Hierarchical Topic Mining via Joint Spherical Tree and Text Embedding

Yu Meng¹*, Yunyi Zhang¹*, Jiaxin Huang¹, Yu Zhang¹, Chao Zhang², Jiawei Han¹
¹University of Illinois at Urbana-Champaign
²Georgia 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
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)
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
(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 $d_{\text{tree}}(\text{sports, arts}) = d_{\text{tree}}(\text{baseball, soccer})$, “baseball” and “soccer” should be embedded closer than “sports” and “arts” to reflect semantic similarity.

Use tree distance in a relative manner: Since $d_{\text{tree}}(\text{sports, baseball}) < d_{\text{tree}}(\text{baseball, soccer})$, “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

  \[
  L_{\text{intra}} = \sum_{c_i \in \mathcal{T}} \sum_{w_j \in c_i} \min(0, u_{w_j}^T c_i - m_{\text{intra}}),
  \]

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

  \[
  L_{\text{inter}} = \sum_{c_i \in \mathcal{T}} \sum_{c_j \in \mathcal{T} \setminus \{c_i\}} \min(0, 1 - c_i^T c_j - m_{\text{inter}}).
  \]

\[\theta_{\text{intra}} \leq \arccos(m_{\text{intra}})\]
\[\theta_{\text{inter}} \geq \arccos(1 - m_{\text{inter}})\]
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.

\[ \mathcal{L}_{\text{inter}} = \sum_{c_i \in T_r} \sum_{c_j \in T_r \setminus \{c_r, c_i\}} \min(0, c_i^T c_r - c_i^T c_j - m_{\text{inter}}), \]
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
    \[
    p(d_i | c_i) = \text{vMF}(d_i; c_i, \kappa_{c_i}) = n_p(\kappa_{c_i}) \exp(\kappa_{c_i} \cdot \cos(d_i, c_i))
    \]
  2. each word $w_j$ is generated conditioned on the semantics of the document $d_i$
    \[
    p(w_j | d_i) \propto \exp(\cos(u_{w_j}, d_i))
    \]
  3. surrounding words $w_{j+k}$ in the local context window of $w_i$ are generated conditioned on the semantics of the center word $w_i$
    \[
    p(w_{j+k} | w_j) \propto \exp(\cos(v_{w_{j+k}}, u_{w_j}))
    \]

1. Topic assignment
2. Global context
3. Local context
Outline

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

- Overall objective: Maximum likelihood estimation

\[ \mathcal{L} = \mathcal{L}_{\text{tree}} + \mathcal{L}_{\text{text}}, \]
\[ \mathcal{L}_{\text{tree}} = \sum_{c_r \in \mathcal{T}} \sum_{c_i \in \mathcal{T}_r} \sum_{c_j \in \mathcal{T}_r \setminus \{c_r, c_i\}} \min(0, c_i^T c_r - c_i^T c_j - m_{\text{inter}}). \quad (9) \]

\[ \mathcal{L}_{\text{text}} = \sum_{d_i \in \mathcal{D}} \sum_{w_j \in \mathcal{D}} \sum_{w_{j+k} \in \mathcal{D}} \min \left(0, \nu_{w_{j+k}}^T u_{w_j} + u_{w_j}^T d_i - \nu_{w_{j+k}}^T u_{w_j} - u_{w_j}^T d_i - m \right) \]

\[ + \sum_{c_i \in \mathcal{T}} \sum_{w_j \in \mathcal{C}_i} \left( \log \left( n_p (\kappa_{c_i}) \right) + \kappa_{c_i} u_{w_j}^T c_i \right) \mathbb{I}(u_{w_j}^T c_i < m_{\text{intra}}). \quad (10) \]

_s.t. \forall w, d, c,\quad \|u_w\| = \|\nu_w\| = \|d\| = \|c\| = 1, \]

Spherical Space Constraint
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
    \[ C_i(t) \leftarrow \text{Top}_t\left(\{w\}; u_w(t), c_i(t), \kappa_{c_i}(t)\right), \]
    the set of terms ranked at the top \( t \) positions via \( \nu\text{MF}(u_w; c_i, \kappa_{c_i}) \)
  - **M-Step:**
    \[ \Theta(t+1) \leftarrow \arg\max \left( L_{\text{ext}}(\Theta(t)) + L_{\text{tree}}(\Theta(t)) \right), \quad \Theta(t) = \{ u_w(t), v_w(t), d(t), c(t) \} \]

Require stochastic optimization technique
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:

\[
\text{Riemannian gradient } \nabla _\theta L(\theta) := (I - \theta \theta^T) \nabla L(\theta),
\]

Euclidean gradient

Riemannian stochastic optimization

\[
\theta^{(t+1)} \leftarrow R_{\theta^{(t)}} \left( \alpha \cdot \nabla L(\theta^{(t)}) \right)
\]

\[
R_x(z) := \frac{x + z}{\|x + z\|}
\]

--- 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 $D$; a category tree $T = \{c_i\}_{i=1}^n$; number of terms $K$ to retrieve per category.

**Output:** Hierarchical Topic Mining results $C_i|_{i=1}^n$.

\[ u_w, v_w, d, c \leftarrow \text{random initialization on } \mathbb{S}^{p-1}; \]

\[ t \leftarrow 1; \]

\[ C_i^{(1)} \leftarrow w_{c_i}|_{i=1}^n \quad \triangleright \text{initialize with category names}; \]

**while** $t < K + 1$ **do**

\[ t \leftarrow t + 1; \]

// Representative term retrieval;

\[ C_i^{(t)}|_{i=1}^n \leftarrow \text{Eq. (10)} \quad \triangleright \text{E-Step}; \]

// Embedding training;

\[ u_w, v_w, d, c \leftarrow \text{Eq. (11)} \quad \triangleright \text{M-Step}; \]

**for** $i \leftarrow 1$ **to** $n$ **do**

\[ C_i^{(t)} \leftarrow C_i^{(t)} \setminus \{w_{c_i}\} \quad \triangleright \text{exclude category names}; \]

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.

<table>
<thead>
<tr>
<th>Corpus</th>
<th># super-categories</th>
<th># sub-categories</th>
<th># documents</th>
</tr>
</thead>
<tbody>
<tr>
<td>NYT</td>
<td>8</td>
<td>12</td>
<td>89,768</td>
</tr>
<tr>
<td>arXiv</td>
<td>3</td>
<td>29</td>
<td>230,105</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Models</th>
<th>NYT</th>
<th>arXiv</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TC</td>
<td>MACC</td>
</tr>
<tr>
<td>hLDA</td>
<td>-0.0070</td>
<td>0.1636</td>
</tr>
<tr>
<td>hPAM</td>
<td>0.0074</td>
<td>0.3091</td>
</tr>
<tr>
<td>JoSE</td>
<td>0.0140</td>
<td>0.6818</td>
</tr>
<tr>
<td>Poincaré GloVe</td>
<td>0.0092</td>
<td>0.6182</td>
</tr>
<tr>
<td>Anchored CorEx</td>
<td>0.0117</td>
<td>0.3909</td>
</tr>
<tr>
<td>CatE</td>
<td>0.0149</td>
<td>0.9000</td>
</tr>
<tr>
<td>JoSH</td>
<td><strong>0.0166</strong></td>
<td><strong>0.9091</strong></td>
</tr>
</tbody>
</table>
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.
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.

<table>
<thead>
<tr>
<th></th>
<th>hLDA</th>
<th>hPAM</th>
<th>JoSE</th>
<th>Poincaré GloVe</th>
<th>Anchored CorEx</th>
<th>CatE</th>
<th>JoSH</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>53</td>
<td>22</td>
<td>5</td>
<td>16</td>
<td>61</td>
<td>52</td>
<td>6</td>
</tr>
</tbody>
</table>
JoSH can be either directly used as a generative classifier, i.e., \( y_d = \arg \max_c \nuMF(d; c, \kappa_c) \), 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.

<table>
<thead>
<tr>
<th>Models</th>
<th>NYT</th>
<th>arXiv</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Macro-F1</td>
<td>Micro-F1</td>
</tr>
<tr>
<td>WeSHClass</td>
<td>0.425</td>
<td>0.581</td>
</tr>
<tr>
<td>JoSH</td>
<td>0.429</td>
<td>0.600</td>
</tr>
<tr>
<td>WeSHClass + CatE</td>
<td>0.503</td>
<td>0.679</td>
</tr>
<tr>
<td>WeSHClass + JoSH</td>
<td>0.582</td>
<td>0.703</td>
</tr>
</tbody>
</table>
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)
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!