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]University of Illinois at Urbana-Champaign
[2]Georgia Institute of Technology

Outline

- Motivation & Introduction
- Spherical Text and Tree Embedding
- Optimization
- Experiments
- Conclusions

Motivation

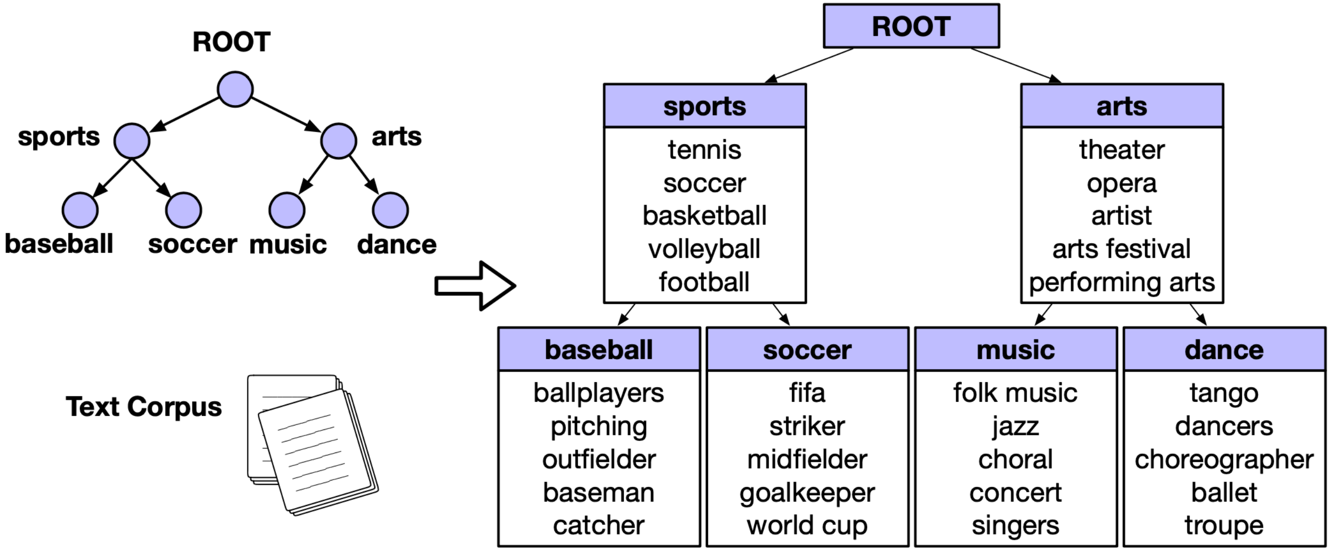
- Mining a set of meaningful topics organized into a **hierarchy** is intuitively appealing and has broad applications
  - Coarse-to-fine topic understanding
  - Hierarchical corpus summarization
  - Hierarchical text classification
  - …
- Hierarchical topic models discover topic structures from text corpora via modeling the text generative process with a latent hierarchy

Motivation

- What are the limitations of (hierarchical) topic models?
- **Failure to incorporate user guidance:** (Hierarchical) topic models tend to retrieve the most general and prominent topics from a text collection
  - may not be of a user's particular interest
  - provide a superficial summarization of the corpus
- **Lack of stability and consistency:** The inference algorithm of (hierarchical) topic models yield local optimum solutions
  - different hierarchy & topic results across different runs
  - this issue even worsens when a larger number of topics and their correlation need to be modeled

Introduction

- **A New Task: Hierarchical Topic Mining**
  - Given a text corpus and a **tree-structured hierarchy** described by **category names**, hierarchical topic mining aims to retrieve a set of terms that provide a clear description of each category
  - e.g. 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



Introduction

- **A New Task: Hierarchical Topic Mining**
  - Connection and difference between hierarchical topic modeling
    - both account for the hierarchical correlations among topics (e.g. "sports" is a super-topic of "soccer")
    - hierarchical topic modeling is purely unsupervised; hierarchical topic mining is weakly-supervised (requires the names of the hierarchy categories)

Outline

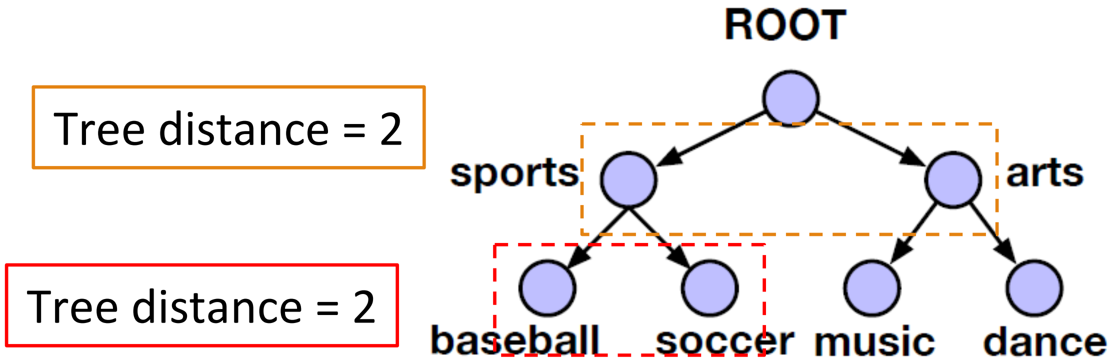
- Motivation & Introduction
- Spherical Text and Tree Embedding
- Optimization
- Experiments
- Conclusions

JoSH Embedding

- Motivation:
  - Directional similarity of text embeddings has proven most effective on estimation of semantic similarity (e.g. cosine similarity between embeddings)
  - Spherical space is a natural choice for modeling directional similarity
- Jointly learn text and tree embedding in the spherical space for hierarchical topic mining (JoSH)
- Difference from hyperbolic models (e.g. Poincare, Lorentz)
  - hyperbolic embeddings preserve absolute tree distance (similar embedding distance => similar tree distance)
  - we do not aim to preserve the absolute tree distance, but rather use it as a relative measure

JoSH Embedding

- (cont'd) Difference from hyperbolic models (e.g. Poincare, Lorentz)
  - hyperbolic embeddings preserve absolute tree distance (similar embedding distance => similar tree distance)
  - we do not aim to preserve the absolute tree distance, but rather use it as a relative measure



Although $d_{\text{tree}}(\text{sports}, \text{arts}) = d_{\text{tree}}(\text{baseball}, \text{soccer})$, "baseball" and "soccer" should be embedded closer than "sports" and "arts" to reflect semantic similarity.

Use tree distance in a relative manner: Since $d_{\text{tree}}(\text{sports}, \text{baseball}) < d_{\text{tree}}(\text{baseball}, \text{soccer})$, "baseball" and "sports" should be embedded closer than "baseball" and "soccer".

JoSH Tree Embedding

- **Intra-Category Coherence**: Representative terms of each category should be highly semantically relevant to each other, reflected by high directional similarity in the spherical space

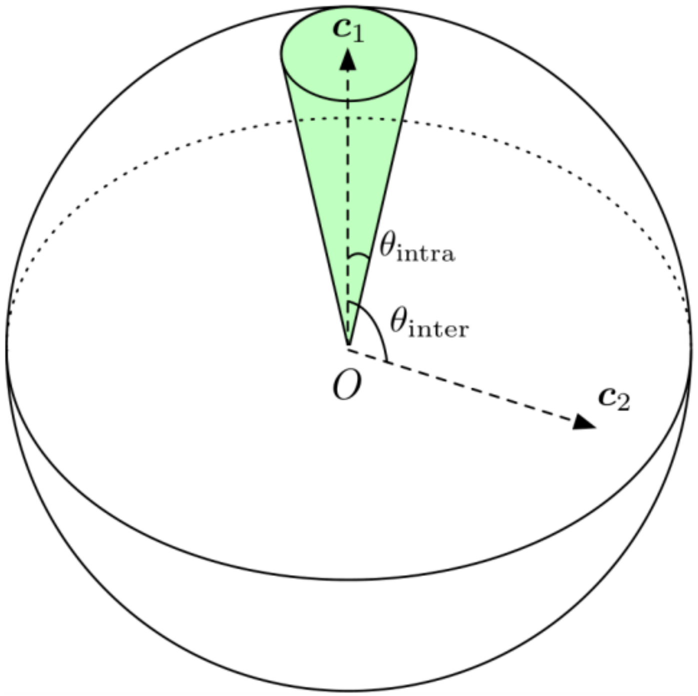
$$\mathcal{L}_{\text{intra}} = \sum_{c_i \in \mathcal{T}} \sum_{w_j \in C_i} \min(0, \boldsymbol{u}_{w_j}^\top \boldsymbol{c}_i - m_{\text{intra}}),$$

- **Inter-Category Distinctiveness**: Encourage distinctiveness across different categories to avoid semantic overlaps so that the retrieved terms provide a clear and distinctive description

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

$$\theta_{\text{intra}} \leq \arccos(m_{\text{intra}})$$

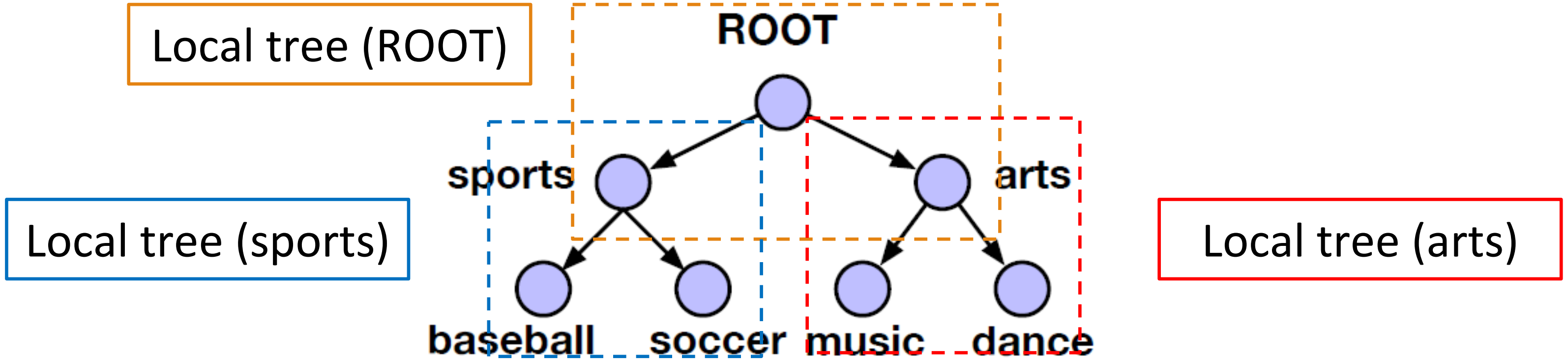
$$\theta_{\text{inter}} \geq \arccos(1 - m_{\text{inter}})$$



(a) Intra- & Inter-Category Configuration.

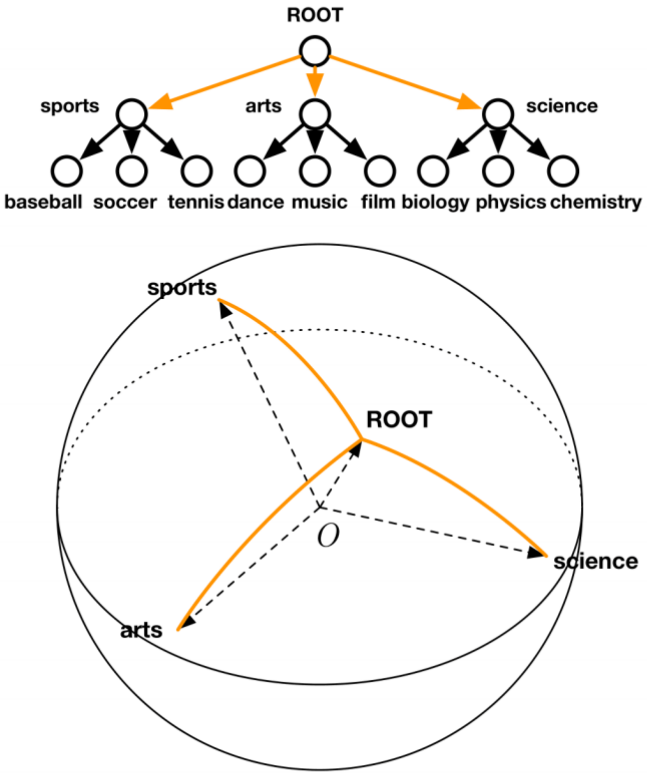
JoSH Tree Embedding

- **Recursive Local Tree Embedding:** Recursively embed local structures of the category tree onto the sphere
- Local tree: A local tree T_r rooted at node $c_r \in T$ consists of node c_r and all its direct children nodes

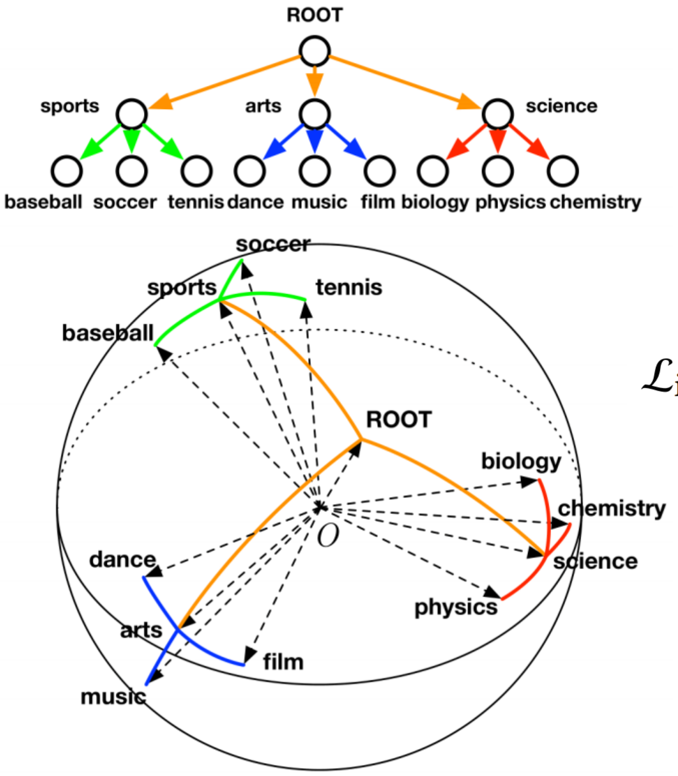


JoSH Tree Embedding

- **Preserving Relative Tree Distance Within Local Trees**: A category should be closer to its parent category than to its sibling categories in the embedding space



(b) Embed First-Level Local Tree.



(c) Embed Second-Level Local Trees.

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

JoSH Text Embedding

- Modeling Text Generation Conditioned on the Category Tree
- A three-step process:
  - 1. a document d_i is generated conditioned on one of the n categories

1. Topic assignment

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

  - 2. each word w_j is generated conditioned on the semantics of the document d_i

2. Global context

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

  - 3. surrounding words w_{j+k} in the local context window of w_i are generated conditioned on the semantics of the center word w_i

3. Local context

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

Outline

- Motivation & Introduction
- Spherical Text and Tree Embedding
- Optimization
- Experiments
- Conclusions

Optimization

- Overall objective: Maximum likelihood estimation

$$\mathcal{L} = \mathcal{L}_{\text{tree}} + \mathcal{L}_{\text{text}},$$

$$\mathcal{L}_{\text{tree}} = \sum_{c_r \in \mathcal{T}} \sum_{c_i \in \mathcal{T}_r} \sum_{c_j \in \mathcal{T}_r \setminus \{c_r, c_i\}} \min(0, \mathbf{c}_i^\top \mathbf{c}_r - \mathbf{c}_i^\top \mathbf{c}_j - m_{\text{inter}}). \quad (9)$$

$$\mathcal{L}_{\text{text}} = \sum_{d_i \in \mathcal{D}} \sum_{w_j \in d_i} \sum_{\substack{w_{j+k} \in d_i \\ -h \le k \le h, k \ne 0}} \min\left(0, \boldsymbol{v}_{w_{j+k}}^\top \boldsymbol{u}_{w_j} + \boldsymbol{u}_{w_j}^\top \boldsymbol{d}_i - \boldsymbol{v}_{w_{j+k}}^\top \boldsymbol{u}_{w_j'} - \boldsymbol{u}_{w_j'}^\top \boldsymbol{d}_i - m\right)$$

$$+ \sum_{c_i \in \mathcal{T}} \sum_{w_j \in C_i} \left(\log\left(n_p(\kappa_{c_i})\right) + \kappa_{c_i} \boldsymbol{u}_{w_j}^\top \boldsymbol{c}_i\right) \mathbb{1}(\boldsymbol{u}_{w_j}^\top \boldsymbol{c}_i < m_{\text{intra}}). \quad (10)$$

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

Spherical Space Constraint

Optimization

- An EM-based algorithm for optimizing the objectives
  - Our objective contains latent variables, i.e., the latent category of words
  - At first, we only know that the category name provided by the user belongs to the corresponding category (e.g. the word “sports” belongs to the category “sports”)
  - 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)

Optimization

- An EM-based algorithm for optimizing the objectives
  - E-Step: Update the estimation of words assigned to each category

$$C_i^{(t)} \leftarrow \mathrm{Top}_t(\{w\}; \boldsymbol{u}_w^{(t)}, \boldsymbol{c}_i^{(t)}, \kappa_{c_i}^{(t)}),$$

the set of terms ranked at the top t positions via $\mathrm{vMF}(\boldsymbol{u}_w; \boldsymbol{c}_i, \kappa_{c_i})$

  - M-Step:

$$\Theta^{(t+1)} \leftarrow \arg\max \left(\mathcal{L}_{\text{text}}\left(\Theta^{(t)}\right) + \mathcal{L}_{\text{tree}}\left(\Theta^{(t)}\right)\right), \quad \Theta^{(t)} = \left\{\boldsymbol{u}_w^{(t)}, \boldsymbol{v}_w^{(t)}, \boldsymbol{d}^{(t)}, \boldsymbol{c}^{(t)}\right\}$$

Require stochastic optimization technique

Optimization

- Riemannian optimization
  - 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
  - Riemannian optimization method in the spherical space:

Riemannian gradient $\boxed{\operatorname{grad} \mathcal{L}(\boldsymbol{\theta})} := \left(I - \boldsymbol{\theta}\boldsymbol{\theta}^\top\right) \boxed{\nabla \mathcal{L}(\boldsymbol{\theta}),}$ Euclidean gradient

  - Riemannian stochastic optimization

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

$$R_{\boldsymbol{x}}(\boldsymbol{z}) := \frac{\boldsymbol{x} + \boldsymbol{z}}{\|\boldsymbol{x} + \boldsymbol{z}\|}$$ --- approximated exponential mapping

Optimization

- Overall algorithm
- Complexity w.r.t. tree size n:
  - $O(nB^2)$ for tree embedding
  - $O(nK)$ for text embedding
- Scales linearly w.r.t tree size

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 $C_i^{(t)}|_{i=1}^{n}$;

Outline

- Motivation & Introduction
- Spherical Text and Tree Embedding
- Optimization
- Experiments
- Conclusions

Experiments: Datasets

- Datasets
  - New York Times annotated corpus (Sandhaus, 2008)
  - ArXiv paper abstracts

Table 1: Dataset statistics.

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

Experiments: Hierarchical Topic Mining

- Baselines
  - hLDA (NIPS 2003) Manual select
  - hPAM (ICML 2007) Manual select
  - JoSE (NeurIPS 2019) Spherical embedding
  - Poincare GloVe (ICLR 2019) Hyperbolic embedding
  - Anchored CorEx (TACL 2017) Seed guided
  - CatE (WWW 2020) Seed guided + embedding
- Metrics:
  - Averaged topic coherence: how coherent the mined topics are
  - Mean accuracy: how accurately the retrieved terms belong to the category

Experiments: Hierarchical Topic Mining

- Quantitative results

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**

Experiments: Hierarchical Topic Mining

- Qualitative results

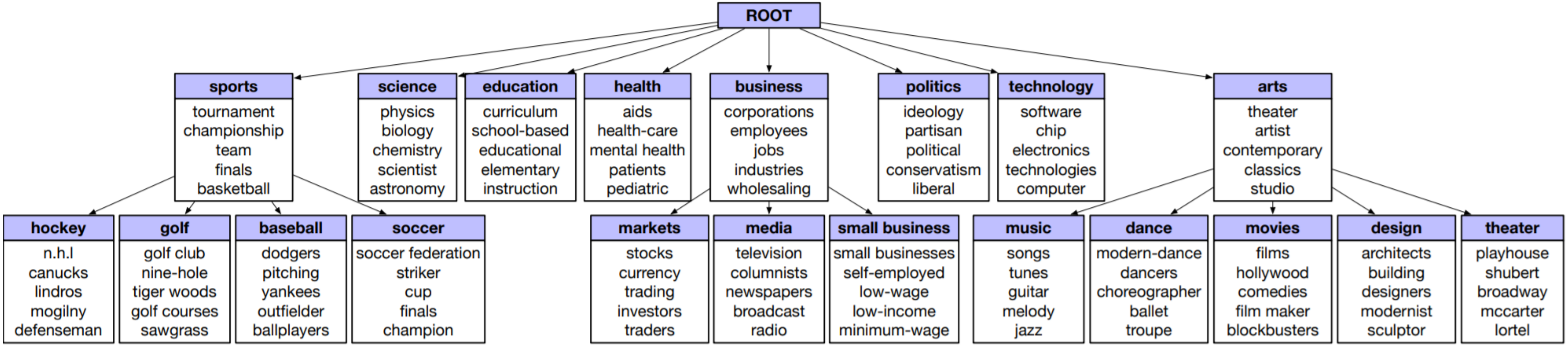
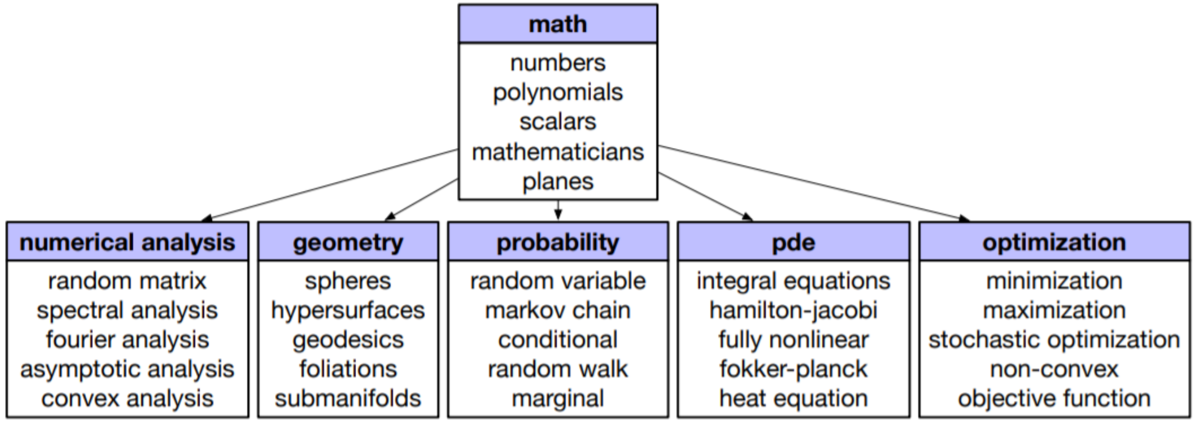


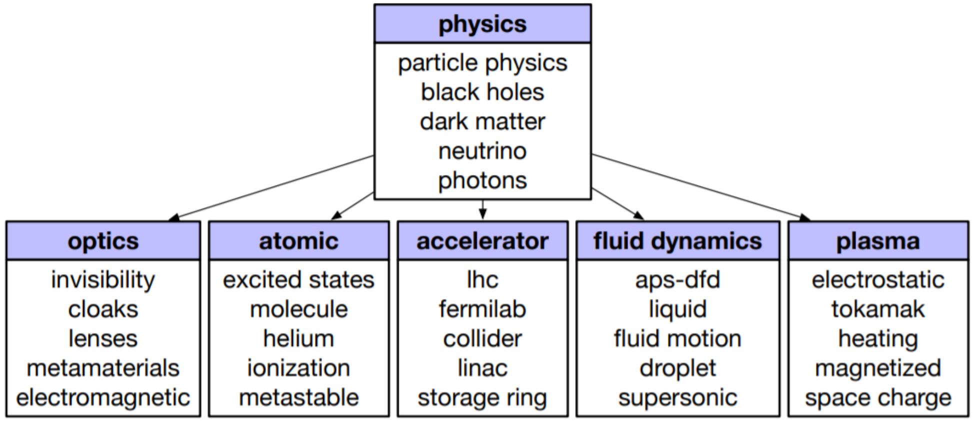
Figure 3: Hierarchical Topic Mining results on NYT.

Experiments: Hierarchical Topic Mining

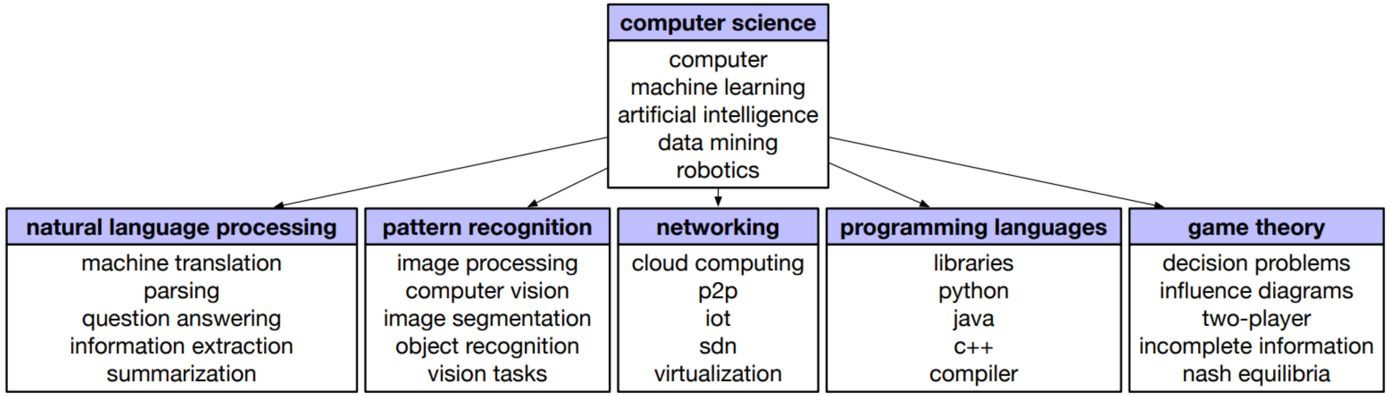
- Qualitative results



(a) "Math" subtree.



(b) "Physics" subtree.



(c) "Computer Science" subtree.

Experiments: Hierarchical Topic Mining

- Run Time
  - JoSH enjoys high efficiency
  - CatE needs to be run recursively on each set of sibling nodes since it requires all the input categories to be mutually semantically exclusive

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

Experiments: Weakly-Supervised Hierarchical Text Classification

- JoSH can be either directly used as a generative classifier, i.e., $y_d = \arg\max_c \text{vMF}(\boldsymbol{d}; \boldsymbol{c}, \kappa_c)$, or its text embedding can be used as features to existing classifiers (e.g. WeSHClass)
- Compared methods:
  - WeSHClass (AAAI 2019)
  - JoSH
  - WeSHClass + CatE (WWW 2020)
  - WeSHClass + JoSH
- Metrics: Micro-F1 and Macro-F1

Experiments: Weakly-Supervised Hierarchical Text Classification

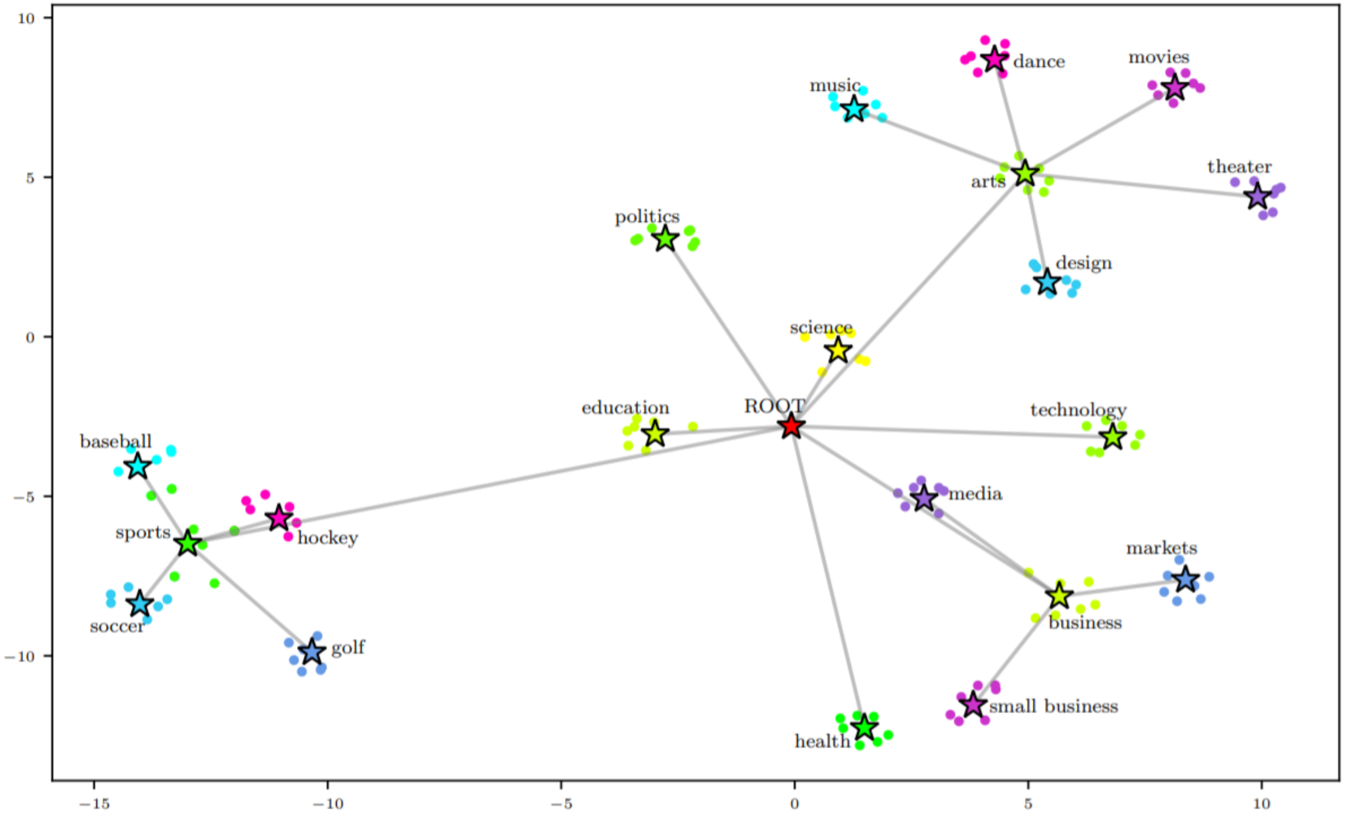
- Quantitative evaluation:

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**

Experiments: Joint Embedding Space Visualization

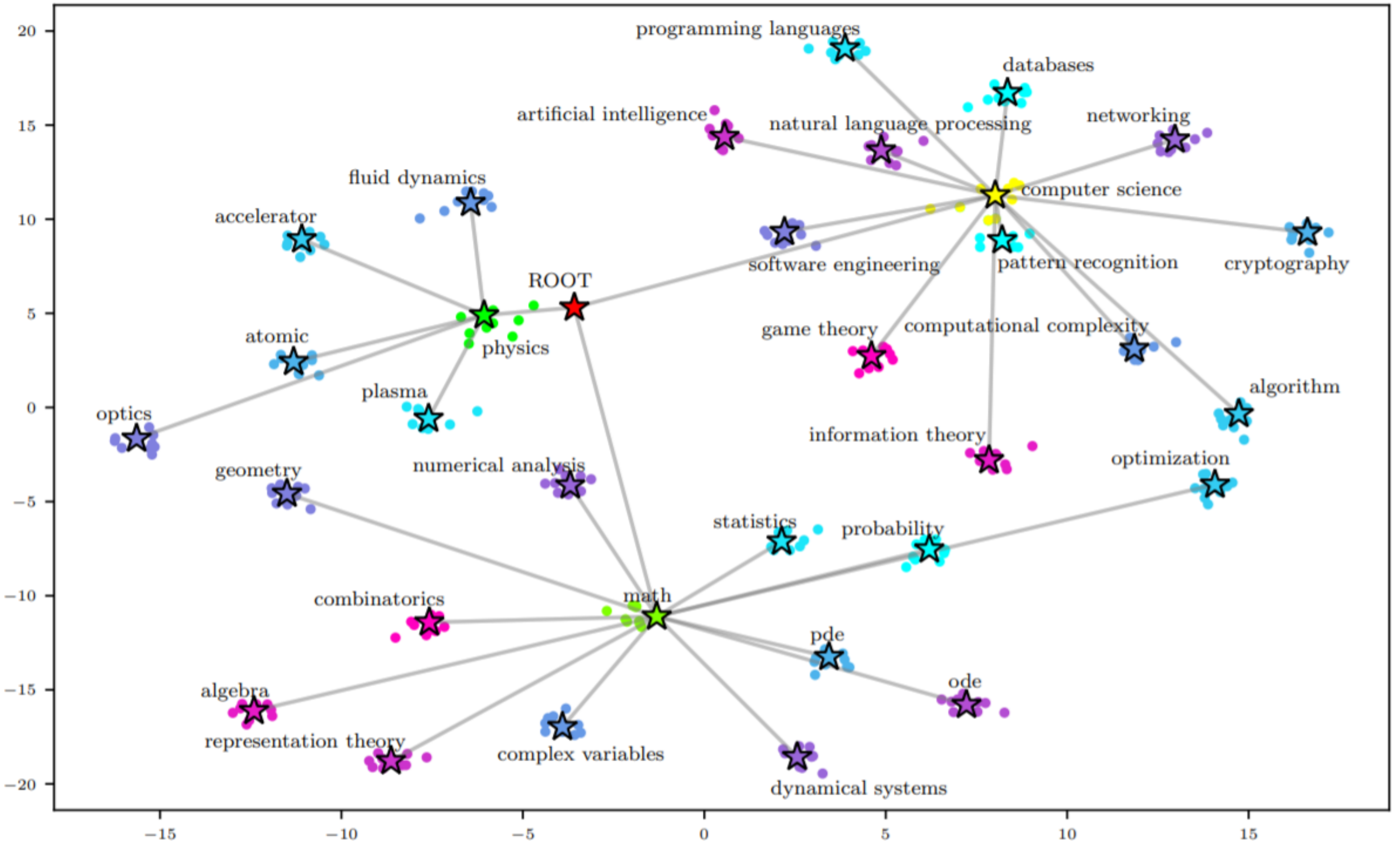
- T-SNE visualization (stars=category embeddings; dots=representative word embeddings)



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

Experiments: Joint Embedding Space Visualization

- T-SNE visualization (stars=category embeddings; dots=representative word embeddings)



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

Outline

- ❑ Motivation & Introduction
- ❑ Spherical Text and Tree Embedding
- ❑ Optimization
- ❑ Experiments
- ❑ Conclusions

Conclusions

- In this work, we introduce
  - A new task for topic discovery: **Hierarchical topic mining**
  - A joint text and tree embedding framework **JoSH**
  - A principled optimization method to train **JoSH**
- Future work:
  - Discover new categories in the hierarchy
  - Taxonomy expansion and enrichment
  - Embedding graph structure jointly with text

Thanks!