Effective Seed-Guided Topic Discovery by Integrating Multiple Types of Contexts

Yu Zhang*
Yunyi Zhang*
University of Illinois at Urbana-Champaign
{yuz9, yzhan238}@illinois.edu

Martin Michalski*
Yucheng Jiang*
University of Illinois at Urbana-Champaign
{martinm6, yj17}@illinois.edu

Yu Meng*
Jiawei Han
University of Illinois at Urbana-Champaign
{yumeng5, hanj}@illinois.edu

ABSTRACT

Instead of mining coherent topics from a given text corpus in a completely unsupervised manner, seed-guided topic discovery methods leverage user-provided seed words to extract distinctive and coherent topics so that the mined topics can better cater to the user's interest. To model the semantic correlation between words and seeds for discovering topic-indicative terms, existing seed-guided approaches utilize different types of context signals, such as document-level word co-occurrences, sliding window-based local contexts, and generic linguistic knowledge brought by pre-trained language models. In this work, we analyze and show empirically that each type of context information has its value and limitation in modeling word semantics under seed guidance, but combining three types of contexts (i.e., word embeddings learned from local contexts, pre-trained language model representations obtained from general-domain training, and topic-indicative sentences retrieved based on seed information) allows them to complement each other for discovering quality topics. We propose an iterative framework, SeedTopicMine, which jointly learns from the three types of contexts and gradually fuses their context signals via an ensemble ranking process. Under various sets of seeds and on multiple datasets, SeedTopicMine consistently yields more coherent and accurate topics than existing seed-guided topic discovery approaches.

CCS CONCEPTS

• **Information systems** → **Clustering**; **Document topic models**;
• **Computing methodologies** → **Natural language processing**.

KEYWORDS

topic discovery; text embedding

ACM Reference Format:
Yu Zhang*, Yunyi Zhang*, Martin Michalski*, Yucheng Jiang*, Yu Meng*, and Jiawei Han. 2023. Effective Seed-Guided Topic Discovery by Integrating Multiple Types of Contexts. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining (WSDM '23), February 27-March 3, 2023, Singapore, Singapore.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3539597.3570475

*Equal Contribution.

WSDM '23, February 27-March 3, 2023, Singapore, Singapore

ACM ISBN 978-1-4503-9407-9/23/02…$15.00
https://doi.org/10.1145/3539597.3570475

1 INTRODUCTION

To efficiently grasp the information in a large collection of documents, it is of great interest to automatically discover a set of coherent topics from the corpus. Besides capturing meaningful structures in massive text data [11], topic discovery also widely benefits downstream text mining tasks such as taxonomy construction [20] and document classification [5].

Unsupervised topic models, from LDA [4] to embedding-based [8, 44] and pre-trained language model-enhanced [29, 39] approaches, have been extensively studied for decades as the mainstream approach to topic discovery. Despite their efficacy in uncovering prominent themes of a corpus, such models tend to retrieve semantically general topics that may not align well with users' specific interests, as explained in [13, 27]. Motivated by this, rather than finding arbitrary topics in a fully unsupervised manner, seed-guided topic discovery [10, 13, 15, 27, 47] aims to extract topics along a certain dimension based on user-provided seeds, and the top-ranked words under each topic should be discriminatively relevant to the corresponding seed. For example, given a collection of restaurant reviews, if a user would like to explore topics of *food types* (e.g., by providing the seeds "*noodles*", "*steak*", and "*pizza*"), then a seed-guided topic discovery model should discover topic-indicative terms for each input seed (e.g., "*ramen*" and "*pasta*" under "*noodles*"), instead of finding terms that are relevant to multiple seeds (e.g., "*beef*") or retrieving topics along other dimensions (e.g., "*good*" or "*bad*" as topics of *sentiments*).

Recent studies on seed-guided topic discovery [13, 20, 27, 47] have been focusing on utilizing different types of context information so that they can go beyond the "bag-of-words" generative assumption in LDA and learn more accurate word semantics for topic discovery. To be specific, there are three major types of context signals used in related studies.

Skip-Gram Word Embeddings. Different from LDA which infers topics based on the global document-word frequency matrix, skip-gram embedding learning [31] assumes that words occurring in similar local contexts (e.g., ±5 words) tend to have similar semantic properties. Following this assumption, each word in the corpus can be represented by an embedding vector in a latent space. To incorporate skip-gram signals in topic discovery, the word embeddings can be injected into the LDA backbone [13] or jointly learned by viewing documents and seeds also as contexts [27]. However, skip-gram embeddings are less helpful in disambiguating word meanings because only one vector is learned for each word given

the whole corpus. Indeed, Sia et al. [39] show that clustering skip-gram embeddings underperforms clustering output representations of contextualized language models such as BERT [7] in unsupervised topic modeling.

Pre-trained Language Model Representations. Pre-trained language models (PLMs) [7, 23, 33] have revolutionized the text mining field by learning contextualized word embeddings. The Transformer architecture [43] used in many PLMs can capture long-range and high-order context signals, and the knowledge learned by PLMs from web-scale corpora can complement contexts in the input corpus in topic discovery [47]. Meanwhile, related studies have observed several cases where PLMs generate noticeably bad topics. For example, Meng et al. [29] show that PLM representations suffer from the curse of dimensionality and do not form clearly separated clusters; Thompson and Mimno [42] find that GPT-2 representations [33] work well only if the outputs of certain layers are taken, and RoBERTa-induced topics [23] are consistently of poor quality.

Topic-Indicative Documents. Although skip-gram embeddings and PLMs are powerful in representing each word based on its contexts, neither of them considers whether the contexts they use are topic-indicative (i.e., semantically close to a certain seed). In fact, skip-gram embedding learning always takes the $\pm x$ words as contexts, regardless of whether they are relevant to any seed; a PLM will always output the same representation for a word if the input corpus is fixed, no matter what the seeds are. To tackle this problem, supervised topic models [17, 25] propose to leverage document-level training data (i.e., each document belongs to which seed or semantic category). However, such information relies on massive human annotation, which may be difficult to obtain in practical applications (e.g., weakly supervised text classification [26, 28, 46]). Moreover, a document may be too broad to be viewed as a context unit because each document can be relevant to multiple topics simultaneously.

To summarize, each type of context signals has its specific advantages and disadvantages. Therefore, a topic discovery method purely relying on one type of context information may not be robust across different datasets or seed dimensions. Meanwhile, it is worth noting that the three types of contexts strongly complement each other. For example, PLMs have contextualization power which skip-gram embeddings are short of; skip-gram embeddings usually have fewer dimensions than PLM representations and are less prone to the curse of dimensionality; topic-indicative documents are not naturally available, but they can be retrieved by applying skip-gram embeddings and PLMs.

Contributions. Motivated by the complementarity of context signals, in this paper, we propose SeedTopicMine, an effective seed-guided topic discovery framework by integrating multiple types of contexts. SeedTopicMine iteratively retrieves and updates the set of topic-discriminative terms for each seed. In each iteration, we first jointly leverage seed-guided skip-gram embeddings and PLM-based representations to discover a set of topic-indicative terms. Then, using these terms, we retrieve a set of topic-indicative sentences. Here, we consider sentences rather than documents because each sentence, as a more fine-grained unit, is more likely to concentrate on one topic. Finally, the derived topic-indicative sentences and the other two types of contexts are cooperatively utilized through an ensemble ranking process, after which the topic-discriminative terms will be updated and used for the next iteration.

Extensive experiments on real-world datasets show that SeedTopicMine effectively discovers discriminative terms under each seed to form coherent topics. Our human evaluation quantitatively validates the superiority of SeedTopicMine over baselines that rely on a single type of contexts. In the ablation study, we observe that even in the same dataset, if we consider different dimensions of seeds, the contributions of different context signals vary significantly, which confirms our key motivation that any single type of context signal is insufficient for discovering seed-discriminative topics stably.

2 PROBLEM DEFINITION

Following [27], we assume a seed can be either a unigram or a phrase. Given an input corpus and a set of seeds, our goal is to find a set of terms under each seed to form a coherent topic. Conforming to the assumption of seeds, each term can also be a unigram or a phrase. In practice, given a raw corpus, one can adopt existing phrase chunking tools [24, 38] to obtain phrases in it.

Definition 2.1. (Problem Definition) Given a corpus $\mathcal{D} = \{d_1, ..., d_{|\mathcal{D}|}\}$ and a set of seeds $\mathcal{S} = \{s_1, ..., s_{|\mathcal{S}|}\}$, seed-guided topic discovery aims to find a set of terms $\mathcal{T}_i = \{t_{i1}, t_{i2}, ..., t_{i|\mathcal{T}_i|}\}$ appearing in $\mathcal{D}$ for each seed s_i $(1 \le i \le |\mathcal{S}|)$, where the term t_{ij} is semantically close to s_i and far from other seeds s_j $(\forall j \ne i)$.

In other words, each seed s_i represents a semantic category c_i, and the task is to find a set of terms $\mathcal{T}_i$ that discriminatively belong to the category c_i $(1 \le i \le |\mathcal{S}|)$.

3 FRAMEWORK

In this section, we first review various types of contexts utilized by previous studies for topic discovery. Then, we present our framework, SeedTopicMine, that iteratively ensembles these types of signals.

3.1 Types of Context Information

Previous studies on topic discovery, either unsupervised, seed-guided, or supervised, propose to leverage different types of information such as skip-gram embeddings [8, 13, 27, 30, 44], pre-trained language model representations [3, 29, 39, 42, 47], and topic-indicative documents [17, 20, 25]. We now introduce three major types of information sources, which are illustrated in Figure 1, and how we propose to use them in SeedTopicMine.

3.1.1 Seed-Guided Text Embeddings. Previous embedding-based topic models [8, 13, 44] propose to incorporate word embeddings to make up for the representation deficiency of the "bag-of-words" generation assumption in LDA. The intuition of text embedding learning is based on the hypothesis that semantically similar terms share similar contexts. In unsupervised topic discovery, the contexts of a term may refer to its skip-grams [31] and the documents it appears in [19, 40]. In our task of seed-guided topic discovery, we can further leverage seeds in the embedding learning process by viewing the category that a term belongs to as its context. To facilitate this goal, in SeedTopicMine, we follow [27, 47] and unify the three types of contexts into one objective. To be specific, we aim to maximize the likelihood of observing a term's skip-gram, document, and category contexts given that term. Formally, the

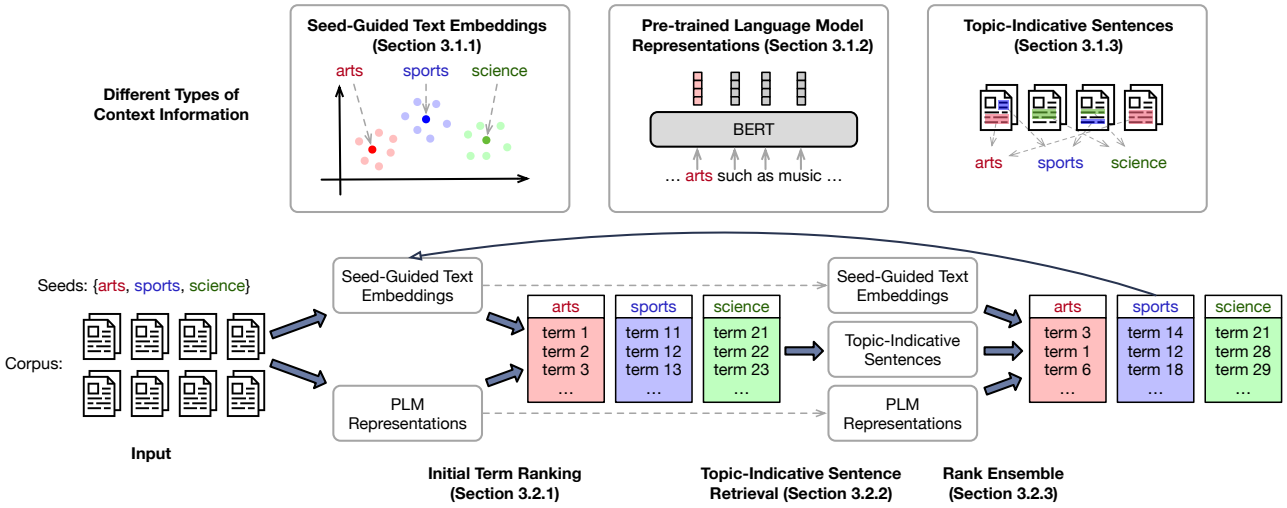


Figure 1: Overview of the SeedTopicMine framework.

embedding learning objective is:

$$\mathcal{J}_{\text{Emb}} = \underbrace{\log \prod_{d\in\mathcal{D}} \prod_{w_i\in d} \prod_{w_j\in C(w_i)} p(w_j|w_i)}_{\text{skip-gram}} + \underbrace{\log \prod_{d\in\mathcal{D}} \prod_{w\in d} p(d|w)}_{\text{document}} + \underbrace{\log \prod_{1\le i\le|\mathcal{S}|} \prod_{w\in\mathcal{T}_i} p(c_i|w)}_{\text{category}}. \quad (1)$$

Here, $C(w_i)$ is the set of terms in w_i's skip-gram window. For example, given a text sequence $w_1 w_2 ... w_X$, we have $C(w_i) = \{w_j | i-x \le j \le i+x,\ j \ne i\}$, where x is the skip-gram window size. $\mathcal{T}_i$, as mentioned in Definition 2.1, is the set of terms related to the seed s_i (i.e., belong to the semantic category c_i). We adopt an iterative framework to gradually expand $\mathcal{T}_i$. At the very beginning, $\mathcal{T}_i = \{s_i\}$ (i.e., the seed initially belongs to its corresponding topic). After each iteration, terms close to the category c_i in the embedding space will be added to $\mathcal{T}_i$.

There are various ways to define each likelihood in Eq. (1). Following previous studies on topic modeling [2, 16, 21, 27], we adopt the von Mises-Fisher (vMF) distribution.

$$p(w_j|w_i) = \frac{\exp(\kappa_{w_i} \cos(\boldsymbol{u}_{w_i}, \boldsymbol{v}_{w_j}))}{\sum_{w'} \exp(\kappa_{w_i} \cos(\boldsymbol{u}_{w_i}, \boldsymbol{v}_{w'}))} \simeq \text{vMF}(\boldsymbol{v}_{w_j}|\boldsymbol{u}_{w_i}, \kappa_{w_i}), \quad (2)$$

where $\kappa_{w_i} \ge 0$ is the concentration parameter, indicating the semantic specificity of w_i; $\boldsymbol{u}_{w_i}$ and $\boldsymbol{v}_{w_j}$ are the embeddings of w_i and w_j, respectively. The vMF distribution can be viewed as an analogue of the Gaussian distribution on a sphere. In Eq. (2), the distribution concentrates around the mean direction $\boldsymbol{u}_{w_i}$, and is more concentrated if κ_{w_i} is larger (i.e., w_i is a more specific term). Similar to Eq. (2), the other two likelihood terms in Eq. (1) can be defined as

$$\begin{aligned} p(d|w) &= \frac{\exp(\kappa_w \cos(\boldsymbol{u}_w, \boldsymbol{v}_d))}{\sum_{d'} \exp(\kappa_w \cos(\boldsymbol{u}_w, \boldsymbol{v}_{d'}))} \simeq \text{vMF}(\boldsymbol{v}_d|\boldsymbol{u}_w, \kappa_w), \\ p(c_i|w) &= \frac{\exp(\kappa_w \cos(\boldsymbol{u}_w, \boldsymbol{v}_{c_i}))}{\sum_{c'} \exp(\kappa_w \cos(\boldsymbol{u}_w, \boldsymbol{v}_{c'}))} \simeq \text{vMF}(\boldsymbol{v}_{c_i}|\boldsymbol{u}_w, \kappa_w), \end{aligned} \quad (3)$$

where $\boldsymbol{v}_d$ and $\boldsymbol{v}_{c_i}$ are the embedding vectors of document d and category c_i, respectively.

To summarize, the seed-guided text embedding learning process is cast as the following optimization problem:

$$\max \mathcal{J}_{\text{Emb}} \ \text{ s.t. } ||\boldsymbol{u}_w|| = ||\boldsymbol{v}_w|| = ||\boldsymbol{v}_d|| = ||\boldsymbol{v}_c|| = 1,\ \kappa_w \ge 0. \quad (4)$$

We follow the optimization process of [27] to optimize Eq. (4).

After embedding learning, for each term w, we obtain two vectors $\boldsymbol{u}_w$ and $\boldsymbol{v}_w$, which, as they do in previous studies [27, 31, 40], carry the semantics of w when it is viewed as a center term and a context term, respectively. Given a term w and a seed s_i, we calculate the cosine similarity between their learned embeddings as the first criterion of their semantic proximity, which will later be used in topic discovery.

$$\text{sim}_{\text{Emb}}(w, s_i) = \cos(\boldsymbol{u}_w, \boldsymbol{u}_{s_i}). \quad (5)$$

3.1.2 Pre-trained Language Model Representations. Recently, PLMs such as BERT [7] have achieved great success in a wide spectrum of text mining tasks. The Transformer architecture [43] used in many PLMs is capable of capturing long-range and high-order context signals. Moreover, the generic knowledge learned by PLMs from web-scale corpora (e.g., Wikipedia) can complement the information one can get from the input corpus. To utilize such signals in topic discovery, for each term appearing in the input corpus, we employ a PLM to derive its representation.

Suppose a term w appears M times in the corpus $\mathcal{D}$. For each of its mentions w^i $(1 \le i \le M)$, we feed the sentence containing this mention into a PLM. Note that w^i may be segmented into multiple word pieces $w^i_1, w^i_2, ..., w^i_L$ according to the PLM tokenizer [36, 37], and each word piece w^i_j will have an output representation vector $\textbf{PLM}(w^i_j)$ after PLM encoding. Following previous studies on topic discovery [39, 42], we take the average of these word piece representations as the representation of the mention.

$$\textbf{PLM}(w^i) = \frac{1}{L} \sum_{j=1}^{L} \textbf{PLM}(w^i_j). \quad (6)$$

The mention representation is contextualized given the sentence it appears in. To get the corpus-level semantics of a term, we average

the representations of all its mentions.

$$\boldsymbol{h}_w = \frac{1}{M}\sum_{i=1}^{M}\mathbf{PLM}(w^i). \tag{7}$$

In this way, for each w, we obtain a vector $\boldsymbol{h}_w$ whose dimension is given by the adopted PLM. For example, if we use $\text{BERT}_{\text{Base}}$ [7], then $\boldsymbol{h}_w \in \mathbb{R}^{768}$. Given a term w and a seed s_i, we calculate the cosine similarity between their PLM-based representations as our second criterion of their semantic proximity.

$$\text{sim}_{\text{PLM}}(w, s_i) = \cos(\boldsymbol{h}_w, \boldsymbol{h}_{s_i}). \tag{8}$$

3.1.3 Topic-Indicative Context. Although seed-guided embedding learning and PLM encoding are both powerful tools to represent each term based on its contexts, neither of them considers whether the utilized context information is topic-indicative or not. To be specific, the PLM-based representation $\boldsymbol{h}_w$ is unaware of the seed space $\mathcal{S}$ (in other words, no matter what the seeds $\mathcal{S} = \{s_1, ..., s_{|\mathcal{S}|}\}$ are, if the corpus $\mathcal{D}$ is fixed, then the same PLM will always generate the same representation vector $\boldsymbol{h}_w$ for w); the embeddings $\boldsymbol{u}_w$ and $\boldsymbol{v}_w$ always take the skip-gram $C(w)$ (i.e., $\pm1, ..., \pm x$ terms) and the document d containing w as contexts during learning, regardless of whether such information is relevant to a certain seed/topic. To alleviate this gap, we propose to use topic-indicative context to derive the correlation between a term w and a seed s_i.

For each seed $s_i \in \mathcal{S}$, we assume it has a set of topic-indicative sentences $\Theta_i = \{\theta_{i1}, ..., \theta_{i|\Theta_i|}\}$. (Initially, Θ_i is not given as input. We will discuss how to obtain and iteratively update Θ_i in Section 3.2.2.) The reason that we consider sentences instead of documents here is because a document is more likely to cover multiple topics. Motivated by [20, 41], we calculate the semantic closeness between w and Θ_i according to the following two criteria: (1) *Popularity*: a term close to Θ_i should appear frequently in the sentences in Θ_i. Formally, $\text{pop}(w, \Theta_i) = \log(1 + \text{tf}(w, \Theta_i))$, where $\text{tf}(\cdot,\cdot)$ denotes term frequency and $\text{tf}(w, \Theta_i) = \sum_{j=1}^{|\Theta_i|}\text{tf}(w, \theta_{ij})$. (2) *Distinctiveness*: a term close to Θ_i should be much more relevant to the sentences in Θ_i than it is to the sentences indicating other topics. This can be characterized by the formula: $\text{dist}(w, \Theta_i) = \frac{\exp(\text{BM25}(w,\Theta_i))}{1+\sum_{i'=1}^{|\mathcal{S}|}\exp(\text{BM25}(w,\Theta_{i'}))}$, where $\text{BM25}(\cdot,\cdot)$ denotes the BM25 relevance function [34].

To jointly consider popularity and distinctiveness, the similarity between a term w and a category c_i based on topic-indicative sentences is defined as follows.

$$\text{sim}_{\text{Sntn}}(w, c_i) = \text{pop}(w, \Theta_i)^{\alpha} \cdot \text{dist}(w, \Theta_i)^{1-\alpha}, \tag{9}$$

where $0 < \alpha < 1$ is a hyperparameter.

3.2 The Iterative SeedTopicMine Framework

We lay out our framework in Figure 1. It has three major modules: initial term ranking, topic-indicative sentence retrieval, and rank ensemble. We now introduce these modules in detail.

3.2.1 Initial Term Ranking. Initially, we only have the seed s_i for each semantic category c_i, and the topic-indicative sentences Θ_i have not been derived yet. Therefore, we first use seed-guided text embeddings (derived in Section 3.1.1) and PLM-based representations (derived in Section 3.1.2) to find terms that are relevant to each category. To be specific, for each category c_i, we calculate the following score for each term w.

$$\text{score}_{\text{Ini}}(w, c_i) = \text{sim}_{\text{Emb}}(w, s_i) \cdot \text{sim}_{\text{PLM}}(w, s_i), \tag{10}$$

where $\text{sim}_{\text{Emb}}(\cdot,\cdot)$ and $\text{sim}_{\text{PLM}}(\cdot,\cdot)$ are given in Eqs. (5) and (8), respectively. As mentioned in Section 3.1.1, the set of topic-related terms $\mathcal{T}_i$ is expanded and updated iteratively. In later iterations, when $\mathcal{T}_i$ is more than just $\{s_i\}$, Eq. (10) can be generalized to

$$\text{score}_{\text{Ini}}(w, c_i) = \sum_{t_{ij}\in\mathcal{T}_i}\text{sim}_{\text{Emb}}(w, t_{ij}) \cdot \sum_{t_{ij}\in\mathcal{T}_i}\text{sim}_{\text{PLM}}(w, t_{ij}). \tag{11}$$

For each category c_i, we find top-τ terms according to $\text{score}_{\text{Ini}}(w, c_i)$ to update the topic-indicative term set $\mathcal{T}_i$.

3.2.2 Topic-Indicative Sentence Retrieval. Based on the set of updated topic-indicative terms $\mathcal{T}_i$, we now retrieve the set of topic-indicative sentences Θ_i from the input corpus so that we can calculate Eq. (9). The retrieval process is inspired by two assumptions: (1) The sentences containing many topic-indicative terms from one category and do not contain any topic-indicative term from other categories should be topic-indicative sentences. We call such sentences "*anchor*" sentences. (2) The "*neighbor*" sentences of topic-indicative "anchor" sentences should also be viewed as topic-indicative if they do not contain topic-indicative terms from other categories.

Algorithm 1: SeedTopicMine

Input: A corpus $\mathcal{D}$; a set of seeds $\mathcal{S} = \{s_1, ..., s_{|\mathcal{S}|}\}$.
Output: A set of terms $\mathcal{T}_i = \{t_{i1}, t_{i2}, ..., t_{i|\mathcal{T}_i|}\}$ appearing in $\mathcal{D}$ for each seed s_i.

$\mathcal{T}_i = \{s_i\}$;
$\boldsymbol{h}_w \leftarrow$ Eq. (7);
for iter $\leftarrow 1$ *to* N **do**
  Learn seed-guided text embedding $\boldsymbol{u}_w$ by optimizing Eq. (4);
  // Initial Term Ranking;
  $\text{score}_{\text{Ini}}(w, c_i) \leftarrow$ Eq. (11);
  $\mathcal{T}_i \leftarrow$ top-ranked terms according to $\text{score}_{\text{Ini}}(w, c_i)$;
  // Topic-Indicative Sentence Retrieval;
  $\text{count}(\theta, c_i) \leftarrow$ Eq. (12);
  $\Theta_i^A \leftarrow$ top-ranked sentences according to Eq. (13);
  $\Theta_i^N \leftarrow \emptyset$;
  for $\theta_{ij} \in \Theta_i^A$ **do**
    for $k \leftarrow 1$ *to* y **do**
      Denote the $+k$ sentence of θ_{ij} as θ_{ij}^{+k};
      if $\forall i' \neq i, \text{count}(\theta_{ij}^{+k}, c_{i'}) = 0$ **then**
        $\Theta_i^N \leftarrow \Theta_i^N \cup \{\theta_{ij}^{+k}\}$;
      else
        break;
    for $k \leftarrow 1$ *to* y **do**
      Denote the $-k$ sentence of θ_{ij} as θ_{ij}^{-k};
      if $\forall i' \neq i, \text{count}(\theta_{ij}^{-k}, c_{i'}) = 0$ **then**
        $\Theta_i^N \leftarrow \Theta_i^N \cup \{\theta_{ij}^{-k}\}$;
      else
        break;
  $\Theta_i \leftarrow \Theta_i^A \cup \Theta_i^N$;
  // Rank Ensemble;
  $\text{score}_{\text{All}}(w, c_i) \leftarrow$ Eq. (14);
  $\text{MRR}(w|c_i) \leftarrow$ Eq. (15);
  $\mathcal{T}_i \leftarrow$ Eq. (16);
$\mathcal{T}_i \leftarrow \mathcal{T}_i \backslash \{s_i\}$;
Return $\mathcal{T}_1, ..., \mathcal{T}_{|\mathcal{S}|}$;

According to Assumption (1), we first retrieve "anchor" sentences by counting the number of topic-indicative terms appearing in each sentence. Formally, given a category c_i, for each sentence θ in $\mathcal{D}$, we calculate

$$\text{count}(\theta, c_i) = \sum_{w \in \mathcal{T}_i} \text{tf}(w, \theta). \tag{12}$$

Because category-indicative "anchor" sentences should have a high count with c_i and a count of 0 with any other category, we rank the sentences using the following criterion.

$$\max_{\theta \in \mathcal{D}} \text{count}(\theta, c_i), \quad \text{where count}(\theta, c_j) = 0 \quad (\forall j \neq i). \tag{13}$$

We use $\Theta_i^A = \{\theta_{i1}, ..., \theta_{i|\Theta_i^A|}\}$ to denote the set of selected "anchor" sentences for c_i.

Then, according to Assumption (2), we find "neighbor" sentences for each "anchor" sentence θ_{ij}. To be specific, given an "anchor" sentence, we check its $\pm 1, \pm 2, ..., \pm y$ sentences in the document (if they exist). If the $+k$ (resp., $-k$) sentence contains topic-indicative terms from other categories, we view it as not topic-indicative, and we do not further check the $+(k+1), ..., +y$ (resp., $-(k+1), ..., -y$) sentences because they may have diverged to other topics. Otherwise, we add the $+k$ (resp., $-k$) sentence into the set of topic-indicative "neighbor" sentences Θ_i^N. A more formal description of this process can be found in Lines 13–24 in Algorithm 1.

Finally, the set of retrieved topic-indicative sentences is the union of "anchor" sentences and "neighbor" sentences (i.e., $\Theta_i = \Theta_i^A \cup \Theta_i^N$).

3.2.3 Ensemble of Multiple Types of Contexts. After obtaining topic-indicative context Θ_i of each category c_i, we can now calculate $\text{sim}_{\text{Sntn}}(w, c_i)$ in Eq. (9). Then, we have a score measuring the semantic proximity between a term w and a category c_i by jointly considering all three types of contexts.

$$\text{score}_{\text{All}}(w, c_i) = \sum_{t_{ij} \in \mathcal{T}_i} \text{sim}_{\text{Emb}}(w, t_{ij}) \cdot \sum_{t_{ij} \in \mathcal{T}_i} \text{sim}_{\text{PLM}}(w, t_{ij}) \cdot \text{sim}_{\text{Sntn}}(w, c_i). \tag{14}$$

By ranking all terms in a descending order of $\text{score}_{\text{All}}(w, c_i)$, we get a ranking list where each term w has a rank position $r_{\text{All}}(w|c_i)$. Besides, instead of incorporating topic-indicative context into ranking, we can consider seed-guided text embeddings alone or PLM-based representations alone. By ranking terms in a descending order of $\sum_{t_{ij} \in \mathcal{T}_i} \text{sim}_{\text{Emb}}(w, t_{ij})$ and $\sum_{t_{ij} \in \mathcal{T}_i} \text{sim}_{\text{PLM}}(w, t_{ij})$, each term w will have two more rank positions $r_{\text{Emb}}(w|c_i)$ and $r_{\text{PLM}}(w|c_i)$, respectively. Based on the three rank positions, we perform rank ensemble by calculating the mean reciprocal rank (MRR).

$$\text{MRR}(w|c_i) = \frac{1}{3}\left(\frac{1}{r_{\text{All}}(w|c_i)} + \frac{1}{r_{\text{Emb}}(w|c_i)} + \frac{1}{r_{\text{PLM}}(w|c_i)}\right). \tag{15}$$

In practice, instead of ranking all terms in the vocabulary, we only check the top-ρ terms in each ranking list. If a term w is not among the top-ρ (e.g., $r_{\text{All}}(w|c_i) > \rho$), we simply set its reciprocal rank to be 0 (e.g., $\frac{1}{r_{\text{All}}(w|c_i)} = 0$). Finally, we update $\mathcal{T}_i$ with the terms whose MRR score exceeds a certain threshold η.

$$\mathcal{T}_i = \{w \mid \text{MRR}(w|c_i) \geq \eta\}, \quad (1 \leq i \leq |\mathcal{S}|). \tag{16}$$

The updated term sets $\mathcal{T}_1, ..., \mathcal{T}_{|\mathcal{S}|}$ are then fed into the next iteration of SeedTopicMine.

We iterate the process of initial term ranking, topic-indicative sentence retrieval, and rank ensemble for N iterations. The entire SeedTopicMine framework is summarized in Algorithm 1.

Table 1: Dataset Statistics.

Dataset	NYT		Yelp	
Dimension	Topic	Location	Food	Sentiment
#Docs	31,997	31,997	29,280	29,280
#Seeds	9	10	8	2
Seeds	arts, technology, health, education, sports, science, business, politics, real estate	united states, iraq, britain, japan, canada, china, france, italy, russia, germany	steak, seafood, pizza, desserts, salad, noodles, sushi, burgers	good, bad

4 EXPERIMENTS

4.1 Setup

4.1.1 Datasets. Following [27], we conduct experiments on two datasets from different domains.

- **NYT**[1] is a collection of news articles written and published by the New York Times. It has two sets of seeds along the *topic* and *location* dimensions, respectively.
- **Yelp**[2] is a corpus of restaurant reviews released by the Yelp Dataset Challenge. It has two sets of seeds along the *food* and *sentiment* dimensions, respectively.

For both datasets, we use AutoPhrase [38] to perform phrase chunking. Following [39], we adopt a 60-40 train-test split for both datasets. The training set is used as the input corpus $\mathcal{D}$, and the testing set is used to calculate the topic coherence metric (see evaluation metrics for details). Dataset statistics are summarized in Table 1.

4.1.2 Compared Methods. We compare our SeedTopicMine with the following baselines including seed-guided topic modeling methods and seed-guided embedding learning methods.

- **SeededLDA [15]** is a seed-guided topic modeling method. It modifies the generative process of LDA by biasing each topic to generate more seeds and by biasing each document to select topics relevant to the seeds appearing in the document.
- **Anchored CorEx [10]** is a seed-guided topic modeling method. It does not rely on generative assumptions. Instead, it leverages seeds by balancing between compressing the input corpus and preserving seed-related information.
- **KeyETM [13]** is an embedding-based topic model (ETM) assisted by keyword seeds. It modifies the objective of ETM [8] to utilize seeds in the form of topic-level priors over the vocabulary.
- **CatE [27]** is a seed-guided embedding learning method for discriminative topic mining. It jointly learns term embedding and specificity from the input corpus. Terms are then selected based on both embedding similarity with the seeds and specificity.

For unsupervised topic discovery approaches (e.g., BERTopic [12] and TopClus [29]), it is difficult to match their generated topics to the given seeds, so we cannot calculate term accuracy-based metrics (see evaluation metrics for details) for their output, and hence we do include them into comparison.

4.1.3 Evaluation Metrics. Given the top-$|\mathcal{T}_i|$ discovered terms under each seed ($|\mathcal{T}_i| = 20$ in our experiments), we evaluate the results based on two different criteria: *topic coherence* and *term accuracy*.

[1] https://catalog.ldc.upenn.edu/LDC2008T19
[2] https://www.yelp.com/dataset/challenge

Table 2: NPMI, P@20, and NDCG@20 scores of compared algorithms. NPMI measures topic coherence; P@20 and NDCG@20 measure term accuracy.

Method	**NYT**-Topic			**NYT**-Location			**Yelp**-Food			**Yelp**-Sentiment		
	NPMI	P@20	NDCG@20	NPMI	P@20	NDCG@20	NPMI	P@20	NDCG@20	NPMI	P@20	NDCG@20
SeededLDA [15]	0.0841	0.2389	0.2979	0.0814	0.1050	0.1873	0.0504	0.1200	0.2132	0.0499	0.1700	0.2410
Anchored CorEx [10]	0.1325	0.2922	0.3627	0.1283	0.2040	0.3003	0.1204	0.3725	0.4531	0.0627	0.1200	0.1997
KeyETM [13]	0.1254	0.1589	0.2342	0.1146	0.0700	0.1676	0.0578	0.1788	0.2940	0.0327	0.4250	0.4994
CatE [27]	0.1941	0.8067	0.8306	0.2165	0.7480	0.7840	**0.2058**	0.6812	0.7312	**0.1509**	0.7150	0.7713
SeedTopicMine	**0.1947**	**0.9456**	**0.9573**	**0.2176**	**0.8360**	**0.8709**	0.2018	**0.7912**	**0.8379**	0.0922	**0.9750**	**0.9811**

- **NPMI [18]** is a widely adopted metric in topic modeling to measure *topic coherence* inside each topic. It is defined as the average normalized pointwise mutual information of each pair of terms in $\mathcal{T}_i$.

$$\text{NPMI} = \frac{1}{|\mathcal{S}|}\sum_{i=1}^{|\mathcal{S}|}\frac{1}{\binom{|\mathcal{T}_i|}{2}}\sum_{t_{ij},t_{ik}\in\mathcal{T}_i}\frac{\log\frac{P(t_{ij},t_{ik})}{P(t_{ij})P(t_{ik})}}{-\log P(t_{ij},t_{ik})}, \tag{17}$$

  where $P(t_{ij}, t_{ik})$ is the probability that t_{ij} and t_{ik} co-occur in a document; $P(t_{ij})$ is the probability that t_{ij} occurs in a document.
- **P@k (also called MACC) [27]** is a metric for *term accuracy*. It measures the proportion of retrieved terms t_{ij} that actually belong to the semantic category c_i.

$$\text{P@}k = \frac{1}{|\mathcal{S}|}\sum_{i=1}^{|\mathcal{S}|}\frac{1}{|\mathcal{T}_i|}\sum_{t_{ij}\in\mathcal{T}_i}\mathbf{1}(t_{ij}\in c_i), \tag{18}$$

  where $\mathbf{1}(t_{ij} \in c_i)$ is the indicator function of whether t_{ij} belongs to c_i (i.e., whether t_{ij} is discriminatively relevant to the seed s_i). This relies on human judgment, so we invite five annotators to perform independent annotation. The reported P@k score is the average P@k of the five annotators. A high inter-annotator agreement is observed, with Fleiss' kappa [9] being 0.896, 0.928, 0.800, and 0.909 on NYT-Topic, NYT-Location, Yelp-Food, and Yelp-Sentiment, respectively. As mentioned above, we set $|\mathcal{T}_i| = 20$ in our experiments, so we report P@20.
- **NDCG@k** is another metric for *term accuracy*. It gives higher weights to higher-ranked terms by applying a logarithmic discount.

$$\text{DCG}_i\text{@}k = \sum_{j=1}^{|\mathcal{T}_i|}\frac{\mathbf{1}(t_{ij}\in c_i)}{\log(j+1)},\quad \text{IDCG@}k = \sum_{j=1}^{|\mathcal{T}_i|}\frac{1}{\log(j+1)},$$
$$\text{NDCG@}k = \frac{1}{|\mathcal{S}|}\sum_{i=1}^{|\mathcal{S}|}\frac{\text{DCG}_i\text{@}k}{\text{IDCG@}k}. \tag{19}$$

  Following the case of P@k, we calculate NDCG@k based on human annotations, and we report NDCG@20.

4.1.4 Hyperparameters and Implementation. The hyperparameter settings of SeedTopicMine are as follows. In seed-guided embedding learning, the context window size $x = 5$; the embedding dimension is 100. In PLM encoding, we use $\text{BERT}_{\text{Base}}$ [7] as the PLM. When computing $\text{sim}_{\text{Sntn}}(w, c_i)$, we set $\alpha = 0.2$. In initial term ranking, we select $\tau = 20$ terms for each seed. In topic-indicative sentence retrieval, we retrieve $|\Theta_i^A| = 500$ "anchor" sentences; the "neighbor" sentence window size $y = 4$. In rank ensemble, there are $\rho = 20$ terms in each ranking list; the MRR threshold $\eta = 0.1$. We run SeedTopicMine for $N = 4$ iterations.

The code, datasets, and annotation results are available at https://github.com/yzhan238/SeedTopicMine.

4.2 Performance Comparison

Table 2 shows the NPMI, P@20, and NDCG@20 scores of compared algorithms on the two datasets. We can observe that: (1) On NYT, SeedTopicMine consistently achieves the best performance in terms of all metrics. Among all the baselines, CatE is the most effective one, significantly outperforming "bag-of-words"-based topic models such as SeededLDA and Anchored CorEx. However, since CatE only uses one type of context information (i.e., skip-gram embeddings), SeedTopicMine can improve CatE by an evident margin on term accuracy through integrating multiple types of signals. (2) On Yelp, SeedTopicMine underperforms CatE in terms of NPMI but significantly outperforms CatE in terms of P@20 and NDCG@20. Note that NPMI is an automatically computed metric, and the other two metrics rely on human annotation. Indeed, a recent study [14] shows that automatic metrics such as NPMI may not align well with human evaluation. From this perspective, we claim that SeedTopicMine performs better than CatE on Yelp, and our qualitative analysis below will validate this claim.

Besides quantitative evaluation, we show the qualitative comparison in Table 3. We randomly select two seeds from NYT-Location, NYT-Topic, Yelp-Food, and Yelp-Sentiment, respectively. For each seed, we show top-5 terms retrieved by each method. A term is marked as incorrect (×) if and only if at least 3 of the 5 annotators judge the term as irrelevant to the seed. Table 3 demonstrates that: (1) SeededLDA, Anchored CorEx, and KeyETM tend to find irrelevant or very general terms. For example, both Anchored CorEx and KeyETM retrieve the term "*fish*" under the seed "*sushi*", but "*fish*" is also relevant to the seed "*seafood*", thus it does not discriminatively belong to the sushi category. (2) Most terms discovered by CatE are accurate. However, CatE still makes mistakes in all four dimensions in Table 3. In contrast, SeedTopicMine achieves higher accuracy. If we further check the mistakes made by CatE, we can find general terms such as "*also*" and "*savoury*", which may co-occur frequently with other top-ranked terms. This possibly explains why CatE achieves higher NPMI than SeedTopicMine on Yelp since NPMI is based on the co-occurrence of retrieved terms.

4.3 Ablation Study

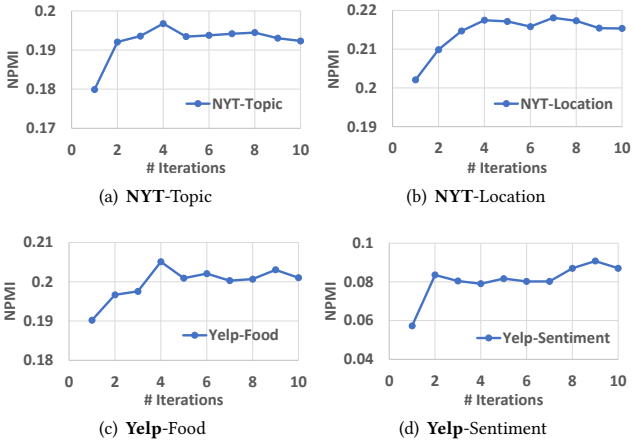
One key design in SeedTopicMine is the ensemble of multiple types of contexts. Specifically, we utilize the context information from three sources: seed-guided text embeddings (**Emb**), pre-trained language model representations (**PLM**), and topic-indicative sentences (**Sntn**). Now, we validate their contribution to the whole framework through an ablation analysis. Specifically, we can ignore one of the three sources while keeping all the other modules unchanged. This yields three ablation versions: **SeedTopicMine-NoEmb**, **SeedTopicMine-NoPLM**, and **SeedTopicMine-NoSntn**.

Table 3: Top-5 terms retrieved by different algorithms. ×: At least 3 of the 5 annotators judge the term as irrelevant to the seed.

Method	**NYT**-Topic		**NYT**-Location		**Yelp**-Food		**Yelp**-Sentiment	
	health	business	france	canada	sushi	desserts	good	bad
SeededLDA	said (×)	said (×)	said (×)	new (×)	roll	food (×)	place (×)	food (×)
	dr (×)	percent (×)	new (×)	city (×)	good (×)	us (×)	food (×)	service (×)
	new (×)	company	state (×)	said (×)	place (×)	order (×)	great	us (×)
	would (×)	year (×)	would (×)	building (×)	food (×)	service (×)	like (×)	order (×)
	hospital	billion (×)	dr (×)	mr (×)	rolls	time (×)	service (×)	time (×)
Anchored CorEx	case (×)	employees	school (×)	market (×)	rolls	also (×)	definitely (×)	one (×)
	court (×)	advertising	students (×)	percent (×)	roll	really (×)	prices (×)	would (×)
	patients	media (×)	children (×)	companies (×)	sashimi	well (×)	strip (×)	like (×)
	cases (×)	businessmen	education (×)	billion (×)	fish (×)	good (×)	selection (×)	could (×)
	lawyer (×)	commerce	schools (×)	investors (×)	tempura	try (×)	value (×)	us (×)
KeyETM	team (×)	percent (×)	city (×)	people (×)	sashimi	food (×)	great	food (×)
	game (×)	japan (×)	state (×)	year (×)	rolls	great (×)	delicious	place (×)
	players (×)	year (×)	york (×)	china (×)	roll	place (×)	amazing	service (×)
	games (×)	japanese (×)	school (×)	years (×)	fish (×)	good (×)	excellent	time (×)
	play (×)	economy	program (×)	time (×)	japanese	service (×)	tasty	restaurant (×)
CatE	public health	diversifying (×)	french	alberta	freshest fish (×)	delicacies (×)	tasty	unforgivable
	health care	clients (×)	corsica	british columbia	sashimi	sundaes	delicious	frustrating
	medical	corporate	spain (×)	ontario	nigiri	savoury (×)	yummy	horrible
	hospitals	investment banking	belgium (×)	manitoba	ayce sushi	pastries	chilaquiles (×)	irritating
	doctors	executives	de (×)	canadian	rolls	custards	also (×)	rude
SeedTopicMine	medical	companies	french	canadian	maki rolls	cheesecakes	great	terrible
	hospitals	businesses	paris	quebec	sashimi	croissants	excellent	horrible
	hospital	corporations	philippe (×)	montreal	ayce sushi	pastries	fantastic	awful
	public health	firms	french state	toronto	revolving sushi	breads (×)	delicious	lousy
	patients	corporate	frenchman	ottawa	nigiri	cheesecake	amazing	shitty

Table 4: Ablation study on different types of context information used in SeedTopicMine.

Method	**Yelp**-Food		**Yelp**-Sentiment	
	P@20	NDCG@20	P@20	NDCG@20
SeedTopicMine	**0.7912**	**0.8379**	**0.9750**	**0.9811**
SeedTopicMine-NoEmb	0.4488	0.5335	0.9550	0.9646
SeedTopicMine-NoPLM	0.6962	0.7602	0.7550	0.8029
SeedTopicMine-NoSntn	0.7488	0.8029	0.9500	0.9631



(a) **NYT**-Topic (b) **NYT**-Location

(c) **Yelp**-Food (d) **Yelp**-Sentiment

Figure 2: Effect of the number of iterations (N) on NPMI.

Table 4 demonstrates the term accuracy scores of the full SeedTopicMine model and the three ablation versions. We can observe that: (1) SeedTopicMine consistently outperforms all three ablation versions, which implies the positive contribution of the three types of context signals. (2) Even for the same dataset (i.e., Yelp), the contribution of a certain type of context information varies significantly with the input seeds. For example, in the food dimension, SeedTopicMine-NoEmb performs the worst, which indicates that seed-guided embeddings are the most helpful signals. Meanwhile, in the sentiment dimension, the contribution of embeddings becomes the smallest. In comparison, pre-trained language model representations have the largest offering. This observation validates the motivation of this work that each type of context information has its specific value and limitation in topic discovery. None of them can dominate the others across all dimensions. Therefore, it becomes necessary to integrate them together, and our results show that the integration does achieve consistently the best performance.

4.4 Parameter Study

Another key design in SeedTopicMine is the iterative framework. To verify the contribution of multiple iterations, we conduct a parameter study by showing the NPMI of the discovered topics if we run SeedTopicMine for different numbers of iterations (i.e., N). Figure 2 shows the effect of N on NPMI across all four dimensions.

From Figure 2, we see that: (1) When N is small (e.g., $N \leq 4$), NPMI increases with N in most cases. When we run SeedTopicMine for only one iteration, the performance is always significantly lower than that when we run 3-4 iterations. This finding validates our design choice that iteratively revising each topic can boost the topic coherence. (2) When N becomes larger, the NPMI curve starts to fluctuate, and the performance gain of running more iterations is subtle. Moreover, more iterations will result in longer

Table 5: Extended qualitative results. ×: At least 3 of the 5 annotators judge the term as irrelevant to the seed.

Dataset	Method	Lower-ranked Terms
NYT-Topic	CatE	**sports**: baseball, football, clubs (×), tennis, coaches, amateur (×), n.b.a, handball
	SeedTopicMine	**sports**: coaches, athletics, players, championships, sportsman, olympians, sporting events, tournament
	CatE	**politics**: rhetoric (×), constituencies (×), vitriolic (×), passivity (×), unprincipled (×), polarized (×), philosophically (×), worldview (×)
	SeedTopicMine	**politics**: democratic, parties, conservative coalition, elected, liberal, electoral, leaders (×), political alliance
Yelp-Food	CatE	**desserts**: churros, chocolate, omelettes (×), crepes, truffles (×), fondue (×), sweets, breakfasts (×)
	SeedTopicMine	**desserts**: candied, scones, truffles (×), tarts, crepes, coffees (×), doughnuts, candies
	CatE	**seafood**: oysters, softshell, paella, fishes, octopus, mussel, mackerel, crawfish
	SeedTopicMine	**seafood**: lobster, clam, seafood, crawfish, blue crab, imitation crab, jumbo shrimp, sardines

running time. Therefore, we believe that setting $N = 4$ strikes a good balance.

4.5 Case Study

We have shown the top-5 terms retrieved by different algorithms in Table 3. One may ask about the quality of lower-ranked terms in each topic. Thus, we conduct an extended case study by showing the 8 lowest-ranked terms among the top 20. These terms are listed in Table 5. Due to space limit, we only show the results of our SeedTopicMine model and the strongest baseline CatE, and two topics are selected for NYT-Topic and Yelp-Food, respectively.

From Table 5, we observe that the accuracy of CatE deteriorates for lower-ranked terms. For example, under the "*politics*" seed from NYT-Topic, all of the 8 shown terms discovered by CatE are judged as irrelevant. By contrast, SeedTopicMine only makes one mistake under the same seed. This observation implies that the efficacy of SeedTopicMine can be generalized to the relatively lower part of the retrieved term list, which also reflects the robustness of SeedTopicMine by integrating multiple types of contexts.

5 RELATED WORK

Seed-Guided Topic Discovery. Different from supervised topic models (e.g., [17, 25]) that rely on a large number of human-annotated documents, seed-guided topic discovery only requires a set of user-interested seeds to find corresponding topics. In [1], seeds are incorporated as prior of topic modeling using must-link and cannot-link constraints. SeededLDA [15] uses seeds to bias topics to produce seed terms and documents to select topics containing them. Anchored CorEx [10] discovers informative topics with correlation maximization and leverages seeds by balancing corpus compression and seed-indicative information. Recent studies also incorporate embedding learning techniques to obtain more accurate semantic representations. For instance, CatE [27] learns category-guided text embeddings by enforcing distinctiveness among seeds in the embedding space; SeeTopic [47] further utilizes the power of pre-trained language models for better text representations and the ability to handle out-of-vocabulary seeds.

Representation-Enhanced Topic Discovery. With the rapid development in text representation learning, recent topic discovery methods incorporate distributed representations to enhance the modeling of text semantics. Earlier approaches incorporate context-free word embeddings [31] into classic probabilistic topic models (e.g., LDA [4]), including Gaussian LDA [6], LFTM [32], Spherical HDP [2], and CGTM [45]. TWE [22] learns embeddings based on associations between words and latent topics obtained by LDA. CLM [44] collaboratively models topics and learns word embeddings by considering both global and local contexts. ETM [8] learns topic embeddings in the word embedding space to improve LDA for a better fit of a large vocabulary. More recent studies leverage the contextualized representations generated by pre-trained language models (e.g., BERT [7]) to facilitate the discovery of coherent topics. These contextualized representations can be used either at token-level for clustering to form topics [29, 39, 42, 49] or at document-level for modeling document-topic correlations [3, 12].

6 CONCLUSIONS AND FUTURE WORK

In this work, we study seed-guided topic discovery by learning from multiple types of contexts, including skip-gram embeddings based on local contexts, pre-trained language model representations upon general-domain pre-training, and topic-indicative sentences retrieved according to seed-distinctive terms. Our proposed SeedTopicMine framework jointly leverages these contexts via an ensemble process for robust topic discovery under different types of seeds. On two real-world datasets and across four sets of seeds, SeedTopicMine consistently outperforms existing seed-guided topic discovery approaches in terms of topic coherence and term accuracy.

For future studies, the promising topic discovery results achieved by SeedTopicMine may further benefit keyword-based text classification [26, 48] via expanding the seed word semantics and prompt-based methods [35] via enriching their verbalizers. Also, SeedTopicMine can be extended to model input seeds organized in a hierarchical manner by injecting hierarchy regularization or discovering topics beyond the provided seeds by incorporating latent topic learning in the corpus modeling process.

ACKNOWLEDGMENTS

We thank Yichen Liu and Ruining Zhao for their help with annotation and anonymous reviewers for their valuable and insightful feedback. Research was supported in part by the IBM-Illinois Discovery Accelerator Institute, US DARPA KAIROS Program No. FA8750-19-2-1004 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

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

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

[3] Federico Bianchi, Silvia Terragni, and Dirk Hovy. 2021. Pre-training is a hot topic: contextualized document embeddings improve topic coherence. In *ACL'21*. 759–766.

[4] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *JMLR* 3 (2003), 993–1022.

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

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

[7] 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*. 4171–4186.

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

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

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

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

[12] Maarten Grootendorst. 2022. BERTopic: Neural topic modeling with a class-based TF-IDF procedure. *arXiv preprint arXiv:2203.05794* (2022).

[13] Bahareh Harandizadeh, J Hunter Priniski, and Fred Morstatter. 2022. Keyword Assisted Embedded Topic Model. In *WSDM'22*. 372–380.

[14] Alexander Hoyle, Pranav Goel, Andrew Hian-Cheong, Denis Peskov, Jordan Boyd-Graber, and Philip Resnik. 2021. Is automated topic model evaluation broken? the incoherence of coherence. In *NeurIPS'21*. 2018–2033.

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

[16] Shoaib Jameel and Steven Schockaert. 2019. Word and Document Embedding with vMF-Mixture Priors on Context Word Vectors. In *ACL'19*. 3319–3328.

[17] Simon Lacoste-Julien, Fei Sha, and Michael I Jordan. 2008. DiscLDA: discriminative learning for dimensionality reduction and classification. In *NIPS'08*. 897–904.

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

[19] Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *ICML'14*. 1188–1196.

[20] Dongha Lee, Jiaming Shen, SeongKu Kang, Susik Yoon, Jiawei Han, and Hwanjo Yu. 2022. TaxoCom: Topic Taxonomy Completion with Hierarchical Discovery of Novel Topic Clusters. In *WWW'22*. 2819–2829.

[21] Ximing Li, Jinjin Chi, Changchun Li, Jihong Ouyang, and Bo Fu. 2016. Integrating topic modeling with word embeddings by mixtures of vMFs. In *COLING'16*. 151–160.

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

[23] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).

[24] 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*. 55–60.

[25] Jon Mcauliffe and David Blei. 2007. Supervised topic models. In *NIPS'07*. 121–128.

[26] Dheeraj Mekala and Jingbo Shang. 2020. Contextualized weak supervision for text classification. In *ACL'20*. 323–333.

[27] Yu Meng, Jiaxin Huang, Guangyuan Wang, Zihan Wang, Chao Zhang, Yu Zhang, and Jiawei Han. 2020. Discriminative topic mining via category-name guided text embedding. In *WWW'20*. 2121–2132.

[28] Yu Meng, Jiaming Shen, Chao Zhang, and Jiawei Han. 2018. Weakly-supervised neural text classification. In *CIKM'18*. 983–992.

[29] Yu Meng, Yunyi Zhang, Jiaxin Huang, Yu Zhang, and Jiawei Han. 2022. Topic Discovery via Latent Space Clustering of Pretrained Language Model Representations. In *WWW'22*. 3143–3152.

[30] Yu Meng, Yunyi Zhang, Jiaxin Huang, Yu Zhang, Chao Zhang, and Jiawei Han. 2020. Hierarchical topic mining via joint spherical tree and text embedding. In *KDD'20*. 1908–1917.

[31] 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*. 3111–3119.

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

[33] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners.

[34] Stephen E Robertson and Steve Walker. 1994. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *SIGIR'94*. 232–241.

[35] Timo Schick and Hinrich Schütze. 2021. Exploiting Cloze-Questions for Few-Shot Text Classification and Natural Language Inference. In *EACL'21*. 255–269.

[36] Mike Schuster and Kaisuke Nakajima. 2012. Japanese and korean voice search. In *ICASSP'12*. 5149–5152.

[37] Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural Machine Translation of Rare Words with Subword Units. In *ACL'16*. 1715–1725.

[38] 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 (2018), 1825–1837.

[39] 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*. 1728–1736.

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

[41] Fangbo Tao, Honglei Zhuang, Chi Wang Yu, Qi Wang, Taylor Cassidy, Lance M Kaplan, Clare R Voss, and Jiawei Han. 2016. Multi-Dimensional, Phrase-Based Summarization in Text Cubes. *IEEE Data Eng. Bull.* 39, 3 (2016), 74–84.

[42] Laure Thompson and David Mimno. 2020. Topic modeling with contextualized word representation clusters. *arXiv preprint arXiv:2010.12626* (2020).

[43] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS'17*. 5998–6008.

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

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

[46] 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*. 1357–1367.

[47] Yu Zhang, Yu Meng, Xuan Wang, Sheng Wang, and Jiawei Han. 2022. Seed-Guided Topic Discovery with Out-of-Vocabulary Seeds. In *NAACL'22*. 279–290.

[48] Yu Zhang, Frank F Xu, Sha Li, Yu Meng, Xuan Wang, Qi Li, and Jiawei Han. 2019. HiGitClass: Keyword-driven hierarchical classification of github repositories. In *ICDM'19*. 876–885.

[49] Zihan Zhang, Meng Fang, Ling Chen, and Mohammad-Reza Namazi-Rad. 2022. Is Neural Topic Modelling Better than Clustering? An Empirical Study on Clustering with Contextual Embeddings for Topics. In *NAACL'22*. 3886–3993.