Hierarchical Topic Mining via Joint Spherical Tree and Text Embedding

Yu Meng[1*], Yunyi Zhang[1*], Jiaxin Huang[1], Yu Zhang[1], Chao Zhang[2], 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, yzhan238, jiaxinh3, yuz9, hanj}@illinois.edu [2]chaozhang@gatech.edu

ABSTRACT

Mining a set of meaningful topics organized into a hierarchy is intuitively appealing since topic correlations are ubiquitous in massive text corpora. To account for potential hierarchical topic structures, hierarchical topic models generalize flat topic models by incorporating latent topic hierarchies into their generative modeling process. However, due to their purely unsupervised nature, the learned topic hierarchy often deviates from users' particular needs or interests. To guide the hierarchical topic discovery process with minimal user supervision, we propose a new task, Hierarchical Topic Mining, which takes a category tree described by category names only, and aims to mine a set of representative terms for each category from a text corpus to help a user comprehend his/her interested topics. We develop a novel joint tree and text embedding method along with a principled optimization procedure that allows simultaneous modeling of the category tree structure and the corpus generative process in the spherical space for effective category-representative term discovery. Our comprehensive experiments show that our model, named JoSH, mines a high-quality set of hierarchical topics with high efficiency and benefits weakly-supervised hierarchical text classification tasks[1].

CCS CONCEPTS

• **Information systems** → **Data mining**; *Document topic models*; *Clustering and classification*; • **Computing methodologies** → **Natural language processing**;

KEYWORDS

Topic Mining; Topic Hierarchy; Text Embedding; Tree Embedding

ACM Reference Format:
Yu Meng[1*], Yunyi Zhang[1*], Jiaxin Huang[1], Yu Zhang[1], Chao Zhang[2], Jiawei Han[1]. 2020. Hierarchical Topic Mining via Joint Spherical Tree and Text Embedding. In *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '20), August 23–27, 2020, Virtual Event, CA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3394486.3403242

[1]Source code can be found at https://github.com/yumeng5/JoSH.

[*]Equal Contribution.

KDD '20, August 23–27, 2020, Virtual Event, CA, USA

ACM ISBN 978-1-4503-7998-4/20/08…$15.00
https://doi.org/10.1145/3394486.3403242

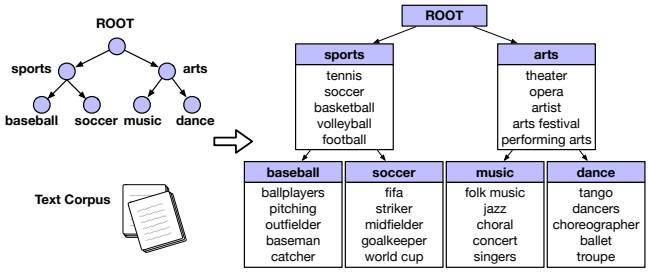


Figure 1: An example of Hierarchical Topic Mining. We aim to retrieve a set of representative terms from a given corpus for each category in a user-provided hierarchy.

1 INTRODUCTION

Topic models [6, 11], which uncover hidden semantic structure in a text corpus via generative modeling, have proven successful on automatic topic discovery. Hierarchical topic models [4, 29] extend the classical ones by considering a latent topic hierarchy during the corpus generative process, motivated by the fact that topics are naturally correlated (*e.g.*, "sports" is a super-topic of "soccer"). Due to their effectiveness of discovering organized topic structures automatically without human supervision, hierarchical topic models have been applied to a wide range of applications including political text analysis [10], entity disambiguation [15] and relation extraction [1].

Despite being able to learn latent topic hierarchies from text corpora, the applicability of hierarchical topic models to learn a user-interested topic structure is limited seriously by their unsupervised nature: Unsupervised generative models maximize the likelihood of the observed data, tending to discover the most general and prominent topics from a text collection, which may not fit a user's particular interest, or provide a superficial summarization of the corpus. Furthermore, the inference algorithms of topic models yield local optimum solutions, resulting in instability and inconsistency across different runs. This issue even worsens in the hierarchical setting where a larger number of topics and their correlations need to be modeled.

In many cases, a user is interested in a specific topic structure, or has prior knowledge about the potential topics in a corpus. These topics, based on a user's interest or prior knowledge, may be easily described via a set of category names with a hierarchical structure. Such a user-provided category hierarchy will facilitate a more stable

topic discovery process, yielding more desirable and consistent results that better cater to a user's needs. Therefore, we propose a new task, **Hierarchical Topic Mining**, which takes only a topic hierarchy described by category names as user guidance, and aims to retrieve a set of coherent and representative terms under each category to help users comprehend his/her interested topics. For example, as shown in Figure 1, a user may provide a hierarchy of interested concepts along with a corpus and rely on hierarchical topic mining to retrieve a set of representative terms from a text corpus (*e.g.*, different music and dance genres, terminologies for different sports, as well as general descriptions for internal nodes) that provide a clear interpretation of the categories.

Several previous studies also focus on guiding topic discovery with word-level supervision. Seed-guided topic modeling [2, 14] incorporates user-provided seed words to bias the generative process towards seed-related topics. A recent study CatE [24] learns discriminative text embeddings guided by category names for representative term retrieval. However, none of the above methods handle hierarchical topic structures. Under the hierarchical setting, there are supervised [33] and semi-supervised [22] models that leverage category labels of documents to regularize the generative process. However, they rely on a large amount of annotated documents which may be costly to obtain. Under our setting, only a set of easy-to-provide category names that form a topic hierarchy is needed to guide the hierarchical topic discovery process.

In this paper, we propose JoSH, a novel **Jo**int **S**pherical tree and text embedding model for **H**ierarchical Topic Mining. The user-provided category tree structure and text corpus statistics are simultaneously modeled via directional similarity in the spherical space, which facilitates effective estimation of category-word semantic correlations for representative term discovery. To train our model in the spherical space, we develop a principled EM optimization procedure based on Riemannian optimization.

Our contributions can be summarized as follows.

(1) We propose a new task for hierarchical topic discovery, Hierarchical Topic Mining, which requires a category hierarchy described by category names as the only supervision to retrieve a set of representative terms per category for effective topic understanding.
(2) We develop a joint embedding framework for hierarchical topic mining by simultaneously modeling the user-provided category tree structure and the text generation process. The model is defined in the spherical space, where directional similarity is employed to characterize semantic correlations among words, documents, and categories for accurate category representative term retrieval.
(3) We develop an EM algorithm to optimize our model in the spherical space that iterates between estimating the latent category of words and maximizing corpus generative likelihood while optimizing the category tree structure in the embedding space.
(4) We conduct a comprehensive set of experiments on two public corpora from different domains on Hierarchical Topic Mining. Our model enjoys high efficiency and mines high-quality topics. The embeddings trained by our model can be directly used for weakly-supervised hierarchical text classification.

2 PROBLEM FORMULATION

Definition 1 (Hierarchical Topic Mining)**.** Given a text corpus $\mathcal{D}$ and a tree-structured hierarchy $\mathcal{T}$ where each node $c_i \in \mathcal{T}$ is represented by the name of the category, **Hierarchical Topic Mining** aims to retrieve a set of terms $C_i = \{w_1, \ldots, w_m\}$ from $\mathcal{D}$ for each category $c_i \in \mathcal{T}$ such that C_i provides a clear description of the category c_i based on $\mathcal{D}$.

Connection and difference between Hierarchical Topic Models. Similar to Hierarchical Topic Modeling [4], we also aim to capture the hierarchical correlations among topics during topic discovery. However, **Hierarchical Topic Mining** is weakly-supervised as it requires the user to provide the names of the hierarchy categories which serve as the minimal supervision and focuses on retrieving representative terms only for the provided categories.

3 SPHERICAL TEXT AND TREE EMBEDDING

In this section, we introduce our model JoSH which jointly learns text embeddings and tree embeddings in the spherical space, where directional similarity is used to effectively characterize semantic correlations among words, documents and categories.

3.1 Motivation

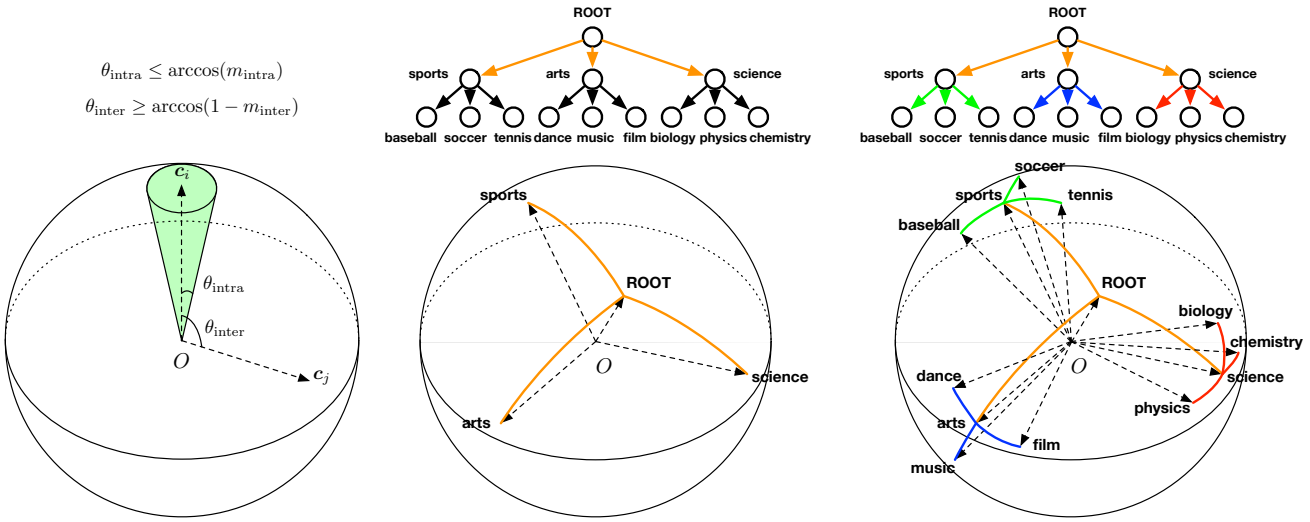
Mining representative terms relevant to a given category relies on accurate estimation of semantic similarity, on which *directional similarity* of text embeddings has proven most effective. For example, cosine similarity is empirically shown [18] to better characterize word semantic similarity and dissimilarity. Motivated by the effectiveness of directional similarity for text analysis, several recent studies employ the spherical space for topic modeling [3], text embedding learning [25] and text sequence generation [16]. To learn text embeddings tailored for the given category tree, we propose to jointly embed the tree structure into the spherical space where each category is surrounded by its representative terms.

Different from recent hyperbolic tree embedding models, such as Poincaré embedding [30], Lorentz model [31] and hyperbolic cones [9], we do not preserve the *absolute* tree distance in the embedding space, but rather the *relative* category relationship reflected in the tree structure. For example, in the category hierarchy given in Figure 1, although the tree distance between "sports" and "arts" and that between "baseball" and "soccer" are both 2, the latter pair of categories should be embedded closer than the former pair due to higher semantic similarity. Therefore, the tree distance in the category hierarchy should not be preserved in an absolute manner, but treated as a relative metric, *e.g.*, for category "soccer", its tree distance to "sports" is smaller than that to "baseball", so "soccer" should be embedded closer to "sports" than to "baseball".

3.2 Spherical Tree Embedding

We propose a novel tree embedding method that preserves the relative category hierarchical structure in the spherical embedding space, meanwhile encouraging inter-category distinctiveness for clear topic interpretation.

3.2.1 The Flat Case. We start with the simplest case where all categories are parallel and do not exhibit hierarchical structures. We aim to jointly embed categories and their representative terms



(a) Intra- & Inter-Category Configuration. (b) Embed First-Level Local Tree. (c) Embed Second-Level Local Trees.

Figure 2: Spherical tree embeddings. All category center vectors reside on the unit sphere. (a) Representative terms are pushed into a spherical sector centered around the category center vector. Directional distance is enforced between categories. (b) & (c) Local trees are recursively embedded onto the sphere.

such that (1) the representative terms selected for each category[2] are semantically coherent and (2) the categories are distinctive from each other, which allows clear category interpretation. For example, in Figure 1, one can clearly recognize and understand "baseball" and "soccer" thanks to the discriminative terms that are exclusively relevant to the corresponding category.

Intra-Category Coherence. The representative terms of each category should be highly semantically relevant to each other, reflected by high directional similarity in the spherical space. To achieve this, we require the embeddings of representative terms to be placed near the category center direction within a local region by maximizing

$$\mathcal{L}_{\text{intra}} = \sum_{c_i \in \mathcal{T}} \sum_{w_j \in C_i} \min(0, \boldsymbol{u}_{w_j}^\top \boldsymbol{c}_i - m_{\text{intra}}), \tag{1}$$

where $\boldsymbol{u}_w$ is the word embedding of w; $\boldsymbol{c}_i$ is the category center vector of c_i. Note that $\boldsymbol{u}_{w_j}^\top \boldsymbol{c}_i = \cos(\boldsymbol{u}_{w_j}, \boldsymbol{c}_i)$ since the vectors reside on the unit sphere $\mathbb{S}^{p-1} \subset \mathbb{R}^p$. We set $m_{\text{intra}} = 0.9$ which works well in general since it requires high cosine similarity between representative words and the category center.

When $\mathcal{L}_{\text{intra}}$ is maximized (*i.e.*, $\forall w_j \in C_i, \boldsymbol{u}_{w_j}^\top \boldsymbol{c}_i \geq m_{\text{intra}}$), the representative word embeddings of the corresponding category reside in a spherical sector centered around the category center vector.

Inter-Category Distinctiveness. We would like to encourage distinctiveness across different categories to avoid semantic overlaps so that the retrieved terms provide a clear and distinctive description of the category. To accomplish this, we enforce inter-category directional dissimilarity by requiring the cosine distance between any two categories to be larger than m_{inter}, *i.e.*,

$$\forall c_i, c_j (c_i \neq c_j), 1 - \boldsymbol{c}_i^\top \boldsymbol{c}_j > m_{\text{inter}}.$$

Therefore, we maximize the following objective:

$$\mathcal{L}_{\text{inter}} = \sum_{c_i \in \mathcal{T}} \sum_{c_j \in \mathcal{T} \setminus \{c_i\}} \min(0, 1 - \boldsymbol{c}_i^\top \boldsymbol{c}_j - m_{\text{inter}}). \tag{2}$$

We will introduce how to set m_{inter} in Section 3.2.2.

Figure 2(a) shows the configuration of category center vectors upon enforcing intra-category coherence and inter-category distinctiveness.

3.2.2 Recursive Local Tree Embedding. We generalize the ideas in the flat case to the hierarchical case and recursively embed local structures of the category tree such that the relative category relationship is preserved.

We first define the local tree structure that we work with at each recursive step:

Definition 2 (Local Tree). A local tree $\mathcal{T}_r$ rooted at node $c_r \in \mathcal{T}$ consists of node c_r and all its *direct* children nodes.

Preserving Relative Tree Distance Within Local Trees. Without a hierarchical structure, pairwise category distance is enforced by Eq. (2). With a local tree structure, the category distance in the embedding space should reflect the tree distance in a *comparative* way. Specifically, since the tree distance between two children nodes is larger than that between a children node and the parent node, a category should be closer to its parent category than to its sibling categories in the embedding space. To achieve this property, we employ the following objective for categories in a local tree $\mathcal{T}_r$:

$$\mathcal{L}_{\text{inter}} = \sum_{c_i \in \mathcal{T}_r \setminus \{c_r\}} \sum_{c_j \in \mathcal{T}_r \setminus \{c_r, c_i\}} \min(0, \boldsymbol{c}_i^\top \boldsymbol{c}_r - \boldsymbol{c}_i^\top \boldsymbol{c}_j - m_{\text{inter}}), \tag{3}$$

[2] We will discuss how to select representative terms in Section 4.

which generalizes Eq. (2) by forcing the directional similarity between a children category center vector and its parent category center vector to be higher than that between two sibling categories by m_{inter}.

Maximizing $\mathcal{L}_{\text{inter}}$ results in two favorable tree embedding properties: (1) The children categories are placed near the parent category (by requiring higher value of $\boldsymbol{c}_i^\top \boldsymbol{c}_r$), which reflects the semantic correlation between a sub-category and a super-category; (2) Any two sibling categories are well-separated (by requiring lower value of $\boldsymbol{c}_i^\top \boldsymbol{c}_j$), which encourages distinction between sibling categories (*e.g.*, "baseball" vs. "soccer").

Recursively Embed Local Trees. We apply the idea of local tree embedding recursively to embed the entire category tree structure in a top-down manner: We first embed the local tree rooted at the ROOT node, and then proceed to the next level to embed the local trees of every node at the current level. We repeat this process until we reach the leaf nodes. Figures 2(b) and 2(c) illustrate the recursive embedding procedure, which can be realized via the following holistic objective which combines the objectives of every local tree:

$$\mathcal{L}_{\text{tree}} = \sum_{c_r \in \mathcal{T}} \sum_{c_i \in \mathcal{T}_r \setminus \{c_r\}} \sum_{c_j \in \mathcal{T}_r \setminus \{c_r, c_i\}} \min(0, \boldsymbol{c}_i^\top \boldsymbol{c}_r - \boldsymbol{c}_i^\top \boldsymbol{c}_j - m_{\text{inter}}).$$

We note that m_{inter} needs to be set differently for different levels: As Figure 2(c) shows, the sibling categories are embedded in more localized regions as we proceed to the lower levels of the hierarchy to reflect their intrinsic semantic similarity. As a result, for each level L of $\mathcal{T}$, we set $m_{\text{inter}}(L)$ to be the average difference between children-parent and inter-sibling embedding similarity across level L, *i.e.*,

$$m_{\text{inter}}(L) = \frac{1}{N_L} \sum_{c_r \in L} \sum_{c_i \in \mathcal{T}_r \setminus \{c_r\}} \sum_{c_j \in \mathcal{T}_r \setminus \{c_r, c_i\}} \boldsymbol{c}_i^\top \boldsymbol{c}_r - \boldsymbol{c}_i^\top \boldsymbol{c}_j,$$

where N_L is the total number of sibling pairs within each local tree in level L. For the simplicity of notations, we omit the argument of m_{inter} in the rest of the paper, but it should be kept in mind that m_{inter} is level-dependent.

Finally, after embedding the category tree, we use the same objective as Eq. (1) to encourage intra-category coherence of retrieved terms so that the category embedding configuration can effectively guide the text embeddings to fit the tree structure.

3.3 Spherical Text Embedding via Modeling Conditional Corpus Generation

We introduce how to learn text embeddings tailored for the given category hierarchy $\mathcal{T}$ in the spherical space by modeling the corpus generation process conditioned on the categories. Specifically, we assume the corpus $\mathcal{D}$ is generated following a three-step process:

(1) First, each document $d_i \in \mathcal{D}$ is generated conditioned on one of the categories in the category hierarchy $\mathcal{T}$. Since a category can cover a wide range of semantics, it is natural to model a category as a distribution in the embedding space instead of as a single vector. Therefore, we extend the previous representation of a category c_i from a single center vector $\boldsymbol{c}_i$ to a spherical distribution centered around $\boldsymbol{c}_i$, *i.e.*, a von Mises-Fisher (vMF) distribution. Specifically, the vMF distribution of a category is parameterized by a mean vector $\boldsymbol{c}_i$ and a concentration parameter κ_{c_i}. The probability density closer to $\boldsymbol{c}_i$ is greater and the spread is controlled by κ_{c_i}. Formally, a unit random vector $\boldsymbol{x} \in \mathbb{S}^{p-1} \subset \mathbb{R}^p$ has the p-variate vMF distribution $\text{vMF}_p(\boldsymbol{x}; \boldsymbol{c}_i, \kappa_{c_i})$ if its probability density function is

$$f(\boldsymbol{x}; \boldsymbol{c}_i, \kappa_{c_i}) = n_p(\kappa_{c_i}) \exp\left(\kappa_{c_i} \cdot \cos(\boldsymbol{x}, \boldsymbol{c}_i)\right),$$

where $\|\boldsymbol{c}_i\| = 1$ is the center direction, $\kappa_{c_i} \geq 0$ is the concentration parameter, and the normalization constant $n_p(\kappa_{c_i})$ is given by

$$n_p(\kappa_{c_i}) = \frac{\kappa_{c_i}^{p/2-1}}{(2\pi)^{p/2} I_{p/2-1}(\kappa_{c_i})},$$

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

We define the generative probability of each document d_i conditioned on its corresponding true category c_i to be:

$$p(d_i \mid c_i) = \text{vMF}(\boldsymbol{d}_i; \boldsymbol{c}_i, \kappa_{c_i}) = n_p(\kappa_{c_i}) \exp\left(\kappa_{c_i} \cdot \cos(\boldsymbol{d}_i, \boldsymbol{c}_i)\right), \quad (4)$$

where $\boldsymbol{d}_i$ is the document embedding of d_i.

However, modeling category distribution via Eq. (4) is not directly helpful for our task, since our goal is to discover representative terms rather than documents for each category. For this reason, we further decompose $p(d_i \mid c_i)$ into category-word distribution:

$$p(d_i \mid c_i) \propto \prod_{w_j \in d_i} p(w_j \mid c_i) \propto \prod_{w_j \in d_i} \text{vMF}(\boldsymbol{u}_{w_j}; \boldsymbol{c}_i, \kappa_{c_i}), \quad (5)$$

where each word is assumed to be generated independently based on the document category. Eq. (5) allows direct modeling of $p(w_j \mid c_i)$, from which category representative terms will be derived.

(2) Second, each word w_j is generated based on the semantics of the document d_i. Intuitively, higher directional similarity implies higher semantic coherence, thus higher probability of co-occurrence. We assume the probability of w_j appearing in document d_i to be:

$$p(w_j \mid d_i) \propto \exp(\cos(\boldsymbol{u}_{w_j}, \boldsymbol{d}_i)). \quad (6)$$

(3) Third, surrounding words w_{j+k} in the local context window ($-h \leq k \leq h, k \neq 0$, h is the local context window size) of w_j are generated conditioned on the semantics of the center word w_j. Similar to (2), we assume the probability of w_{j+k} appearing in the local context window of w_j to be:

$$p(w_{j+k} \mid w_j) \propto \exp(\cos(\boldsymbol{v}_{w_{j+k}}, \boldsymbol{u}_{w_j})), \quad (7)$$

where $\boldsymbol{v}_w$ is the context word representation of w.

We summarize how the above three steps jointly model the text generation process by capturing both global and local textual contexts, conditioned on the given categories: Step (1) draws a connection between each document and one of the categories in $\mathcal{T}$ (*i.e., topic assignment*). Step (2) models the semantic coherence between a word and the document it appears in (*i.e., global contexts*). Step (3) models the semantic correlations of co-occurring words within a local context window (*i.e., local contexts*). We note that all three steps use directional similarity to model the correlations among categories, documents, and words.

4 OPTIMIZATION

In this section, we introduce the optimization procedure for learning embedding in the spherical space via our model defined in the previous section.

4.1 Overview

We first summarize the objectives of our optimization problem as follows (the derivation is based on maximum likelihood estimation; details can be found at Appendix B):

$$\mathcal{L} = \mathcal{L}_{\text{tree}} + \mathcal{L}_{\text{text}},$$
$$\mathcal{L}_{\text{tree}} = \sum_{c_r \in \mathcal{T}} \sum_{c_i \in \mathcal{T}_r \setminus \{c_r\}} \sum_{c_j \in \mathcal{T}_r \setminus \{c_r, c_i\}} \min(0, \boldsymbol{c}_i^\top \boldsymbol{c}_r - \boldsymbol{c}_i^\top \boldsymbol{c}_j - m_{\text{inter}}). \tag{8}$$

$$\begin{aligned}\mathcal{L}_{\text{text}} = &\sum_{d_i \in \mathcal{D}} \sum_{w_j \in d_i} \sum_{\substack{w_{j+k} \in d_i \\ -h \le k \le h, k \ne 0}} \min\Big(0, \boldsymbol{v}_{w_{j+k}}^\top \boldsymbol{u}_{w_j} + \boldsymbol{u}_{w_j}^\top \boldsymbol{d}_i \\ &- \boldsymbol{v}_{w_{j+k}}^\top \boldsymbol{u}_{w_j'} - \boldsymbol{u}_{w_j'}^\top \boldsymbol{d}_i - m\Big) \\ &+ \sum_{c_i \in \mathcal{T}} \sum_{w_j \in C_i} \Big(\log\left(n_p(\kappa_{c_i})\right) + \kappa_{c_i} \boldsymbol{u}_{w_j}^\top \boldsymbol{c}_i\Big) \mathbb{1}(\boldsymbol{u}_{w_j}^\top \boldsymbol{c}_i < m_{\text{intra}}).\end{aligned} \tag{9}$$

$$s.t. \quad \forall w, d, c, \quad \|\boldsymbol{u}_w\| = \|\boldsymbol{v}_w\| = \|\boldsymbol{d}\| = \|\boldsymbol{c}\| = 1, \kappa_c \ge 0,$$

where $\mathbb{1}(\cdot)$ is the indicator function; we set $m = 0.25$.

We note that our objective contains latent variables, *i.e.*, the second term in Eq. (9) requires knowledge about the latent category of words. At the beginning, we only know that the category name provided by the user belongs to the corresponding category (*e.g.*, $w_{\text{sports}} \in C_{\text{sports}}$). The goal of Hierarchical Topic Mining is to discover the latent category assignment of more words such that they form a clear description of the category.

To solve the optimization problem involving latent variables, we develop an EM algorithm that iterates between the estimation of the latent category assignment of words (*i.e.*, **E-Step**) and maximization of the embedding training objectives (*i.e.*, **M-Step**). We detail the design of the EM algorithm below:

E-Step. We update the estimation of words assigned to each category by

$$C_i^{(t)} \leftarrow \text{Top}_t\left(\{w\}; \boldsymbol{u}_w^{(t)}, \boldsymbol{c}_i^{(t)}, \kappa_{c_i}^{(t)}\right), \tag{10}$$

where $\text{Top}_t(\{w\}; \boldsymbol{u}_w, \boldsymbol{c}_i, \kappa_{c_i})$ denotes the set of terms ranked at the top t positions according to $\text{vMF}(\boldsymbol{u}_w; \boldsymbol{c}_i, \kappa_{c_i})$ (*i.e.*, we assign the t terms to c_i that are most likely generated from its current estimated category distribution). In practice, we find that gradually increasing t (*i.e.*, set $t = 1$ at the first iteration where $C_i^{(1)}$ contains only the category name w_{c_i} by initializing $\boldsymbol{c}_i^{(1)} = \boldsymbol{u}_{w_{c_i}}^{(1)}$; increment t by 1 for the following iterations) works well. Therefore, here t also denotes the iteration index.

Note here that we only update the estimation of category assignment for the top t words per category, which will become the representative terms retrieved. The reason is that most of the terms in the vocabulary are not representative for any of the categories; assigning them to one of the category will have negative impact on accurate estimation of the category distribution.

M-Step. We update the text embeddings and category embeddings by maximizing $\mathcal{L}_{\text{tree}}$ and $\mathcal{L}_{\text{text}}$:

$$\Theta^{(t+1)} \leftarrow \arg\max\left(\mathcal{L}_{\text{text}}\left(\Theta^{(t)}\right) + \mathcal{L}_{\text{tree}}\left(\Theta^{(t)}\right)\right), \tag{11}$$

where $\Theta^{(t)} = \left\{\boldsymbol{u}_w^{(t)}, \boldsymbol{v}_w^{(t)}, \boldsymbol{d}^{(t)}, \boldsymbol{c}^{(t)}\right\}$.

Eq. (11) requires non-Euclidean stochastic optimization methods, which will be introduced in the next subsection.

4.2 Riemannian Optimization

Embedding learning is usually based on stochastic optimization techniques, but Euclidean optimization methods like SGD cannot be directly applied to our case, because the Euclidean gradient provides update directions in a non-curvature space, while the embeddings in our model must be updated on the spherical surface $\mathbb{S}^{p-1}$ with constant positive curvature.

For the above reason, we apply the Riemannian optimization method in the spherical space as described in [25] to train text and tree embeddings. Specifically, the Riemannian gradient of a parameter $\boldsymbol{\theta}$ is computed as

$$\text{grad}\, \mathcal{L}(\boldsymbol{\theta}) := \left(I - \boldsymbol{\theta}\boldsymbol{\theta}^\top\right) \nabla \mathcal{L}(\boldsymbol{\theta}),$$

where $\nabla \mathcal{L}(\boldsymbol{\theta})$ is the Euclidean gradient of $\boldsymbol{\theta}$.

For example, the Riemannian gradient of $\boldsymbol{u}_w$ is computed as

$$\text{grad}\, \mathcal{L}(\boldsymbol{u}_{w_j}) = \left(I - \boldsymbol{u}_{w_j}\boldsymbol{u}_{w_j}^\top\right)\Bigg(\sum_{c \in \mathcal{T}} \mathbb{1}(w_j \in C)\kappa_c \boldsymbol{c} + \sum_{d_i, w_{j+k}} \mathbb{1}(\text{pos}_{d_i, w_j, w_{j+k}} - \text{neg} < m)\left(\boldsymbol{v}_{w_{j+k}} + \boldsymbol{d}_i\right)\Bigg),$$

where $\mathbb{1}(w_j \in C)$ is the indicator function of whether w_j belongs to category c; $\mathbb{1}(\text{pos}_{d_i, w_j, w_{j+k}} - \text{neg} < m)$ is the indicator function of whether the margin of the positive tuple over the negative one is achieved.

The Riemannian gradient of the other embeddings can be derived similarly. Since we aim to maximize our objective, we update the parameters following the Riemannian gradient direction:

$$\boldsymbol{\theta}^{(t+1)} \leftarrow R_{\boldsymbol{\theta}^{(t)}}\left(\alpha \cdot \text{grad}\, \mathcal{L}\left(\boldsymbol{\theta}^{(t)}\right)\right),$$

where α is the learning rate; $R_{\boldsymbol{x}}(\boldsymbol{z})$ is a first-order approximation of the exponential mapping at $\boldsymbol{x}$ which maps the updated parameters back to the sphere. We follow the definition in [25]:

$$R_{\boldsymbol{x}}(\boldsymbol{z}) := \frac{\boldsymbol{x} + \boldsymbol{z}}{\|\boldsymbol{x} + \boldsymbol{z}\|}.$$

4.3 Overall Algorithm

We summarize the overall algorithm of Hierarchical Topic Mining in Algorithm 1.

Complexity. We analyze the computation cost of our algorithm with respect to the tree size n. The tree embedding objective (Eq. (8)) loops over every local tree $\mathcal{T}_r \in \mathcal{T}$ and every pair of sibling nodes in $\mathcal{T}_r$. Since the number of local trees is upper bounded by the number of total tree nodes, the complexity is $O(nB^2)$ where B is the maximum branching factor in $\mathcal{T}$. The text embedding objective (Eq. (9)) pushes each representative term into the spherical sector centered around the category center vector, whose complexity is $O(nK)$. Overall, our algorithm scales linearly with the tree size.

5 EXPERIMENTS

In this section, we conduct empirical evaluations to demonstrate the effectiveness of our model. We also carry out case studies to

Algorithm 1: Hierarchical Topic Mining.

Input: A text corpus $\mathcal{D}$; a category tree $\mathcal{T} = \{c_i\}|_{i=1}^{n}$; number of terms K to retrieve per category .
Output: Hierarchical Topic Mining results $C_i|_{i=1}^{n}$.

$\boldsymbol{u}_w, \boldsymbol{v}_w, \boldsymbol{d}, \boldsymbol{c} \leftarrow$ random initialization on $\mathbb{S}^{p-1}$;
$t \leftarrow 1$;
$C_i^{(1)} \leftarrow w_{c_i}|_{i=1}^{n}$ ▷ initialize with category names;
while $t < K + 1$ **do**
  $t \leftarrow t + 1$;
  // Representative term retrieval;
  $C_i^{(t)}|_{i=1}^{n} \leftarrow$ Eq. (10) ▷ E-Step;
  // Embedding training;
  $\boldsymbol{u}_w, \boldsymbol{v}_w, \boldsymbol{d}, \boldsymbol{c} \leftarrow$ Eq. (11) ▷ M-Step;
for $i \leftarrow 1$ *to* n **do**
  $C_i^{(t)} \leftarrow C_i^{(t)} \setminus \{w_{c_i}\}$ ▷ exclude category names;
Return $C_i^{(t)}|_{i=1}^{n}$;

Table 1: Dataset statistics.

Corpus	# super-categories	# sub-categories	# documents
NYT	8	12	89,768
arXiv	3	29	230,105

show how the joint embedding space effectively models category tree structure and textual semantics.

5.1 Experiment Setup

Datasets. We use two datasets from different domains with ground-truth category hierarchy: (1) The New York Times annotated corpus (**NYT**) [34]; (2) arXiv paper abstracts (**arXiv**)[3]. For both datasets, we first select the major categories (with more than 1,000 documents) and then collect documents with exactly one ground truth category label. The dataset statistics can be found at Table 1.

Implementation Details and Parameters. We pre-process the corpora by discarding infrequent words that appear less than 5 times. We use AutoPhrase [35] to extract quality phrases, which are treated as single words during embedding training. For fair comparisons with baselines, we set hyperparameters as below for all methods: Embedding dimension $p = 100$; local context window size $h = 5$; number of representative terms to retrieve per category $K = 5$; learning rate α is set to be 0.025 initially with linear decay. Other parameters (if any) are set to be the default values of the corresponding algorithm.

5.2 Hierarchical Topic Mining

Compared Methods. We compare our model with the following baselines including unsupervised/seed-guided hierarchical topic models and unsupervised/seed-guided text embedding models. For baseline methods that require the number of topics n_L at each level L as input, we vary n_L in $[n_L, 2n_L, \ldots, 5n_L]$ where n_L is the actual number of categories at level L and report the best performance of the method.

- hLDA [4]: hLDA is a non-parametric hierarchical topic model. It assumes that documents are generated from the word distribution of a path of topics induced by the nested Chinese restaurant process. Since hLDA is unsupervised and cannot take given category names as supervision, we manually match the most relevant topics to the provided category hierarchy.
- hPAM [29]: hPAM generalizes the Pachinko Allocation Model [19] by sampling topic paths from the Dirichlet-multinomial distributions of internal nodes. We perform manual matching of topics as we do for hLDA.
- JoSE [25]: JoSE trains spherical text embeddings with Riemannian optimization. It outperforms Euclidean embedding models on textual similarity measurement. We retrieve the nearest-neighbor words of the category name in the spherical space as category representative words.
- Poincaré GloVe [36]: Poincaré GloVe learns hyperbolic word embeddings based on the Euclidean GloVe model. It naturally encodes the latent hierarchical word semantic correlations (*e.g.*, hypernym-hyponym). We retrieve the nearest-neighbor words of the category name in the Poincaré space as category representative words.
- Anchored CorEx [8]: CorEx discovers informative topics via total correlation maximization and can naturally model topic hierarchy via latent factor dependencies. Its anchored version 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.
- CatE [24]: CatE takes category names as input and learns discriminative text embeddings by enforcing distinctiveness among categories. We recursively run CatE on local trees since CatE assumes that the provided categories are mutually-exclusive semantically.

Table 2: Quantitative evaluation: Hierarchical Topic Mining.

Models	NYT		arXiv	
	TC	MACC	TC	MACC
hLDA	-0.0070	0.1636	-0.0124	0.1471
hPAM	0.0074	0.3091	0.0037	0.1824
JoSE	0.0140	0.6818	0.0051	0.7412
Poincaré GloVe	0.0092	0.6182	-0.0050	0.5588
Anchored CorEx	0.0117	0.3909	0.0060	0.4941
CatE	0.0149	0.9000	0.0066	0.8176
JoSH	**0.0166**	**0.9091**	**0.0074**	**0.8324**

Quantitative Evaluation. We apply two metrics on the top-K ($K = 5$ in our experiments) words/phrases retrieved under each category to evaluate all methods: Topic coherence (TC) and Mean accuracy (MACC) as defined in [24]. The accuracy metric is obtained from the averaged results given by five graduate students who independently label whether each retrieved term is highly relevant to the corresponding category. The quantitative results are reported in Table 2.

Qualitative Results. We demonstrate the qualitative results of **NYT** in Figure 3 and **arXiv** in Figure 5 in Appendix A. Words in

[3]Data crawled from https://arxiv.org/.

Table 3: Run time (in minutes) on NYT. Models are run on a machine with 20 cores of Intel(R) Xeon(R) CPU E5-2680 v2 @ 2.80 GHz.

hLDA	hPAM	JoSE	Poincaré GloVe	Anchored CorEx	CatE	JoSH
53	22	5	16	61	52	6

blue boxes are input category names; words in white boxes are retrieved representative terms of the corresponding category.

Run Time. Since topic discovery is usually performed on large-scale text corpus, algorithm efficiency is of great importance. Therefore, we report the run time of all methods in Table 3. JoSH takes only slightly longer to train than JoSE, which only learns text embeddings.

Discussions. The two unsupervised baselines (hLDA and hPAM) do not perform well. Despite running with different parameters for multiple times, they still fail to generate high quality topics similar to the ground-truth category hierarchy, showing the limitations of unsupervised approaches. For the two unsupervised embedding baselines, JoSE outperforms Poincaré GloVe by a large margin, demonstrating that the spherical space is more suitable than the hyperbolic space on capturing textual semantic correlations for category representative term retrieval. CatE has strong performance on the two datasets, but it has to be run recursively on each set of sibling nodes since it requires all the input categories to be mutually exclusive. Therefore, run time will become a potential bottleneck of applying CatE to large-scale hierarchies. JoSH not only outperforms all models on Hierarchical Topic Mining quality, but also enjoys high efficiency via efficient joint modeling of category tree structure and text corpus statistics.

5.3 Weakly-Supervised Hierarchical Text Classification

Hierarchical Topic Mining is also closely related to the task of text classification. Intuitively, having a good understanding of topics should lead to better categorization of documents. Similar to Hierarchical Topic Mining, the input to weakly-supervised hierarchical classification is also a word-described category tree. Since weakly-supervised classification [26, 27, 37] does not require training documents, it is especially favorable when manual annotation is expensive.

Compared Methods. We compare the following weakly-supervised hierarchical models on their classification performance, evaluated on the two datasets.

• WeSHClass [27]: WeSHClass leverages the provided keywords of each category to generate a set of pseudo documents for pre-training a hierarchical deep classifier, and self-trains the ensembled local classifiers on unlabeled data. It uses Word2Vec [28] as word representation.

• JoSH: Since our model makes explicit generative assumption between topics and documents (Eq. (4)), we are able to build a generative classifier by assigning the document to the category with the highest probability that it gets generated from, *i.e.*,

$$y_d = \arg\max_c \text{vMF}(\boldsymbol{d}; \boldsymbol{c}, \kappa_c),$$

where y_d is the predicted category label for document d.

Table 4: Quantitative evaluation: Weakly-supervised hierarchical classification.

Models	NYT		arXiv	
	Macro-F1	Micro-F1	Macro-F1	Micro-F1
WeSHClass	0.425	0.581	0.320	0.542
JoSH	0.429	0.600	0.367	0.610
WeSHClass + CatE	0.503	0.679	0.401	0.622
WeSHClass + JoSH	**0.582**	**0.703**	**0.412**	**0.673**

• WeSHClass + CatE [24]: It is shown in [24] that the learned discriminative text embedding can be used as input feature to benefit classification model. We replace the Word2Vec embedding used in WeSHClass with CatE embeddings.

• WeSHClass + JoSH: We replace the Word2Vec embedding used in WeSHClass with word embeddings learned by JoSH. Since JoSH effectively leverages the category tree structure to guide text embedding configuration, it is expected to benefit hierarchical classification model as input features.

Quantitative Evaluation. We use two metrics for classification evaluation, Macro-F1 and Micro-F1, which are commonly used in multi-class classification evaluations. The results are reported in Table 4.

Discussions. We demonstrate two potential usage of JoSH in weakly-supervised hierarchical text classification: (1) Directly build a generative classifier based on the model assumption; (2) Use the learned embedding as input features to existing classification models. JoSH alone as a generative classifier even outperforms the WeSHClass model; when used as features to WeSHClass, JoSH significantly boosts the classification performance, proved to be more effective than CatE which does not model the category hierarchy.

5.4 Joint Embedding Space Visualization

To understand how categories and words are distributed in the joint embedding space and how the category tree structure is modeled, we apply t-SNE [21] to visualize the embedding space in Figure 4. Representative terms surround their category centers; sub-categories surround their super-categories which form a category tree structure. An interesting observation is that some sub-categories under different super-categories are embedded closer, *e.g.*, in Figure 4(b), "optimization" under "math" and "algorithm" under "computer science". Indeed, these two sub-categories are somewhat cross-domain—"optimization" and "algorithm" are relevant to both mathematics and computer science. This shows that JoSH not only models the given category tree structure, but also captures semantic correlation among categories via jointly training tree and text embedding.

6 RELATED WORK

6.1 Hierarchical Topic Modeling

Hierarchical topic models extend their flat counterparts by capturing the correlations among topics and generate topic hierarchies. hLDA [4] generalizes LDA [6] with a non-parametric probabilistic model, the nested Chinese restaurant process, which induces a path from the root topic to a leaf topic. The documents are assumed to be generated by sampling words from the topics along this path.

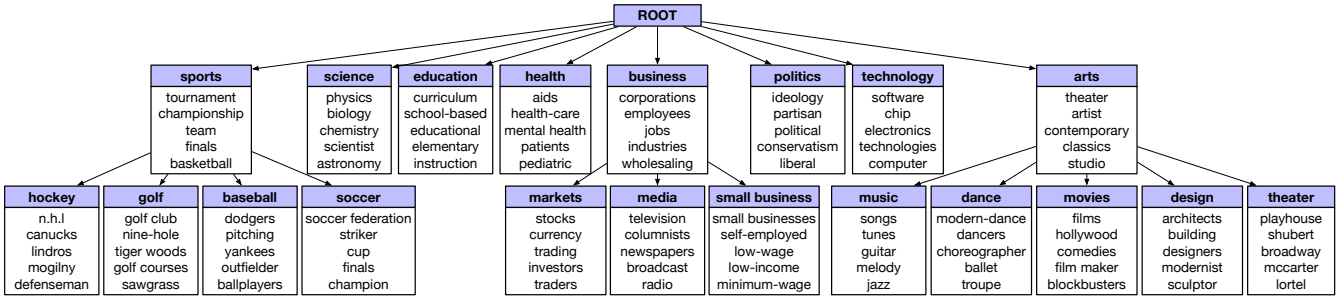
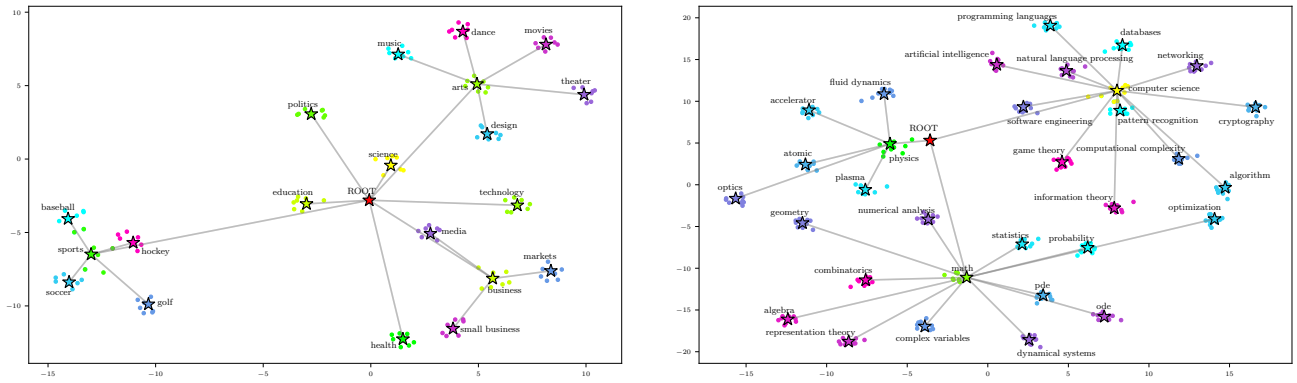


Figure 3: Hierarchical Topic Mining results on NYT.



(a) **NYT** joint embedding space.

(b) **arXiv** joint embedding space.

Figure 4: Joint embedding space visualization. Category center vectors are denoted as stars; representative words are denoted as dots in the same color with corresponding category.

Another famous hierarchical topic model, hPAM [29] is built on the Pachinko Allocation Model [19] which models documents as a mixture of distributions over a set of topics; the co-occurrences of topics are further represented via a directed acyclic graph. hPAM represents the topic hierarchical structure through the Dirichlet-multinomial parameters of the internal node distributions. There are also supervised hierarchical topic models. HSLDA [33] extends sLDA [5] by incorporating a breadth first traversal in the label space during document generation. SSHLDA [22] is a semi-supervised hierarchical topic model that not only explores new latent topics in the label space, but also makes use of the information from the hierarchy of observed labels. A seed-guided topic modeling framework, CorEx [8], learns informative topics that maximize total correlation. It is similar to our setting as it incorporates seed words by preserving seed relevant information. CorEx is able to generate topic hierarchy via latent factor dependencies. Different from the previous unsupervised and supervised topic models, our framework takes as guidance only a category hierarchy described by category names, and models category-word semantic correlation via joint spherical text and tree embedding.

6.2 Text Embedding and Tree Embedding

Text embeddings [17, 23, 25, 28, 32] effectively capture textual semantic similarity via distributed representation learning of words, phrases, sentences, etc. Several topic modeling frameworks, such as [3, 7, 20] leverage text embeddings to model contextualized semantic similarity of words, making up the bag-of-words generative assumption in classical topic models. Poincaré GloVe [36] adapts the original GloVe model by training word embedding in the Poincaré space where the latent hierarchical semantic relations between words are naturally captured. A recent text embedding model CatE [24] proposes to learn discriminative text embeddings for category representative term retrieval given a set of category names as user guidance, which is similar to our setting. CatE makes mutual exclusive assumption on category semantics, which does not hold when categories exhibit a hierarchical structure. None of the previous text embedding framework is able to model a given hierarchical category structure in the embedding space to guide text embedding learning.

With the recent advances in hyperbolic embedding space, several frameworks have been developed to model tree structures. Poincaré embedding [30] learns to model hierarchical structure in the Poincaré ball. Since the embedding distance directly corresponds to tree distance, Poincaré embedding can be used to infer lexical entailment relationship by embedding the tree structure of WordNet or perform link prediction by embedding networks. Later, Lorentz model [31] brings a more principled optimization approach in the hyperbolic space to learn tree structures; hyperbolic cones [9] are proposed to model hierarchical relations and

admit an optimal shape with a closed form expression. These hyperbolic tree embedding methods, however, are not suitable for embedding category trees in a joint space with words. The reason is that hyperbolic embeddings preserve the *absolute* tree distance, *i.e.*, similar embedding distances imply similar tree distances. In a category tree, lower-level sibling categories are generally more semantically similar than higher-level ones despite the same tree distance. Therefore, category embedding distances should not be solely determined by tree distances. In our model, text and category tree are jointly embedded, allowing the tree structure to better reflect the textual semantics of the categories.

7 CONCLUSIONS AND FUTURE WORK

In this paper, we propose a new task for hierarchical topic discovery guided by a user-provided category tree described with category names only. To effectively model the category tree structure while capturing text corpus statistics, we propose a joint spherical space embedding model JoSH that uses directional similarity to characterize semantic correlations among words, documents, and categories. We develop an EM algorithm based on Riemannian optimization for training the model in the spherical space. JoSH mines high-quality topics and enjoys high efficiency. We also show that JoSH can be applied to the task of weakly-supervised hierarchical classification, serving as either a generative classifier on its own, or input features to existing classification models.

In the future, we aim to extend JoSH to not only focus on a user-given category structure, but also be able to discover other latent topics from a text corpus, probably by relaxing the assumption that a document is generated from one of the given topics or collaborating with other taxonomy construction algorithms [12, 13]. Also, the promising results of our joint spherical space embedding model may shed light on future studies of embedding tree or graph structures along with textual data in the spherical space for mining structured knowledge from text corpora.

ACKNOWLEDGMENTS

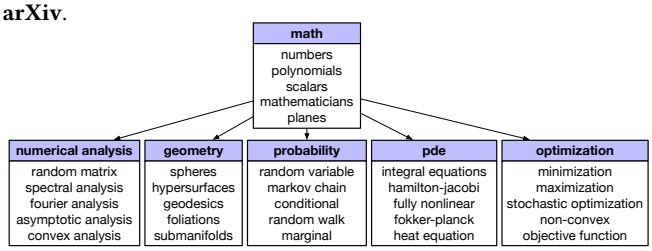
Research was sponsored in part by US DARPA KAIROS Program No. FA8750-19-2-1004 and SocialSim Program No. W911NF-17-C-0099, National Science Foundation IIS 16-18481, IIS 17-04532, and IIS 17-41317, and DTRA HDTRA11810026. Any opinions, findings, and conclusions or recommendations expressed herein are those of the authors and should not be interpreted as necessarily representing the views, either expressed or implied, of DARPA or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for government purposes notwithstanding any copyright annotation hereon. We thank anonymous reviewers for valuable and insightful feedback.

REFERENCES

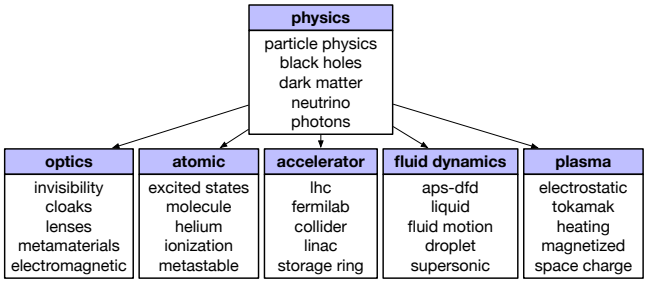
[1] Enrique Alfonseca, Katja Filippova, Jean-Yves Delort, and Guillermo Garrido. 2012. Pattern Learning for Relation Extraction with a Hierarchical Topic Model. In *ACL*.
[2] David Andrzejewski and Xiaojin Zhu. 2009. Latent Dirichlet Allocation with Topic-in-Set Knowledge. In *HLT-NAACL*.
[3] Kayhan Batmanghelich, Ardavan Saeedi, Karthik Narasimhan, and Sam Gershman. 2016. Nonparametric spherical topic modeling with word embeddings. In *ACL*. 537.
[4] David M. Blei, Thomas L. Griffiths, Michael I. Jordan, and Joshua B. Tenenbaum. 2003. Hierarchical Topic Models and the Nested Chinese Restaurant Process. In *NIPS*.
[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] Adji B. Dieng, Francisco J. R. Ruiz, and David M. Blei. 2019. Topic Modeling in Embedding Spaces. *ArXiv* abs/1907.04907 (2019).
[8] Ryan J. Gallagher, Kyle Reing, David C. Kale, and Greg Ver Steeg. 2017. Anchored Correlation Explanation: Topic Modeling with Minimal Domain Knowledge. *TACL* (2017).
[9] Octavian-Eugen Ganea, Gary Bécigneul, and Thomas Hofmann. 2018. Hyperbolic Entailment Cones for Learning Hierarchical Embeddings. In *ICML*.
[10] Justin Grimmer. 2010. A Bayesian hierarchical topic model for political texts: Measuring expressed agendas in Senate press releases. *Political Analysis* 18, 1 (2010), 1–35.
[11] Thomas Hofmann. 1999. Probabilistic Latent Semantic Indexing. In *SIGIR*.
[12] 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*.
[13] Jiaxin Huang, Yiqing Xie, Yu Meng, Yunyi Zhang, and Jiawei Han. 2020. CoRel: Seed-Guided Topical Taxonomy Construction by Concept Learning and Relation Transferring. In *KDD*.
[14] Jagadeesh Jagarlamudi, Hal Daumé, and Raghavendra Udupa. 2012. Incorporating Lexical Priors into Topic Models. In *EACL*.
[15] Saurabh S Kataria, Krishnan S Kumar, Rajeev R Rastogi, Prithviraj Sen, and Srinivasan H Sengamedu. 2011. Entity disambiguation with hierarchical topic models. In *KDD*.
[16] Sachin Kumar and Yulia Tsvetkov. 2019. Von Mises-Fisher Loss for Training Sequence to Sequence Models with Continuous Outputs. In *ICLR*.
[17] Quoc V. Le and Tomas Mikolov. 2014. Distributed Representations of Sentences and Documents. In *ICML*.
[18] Omer Levy and Yoav Goldberg. 2014. Linguistic Regularities in Sparse and Explicit Word Representations. In *CoNLL*.
[19] Wei Li and Andrew McCallum. 2006. Pachinko allocation: DAG-structured mixture models of topic correlations. In *ICML*. 577–584.
[20] Yang Liu, Zhiyuan Liu, Tat-Seng Chua, and Maosong Sun. 2015. Topical Word Embeddings. In *AAAI*.
[21] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, Nov (2008), 2579–2605.
[22] Xianling Mao, Zhaoyan Ming, Tat-Seng Chua, Si Kan Li, Hongfei Yan, and Xiaoming Li. 2012. SSHLDA: A Semi-Supervised Hierarchical Topic Model. In *EMNLP-CoNLL*.
[23] Yu Meng, Jiaxin Huang, Guangyuan Wang, Zihan Wang, Chao Zhang, and Jiawei Han. 2020. Unsupervised Word Embedding Learning by Incorporating Local and Global Contexts. *Frontiers in Big Data* (2020).
[24] 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*.
[25] Yu Meng, Jiaxin Huang, Guangyuan Wang, Chao Zhang, Honglei Zhuang, Lance Kaplan, and Jiawei Han. 2019. Spherical Text Embedding. In *NeurIPS*.
[26] Yu Meng, Jiaming Shen, Chao Zhang, and Jiawei Han. 2018. Weakly-Supervised Neural Text Classification. In *CIKM*.
[27] Yu Meng, Jiaming Shen, Chao Zhang, and Jiawei Han. 2019. Weakly-Supervised Hierarchical Text Classification. In *AAAI*.
[28] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *NIPS*.
[29] David M. Mimno, Wei Li, and Andrew McCallum. 2007. Mixtures of hierarchical topics with Pachinko allocation. In *ICML '07*.
[30] Maximilian Nickel and Douwe Kiela. 2017. Poincaré Embeddings for Learning Hierarchical Representations. In *NIPS*.
[31] Maximilian Nickel and Douwe Kiela. 2018. Learning Continuous Hierarchies in the Lorentz Model of Hyperbolic Geometry. In *ICML*.
[32] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global Vectors for Word Representation. In *EMNLP*.
[33] Adler J. Perotte, Frank D. Wood, Noémie Elhadad, and Nicholas Bartlett. 2011. Hierarchically Supervised Latent Dirichlet Allocation. In *NIPS*.
[34] Evan Sandhaus. 2008. The New York Times Annotated Corpus.
[35] 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.
[36] Alexandru Tifrea, Gary Bécigneul, and Octavian-Eugen Ganea. 2019. Poincaré Glove: Hyperbolic Word Embeddings. In *ICLR*.
[37] 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*.

A HIERARCHICAL TOPIC MINING RESULTS ON ARXIV

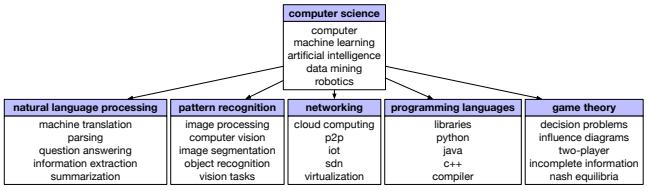
Figure 5 shows part of the Hierarchical Topic Mining results on **arXiv**.



(a) "Math" subtree.



(b) "Physics" subtree.



(c) "Computer Science" subtree.

Figure 5: Results of Hierarchical Topic Mining on arXiv: Only 5 sub-categories per super-category are shown here.

B DERIVATION OF OBJECTIVE

The derivation of Eqs. (8) and (9) is provided as follows.

The conditional likelihood of the corpus given the category hierarchy is obtained by combining the assumptions described in Eqs. (5), (6) and (7):

$$\begin{aligned} P(\mathcal{D} \mid \mathcal{T}) &= \prod_{d_i \in \mathcal{D}} p(d_i \mid c_i) \prod_{w_j \in d_i} p(w_j \mid d_i) \prod_{\substack{w_{j+k} \in d \\ -h \le k \le h, k \neq 0}} p(w_{j+k} \mid w_j) \\ &\propto \prod_{d_i \in \mathcal{D}} \prod_{w_j \in d_i} p(w_j \mid c_i) p(w_j \mid d_i) \prod_{\substack{w_{j+k} \in d_i \\ -h \le k \le h, k \neq 0}} p(w_{j+k} \mid w_j), \end{aligned} \tag{12}$$

where c_i is the latent true category of d_i.

To make the learning of text embedding and category distribution explicit, we re-write Eq. (12) by re-arranging the product of $p(w \mid c)$ over categories:

$$\begin{aligned} P(\mathcal{D} \mid \mathcal{T}) \propto & \prod_{c_i \in \mathcal{T}} \prod_{w_j \in c_i} p(w_j \mid c_i) \\ & \cdot \prod_{d_i \in \mathcal{D}} \prod_{w_j \in d_i} p(w_j \mid d_i) \prod_{\substack{w_{j+k} \in d_i \\ -h \le k \le h, k \neq 0}} p(w_{j+k} \mid w_j), \end{aligned}$$

Taking the log-likelihood as our objective to maximize, we have

$$\begin{aligned} \mathcal{L} = & \sum_{c_i \in \mathcal{T}} \sum_{w_j \in c_i} \log p(w_j \mid c_i) \\ & + \sum_{d_i \in \mathcal{D}} \sum_{w_j \in d_i} \log p(w_j \mid d_i) \\ & + \sum_{d_i \in \mathcal{D}} \sum_{w_j \in d_i} \sum_{\substack{w_{j+k} \in d_i \\ -h \le k \le h, k \neq 0}} \log p(w_{j+k} \mid w_j) \\ & + \text{constant}. \end{aligned} \tag{13}$$

We omit the constant term and split Eq. (13) into category distribution modeling and corpus-based embedding learning objectives, plugging in the definition of the probability expressions given by Eqs. (5), (6) and (7). For category distribution modeling, we have:

$$\begin{aligned} \mathcal{L}_{\text{cat}} &= \sum_{c_i \in \mathcal{T}} \sum_{w_j \in c_i} \log p(w_j \mid c_i) \\ &= \sum_{c_i \in \mathcal{T}} \sum_{w_j \in c_i} \log \left(n_p(\kappa_{c_i}) \right) + \kappa_{c_i} \cdot \cos(\boldsymbol{u}_{w_j}, \boldsymbol{c}_i). \end{aligned} \tag{14}$$

Eq. (14) achieves the same effect as Eq. (1) on encouraging word representative terms to have high directional similarity with the category center vector, except that Eq. (14) does not incorporate an intra-category margin. Thus we extend Eq. (14) into the following:

$$\mathcal{L}^*_{\text{cat}} = \sum_{c_i \in \mathcal{T}} \sum_{w_j \in C_i} \left(\log \left(n_p(\kappa_{c_i}) \right) + \kappa_{c_i} \boldsymbol{u}_{w_j}^\top \boldsymbol{c}_i \right) \mathbb{1}(\boldsymbol{u}_{w_j}^\top \boldsymbol{c}_i < m_{\text{intra}}), \tag{15}$$

where $\mathbb{1}(\cdot)$ is the indicator function.

For corpus-based embedding learning, we have:

$$\begin{aligned} \mathcal{L}_{\text{corpus}} &= \sum_{d_i \in \mathcal{D}} \sum_{w_j \in d_i} \left(\log p(w_j \mid d_i) + \sum_{\substack{w_{j+k} \in d_i \\ -h \le k \le h, k \neq 0}} \log p(w_{j+k} \mid w_j) \right) \\ &= \sum_{d_i \in \mathcal{D}} \sum_{w_j \in d_i} \left(\cos(\boldsymbol{u}_{w_j}, \boldsymbol{d}_i) + \sum_{\substack{w_{j+k} \in d_i \\ -h \le k \le h, k \neq 0}} \cos(\boldsymbol{v}_{w_{j+k}}, \boldsymbol{u}_{w_j}) \right). \end{aligned}$$

Directly maximizing the above objective results in trivial solution that all text embedding vectors are converged to the same point (so that the cosine similarity term is always maximized). To tackle this issue, we employ the same technique used in [25] where the log-likelihood of a positive co-occurring tuple (w_j, w_{j+k}, d_i) is pushed over that of a negative tuple (w'_j, w_{j+k}, d_i) by a margin m, where w'_j is a randomly sampled word from the vocabulary.

$$\begin{aligned} \mathcal{L}_{\text{corpus}} = \sum_{d_i \in \mathcal{D}} \sum_{w_j \in d_i} \sum_{\substack{w_{j+k} \in d_i \\ -h \le k \le h, k \neq 0}} \min \Big(0, -m + \cos(\boldsymbol{v}_{w_{j+k}}, \boldsymbol{u}_{w_j}) + \\ \cos(\boldsymbol{u}_{w_j}, \boldsymbol{d}_i)) - \cos(\boldsymbol{v}_{w_{j+k}}, \boldsymbol{u}_{w'_j}) - \cos(\boldsymbol{u}_{w'_j}, \boldsymbol{d}_i) \Big). \end{aligned}$$

Finally, $\mathcal{L}_{\text{text}} = \mathcal{L}^*_{\text{cat}} + \mathcal{L}_{\text{corpus}}$.