 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; Lau, Newman, and Baldwin 2014), our objective aims to maximize 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) *Conceptual*: a novel framework, SEETOPIC, is proposed for the seed-guided topic mining task in the presence of out-of-vocabulary seeds. (2) *Methodological*: an ensemble ranking approach is designed to jointly leverage PLMs and seed-guided embedding learning to model general and local text semantics; and (3) *Experimental*: extensive experiments conducted on three real-world datasets from different domains show the effectiveness of SEETOPIC in terms of topic coherence, term accuracy, and topic distinctiveness.

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 $\mathcal{S}_i = \{w_{i1}, ..., w_{iS}\} \subseteq \mathcal{V}_\mathcal{D}$ 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 ($\forall j \neq i$).*

The SEETOPIC Framework

In this section, we first introduce how we model general and local text semantics using the PLM and our proposed seed-guided ensemble learning method. Then, we present the iterative ensemble ranking process and our overall framework.

Modeling General Text Semantics using the PLM

PLMs such as GPT-2 (Peters et al. 2018), BERT (Devlin et al. 2019), and ELECTRA (Clark et al. 2020) aim to learn generic language representations from web-scale corpora (e.g., Wikipedia (Devlin et al. 2019), PubMed (Lee et al. 2020), and GitHub (Feng et al. 2020)) that can be applied to a wide variety of text-related applications.

To transfer such general knowledge to our topic mining 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, [CLS] and [SEP] are two special tokens used by PLMs at the beginning and the end of a sentence, respectively.) Note that w can be a phrase containing multiple words, and each word can be out of the PLM's vocabulary. Thus, most PLMs have a pre-trained tokenizer (Sennrich, Haddow, and Birch 2016; Wu et al. 2016) to segment each unseen word into frequent subwords. For example, for the category name "*nightlife spot*" in Table 1, the

[2] One possible idea to deal with composite seeds is to split them into multiple seeds. However, we can see flexible ways of conjunction in Table 1 such as "/", "()", "," and "and". Also, we have cases like "professional and other places" which should be split into "professional places" 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 related to "*hepatitis b*" and "*hepatitis c*".

input sequence will be "[CLS] *night ##life spot* [SEP]" if we adopt BERT as the PLM. After tokenization, the contextualization power of the PLM will help model the semantics of each word/subword. To give another example, the PLM can infer the context-aware meaning of "*care*" in "*health and personal care*" after seeing "*health*", so as to provide a more precise representation of the whole category.

After LM encoding, following (Sia, Dalmia, and Mielke 2020; Thompson and Mimno 2020), we take the output of all tokens (including "[CLS]" and "[SEP]") from the last layer and average them to get the term embedding $\boldsymbol{e}_w$. In this way, even if a seed category name c_i is **out-of-vocabulary**, we can still obtain its representation $\boldsymbol{e}_{c_i}$ according to the general knowledge learned by the PLM from web-scale corpora.

Modeling Local Text Semantics in the Corpus

The motivation of topic mining is to discover latent topic structures from the input corpus. Therefore, purely relying on general knowledge in the PLM is insufficient because topic mining results should adapt to the input corpus $\mathcal{D}$. Therefore, this section introduces how we learn another set of embeddings $\{\boldsymbol{u}_w | w \in \mathcal{V}_\mathcal{D}\}$ from $\mathcal{D}$. Inspired by some standard evaluation metrics for topic modeling based on pointwise mutual information (PMI) (Newman et al. 2010; Lau, Newman, and Baldwin 2014), we aim to maximize PMI during embedding learning.

Our starting point, as suggested by the Skip-Gram model (Mikolov et al. 2013), is that similar terms share similar contexts in the input corpus. For each term $w \in \mathcal{V}_\mathcal{D}$, we assume $\boldsymbol{u}_w$ represent the embedding of w as a "center", and we use $\boldsymbol{v}_w$ to denote the embedding of w as a "context". Levy and Goldberg (2014) prove that the Skip-Gram model is implicitly factorizing the PMI matrix. Formally, it is expected that

$$\boldsymbol{u}_w^T \boldsymbol{v}_z = \mathbf{X}_{wz} = \log\Big(\frac{\#_\mathcal{D}(w,z)\cdot|\mathcal{D}|}{\#_\mathcal{D}(w)\cdot\#_\mathcal{D}(z)\cdot b}\Big), \ \ (\forall w,z \in \mathcal{V}_\mathcal{D}) \tag{1}$$

where $\#_\mathcal{D}(w,z)$ denotes the number of co-occurrences of w and z in the context window in $\mathcal{D}$; $\#_\mathcal{D}(w)$ denotes the number of occurrences of w in $\mathcal{D}$; $|\mathcal{D}|$ is the total number of terms in $\mathcal{D}$; b is the number of negative samples.

Previous studies (Tang, Qu, and Mei 2015; Xun et al. 2017a; Meng et al. 2020a) also demonstrate the importance of modeling document-word proximity. They assume that words in similar documents are topic-coherent. Let $\boldsymbol{v}_d$ be the embedding of document d. One can revise Eq. (1) to describe this assumption:

$$\boldsymbol{u}_w^T \boldsymbol{v}_d = \mathbf{Y}_{wd} = \log\Big(\frac{\#_d(w)\cdot|d|}{\#_\mathcal{D}(w)\cdot|\mathcal{D}|\cdot b}\Big), \ \ (\forall w \in \mathcal{V}_\mathcal{D}, d \in \mathcal{D}) \tag{2}$$

where $\#_d(w)$ denotes the number of times w occurs in d; $|d|$ is the total number of terms in d.

Note that our topic mining task is guided by user-provided seeds. Therefore, the embedding space should be regularized to encourage distinctiveness of different categories. In other words, we expect each category c_i to be surrounded by its representative terms $\mathcal{S}_i$ in the embedding space. We adopt an iterative process to gradually update category-representative terms. Initially, $\mathcal{S}_i$ consists of just a few in-vocabulary terms similar with 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. By revising Eq. (1), we have

$$\boldsymbol{u}_w^T \boldsymbol{v}_{c_i} = \mathbf{Z}_{w,c_i} = \begin{cases} 0, & \text{if } w \notin \mathcal{S}_1 \cup ... \cup \mathcal{S}_{|\mathcal{C}|}, \\ -\infty, & \text{if } w \in \mathcal{S}_j \ (\forall j \neq i), \\ \log\frac{|\mathcal{C}|}{b}, & \text{if } w \in \mathcal{S}_i. \end{cases} \tag{3}$$

Directly learning embeddings using Eqs. (1), (2) and (3) is not easy because some entries in $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{Z}$ are $-\infty$ (e.g., if $\#_\mathcal{D}(w,z) = 0$ or $\#_d(w) = 0$). Therefore, we try to optimize the negative sampling loss instead. Given a pair (w,z), its local objective is

$$\begin{aligned} l(w,z) &= \#_\mathcal{D}(w,z)\log\sigma(\boldsymbol{u}_w^T\boldsymbol{v}_z) + \\ &\quad b\cdot\#_\mathcal{D}(w)\cdot\frac{\#_\mathcal{D}(z)}{|\mathcal{D}|}\log\sigma(-\boldsymbol{u}_w^T\boldsymbol{v}_z) \\ &\propto \exp(\mathbf{X}_{wz})\log\sigma(\boldsymbol{u}_w^T\boldsymbol{v}_z) + \sigma(-\boldsymbol{u}_w^T\boldsymbol{v}_z). \end{aligned} \tag{4}$$

Solving $\frac{\partial l(w,z)}{\partial \boldsymbol{u}_w} = 0$ or $\frac{\partial l(w,z)}{\partial \boldsymbol{v}_z} = 0$ will give us Eq. (1). Note that we no longer have $-\infty$ in this objective because $\exp(-\infty) = 0$. Similarly, we can have

$$\begin{aligned} l(w,d) &\propto \exp(\mathbf{Y}_{wd})\log\sigma(\boldsymbol{u}_w^T\boldsymbol{v}_d) + \sigma(-\boldsymbol{u}_w^T\boldsymbol{v}_d), \\ l(w,c_i) &\propto \exp(\mathbf{Z}_{w,c_i})\log\sigma(\boldsymbol{u}_w^T\boldsymbol{v}_{c_i}) + \sigma(-\boldsymbol{u}_w^T\boldsymbol{v}_{c_i}). \end{aligned} \tag{5}$$

To learn the embeddings $\boldsymbol{u}_w$, $\boldsymbol{v}_w$, $\boldsymbol{v}_d$, and $\boldsymbol{v}_{c_i}$, we optimize the three objectives $l(w,z)$, $l(w,d)$, and $l(w,c_i)$ by turns. Through optimizing one objective (e.g., $l(w,z)$), we can update embeddings (e.g., $\boldsymbol{u}_w$ and $\boldsymbol{v}_z$) using gradient descent.

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 general knowledge learned by the PLM, while $\{\boldsymbol{u}_w | w \in \mathcal{V}_\mathcal{D}\}$ carries local information from 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 $w \in \mathcal{V}_\mathcal{D}$ as:

$$\begin{aligned} \text{score}_G(w|\mathcal{S}_i) &= \frac{1}{|\mathcal{S}_i|}\sum_{w'\in\mathcal{S}_i}\cos(\boldsymbol{e}_w, \boldsymbol{e}_{w'}), \\ \text{score}_L(w|\mathcal{S}_i) &= \frac{1}{|\mathcal{S}_i|}\sum_{w'\in\mathcal{S}_i}\cos(\boldsymbol{u}_w, \boldsymbol{u}_{w'}). \end{aligned} \tag{6}$$

Here, the subscript "G" means "according to general knowledge", while "L" means "according to the local corpus". 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:

$$\text{score}(w|\mathcal{S}_i) = \Big(\frac{1}{2}\Big(\frac{1}{\text{rank}_G(w)}\Big)^\rho + \frac{1}{2}\Big(\frac{1}{\text{rank}_L(w)}\Big)^\rho\Big)^{1/\rho}. \tag{7}$$

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

Algorithm 1: SEETOPIC

Input: A text corpus $\mathcal{D} = \{d_1, ..., d_{|\mathcal{D}|}\}$, a set of category names $\mathcal{C} = \{c_1, ..., c_{|\mathcal{C}|}\}$, 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 $\mathcal{S}_i \leftarrow \mathcal{S}_i \cup \{\underset{w \in \mathcal{V}_\mathcal{D} \setminus (\mathcal{S}_1 \cup ... \cup \mathcal{S}_{|\mathcal{C}|})}{\arg\max} \cos(\boldsymbol{e}_w, \boldsymbol{e}_{c_i})\}$;
7 // Update $\mathcal{S}_i$ for T iterations;
8 **for** $t \leftarrow 1$ *to* T **do**
9 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 Eqs. (4) and (5);
10 $\text{score}_G(w|\mathcal{S}_i)$ and $\text{score}_L(w|\mathcal{S}_i) \leftarrow$ Eq. (6);
11 $\text{score}(w|\mathcal{S}_i) \leftarrow$ Eq. (7);
12 $\mathcal{S}_1, ..., \mathcal{S}_{|\mathcal{C}|} \leftarrow \emptyset$;
13 **for** $n \leftarrow 1$ *to* $(t+1)N$ **do**
14 **for** $i \leftarrow 1$ *to* $|\mathcal{C}|$ **do**
15 $\mathcal{S}_i \leftarrow \mathcal{S}_i \cup \{\underset{w \in \mathcal{V}_\mathcal{D} \setminus (\mathcal{S}_1 \cup ... \cup \mathcal{S}_{|\mathcal{C}|})}{\arg\max} \text{score}(w|\mathcal{S}_i)\}$;
16 Return $(\mathcal{S}_1, ..., \mathcal{S}_{|\mathcal{C}|})$;

top-M according to $\text{score}_G(w)$, we set $\text{rank}_G(w) = +\infty$ (i.e., $\frac{1}{\text{rank}_G(w)} = 0$). Similarly, if $\text{rank}_L(w) > M$, we set $\text{rank}_L(w) = +\infty$ (i.e., $\frac{1}{\text{rank}_L(w)} = 0$).

Eq. (7) is in the form of *Hölder* mean. In fact,

$$\begin{aligned} &\text{if } \rho = 1,\ \text{score}(w|\mathcal{S}_i) = \frac{1}{2}\Big(\frac{1}{\text{rank}_G(w)} + \frac{1}{\text{rank}_L(w)}\Big); \\ &\text{if } \rho \to 0,\ \text{score}(w|\mathcal{S}_i) \to \Big(\frac{1}{\text{rank}_G(w)} \cdot \frac{1}{\text{rank}_L(w)}\Big)^{1/2}. \end{aligned} \tag{8}$$

In other words, when $\rho = 1$, Eq. (7) becomes the *arithmetic* mean of the two reciprocal ranks (i.e., 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. (7) becomes the *geometric* mean of the two reciprocal ranks, 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 $\rho = 0.5$ and show it outperforms MRR (i.e., $\rho = 1$) in our task.

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:

$$\mathcal{S}_i = \mathcal{S}_i \cup \{\underset{w \in \mathcal{V}_\mathcal{D} \setminus (\mathcal{S}_1 \cup ... \cup \mathcal{S}_{|\mathcal{C}|})}{\arg\max} \text{score}(w|\mathcal{S}_i)\}. \tag{9}$$

Overall Framework

We summarize the entire SEETOPIC framework in Algorithm 1. To deal with **out-of-vocabulary** category names, we first utilize the PLM to find their nearest in-vocabulary terms as the initial category-discriminative term set $\mathcal{S}_i$ (Lines 1-6). After initialization, $|\mathcal{S}_i| = N$ $(\forall 1 \le i \le |\mathcal{C}|)$. 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 7-15). At each iteration, according to the up-to-date $\mathcal{S}_1, \mathcal{S}_2, ..., \mathcal{S}_{|\mathcal{C}|}$, the embeddings $\boldsymbol{u}_w$ and $\boldsymbol{v}_{c_i}$ learned in previous iterations no longer satisfy Eq. (3). Therefore, we relearn embeddings $\boldsymbol{u}_w$, $\boldsymbol{v}_w$, $\boldsymbol{v}_d$, and $\boldsymbol{v}_{c_i}$ based on the input corpus $\mathcal{D}$ and the up-to-date $\mathcal{S}_1, ..., \mathcal{S}_{|\mathcal{C}|}$ (Line 9). 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 9), are then leveraged to perform ensemble ranking (Lines 10-11). Based on the ensemble score $\text{score}(w|\mathcal{S}_i)$, we update $\mathcal{S}_i$ using Eq. (9) (Lines 12-15). After the t-th iteration, $|\mathcal{S}_i| = (t+1)N$ $(\forall 1 \le i \le |\mathcal{C}|)$.

Table 2: Dataset Statistics. OOV: Out-of-vocabulary.

Dataset	SciDocs	Amazon	Twitter
#Documents	23,473	100,000	135,529
Avg Doc Length	239.8	119.0	6.7
#Categories	11	10	9
#OOV Category Names (After Phrase Chunking)	5	6	7

Experiments

Experimental Setup

Datasets. We conduct experiments on three public datasets from different domains.

- **SciDocs (Cohan et al. 2020)**[3] 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 these papers (title + abstract) as the input corpus and the 11 category names as seeds.
- **Amazon (McAuley and Leskovec 2013)**[4] contains product reviews spanning May 1996 – July 2014. Each Amazon review belongs to one or more product categories. We select 10 large categories and sample 10K reviews from each category.
- **Twitter (Zhang et al. 2017)**[5] contains 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 have been shown in Table 1. Dataset statistics are summarized in Table 2. For all three datasets, we use AutoPhrase (Shang et al. 2018)[6] to detect phrases in the corpus.

Previous studies (Jagarlamudi, Daumé, and Udupa 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

[3] https://github.com/allenai/scidocs
[4] http://jmcauley.ucsd.edu/data/amazon/index.html
[5] https://github.com/franticnerd/geoburst
[6] https://github.com/shangjingbo1226/AutoPhrase

Table 3: PMI, NPMI, and MACC of compared algorithms on three datasets. PMI and NPMI measure topic coherence; MACC measures term accuracy. BERT and BioBERT do not have the standard deviation because they are deterministic according to our usage. *: significantly worse than SEETOPIC (p-value < 0.05). **: significantly worse than SEETOPIC (p-value < 0.01).

Methods	SciDocs			Amazon			Twitter		
	PMI	NPMI	MACC	PMI	NPMI	MACC	PMI	NPMI	MACC
SeededLDA	$10.5 \pm 0.6^{**}$	$4.98 \pm 0.14^{**}$	$0.158 \pm 0.019^{**}$	$20.1 \pm 0.8^{**}$	$6.24 \pm 0.14^{**}$	$0.173 \pm 0.055^{**}$	$55.4 \pm 4.8^{**}$	$5.08 \pm 0.34^{**}$	$0.196 \pm 0.045^{**}$
Anchored CorEx	$43.1 \pm 4.5^{**}$	$9.02 \pm 0.72^{**}$	$0.291 \pm 0.072^{**}$	$59.8 \pm 3.6^{**}$	$11.89 \pm 0.35^{**}$	$0.340 \pm 0.020^{**}$	$71.9 \pm 9.9^{**}$	$7.43 \pm 0.74^{**}$	$0.211 \pm 0.051^{**}$
Labeled ETM	185.4 ± 3.3	$18.78 \pm 0.31^{*}$	$0.458 \pm 0.037^{**}$	217.0 ± 1.0	19.56 ± 0.07	$0.550 \pm 0.017^{**}$	$248.4 \pm 4.1^{*}$	$21.10 \pm 0.32^{*}$	$0.304 \pm 0.017^{**}$
CatE	$172.3 \pm 2.3^{**}$	$18.66 \pm 0.27^{*}$	$0.670 \pm 0.010^{**}$	$194.7 \pm 3.0^{**}$	$18.51 \pm 0.30^{**}$	$0.797 \pm 0.021^{*}$	252.0 ± 3.3	21.39 ± 0.27	$0.374 \pm 0.053^{**}$
BERT	179.9^{**}	18.19^{**}	0.700^{**}	195.2^{**}	17.62^{**}	0.840^{*}	241.3^{**}	20.49^{**}	**0.667**
BioBERT	172.3^{**}	17.75^{**}	0.891^{*}	–	–	–	–	–	–
SEETOPIC-NoIter	$185.3 \pm 1.2^{*}$	$19.09 \pm 0.12^{*}$	0.903 ± 0.014	218.8 ± 0.5	19.57 ± 0.04	0.833 ± 0.015	$252.2 \pm 1.2^{*}$	$21.45 \pm 0.12^{*}$	0.611 ± 0.011
SEETOPIC	**189.7 ± 2.0**	**19.40 ± 0.15**	**0.915 ± 0.019**	**221.6 ± 4.1**	**19.75 ± 0.33**	**0.847 ± 0.006**	**256.4 ± 2.0**	**21.80 ± 0.17**	0.622 ± 0.011

in-vocabulary terms. Therefore, we do not consider these datasets for evaluation in our task settings.

Compared Methods. We compare our SEETOPIC framework with the following methods, including seed-guided topic modeling methods, seed-guided embedding learning methods, and PLMs.

- **SeededLDA (Jagarlamudi, Daumé, and Udupa 2012)**[7] is a seed-guided topic modeling method. It improves LDA by biasing topics to produce input seeds and by biasing documents to select topics related to the seeds they contain, so that the learned topics are of interest to the user.
- **Anchored CorEx (Gallagher et al. 2017)**[8] is a seed-guided topic modeling method. It incorporates user-provided seeds by balancing between compressing the input corpus and preserving seed-related information.
- **Labeled ETM (Dieng, Ruiz, and Blei 2020)**[9] is an embedding-based topic model. It leverages distributed representation of each term to make the model more robust to rare words and stop words. Following (Meng et al. 2020a), we retrieve representative terms according to their embedding similarity with the category name.
- **CatE (Meng et al. 2020a)**[10] is a seed-guided embedding learning method for discriminative topic mining. 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.
- **BERT (Devlin et al. 2019)**[11] is a PLM. Following Lines 1-6 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 representative terms.
- **BioBERT (Lee et al. 2020)**[12] 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.
- **SEETOPIC-NoIter** is a variant of our SEETOPIC framework. In Algorithm 1, after initialization (Lines 1-6), it executes Lines 8-15 only once (i.e., $T = 1$) to find all representative terms.

[7] https://github.com/vi3k6i5/GuidedLDA

[8] https://github.com/gregversteeg/corex_topic

[9] https://github.com/adjidieng/ETM

[10] https://github.com/yumeng5/CatE

[11] https://huggingface.co/bert-base-uncased

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

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 1-6 in Algorithm 1 to get the initial representative in-vocabulary terms for each category, and input these terms as seeds into the baselines. [13]

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

- **PMI (Newman et al. 2010)** is a standard metric in topic modeling to measure topic coherence. Within each topic, it calculates the pointwise mutual information for each pair of terms in $\mathcal{S}_i$:

$$\text{PMI} = \frac{1}{|\mathcal{C}|}\sum_{i=1}^{|\mathcal{C}|}\sum_{w_j,w_k\in\mathcal{S}_i}\log\frac{P(w_j,w_k)}{P(w_j)P(w_k)}, \tag{10}$$

  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.

- **NPMI (Lau, Newman, and Baldwin 2014)** is another standard metric to evaluate topic coherence. It revises PMI with the normalized pointwise mutual information:

$$\text{NPMI} = \frac{1}{|\mathcal{C}|}\sum_{i=1}^{|\mathcal{C}|}\sum_{w_j,w_k\in\mathcal{S}_i}\frac{\log\frac{P(w_j,w_k)}{P(w_j)P(w_k)}}{-\log P(w_j,w_k)}. \tag{11}$$

- **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:

$$\text{MACC} = \frac{1}{|\mathcal{C}|}\sum_{i=1}^{|\mathcal{C}|}\frac{1}{|\mathcal{S}_i|}\sum_{w_j\in\mathcal{S}_i}\mathbf{1}(w_j\in c_i), \tag{12}$$

  where $\mathbf{1}(w_j \in c_i)$ is the indicator function of whether w_j is related to category c_i. MACC requires human evaluation, so we invite two graduate students to perform independent annotation. We set $\mathbf{1}(w_j \in c_i) = 1$ if and only if both annotators agree on it.

- **Distinctiveness** measures how well the topics are distinc-

[13] This means ALL compared methods use BERT/BioBERT to initialize their term sets. Without the PLM, the baselines can hardly deal with out-of-vocabulary seeds and should have lower scores.

Table 4: Distinctiveness of compared algorithms on three datasets. BERT and BioBERT do not have the standard deviation because they are deterministic according to our usage.

Methods	SciDocs	Amazon	Twitter
	Distinctiveness	Distinctiveness	Distinctiveness
SeededLDA	$0.451 \pm 0.019^{**}$	$0.393 \pm 0.012^{**}$	$0.696 \pm 0.028^{**}$
Anchored CorEx	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$
Labeled ETM	$0.961 \pm 0.014^{**}$	$\mathbf{1.000 \pm 0.000}$	0.989 ± 0.011
CatE	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$
BERT	0.909^{**}	**1.000**	0.978^{**}
BioBERT	**1.000**	–	–
SeeTopic-NoIter	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$
SeeTopic	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$

tive from each other. It is defined as

$$\text{Distinctiveness} = \frac{|\bigcup_{i=1}^{|\mathcal{C}|} \mathcal{S}_i|}{\sum_{i=1}^{|\mathcal{C}|} |\mathcal{S}_i|}. \quad (13)$$

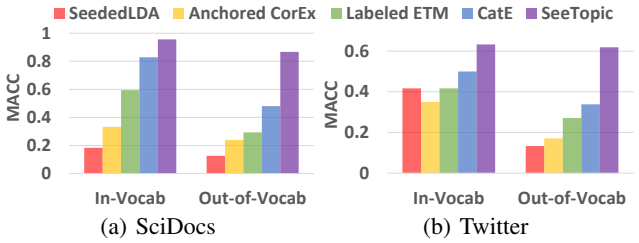
Parameter Settings. We use BioBERT as the PLM on SciDocs, and BERT-base as the PLM on Amazon and Twitter. For embedding learning in the input corpus, embedding dimension $= 768$ (the same as PLMs); the number of negative samples $b = 5$. For ensemble ranking, the length of the general/local ranking list $M = 100$; the hyperparameter ρ in Eq. (7) is set as 0.5; the number of iterations $T = 3$; 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. (10)-(13)).

Performance Comparison

Tables 3 and 4 show the performance of all methods. We run each experiment 3 times with mean and standard deviation reported. BERT and BioBERT do not have the standard deviation because they are deterministic according to our usage. To show statistical significance, we conduct a two-tailed unpaired t-test to compare SeeTopic and each baseline. (When comparing SeeTopic with the deterministic BERT and BioBERT models, we conduct a two-tailed Z-test instead.) The significance level of each result is marked in the Tables.

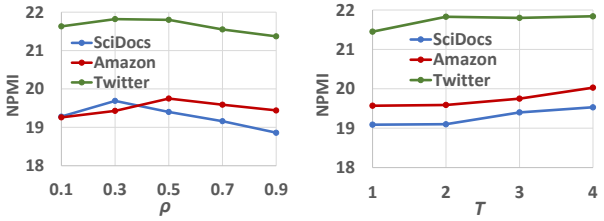
In Table 3, classical seed-guided topic modeling baselines (i.e., SeededLDA and Anchored CorEx) perform not well in respect of topic coherence and term accuracy. Embedding-based topic mining approaches make some progress: Labeled ETM has competitive PMI and NPMI scores; CatE achieves higher MACC scores, but still significantly lower than models using PLMs. Our SeeTopic has the highest PMI, NPMI, and MACC scores in most cases. It also consistently outperforms SeeTopic-NoIter on all three datasets, indicating the positive contribution of the proposed iterative process. From Table 4, we see that SeeTopic guarantees the mutual exclusivity of $\mathcal{S}_1, ..., \mathcal{S}_{|\mathcal{C}|}$. This is well-aligned with our task requirement that each retrieved term is discriminatively close to one category and far from the others. In comparison, SeededLDA, Labeled ETM, and BERT cannot guarantee such topic distinctiveness.

Figure 1 compares the MACC scores of different seed-guided topic mining methods on in-vocabulary categories and out-of-vocabulary categories. We find that, in comparison with the baselines, the performance improvement of SeeTopic on out-of-vocabulary categories is significantly larger than that on in-vocabulary categories. The MACC scores of SeeTopic on out-of-vocabulary categories are already close to those on in-vocabulary ones. Note that all baselines compared 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) Twitter

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



(a) Effect of ρ (b) Effect of T

Figure 2: Parameter study of SeeTopic measured by NPMI.

Parameter Study

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

According to Figure 2(a), SeeTopic achieves the best performance with different ρ values on the three datasets, but setting $\rho = 0.5$ always leads to competitive NPMI scores. In contrast, when $\rho \to 1$, the performance is suboptimal. This finding indicates that replacing the mean reciprocal rank (i.e., $\rho = 1$) with our proposed Eq. (7) is reasonable. According to Figure 2(b), SeeTopic has higher NPMI scores when there are more iterations. On SciDocs and Twitter, the scores start to converge after $T = 3$. Besides, more iterations will result in longer running time. Overall, we believe setting $T = 3$ strikes a good balance.

Case Study

Finally, we perform a qualitative comparison of different algorithms by showing their retrieved terms. Due to space limit, we select two out-of-vocabulary categories "*hepatitis a/b/c/e*" and "*sports and outdoors*" from SciDocs and Amazon, respectively. The results are shown in Table 5.

For the category "*hepatitis a/b/c/e*", SeededLDA and Anchored CorEx can only find very general medical terms, which is relevant to all seeds in SciDocs and not category-discriminative; Labeled ETM and CatE pick some terms related to "*alanine aminotransferase*", whose elevation suggest not only hepatitis but also other diseases like diabetes and heart failure, thus not discriminative either; BioBERT

Table 5: Top-5 representative terms retrieved by different algorithms on two out-of-vocabulary categories.

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	hepatitis b virus hbv dna (✓), normal alanine aminotransferase (✗), naive chronic hepatitis b patients (✓), hbeag-negative (✓), hbv dna (✓)
BioBERT	hepatitis b virus hbv dna (✓), hepatitis b virus hbv infection (✓), chronic hepatitis b virus hbv infection (✓), hepatitis b e antigen hbeag (✓), hepatitis c virus hcv infection (✓)
SeeTopic-NoIter	hepatitis b virus hbv dna (✓), hepatitis b e antigen hbeag (✓), chronic hepatitis b virus hbv infection (✓), hepatitis b virus hbv infection (✓), hepatitis b surface antigen hbsag (✓)
SeeTopic	hepatitis b virus hbv infection (✓), naive chronic hepatitis b patients (✓), chronic hepatitis b virus hbv infection (✓), hepatitis b virus hbv dna (✓), hepatitis b e antigen hbeag (✓)
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	cars and tracks (✓), outdoorsy (✓), mma (✓), quick drying (✗), sport (✓)
BERT	sports (✓), outdoor activities (✓), cars and tracks (✓), athletics (✓), special events (✗)
SeeTopic-NoIter	outdoor activities (✓), sports (✓), cars and tracks (✓), mma (✓), special events (✗)
SeeTopic	mma (✓), sports (✓), volleyball (✓), basketball (✓), weightlifting (✓)

and SeeTopic, with the power of PLMs, can accurately find terms related to "*hepatitis b*" and "*hepatitis c*". For the category "*sports and outdoors*", BERT makes a mistake by including a general term "*special events*"; SeeTopic-NoIter, without an iterative update process, also includes this error; the full SeeTopic model lets BERT and local text semantics iteratively benefit each other and finally excludes the error. Moreover, most terms retrieved by BERT are not quite specific (e.g., "*sports*", "*outdoor activities*"), while SeeTopic can find more specific and informative terms (e.g., "*mma*", "*volleyball*", "*basketball*", "*weightlifting*").

Related Work

We review two lines of related work: seed-guided topic modeling and embedding-based topic mining.

Seed-Guided Topic Modeling

Seed-guided topic models aim to leverage user-provided seeds to discover underlying topics according to users' interests. Early studies along this direction take LDA (Blei, Ng, and Jordan 2003) as the backbone and incorporate seed information into model learning. For example, Andrzejewski et al. (2009) consider must-link and cannot-link constraints among seeds as priors. SeededLDA (Jagarlamudi, Daumé, and Udupa 2012) encourages topics to contain more seeds and encourages documents to select topics related 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. JoSH (Meng et al. 2020b) further proposes taxonomy-guided text embedding for hierarchical topic mining. 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 Mining

A number of studies have proposed to 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, Zaheer, and Dyer 2015), LFTM (Nguyen et al. 2015), Spherical HDP (Batmanghelich et al. 2016), and CGTM (Xun et al. 2017b)). There are also studies thinking 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, Ruiz, and Blei 2020) models word-topic correlations via word embeddings to improve the expressiveness of topic models. More recent studies (Sia, Dalmia, and Mielke 2020; Thompson and Mimno 2020) show that directly clustering word embeddings (e.g., word2vec or BERT) also generates good topics. However, these models are unsupervised and hard to be applied to seed-guided settings. In contrast, our SeeTopic framework joint leverages PLMs, word embeddings, topics, and seed information.

Conclusions and Future Work

In this paper, we study seed-guided topic mining in the presence of out-of-vocabulary seeds. To understand and make use of the in-vocabulary components in each seed, we utilize the tokenization and contextualization power of the PLM. 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 the general knowledge of the PLM and the local signal learned from the input corpus. Experimental results show that SeeTopic outperforms seed-guided topic mining baselines and PLMs in terms of topic coherence, term accuracy, and topic distinctiveness. Parameter study and case study further validate some design choices in SeeTopic.

There are two possible future directions we would like to explore. First, we are interested in extending SeeTopic to seed-guided hierarchical topic mining, where the parent and child information in the input category hierarchy can help infer the meaning of out-of-vocabulary nodes. Second, we may extend SeeTopic to not only focus on user-provided seeds but also discover other latent topics that are "siblings" of the provided categories, and it is worth exploring whether and how PLMs can play a role in this setting.

References

Andrzejewski, D.; Zhu, X.; and Craven, M. 2009. Incorporating domain knowledge into topic modeling via dirichlet forest priors. In *ICML'09*, 25–32.

Batmanghelich, K.; Saeedi, A.; Narasimhan, K.; and Gershman, S. 2016. Nonparametric spherical topic modeling with word embeddings. In *ACL'16*, 537–542.

Blei, D. M.; Ng, A. Y.; and Jordan, M. I. 2003. Latent dirichlet allocation. *JMLR* 3:993–1022.

Chen, X.; Xia, Y.; Jin, P.; and Carroll, J. 2015. Dataless text classification with descriptive lda. In *AAAI'15*, 2224–2231.

Clark, K.; Luong, M.; Le, Q. V.; and Manning, C. D. 2020. ELECTRA: pre-training text encoders as discriminators rather than generators. In *ICLR'20*.

Cohan, A.; Feldman, S.; Beltagy, I.; Downey, D.; and Weld, D. S. 2020. Specter: Document-level representation learning using citation-informed transformers. In *ACL'20*, 2270–2282.

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

Das, R.; Zaheer, M.; and Dyer, C. 2015. Gaussian lda for topic models with word embeddings. In *ACL'15*, 795–804.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT'19*, 4171–4186.

Dieng, A. B.; Ruiz, F. J.; and Blei, D. M. 2020. Topic modeling in embedding spaces. *TACL* 8:439–453.

Feng, Z.; Guo, D.; Tang, D.; Duan, N.; Feng, X.; Gong, M.; Shou, L.; Qin, B.; Liu, T.; Jiang, D.; et al. 2020. Codebert: A pre-trained model for programming and natural languages. In *EMNLP'20, Findings*, 1536–1547.

Gallagher, R. J.; Reing, K.; Kale, D.; and Ver Steeg, G. 2017. Anchored correlation explanation: Topic modeling with minimal domain knowledge. *TACL* 5:529–542.

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

Hofmann, T. 1999. Probabilistic latent semantic indexing. In *SIGIR'99*, 50–57.

Jagarlamudi, J.; Daumé, H.; and Udupa, R. 2012. Incorporating lexical priors into topic models. In *EACL'12*, 204–213.

Ji, Z.; Xu, F.; Wang, B.; and He, B. 2012. Question-answer topic model for question retrieval in community question answering. In *CIKM'12*, 2471–2474.

Lau, J. H.; Newman, D.; and Baldwin, T. 2014. Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality. In *EACL'14*, 530–539.

Lee, J.; Yoon, W.; Kim, S.; Kim, D.; Kim, S.; So, C. H.; and Kang, J. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics* 36(4):1234–1240.

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

Liu, Y.; Liu, Z.; Chua, T.-S.; and Sun, M. 2015. Topical word embeddings. In *AAAI'15*, 2418–2424.

Manning, C. D.; Surdeanu, M.; Bauer, J.; Finkel, J. R.; Bethard, S.; and McClosky, D. 2014. The stanford corenlp natural language processing toolkit. In *ACL'14, System Demonstrations*, 55–60.

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

Meng, Y.; Huang, J.; Wang, G.; Wang, Z.; Zhang, C.; Zhang, Y.; and Han, J. 2020a. Discriminative topic mining via category-name guided text embedding. In *WWW'20*, 2121–2132.

Meng, Y.; Zhang, Y.; Huang, J.; Zhang, Y.; Zhang, C.; and Han, J. 2020b. Hierarchical topic mining via joint spherical tree and text embedding. In *KDD'20*, 1908–1917.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS'13*, 3111–3119.

Newman, D.; Lau, J. H.; Grieser, K.; and Baldwin, T. 2010. Automatic evaluation of topic coherence. In *NAACL-HLT'10*, 100–108.

Nguyen, D. Q.; Billingsley, R.; Du, L.; and Johnson, M. 2015. Improving topic models with latent feature word representations. *TACL* 3:299–313.

Peters, M. E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; and Zettlemoyer, L. 2018. Deep contextualized word representations. In *NAACL-HLT'18*, 2227–2237.

Sennrich, R.; Haddow, B.; and Birch, A. 2016. Neural machine translation of rare words with subword units. In *ACL'16*, 1715–1725.

Shang, J.; Liu, J.; Jiang, M.; Ren, X.; Voss, C. R.; and Han, J. 2018. Automated phrase mining from massive text corpora. *IEEE TKDE* 30(10):1825–1837.

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

Tang, J.; Qu, M.; and Mei, Q. 2015. Pte: Predictive text embedding through large-scale heterogeneous text networks. In *KDD'15*, 1165–1174.

Thompson, L., and Mimno, D. 2020. Topic modeling with contextualized word representation clusters. *arXiv preprint arXiv:2010.12626*.

Wang, D.; Zhu, S.; Li, T.; and Gong, Y. 2009. Multi-document summarization using sentence-based topic models. In *ACL'09*, 297–300.

Wu, Y.; Schuster, M.; Chen, Z.; Le, Q. V.; Norouzi, M.; Macherey, W.; Krikun, M.; Cao, Y.; Gao, Q.; Macherey, K.; et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Xun, G.; Li, Y.; Gao, J.; and Zhang, A. 2017a. Collaboratively improving topic discovery and word embeddings by coordinating global and local contexts. In *KDD'17*, 535–543.

Xun, G.; Li, Y.; Zhao, W. X.; Gao, J.; and Zhang, A. 2017b. A correlated topic model using word embeddings. In *IJCAI'17*, 4207–4213.

Zhang, C.; Zhang, K.; Yuan, Q.; Tao, F.; Zhang, L.; Hanratty, T.; and Han, J. 2017. React: Online multimodal embedding for recency-aware spatiotemporal activity modeling. In *SIGIR'17*, 245–254.

Zhang, C.; Tao, F.; Chen, X.; Shen, J.; Jiang, M.; Sadler, B.; Vanni, M.; and Han, J. 2018. Taxogen: Unsupervised topic taxonomy construction by adaptive term embedding and clustering. In *KDD'18*, 2701–2709.