Discriminative Topic Mining via Category-Name Guided Text Embedding

Yu Meng[1*], Jiaxin Huang[1*], Guangyuan Wang[1], Zihan Wang[1],
Chao Zhang[2], Yu Zhang[1], Jiawei Han[1]
[1]Department of Computer Science, University of Illinois at Urbana-Champaign, IL, USA
[2]College of Computing, Georgia Institute of Technology, GA, USA
[1]{yumeng5, jiaxinh3, gwang10, zihanw2, yuz9, hanj}@illinois.edu [2]chaozhang@gatech.edu

ABSTRACT

Mining a set of meaningful and distinctive topics automatically from massive text corpora has broad applications. Existing topic models, however, typically work in a purely unsupervised way, which often generate topics that do not fit users' particular needs and yield suboptimal performance on downstream tasks. We propose a new task, discriminative topic mining, which leverages a set of user-provided category names to mine discriminative topics from text corpora. This new task not only helps a user understand clearly and distinctively the topics he/she is most interested in, but also benefits directly keyword-driven classification tasks. We develop CatE, a novel category-name guided text embedding method for discriminative topic mining, which effectively leverages minimal user guidance to learn a discriminative embedding space and discover category representative terms in an iterative manner. We conduct a comprehensive set of experiments to show that CatE mines high-quality set of topics guided by category names only, and benefits a variety of downstream applications including weakly-supervised classification and lexical entailment direction identification.

CCS CONCEPTS

• **Information systems** → **Data mining**; *Document topic models*; *Clustering and classification*;

KEYWORDS

Topic Mining, Discriminative Analysis, Text Embedding, Text Classification

ACM Reference Format:
Yu Meng[1*], Jiaxin Huang[1*], Guangyuan Wang[1], Zihan Wang[1], Chao Zhang[2], Yu Zhang[1], Jiawei Han[1]. 2020. Discriminative Topic Mining via Category-Name Guided Text Embedding. In *Proceedings of The Web Conference 2020 (WWW '20), April 20–24, 2020, Taipei, Taiwan.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3366423.3380278

*Equal Contribution.

WWW '20, April 20–24, 2020, Taipei, Taiwan

ACM ISBN 978-1-4503-7023-3/20/04.
https://doi.org/10.1145/3366423.3380278

1 INTRODUCTION

To help users effectively and efficiently comprehend a large set of text documents, it is of great interest to generate a set of meaningful and coherent topics automatically from a given corpus. Topic models [6, 19] are such unsupervised statistical tools that discover latent topics from text corpora. Due to their effectiveness in uncovering hidden semantic structure in text collections, topic models are widely used in text mining [15, 29] and information retrieval tasks [14, 52].

Despite of their effectiveness, traditional topic models suffer from two noteworthy limitations: (1) *Failure to incorporate user guidance.* Topic models tend to retrieve the most general and prominent topics from a text collection, which may not be of a user's particular interest, or provide a skewed and biased summarization of the corpus. (2) *Failure to enforce distinctiveness among retrieved topics.* Concepts are most effectively interpreted via their uniquely defining features. For example, Egypt is known for pyramids and China is known for the Great Wall. Topic models, however, do not impose disriminative constraints, resulting in vague interpretations of the retrieved topics. Table 1 demonstrates three retrieved topics from the New York Times (**NYT**) annotated corpus [42] via LDA [6]. We can see that it is difficult to clearly define the meaning of the three topics due to an overlap of their semantics (*e.g.*, the term "united states" appears in all three topics).

Table 1: LDA retrieved topics on NYT dataset. The meanings of the retrieved topics have overlap with each other.

Topic 1	Topic 2	Topic 3
canada, united states canadian, economy	sports, united states olympic, games	united states, iraq government, president

In order to incorporate user knowledge or preference into topic discovery for mining distinctive topics from a text corpus, we propose a new task, **Discriminative Topic Mining**, which takes only a set of category names as user guidance, and aims to retrieve a set of representative and discriminative terms under each provided category. In many cases, a user may have a specific set of interested topics in mind, or have prior knowledge about the potential topics in a corpus. Such user interest or prior knowledge may come naturally in the form of a set of category names that could be used to guide the topic discovery process, resulting in more desirable results that better cater to a user's need and fit specific downstream applications. For example, a user may provide several country names and rely on discriminative topic mining to retrieve each country's provinces, cities, currency, etc. from a text

corpus. We will show that this new task not only helps the user to clearly and distinctively understand his/her interested topics, but also benefits keywords-driven classification tasks.

There exist previous studies that attempt to incorporate prior knowledge into topic models. Along one line of work, supervised topic models such as Supervised LDA [5] and DiscLDA [23] guide the model to predict category labels based on document-level training data. While they do improve the discriminative power of unsupervised topic models on classification tasks, they rely on massive hand-labeled documents, which may be difficult to obtain in practical applications. Along another line of work that is more similar to our setting, users are asked to provide a set of seed words to guide the topic discovery process, which is referred to as seed-guided topic modeling [1, 21]. However, they still do not impose requirements on the distinctiveness of the retrieved topics and thus are not optimized for discriminative topic presentation and other applications such as keyword-driven classification.

We develop a novel category-name guided text embedding method, CatE, for discriminative topic mining. CatE consists of two modules: (1) A *category-name guided text embedding learning module* that takes a set of category names to learn category distinctive word embeddings by modeling the text generative process conditioned on the user provided categories, and (2) a *category representative word retrieval module* that selects category representative words based on both word embedding similarity and word distributional specificity. The two modules collaborate in an iterative way: At each iteration, the former refines word embeddings and category embeddings for accurate representative word retrieval; and the latter selects representative words that will be used by the former at the next iteration.

Our contributions can be summarized as follows.

(1) We propose discriminative topic mining, a new task for topic discovery from text corpora with a set of category names as the only supervision. We show qualitatively and quantitatively that this new task helps users obtain a clear and distinctive understanding of interested topics, and directly benefits keyword-driven classification tasks.
(2) We develop a category-name guided text embedding framework for discriminative topic mining by modeling the text generation process. The model effectively learns a category distinctive embedding space that best separates the given set of categories based on word-level supervision.
(3) We propose an unsupervised method that jointly learns word embedding and word distributional specificity, which allow us to consider both relatedness and specificity when retrieving category representative terms. We also provide theoretical interpretations of the model.
(4) We conduct a comprehensive set of experiments on a variety of tasks including topic mining, weakly-supervised classification and lexical entailment direction identification to demonstrate the effectiveness of our model on these tasks.

2 PROBLEM FORMULATION

Definition 1 (Discriminative Topic Mining)**.** Given a text corpus $\mathcal{D}$ and a set of category names $C = \{c_1, \ldots, c_n\}$, discriminative topic mining aims to retrieve a set of terms $\mathcal{S}_i = \{w_1, \ldots, w_m\}$ from $\mathcal{D}$ for each category c_i such that each term in $\mathcal{S}_i$ semantically belongs to and only belongs to category c_i.

Example 1. Given a set of country names, c_1: "The United States", c_2: "France" and c_3: "Canada", it is correct to retrieve "Ontario" as an element in $\mathcal{S}_3$, because Ontario is a province in Canada and exclusively belongs to Canada semantically. However, it is incorrect to retrieve "North America" as an element in $\mathcal{S}_3$, because North America is a continent and does not belong to any countries. It is also incorrect to retrieve "English" as an element in $\mathcal{S}_3$, because English is also the national language of the United States.

The differences between discriminative topic mining and standard topic modeling are mainly two-fold: (1) Discriminative topic mining requires a set of user provided category names and only focuses on retrieving terms belonging to the given categories. (2) Discriminative topic mining imposes strong discriminative requirements that each retrieved term under the corresponding category must belong to and only belong to that category semantically.

3 CATEGORY-NAME GUIDED EMBEDDING

In this section, we first formulate a text generative process under user guidance, and then cast the learning of the generative process as a category-name guided text embedding model. Words, documents and categories are jointly embedded into a shared space where embeddings are not only learned according to the corpus generative assumption, but also encouraged to incorporate category distinctive information.

3.1 Motivation

Traditional topic models like LDA [6] use document-topic and topic-word distributions to model the text generation process, where an obvious defect exists due to the bag-of-words generation assumption—each word is drawn independently from the topic-word distribution without considering the correlations between adjacent words. In addition, topic models make explicit probabilistic assumptions regarding the text generation mechanism, resulting in high model complexity and inflexibility [16].

Along another line of text representation research, word embeddings like Word2Vec [33] effectively capture word semantic correlations by mapping words with similar *local contexts* closer in the embedding space. They do not impose particular assumptions on the type of data distribution of the corpus and enjoy greater flexibility and higher efficiency. However, word embeddings usually do not exploit document-level co-occurrences of words (*i.e.*, *global contexts*) and also cannot naturally incorporate latent topics into the model without making topic-relevant generative assumptions.

To take advantage of both lines of work for mining topics from text corpora, we propose to explicitly model the text generation process and cast it as an embedding learning problem.

3.2 Modeling Text Generation Under User Guidance

When the user provides n category names, we assume text generation is a three-step process: (1) First, a document d is generated conditioned on one of the n categories (this is similar to the assumption in multi-class classification problems where each document

belongs to exactly one of the categories); (2) second, each word w_i is generated conditioned on the semantics of the document d; and (3) third, surrounding words w_{i+j} in the local context window ($-h \le j \le h, j \neq 0$, h is the local context window size) of w_i are generated conditioned on the semantics of the center word w_i. Step (1) explicitly models the associations between each document and user-interested categories (*i.e.*, *topic assignment*). Step (2) makes sure each word is generated in consistency with the semantics of its belonging document (*i.e.*, *global contexts*). Step (3) models the correlations of adjacent words in the corpus (*i.e.*, *local contexts*). Putting the above pieces together, we have the following expression for the likelihood of corpus generation conditioned on a specific set of user-interested categories C:

$$P(\mathcal{D} \mid C) = \prod_{d \in \mathcal{D}} p(d \mid c_d) \prod_{w_i \in d} p(w_i \mid d) \prod_{\substack{w_{i+j} \in d \\ -h \le j \le h, j \neq 0}} p(w_{i+j} \mid w_i) \tag{1}$$

where c_d is the latent category of d.

Taking the negative log-likelihood as our objective $\mathcal{L}$, we have

$$\begin{aligned} \mathcal{L} = & -\sum_{d \in \mathcal{D}} \log p(d \mid c_d) \qquad (\mathcal{L}_{\text{topic}}) \\ & -\sum_{d \in \mathcal{D}} \sum_{w_i \in d} \log p(w_i \mid d) \qquad (\mathcal{L}_{\text{global}}) \\ & -\sum_{d \in \mathcal{D}} \sum_{w_i \in d} \sum_{\substack{w_{i+j} \in d \\ -h \le j \le h, j \neq 0}} \log p(w_{i+j} \mid w_i). \qquad (\mathcal{L}_{\text{local}}) \end{aligned} \tag{2}$$

In Eq. (2), $p(w_i \mid d)$ and $p(w_{i+j} \mid w_i)$ are observable (*e.g.*, $p(w_i \mid d) = 1$ if w_i appears in d, and $p(w_i \mid d) = 0$ otherwise), while $p(d \mid c_d)$ is latent (*i.e.*, we do not know which category d belongs to). To directly leverage the word level user supervisions (*i.e.*, category names), a natural solution is to decompose $p(d \mid c_d)$ into word-topic distributions:

$$p(d \mid c_d) \propto p(c_d \mid d) p(d) \propto p(c_d \mid d) \propto \prod_{w \in d} p(c_d \mid w),$$

where the first proportionality is derived via Bayes rule; the second derived assuming $p(d)$ is constant; and the third assumes $p(c_d \mid d)$ is jointly decided by all words in d.

Next, we rewrite the first term in Eq. (2) (*i.e.*, $\mathcal{L}_{\text{topic}}$) by reorganizing the summation over categories instead of documents:

$$\mathcal{L}_{\text{topic}} = -\sum_{d \in \mathcal{D}} \log p(d \mid c_d) = -\sum_{c \in C} \sum_{w \in c} p(c \mid w) + \text{const.}$$

Now $\mathcal{L}_{\text{topic}}$ is expressed in $p(c \mid w)$, the category assignment of words. This is exactly the task we aim for—finding words that belong to the categories.

3.3 Embedding Learning

In this subsection, we introduce how to formulate the optimization of the objective in Eq. (2) as an embedding learning problem.

Similar to previous work [7, 33], we define the three probability expressions in Eq. (2) via log-linear models in the embedding space:

$$p(c_i \mid w) = \frac{\exp(\boldsymbol{c}_i^\top \boldsymbol{u}_w)}{\sum_{c_j \in C} \exp(\boldsymbol{c}_j^\top \boldsymbol{u}_w)}, \tag{3}$$

$$p(w_i \mid d) = \frac{\exp(\boldsymbol{u}_{w_i}^\top \boldsymbol{d})}{\sum_{d' \in \mathcal{D}} \exp(\boldsymbol{u}_{w_i}^\top \boldsymbol{d'})}, \tag{4}$$

$$p(w_{i+j} \mid w_i) = \frac{\exp(\boldsymbol{u}_{w_i}^\top \boldsymbol{v}_{w_{i+j}})}{\sum_{w' \in V} \exp(\boldsymbol{u}_{w_i}^\top \boldsymbol{v}_{w'})}, \tag{5}$$

where $\boldsymbol{u}_w$ is the input vector representation of w (usually used as the word embedding); $\boldsymbol{v}_w$ is the output vector representation that serves as w's contextual representation; $\boldsymbol{d}$ is the document embedding; $\boldsymbol{c}_i$ is the category embedding. Please note that Eqs. (4) and (5) are not yet the final design of our embedding model, as we will propose an extension of them in Section 4.2 that leads to a more effective and suitable model for discriminative topic mining.

While Eqs. (4) and (5) can be directly plugged into Eq. (2) to train word and document embeddings, Eq. (3) requires knowledge about the latent topic (*i.e.*, the category that w belongs to) of a word w. Initially, we only know the user-provided category names belong to their corresponding categories, but during the iterative topic mining process, we will retrieve more terms under each category, gradually discovering the latent topic of more words.

To this end, we design the following for learning $\mathcal{L}_{\text{topic}}$ in Eq. (2): Let $\boldsymbol{p}_w = \begin{bmatrix} p(c_1 \mid w) & \dots & p(c_n \mid w) \end{bmatrix}^\top$ be the probability distribution of w over all classes. If a word w is known to belong to class c_i, $\boldsymbol{p}_w$ computed from Eq. (3) should become a one-hot vector $\boldsymbol{l}_w$ (*i.e.*, the category label of w) with $p(c_i \mid w) = 1$. To achieve this property, we minimize the KL divergence from each category representative word's distribution $\boldsymbol{p}_w$ to its corresponding discrete delta distribution $\boldsymbol{l}_w$. Formally, given a set of class representative words $\mathcal{S}_i$ (we will introduce how to retrieve $\mathcal{S}_i$ in Section 4) for category c_i, the $\mathcal{L}_{\text{topic}}$ term is implemented as:

$$\mathcal{L}_{\text{topic}} = \sum_{c_i \in C} \sum_{w \in \mathcal{S}_i} KL\left(\boldsymbol{l}_w \| \boldsymbol{p}_w\right). \tag{6}$$

From the embedding learning perspective, Eq. (6) is equivalent to a cross-entropy regularization loss, encouraging the category embeddings to become distinctive anchor points in the embedding space that are far from each other and are surrounded by their current retrieved class representative terms.

4 CATEGORY REPRESENTATIVE WORD RETRIEVAL

In this section, we detail how to retrieve category representative words (*i.e.*, the words that belong to and only belong to a category) for topic mining.

As a starting point, we propose to retrieve category representative terms by jointly considering two separate aspects: Relatedness and specificity. In particular, a representative word w of category c should satisfy simultaneously two constraints: (1) w is semantically related to c, and (2) w is semantically more specific than the category name of c. Constraint (1) can be imposed by simply requiring high cosine similarity between a candidate word embedding and the category embedding. However, constraint (2) is not naturally captured by the text embedding space. Hence, we are motivated to improve the previous text embedding model by incorporating word specificity signals.

In the following, we first present the concept of word distributional specificity, and then introduce how to capture the signal effectively in our model. Finally, we describe how to retrieve category representative words by jointly considering the two constraints.

4.1 Word Distributional Specificity

We adapt the concept of distributional generality in [51] and define word distributional specificity as below.

Definition 2 (Word Distributional Specificity)**.** We assume there is a scalar $\kappa_w \geq 0$ correlated with each word w indicating how specific the word meaning is. The bigger κ_w is, the more specific meaning word w has, and the less varying contexts w appears in.

The above definition is grounded on the distributional inclusion hypothesis [59] which states that hyponyms are expected to occur in a subset of the contexts of their hypernyms.

For example, "*seafood*" has a higher word distributional specificity than "*food*", because seafood is a specific type of food.

4.2 Jointly Learning Word Embedding and Distributional Specificity

In this subsection, we propose an extension of Eqs. (4) and (5) to jointly learn word embedding and word distributional specificity in an *unsupervised* way.

Specifically, we modify Eqs. (4) and (5) to incorporate an additional learnable scalar κ_w for each word w, while constraining the embeddings to be on the unit hyper-sphere $\mathbb{S}^{p-1} \subset \mathbb{R}^p$, motivated by the fact that directional similarity is more effective in capturing semantics [30].

Formally, we re-define the probability expressions in Eqs. (4) and (5) to be[1]:

$$p(w_i \mid d) = \frac{\exp(\kappa_{w_i} \boldsymbol{u}_{w_i}^\top \boldsymbol{d})}{\sum_{d' \in \mathcal{D}} \exp(\kappa_{w_i} \boldsymbol{u}_{w_i}^\top \boldsymbol{d'})}, \tag{7}$$

$$p(w_{i+j} \mid w_i) = \frac{\exp(\kappa_{w_i} \boldsymbol{u}_{w_i}^\top \boldsymbol{v}_{w_{i+j}})}{\sum_{w' \in V} \exp(\kappa_{w_i} \boldsymbol{u}_{w_i}^\top \boldsymbol{v}_{w'})}, \tag{8}$$

$$s.t. \quad \forall w, d, c, \quad \|\boldsymbol{u}_w\| = \|\boldsymbol{v}_w\| = \|\boldsymbol{d}\| = \|\boldsymbol{c}\| = 1.$$

In practice, the unit norm constraints can be satisfied by simply normalizing the embedding vectors after each update[2]. Under the above setting, the parameter κ_w learned is the distributional specificity of w.

4.3 Explaining the Model

We explain here why the additional parameter κ_w in Eqs. (7) and (8) effectively captures word distributional specificity. We first introduce a spherical distribution, and then show how our model is connected to the properties of the distribution.

Definition 3 (The von Mises Fisher (vMF) distribution)**.** A unit random vector $\boldsymbol{x} \in \mathbb{S}^{p-1} \subset \mathbb{R}^p$ has the p-variate von Mises Fisher distribution $vMF_p(\boldsymbol{\mu}, \kappa)$ if its probability dense function is

$$f(\boldsymbol{x}; \boldsymbol{\mu}, \kappa) = c_p(\kappa) \exp(\kappa \boldsymbol{\mu}^\top \boldsymbol{x}),$$

where $\kappa \geq 0$ is the concentration parameter, $\|\boldsymbol{\mu}\| = 1$ is the mean direction, and the normalization constant $c_p(\kappa)$ is given by

$$c_p(\kappa) = \frac{\kappa^{p/2-1}}{(2\pi)^{p/2} I_{p/2-1}(\kappa)},$$

where $I_r(\cdot)$ represents the modified Bessel function of the first kind at order r.

Theorem 1. When the corpus size and vocabulary size are infinite (*i.e.*, $|\mathcal{D}| \to \infty$ and $|V| \to \infty$) and all p-dimensional word vectors and document vectors are unit vectors, generalizing Eqs. (7) and (8) to the continuous cases results in the p-variate vMF distribution with the center word vector $\boldsymbol{u}_{w_i}$ as the mean direction and κ_{w_i} as the concentration parameter, *i.e.*,

$$\lim_{|V| \to \infty} p(w_{i+j} \mid w_i) = c_p(\kappa_{w_i}) \exp(\kappa_{w_i} \boldsymbol{u}_{w_i}^\top \boldsymbol{v}_{w_{i+j}}), \tag{9}$$

$$\lim_{|V| \to \infty} p(w_i \mid d) = c_p(\kappa_{w_i}) \exp(\kappa_{w_i} \boldsymbol{u}_{w_i}^\top \boldsymbol{d}). \tag{10}$$

Proof. We give the proof for Eq. (9). The proof for Eq. (10) can be derived similarly.

We generalize the relationship proportionality $p(w_{i+j} \mid w_i) \propto \exp(\kappa_{w_i} \boldsymbol{u}_{w_i}^\top \boldsymbol{v}_{w_{i+j}})$ in Eq. (8) to the continuous case and obtain the following probability density distribution:

$$\begin{aligned} \lim_{|V| \to \infty} p(w_{i+j} \mid w_i) &= \frac{\exp(\kappa_{w_i} \boldsymbol{u}_{w_i}^\top \boldsymbol{v}_{w_{i+j}})}{\int_{\mathbb{S}^{p-1}} \exp(\kappa_{w_i} \boldsymbol{u}_{w_i}^\top \boldsymbol{v}_{w'}) d\boldsymbol{v}_{w'}} \\ &\triangleq \frac{\exp(\kappa_{w_i} \boldsymbol{u}_{w_i}^\top \boldsymbol{v}_{w_{i+j}})}{Z}, \end{aligned} \tag{11}$$

where Z denotes the integral in the denominator.

The probability density function of vMF distribution integrates to 1 over the entire sphere, *i.e.*,

$$\int_{\mathbb{S}^{p-1}} c_p(\kappa_{w_i}) \exp(\kappa_{w_i} \boldsymbol{u}_{w_i}^\top \boldsymbol{v}_{w'}) d\boldsymbol{v}_{w'} = 1,$$

we have

$$Z = \int_{\mathbb{S}^{p-1}} \exp(\kappa_{w_i} \boldsymbol{u}_{w_i}^\top \boldsymbol{v}_{w'}) d\boldsymbol{v}_{w'} = \frac{1}{c_p(\kappa_{w_i})}.$$

Plugging Z back to Eq. (11), we obtain

$$\lim_{|V| \to \infty} p(w_{i+j} \mid w_i) = c_p(\kappa_{w_i}) \exp(\kappa_{w_i} \boldsymbol{u}_{w_i}^\top \boldsymbol{v}_{w_{i+j}}).$$

□

Theorem 1 reveals the underlying generative assumption of the joint learning model defined in Section 4.2—the contexts vectors are assumed to be generated from the vMF distribution with the center word vector $\boldsymbol{u}_{w_i}$ as the mean direction and κ_{w_i} as the concentration parameter. Our model essentially learns both word embedding and word distributional specificity that maximize the probability of the context vectors getting generated by the center word's vMF distribution. Figure 1 shows two words with different distributional specificity. "*Food*" has more general meaning than "*seafood*" and appears in more diverse contexts. Therefore, the learned vMF distribution of "*food*" will have a lower concentration parameter than that of "*seafood*". In other words, "*food*" has a lower distributional specificity than "*seafood*".

[1] Eq. (3) is not refined with the κ parameter because we do not aim to learn category specificity.

[2] Alternatively, one may apply the Riemannian optimization techniques in the spherical space as described in [30].

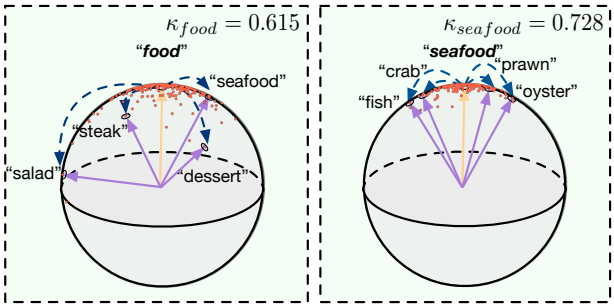


Figure 1: Word Distributional Specificity.

4.4 Selecting Category Representative Words

Finally, the learned distributional specificity can be used to impose the constraint that class representative words should belong to the category. Specifically, a category representative word must have higher distributional specificity than the category name. However, we also want to avoid selecting too specific terms as category representative words. From the embedding learning perspective, words with higher semantic specificity may appear fewer times in the corpus and suffer from lower embedding quality and higher variance due to insufficient training, which can lead to the distortion of the category embedding manifold if they are selected as category representative words.

Therefore, among all the words that are more specific than the category name, we prefer words that (1) have high embedding cosine similarity with the category name, and (2) have low distributional specificity, which indicates wider semantic coverage. Formally, we find a representative word of category c_i and add it to the set $\mathcal{S}$ by

$$w = \arg\min_{w} \text{rank}_{sim}(w, c_i) \cdot \text{rank}_{spec}(w) \quad s.t. \quad w \notin \mathcal{S} \quad \text{and} \quad \kappa_w > \kappa_{c_i}, \tag{12}$$

where $\text{rank}_{sim}(w, c_i)$ is the ranking of w by embedding cosine similarity with category c_i, *i.e.*, $\cos(\boldsymbol{u}_w, \boldsymbol{c}_i)$, from high to low; $\text{rank}_{spec}(w)$ is the ranking of w by distributional specificity, *i.e.*, κ_w, from low to high.

4.5 Overall Algorithm

We summarize the overall algorithm of discriminative topic mining in Algorithm 1.

Initially, the set of class representative words $\mathcal{S}_i$ is simply the category name. During training, $\mathcal{S}_i$ gradually incorporates more class representative words so that the category embedding models more accurate and complete class semantics. The embeddings of class representative words are directly enforced by Eq. (6) to encode category distinctive information, and this weak supervision signal will pass to other words through Eqs. (7) and (8) so that the resulting embedding space is specifically fine-tuned to distinguish the given set of categories.

Algorithm 1: Discriminative Topic Mining.

Input: A text corpus $\mathcal{D}$; a set of category names $C = \{c_i\}|_{i=1}^{n}$.
Output: Discriminative topic mining results $\mathcal{S}_i|_{i=1}^{n}$.
for $i \leftarrow 1$ *to* n **do**
  $\mathcal{S}_i \leftarrow \{c_i\}$ ▷ initialize $\mathcal{S}_i$ with category names;
for $t \leftarrow 1$ *to* max_iter **do**
  Train $\boldsymbol{W}, \boldsymbol{C}$ on $\mathcal{D}$ according to Equation (2);
  for $i \leftarrow 1$ *to* n **do**
    $w \leftarrow$ Select representative word of c_i by Eq. (12);
    $\mathcal{S}_i \leftarrow \mathcal{S}_i \cup \{w\}$;
for $i \leftarrow 1$ *to* n **do**
  $\mathcal{S}_i \leftarrow \mathcal{S}_i \setminus \{c_i\}$ ▷ exclude category names;
Return $\mathcal{S}_i|_{i=1}^{n}$;

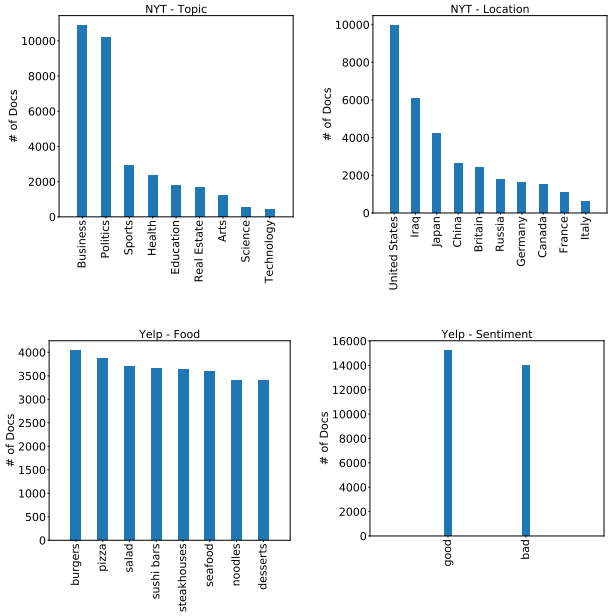


Figure 2: Dataset statistics.

5 EXPERIMENTS

5.1 Experiment Setup

Datasets. We use two datasets, the New York Times annotated corpus (**NYT**) [42], the recently released *Yelp Dataset Challenge* (**Yelp**)[3]. **NYT** and **Yelp** each has two sets of categories: **NYT**: *topic* and *location*; **Yelp**: *food type* and *sentiment*. For **NYT**, we first select the major categories (with more than 100 documents) from topics and locations, and then collect documents that are single-labeled on both set of categories, *i.e.*, each document has exactly one ground truth topic label and one ground truth location label. We do the same for **Yelp**. The category names and the number of documents in each category can be found in Figure 2.

Implementation Details and Parameters. Since the full softmax in Eqs. (7) and (8) results in computational complexity proportional to the vocabulary size, we adopt the negative sampling strategy

[3] https://www.yelp.com/dataset/challenge

[33] for efficient approximation. The training objective (Eq. (2)) is optimized with SGD. We pre-process the corpus by discarding infrequent words that appear less than 5 times in the corpus. We use AutoPhrase [44] to extract quality phrases, and the phrases are treated as single words during embedding training. For fair comparisons with baselines, we set the hyperparameters as below for all methods: word embedding dimension $p = 100$, local context window size $h = 5$, number of negative samples $k = 5$, training iterations on the corpus $max\_iter = 10$. Other parameters (if any) are set to be the default values of the corresponding algorithm. In CatE, the word distributional specificity parameter κ_w is initialized to 1 for each word.

5.2 Discriminative Topic Mining

Compared Methods. We compare CatE with the following baselines including traditional topic modeling, seed-guided topic modeling and embedding-based topic modeling. For all the baseline methods that require the number of topics N as input, we vary N in $[n, 2n, \ldots, 10n]$ where n is the actual number of categories, and report the best performance of the method. Note that our method CatE does not require any tuning of N and directly uses the provided category names as the only supervision.

- LDA [6]: LDA is the standard topic model that learns topic-word and document-topic distributions by modeling the generative process of the corpus. It is unsupervised and cannot incorporate seed words as supervision. We manually select the most relevant topics to the provided category names.
- Seeded LDA [21]: Seeded LDA biases the regular topic model generative process by introducing a seed topic distribution induced by input seed words to encourage the model to focus on user-interested topics. We provide the category names as seed words.
- TWE [26]: TWE has three models for learning word embedding under a set of topics. For all three models, we use the topic specific word representation of the category names to retrieve representative phrases under each category, and report the best performance of the three models.
- Anchored CorEx [16]: CorEx does not rely on generative assumptions and learns maximally informative topics measured by total correlation. Anchored CorEx incorporates user-provided seed words by balancing between compressing the original corpus and preserving anchor words related information. We provide the category names as seed words.
- Labeled ETM [13]: ETM uses the distributed representation of word embedding to enhance the robustness of topic models to rare words. We use the labeled ETM version which is more robust to stop words. The top phrases are retrieved according to embedding similarity with the category name.
- CatE: Our proposed method retrieves category representative terms according to both embedding similarity and distributional specificity, as described by Eq. (12).

Evaluation Metrics. We apply two metrics on the top-m ($m = 10$ in our experiments) words/phrases retrieved under each category to evaluate all methods:

- Topic coherence (TC) is a standard metric [24] in topic modeling which measures the coherence of terms inside each topic, and is defined as the average normalized pointwise mutual information of two words randomly drawn from the same document, *i.e.*,

$$\text{TC} = \frac{1}{\binom{m}{2} \cdot n} \sum_{k=1}^{n} \sum_{i=1}^{m} \sum_{j=i+1}^{m} -\frac{\log \frac{P(w_i, w_j)}{P(w_i)P(w_j)}}{\log P(w_i, w_j)},$$

  where $P(w_i, w_j)$ is the probability of w_i and w_j co-occurring in a document; $P(w_i)$ is the marginal probability of w_i.
- Mean accuracy (MACC) measures the proportion of retrieved top words that actually belong to the category defined by user-provided category names, *i.e.*,

$$\text{MACC} = \frac{1}{n} \sum_{k=1}^{n} \frac{1}{m} \sum_{i=1}^{m} \mathbb{1}(w_i \in c_k),$$

  where $\mathbb{1}(w_i \in c_k)$ is the indicator function of whether w_i belongs to category c_k. We invite five graduate students to independently label whether each retrieved term belongs to the corresponding category. The final results are the averaged labeling of the five annotators.

Results. We show both qualitative and quantitative evaluation results. We randomly select two categories from **NYT**-Location, **NYT**-Topic, **Yelp**-Food Type and **Yelp**-Sentiment respectively, and show top-5 words/phrases retrieved by all methods under each category in Table 2. Terms that are determined by more than half of the human annotators to not belong to the corresponding category are marked with (×). We measure the topic modeling quality by TC and MACC across all categories and report the results in Table 3.

Discussions. From Tables 2 and 3, we observe that the standard topic model (LDA) retrieves reasonably good topics (even better than Seeded LDA and Anchored CorEx in some cases) relevant to category names, as long as careful manual selection of topics is performed. However, inspecting all the topics to select one's interested topics is inefficient and costly for users, especially when the number of topics is large.

When users can provide a set of category names, seed guided topic modeling methods can directly retrieve relevant topics of user's interest, alleviating the burden of manual selection. Among the four guided topic modeling baselines, Seeded LDA and Anchored CorEx suffer from noisy retrieval results—some categories are dominated by off-topic terms. For example, Anchored CorEx retrieves words related to sports under the location category "canada", which should have been put under the topic category "sports".

The above issue is rarely observed in the results of the other two embedding-based topic modeling baselines, TWE and Labeled ETM, because they employ distributed word representations when modeling topic word correlation, requiring the retrieved words to be semantically relevant to the category names. However, their results contain terms that are relevant but do not actually belong to the corresponding category. For example, Labeled ETM retrieves "france", "germany" and "europe" under the location category "britain". In short, TWE and Labeled ETM lacks *discriminative* power over the set of provided categories, and fails to compare the relative *generality/specificity* between a pair of terms (*e.g.*, it is correct to put "british" under "europe", but incorrect vice versa).

Our proposed method CatE enjoys the benefits brought by word embeddings, and explicitly regularizes the embedding space to become discriminative for the provided set of categories. In addition,

Table 2: Qualitative evaluation on discriminative topic mining.

Methods	**NYT**-Location		**NYT**-Topic		**Yelp**-Food		**Yelp**-Sentiment	
	britain	canada	education	politics	burger	desserts	good	bad
LDA	company (×)	percent (×)	school	campaign	fatburger	ice cream	great	valet (×)
	companies (×)	economy (×)	students	clinton	dos (×)	chocolate	place (×)	peter (×)
	british	canadian	city (×)	mayor	liar (×)	gelato	love	aid (×)
	shares (×)	united states (×)	state (×)	election	cheeseburgers	tea (×)	friendly	relief (×)
	great britain	trade (×)	schools	political	bearing (×)	sweet	breakfast	rowdy
Seeded LDA	british	city (×)	state (×)	republican	like (×)	great (×)	place (×)	service (×)
	industry (×)	building (×)	school	political	fries	like (×)	great	did (×)
	deal (×)	street (×)	students	senator	just (×)	ice cream	service (×)	order (×)
	billion (×)	buildings (×)	city (×)	president	great (×)	delicious (×)	just (×)	time (×)
	business (×)	york (×)	board (×)	democrats	time (×)	just (×)	ordered (×)	ordered (×)
TWE	germany (×)	toronto	arts (×)	religion	burgers	chocolate	tasty	subpar
	spain (×)	osaka (×)	fourth graders	race	fries	complimentary (×)	decent	positive (×)
	manufacturing (×)	booming (×)	musicians (×)	attraction (×)	hamburger	green tea (×)	darned (×)	awful
	south korea (×)	asia (×)	advisors	era (×)	cheeseburger	sundae	great	crappy
	markets (×)	alberta	regents	tale (×)	patty	whipped cream	suffered (×)	honest (×)
Anchored CorEx	moscow (×)	sports (×)	republican (×)	military (×)	order (×)	make (×)	selection (×)	did (×)
	british	games (×)	senator (×)	war (×)	know (×)	chocolate	prices (×)	just (×)
	london	players (×)	democratic (×)	troops (×)	called (×)	people (×)	great	came (×)
	german (×)	canadian	school	baghdad (×)	fries	right (×)	reasonable	asked (×)
	russian (×)	coach	schools	iraq (×)	going (×)	want (×)	mac (×)	table (×)
Labeled ETM	france (×)	canadian	higher education	political	hamburger	pana	decent	horrible
	germany (×)	british columbia	educational	expediency (×)	cheeseburger	gelato	great	terrible
	canada (×)	britain (×)	school	perceptions (×)	burgers	tiramisu	tasty	good (×)
	british	quebec	schools	foreign affairs	patty	cheesecake	bad (×)	awful
	europe (×)	north america (×)	regents	ideology	steak (×)	ice cream	delicious	appallingly
CatE	england	ontario	educational	political	burgers	dessert	delicious	sickening
	london	toronto	schools	international politics	cheeseburger	pastries	mindful	nasty
	britons	quebec	higher education	liberalism	hamburger	cheesecakes	excellent	dreadful
	scottish	montreal	secondary education	political philosophy	burger king	scones	wonderful	freaks
	great britain	ottawa	teachers	geopolitics	smash burger	ice cream	faithful	cheapskates

Table 3: Quantitative evaluation on discriminative topic mining.

Methods	**NYT**-Location		**NYT**-Topic		**Yelp**-Food		**Yelp**-Sentiment	
	TC	MACC	TC	MACC	TC	MACC	TC	MACC
LDA	0.007	0.489	0.027	0.744	-0.033	0.213	-0.197	0.350
Seeded LDA	0.024	0.168	0.031	0.456	0.016	0.188	0.049	0.223
TWE	0.002	0.171	-0.011	0.289	0.004	0.688	-0.077	0.748
Anchored CorEx	0.029	0.190	0.035	0.533	0.025	0.313	0.067	0.250
Labeled ETM	0.032	0.493	0.025	0.889	0.012	0.775	0.026	0.852
CatE	**0.049**	**0.972**	**0.048**	**0.967**	**0.034**	**0.913**	**0.086**	**1.000**

CatE learns the semantic specificity of each term in the corpus and enforces the words/phrases retrieved to be more specific than the category names. As shown in Tables 2 and 3, CatE correctly retrieves distinctive terms that indeed belong to the category.

5.3 Weakly-Supervised Text Classification

In this subsection, we show that the discriminative power of CatE benefits document-level classification, and we explore the application of CatE to document classification under weak supervision.

Weakly-supervised text classification [8, 31, 32, 45, 58] uses category names or a set of keywords from each category instead of human annotated documents to train a classifier. It is especially preferable when manually labeling massive training documents is costly or difficult.

Previous weakly-supervised document classification studies use unsupervised word representations to either retrieve from knowledge base relevant articles to category names as training data [45], or derive similar words and form pseudo training data for pre-training classifiers [31, 32]. In this work, we do not propose new models for weakly-supervised document classification, but simply replace the unsupervised embeddings used in previous systems with CatE, based on the intuition that when the supervision is given on word-level, deriving discriminative word embeddings at the early stage is beneficial for all subsequent steps in weakly-supervised classification.

In particular, we use WeSTClass [31, 32] as the weakly-supervised classification model. WeSTClass first models topic distribution in the word embedding space to retrieve relevant words to the given category names, and applies self-training to bootstrap the model on unlabeled documents. It uses Word2Vec [33] as the word representation. In the following, we experiment with different word embedding models as input features to WeSTClass.

Compared Methods. We note here that our goal is *not* designing a weakly-supervised classification method; instead, our purpose is to show that CatE benefits classification tasks with stronger discriminative power than unsupervised text embedding models by only leveraging category names. In this sense, our contribution is improving the input text feature quality for document classification when

Table 4: Weakly-supervised text classification evaluation based on WeSTClass [31] model.

Embedding	**NYT**-Location		**NYT**-Topic		**Yelp**-Food		**Yelp**-Sentiment	
	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1
Word2Vec	0.533	0.467	0.588	0.695	0.540	0.528	0.723	0.715
GloVe	0.521	0.455	0.563	0.688	0.515	0.503	0.720	0.711
fastText	0.543	0.485	0.575	0.693	0.544	0.529	0.738	0.743
BERT	0.301	0.288	0.328	0.451	0.330	0.404	0.695	0.674
CatE	**0.655**	**0.613**	**0.611**	**0.739**	**0.656**	**0.648**	**0.838**	**0.836**

category names are available. To the best of our knowledge, this is the first work that proposes to learn discriminative text embedding only from category names (*i.e.*, without requiring additional information other than the supervision given for weakly-supervised classification). We compare CatE with the following unsupervised text embedding baselines as input features to the state-of-the-art weakly-supervised classification model WeSTClass [31, 32].

- Word2Vec [33]: Word2Vec is a predictive word embedding model that learns distributed representations by maximizing the probability of using the center word to predict its local context words or in the opposite way.
- GloVe [38]: GloVe learns word embedding by factorizing a global word-word co-occurrence matrix where the co-occurrence is defined upon a fix-sized context window.
- fastText [7]: fastText is an extension of Word2Vec which learns word embedding efficiently by incorporating subword information. It uses the sum of vector representations of all n-grams in a word to predict context words in a fix-sized window.
- BERT [11]: BERT is a state-of-the-art pretrained language model that provides contextualized word representations. It trains bidirectional Transformers [48] to predict randomly masked words and consecutive sentence relationships.

Evaluation Metrics. We employ the Micro-F1 and Macro-F1 scores that are commonly used in multi-class classification evaluations [31, 32] as the metrics.

Results. We first train all the embedding models on the corresponding corpus (except BERT which we take its pre-trained model and fine-tune it on the corpus), and use the trained embedding as the word representation to WeSTClass [31, 32]. The weakly-supervised classification results on **NYT**-Location, **NYT**-Topic, **Yelp**-Food Type and **Yelp**-Sentiment are shown in Table 4.

Discussions. From Table 4, we observe that: (1) Unsupervised embeddings (Word2Vec, GloVe and fastText) do not really have notable differences as word representations to WeSTClass; (2) Despite its great effectiveness as a pre-trained deep language model for supervised tasks, BERT is not suitable for classification without sufficient training data, probably because BERT embedding has higher dimensionality (even the base model of BERT is 768-dimensional) which might require stronger supervision signals to tune; (3) CatE outperforms all unsupervised embeddings on **NYT**-Location and **Yelp**-Food Type and **Yelp**-Sentiment categories by a large margin, and have marginal advantage on **NYT**-Topic. This is probably because different locations (*e.g.*, "Canada" vs. "The United States"), food types (*e.g.*, "burgers" vs. "pizza"), and sentiment polarities (*e.g.*, "good" vs. "bad") can have highly similar local contexts, and are more difficult to be differentiated than themes. CatE explicitly regularizes the embedding space for the specific categories and becomes especially advantageous when the given category names are semantically similar.

There have been very few previous efforts in the text classification literature that dedicate to learning discriminative word embeddings from word-level supervisions, and word embeddings are typically fine-tuned jointly with classification models [22, 50, 56] via document-level supervisions. However, our study shows that under label scarcity scenarios, using word-level supervision only can bring significant improvements to weakly-supervised models. Therefore, it might be beneficial for future weakly-supervised/semi-supervised studies to also consider leveraging word-level supervision to gain a performance boost.

5.4 Unsupervised Lexical Entailment Direction Identification

In CatE, we enforce the retrieved terms to be more specific than the given category name by comparing their learned distributional specificity values κ. Since κ characterizes the semantic generality of a term, it can be directly applied to identify the direction in lexical entailment.

Lexical entailment (LE) [49] refers to the "type-of" relation, also known as hyponymy-hypernymy relation in NLP. LE typically includes two tasks: (1) Discriminate hypernymy from other relations (detection) and (2) Identify from a hyponymy-hypernymy pair which one is hyponymy (direction identification). Recently, there has been a line of supervised (*i.e.*, require labeled hyponymy-hypernymy pairs as training data) embedding studies [36, 37, 47] that learn hyperbolic word embeddings to capture the lexical entailment relationships.

In our evaluation, we focus on unsupervised methods for LE direction identification, which is closer to the application of CatE.

Compared Methods. We compare CatE with the following unsupervised baselines that can identify the direction in a given hyponymy-hypernymy pair.

- Frequency [51]: This baseline simply uses the frequency of a term in the corpus to characterize its generality. It hypothesizes that hypernyms are more frequent than hyponyms in the corpus.
- SLQS [43]: SLQS measures the generality of a term via the entropy of its statistically most prominent context.
- Vec-Norm: It is shown in [35] that the L-2 norm of word embedding indicates the generality of a term, *i.e.*, a general term tends to have a lower embedding norm, because it co-occurs with many different terms and its vector is dragged from different directions.

Benchmark Test Set. Following [43], we use the BLESS [2] dataset for LE direction identification. BLESS contains 1,337 unique hyponym-hypernym pairs. The task is to predict the directionality of hypernymy within each pair.

Results. We train all models on the latest Wikipedia dump[4] containing 2.4 billion tokens and report the accuracy for hypernymy direction identification in Table 5.

Table 5: Lexical entailment direction identification.

Methods	Frequency	SLQS	Vec-Norm	CatE
Accuracy	0.659	0.861	0.562	**0.895**

Discussions. Our method achieves the highest accuracy on identifying the direction of lexical entailment among a pair of words, which explains the great effectiveness of CatE on retrieving terms that belong to a category. Another desirable property of CatE is that distributional specificity is jointly trained along with the text embedding, and can be directly obtained as a by-product. Our learning of word distributional specificity is based on the distributional inclusion hypothesis [59] and has a probabilistic interpretation presented in Section 4.3.

5.5 Case Study

Discriminative Embedding Space. In this case study, we demonstrate the effect of regularizing the embedding space with category representative words. Specifically, we apply t-SNE [27] to visualize the embeddings trained on **NYT**-Location in Figure 3 where category embeddings are denoted as stars, and the retrieved class representative phrases are denoted as points with the same color as their ground-truth corresponding categories. At the early stage of training (Figure 3(a)), words from different categories are mixed together. During training, the categories are becoming well-separated. Category representative words gather around their corresponding category in the embedding space, which encourages other semantically similar words to move towards their belonging categories as well (Figure 3 shows more words than class representative words retrieved by our model during training).

Coarse-to-Fine Topic Presentation. In the second set of case studies, we demonstrate the learned word distributional specificity with concrete examples from **NYT**-Topic, and illustrate how it helps present a topic in a coarse-to-fine manner. Table 6 lists the most similar phrases with each category (measured by embedding cosine similarity) from different ranges of distributional specificity. When κ is smaller, the retrieved words have wider semantic coverage.

A drawback of traditional topic modeling is that it presents each category via a top ranked list according to topic-word distribution, which usually seems randomly-ordered because latent probability distribution is generally hard to be interpreted by humans. In our model, however, one can sort the retrieved phrases under each topic according to distributional specificity, so that the topic mining results can be viewed in a coarse-to-fine manner.

6 RELATED WORK

We review two lines of related work that are most relevant to our task: Topic modeling and task-oriented text embedding.

[4] https://dumps.wikimedia.org/enwiki/latest/enwiki-latest-pages-articles.xml.bz2

6.1 Topic Modeling

Topic models discover semantically relevant terms that form coherent topics via probabilistic generative models. Unsupervised topic models have been studied for decades, among which pLSA [19] and LDA [6] are the most famous ones, serving as the backbone for many future variants. The basic idea is to represent documents via mixtures over latent topics, where each topic is characterized by a distribution over words. Subsequent studies lead to a large number of variants such as Hierarchical LDA [18], Correlated Topic Models [4], Pachinko Allocation [25] and Concept Topic Models [9]. Although unsupervised topic models are sufficiently expressive to model multiple topics per document, they are unable to incorporate the category and label information into their learning procedure.

Several modifications of topic models have been proposed to incorporate supervision. Supervised LDA [5] and DiscLDA [23] assume each document is associated with a label and train the model by predicting the document category label. Author Topic Models [40] and Multi-Label Topic Models [41] further model each document as a bag of words with a bag of labels. However, these models obtain topics that do not correspond directly to the labels. Labeled LDA [39] and SSHLDA [28] can be used to solve this problem. However, all the *supervised* models mentioned above requires sufficient annotated documents, which are expensive to obtain in some domains. In contrast, our model relies on very weak supervisions (*i.e.*, a set of category names) which are much easier to obtain.

Several studies leverage word-level supervision to build topic models. For example, Dirichlet Forest [1] has been used as priors to incorporate must-link and cannot-link constraints among seed words. Seeded LDA [21] takes user-provided seed words as supervision to learn seed-related topics via a seed topic distribution. CorEx [16] learns maximally informative topics from the corpus and uses total correlation as the measure. It can incorporate seed words by jointly compressing the text corpus and preserving seed relevant information. However, none of the above systems *explicitly* model distinction among different topics, and they also do not require the retrieved terms to belong to the provided categories. As a result, the retrieved topics still suffer from irrelevant term intrusion, as we will demonstrate in the experiment section.

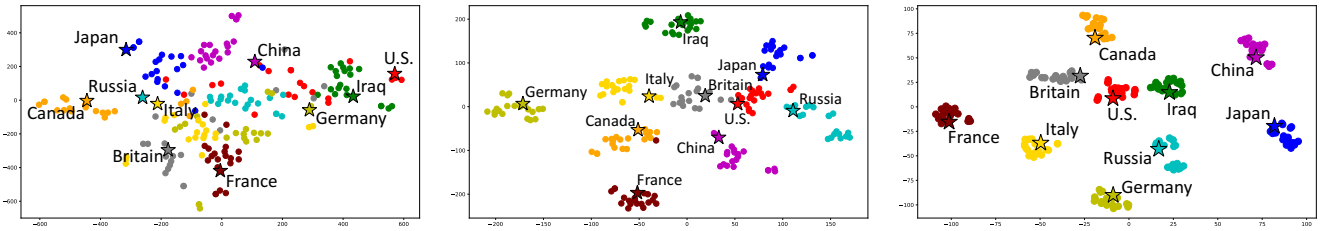
With the development of word embeddings [7, 33, 38], several studies propose to extend LDA to involve word embeddings. One common strategy is to convert the discrete text into continuous representations of embeddings, and then adapt LDA to generate real-valued data [3, 10, 54, 55]. There are a few other ways of combining LDA and embeddings. For example, [34] mixes the likelihood defined by LDA with a log-linear model that uses pre-fitted word embeddings; [53] adopts a geometric perspective, using Wasserstein distances to learn topics and word embeddings jointly; [13] uses the distributed representation of word embedding to enhance the robustness of topic models to rare words. Motivated by the success of these recent topic models, we model the text generation process in the embedding space, and propose several designs to tailor our model for the task of discriminative topic mining.

6.2 Task-Oriented Text Embedding

Discriminative text embeddings are typically trained under a supervised manner with task specific training data, such as training

Table 6: Coarse-to-fine topic presentation on NYT-Topic.

Range of κ	Science ($\kappa_c = 0.539$)	Technology ($\kappa_c = 0.566$)	Health ($\kappa_c = 0.527$)
$\kappa_c < \kappa < 1.25\kappa_c$	scientist, academic, research, laboratory	machine, equipment, devices, engineering	medical, hospitals, patients, treatment
$1.25\kappa_c < \kappa < 1.5\kappa_c$	physics, sociology, biology, astronomy	information technology, computing, telecommunication, biotechnology	mental hygiene, infectious diseases, hospitalizations, immunizations
$1.5\kappa_c < \kappa < 1.75\kappa_c$	microbiology, anthropology, physiology, cosmology	wireless technology, nanotechnology, semiconductor industry, microelectronics	dental care, chronic illnesses, cardiovascular disease, diabetes
$\kappa > 1.75\kappa_c$	national science foundation, george washington university, hong kong university, american academy	integrated circuits, assemblers, circuit board, advanced micro devices	juvenile diabetes, high blood pressure, family violence, kidney failure



(a) Epoch 1 (b) Epoch 3 (c) Epoch 5

Figure 3: Discriminative embedding space training for topic mining.

CNNs [22] or RNNs [56] for text classification. Among supervised word embedding models, some previous studies are more relevant because they explicitly leverage the category information to optimize embedding for classification tasks. Predictive Text Embedding (PTE) [46] constructs a heterogeneous text network and jointly embeds words, documents and labels based on word-word and word-document co-occurrences as well as labeled documents. Label-Embedding Attentive Model [50] jointly embeds words and labels so that attention mechanisms can be employed to discover category distinctive words. All the above frameworks require labeled training documents for fine-tuning word embeddings. Our method only requires category names to learn a discriminative embedding space over the categories, which are much easier to obtain.

Some recent studies propose to learn embeddings for lexical entailment, which is relevant to our task because it may help determine which terms belong to a category. Hyperbolic models such as Poincaré [12, 36, 47], Lorentz [37] and hyperbolic cone [17] have proven successful in graded lexical entailment detection. However, the above models are supervised and require hypernym-hyponym training pairs, which may not be available under the setting of topic discovery. Our model jointly learns the word vector representation in the embedding space and its distributional specificity without requiring supervision, and simultaneously considers relatedness and specificity of words when retrieving category representative terms.

7 CONCLUSIONS AND FUTURE WORK

In this paper, we first propose a new task for topic discovery, discriminative topic mining, which aims to mine distinctive topics from text corpora guided by category names only. Then we introduce a category-name guided word embedding framework CatE that learns category distinctive text embedding by modeling the text generation process conditioned on the user provided categories. CatE effectively retrieves class representative terms based on both relatedness and specificity of words, by jointly learning word embedding and word distributional specificity. Experiments show that CatE retrieves high-quality distinctive topics, and benefits downstream tasks including weakly-supervised document classification and unsupervised lexical entailment direction identification.

In the future, we are interested in extending CatE to not only focus on user provided categories, but also have the potential to discover other latent topics in a text corpus, probably via distant supervision from knowledge bases. There are a wide range of downstream tasks that may benefit from CatE. For example, we would like to exploit CatE for unsupervised taxonomy construction [57] by applying CatE recursively at each level of the taxonomy to find potential children nodes. Furthermore, CatE might help entity set expansion via generating auxiliary sets consisting of relevant words to seed words [20].

ACKNOWLEDGMENTS

Research was sponsored in part by DARPA under Agreements No. W911NF-17-C-0099 and FA8750-19-2-1004, National Science Foundation IIS 16-18481, IIS 17-04532, and IIS-17-41317, and DTRA HDTRA11810026. Any opinions, findings, and conclusions or recommendations expressed in this document are those of the author(s) and should not be interpreted as the views of any U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon. We thank anonymous reviewers for valuable and insightful feedback.

REFERENCES

[1] David Andrzejewski and Xiaojin Zhu. 2009. Latent Dirichlet Allocation with Topic-in-Set Knowledge. In *HLT-NAACL*.

[2] Marco Baroni and Alessandro Lenci. 2011. How we BLESSed distributional semantic evaluation. In *EMNLP*.

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

[4] David Blei and John Lafferty. 2006. Correlated topic models. In *NIPS*. 147.

[5] David M Blei and Jon D Mcauliffe. 2008. Supervised topic models. In *NIPS*. 121–128.

[6] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. In *NIPS*.

[7] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics* 5 (2017), 135–146.

[8] Ming-Wei Chang, Lev-Arie Ratinov, Dan Roth, and Vivek Srikumar. 2008. Importance of Semantic Representation: Dataless Classification. In *AAAI*.

[9] Chaitanya Chemudugunta, Padhraic Smyth, and Mark Steyvers. 2008. Combining concept hierarchies and statistical topic models. In *CIKM*. 1469–1470.

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

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT*.

[12] Bhuwan Dhingra, Christopher J. Shallue, Mohammad Norouzi, Andrew M. Dai, and George E. Dahl. 2018. Embedding Text in Hyperbolic Spaces. In *TextGraphs@NAACL-HLT*.

[13] Adji B. Dieng, Francisco J. R. Ruiz, and David M. Blei. 2019. Topic Modeling in Embedding Spaces. *ArXiv* abs/1907.04907 (2019).

[14] Zhicheng Dou, Ruihua Song, and Ji-Rong Wen. 2007. A large-scale evaluation and analysis of personalized search strategies. In *WWW*.

[15] George F. Foster and Roland Kuhn. 2007. Mixture-Model Adaptation for SMT. In *WMT@ACL*.

[16] Ryan J. Gallagher, Kyle Reing, David C. Kale, and Greg Ver Steeg. 2017. Anchored Correlation Explanation: Topic Modeling with Minimal Domain Knowledge. *TACL* (2017).

[17] Octavian-Eugen Ganea, Gary Bécigneul, and Thomas Hofmann. 2018. Hyperbolic Entailment Cones for Learning Hierarchical Embeddings. In *ICML*.

[18] Thomas L Griffiths, Michael I Jordan, Joshua B Tenenbaum, and David M Blei. 2004. Hierarchical topic models and the nested Chinese restaurant process. In *NIPS*. 17–24.

[19] Thomas Hofmann. 1999. Probabilistic Latent Semantic Indexing. In *SIGIR*.

[20] Jiaxin Huang, Yiqing Xie, Yu Meng, Jiaming Shen, Yunyi Zhang, and Jiawei Han. 2020. Guiding Corpus-based Set Expansion by Auxiliary Sets Generation and Co-Expansion. In *WWW*.

[21] Jagadeesh Jagarlamudi, Hal Daumé, and Raghavendra Udupa. 2012. Incorporating Lexical Priors into Topic Models. In *EACL*.

[22] Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *EMNLP*.

[23] Simon Lacoste-Julien, Fei Sha, and Michael I Jordan. 2009. DiscLDA: Discriminative learning for dimensionality reduction and classification. In *NIPS*. 897–904.

[24] Jey Han Lau, David Newman, and Timothy Baldwin. 2014. Machine Reading Tea Leaves: Automatically Evaluating Topic Coherence and Topic Model Quality. In *EACL*.

[25] Wei Li and Andrew McCallum. 2006. Pachinko allocation: DAG-structured mixture models of topic correlations. In *ICML*. 577–584.

[26] Yang Liu, Zhiyuan Liu, Tat-Seng Chua, and Maosong Sun. 2015. Topical Word Embeddings. In *AAAI*.

[27] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, Nov (2008), 2579–2605.

[28] Xian-Ling Mao, Zhao-Yan Ming, Tat-Seng Chua, Si Li, Hongfei Yan, and Xiaoming Li. 2012. SSHLDA: a semi-supervised hierarchical topic model. In *EMNLP*. 800–809.

[29] Qiaozhu Mei, Xuehua Shen, and ChengXiang Zhai. 2007. Automatic labeling of multinomial topic models. In *KDD*.

[30] Yu Meng, Jiaxin Huang, Guangyuan Wang, Chao Zhang, Honglei Zhuang, Lance Kaplan, and Jiawei Han. 2019. Spherical Text Embedding. In *NeurIPS*.

[31] Yu Meng, Jiaming Shen, Chao Zhang, and Jiawei Han. 2018. Weakly-Supervised Neural Text Classification. In *CIKM*.

[32] Yu Meng, Jiaming Shen, Chao Zhang, and Jiawei Han. 2019. Weakly-Supervised Hierarchical Text Classification. In *AAAI*.

[33] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *NIPS*.

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

[35] Kim Anh Nguyen, Maximilian Köper, Sabine Schulte im Walde, and Ngoc Thang Vu. 2017. Hierarchical Embeddings for Hypernymy Detection and Directionality. In *EMNLP*.

[36] Maximilian Nickel and Douwe Kiela. 2017. Poincaré Embeddings for Learning Hierarchical Representations. In *NIPS*.

[37] Maximilian Nickel and Douwe Kiela. 2018. Learning Continuous Hierarchies in the Lorentz Model of Hyperbolic Geometry. In *ICML*.

[38] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global Vectors for Word Representation. In *EMNLP*.

[39] Daniel Ramage, David Hall, Ramesh Nallapati, and Christopher D Manning. 2009. Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora. In *EMNLP*. 248–256.

[40] Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth. 2004. The author-topic model for authors and documents. In *UAI*. 487–494.

[41] Timothy N Rubin, America Chambers, Padhraic Smyth, and Mark Steyvers. 2012. Statistical topic models for multi-label document classification. *Machine learning* 88, 1-2 (2012), 157–208.

[42] Evan Sandhaus. 2008. The New York Times Annotated Corpus.

[43] Enrico Santus, Alessandro Lenci, Qin Lu, and Sabine Schulte im Walde. 2014. Chasing Hypernyms in Vector Spaces with Entropy. In *EACL*.

[44] Jingbo Shang, Jialu Liu, Meng Jiang, Xiang Ren, Clare R. Voss, and Jiawei Han. 2018. Automated Phrase Mining from Massive Text Corpora. *IEEE Transactions on Knowledge and Data Engineering* 30 (2018), 1825–1837.

[45] Yangqiu Song and Dan Roth. 2014. On Dataless Hierarchical Text Classification. In *AAAI*.

[46] Jian Tang, Meng Qu, and Qiaozhu Mei. 2015. PTE: Predictive Text Embedding through Large-scale Heterogeneous Text Networks. In *KDD*.

[47] Alexandru Tifrea, Gary Bécigneul, and Octavian-Eugen Ganea. 2019. Poincaré Glove: Hyperbolic Word Embeddings. In *ICLR*.

[48] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. In *NIPS*.

[49] Ivan Vulic, Daniela Gerz, Douwe Kiela, Felix Hill, and Anna Korhonen. 2017. HyperLex: A Large-Scale Evaluation of Graded Lexical Entailment. *Computational Linguistics* (2017).

[50] Guoyin Wang, Chunyuan Li, Wenlin Wang, Yizhe Zhang, Dinghan Shen, Xinyuan Zhang, Ricardo Henao, and Lawrence Carin. 2018. Joint Embedding of Words and Labels for Text Classification. In *ACL*.

[51] Julie Weeds, David J. Weir, and Diana McCarthy. 2004. Characterising Measures of Lexical Distributional Similarity. In *COLING*.

[52] Xing Wei and W. Bruce Croft. 2006. LDA-based document models for ad-hoc retrieval. In *SIGIR*.

[53] Hongteng Xu, Wenlin Wang, Wei Liu, and Lawrence Carin. 2018. Distilled wasserstein learning for word embedding and topic modeling. In *NIPS*. 1716–1725.

[54] Guangxu Xun, Vishrawas Gopalakrishnan, Fenglong Ma, Yaliang Li, Jing Gao, and Aidong Zhang. 2016. Topic discovery for short texts using word embeddings. In *ICDM*. 1299–1304.

[55] 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*.

[56] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alexander J. Smola, and Eduard H. Hovy. 2016. Hierarchical Attention Networks for Document Classification. In *HLT-NAACL*.

[57] Chao Zhang, Fangbo Tao, Xiusi Chen, Jiaming Shen, Meng Jiang, Brian M. Sadler, Michelle T. Vanni, and Jiawei Han. 2018. TaxoGen: Constructing Topical Concept Taxonomy by Adaptive Term Embedding and Clustering. In *KDD*.

[58] 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*.

[59] Maayan Zhitomirsky-Geffet and Ido Dagan. 2005. The Distributional Inclusion Hypotheses and Lexical Entailment. In *ACL*.