Part III: Embedding-Driven Topic Discovery

KDD 2021 Tutorial

On the Power of Pre-Trained Text Representations: Models and Applications in Text Mining

Yu Meng, Jiaxin Huang, Yu Zhang, Jiawei Han

Computer Science, University of Illinois at Urbana-Champaign

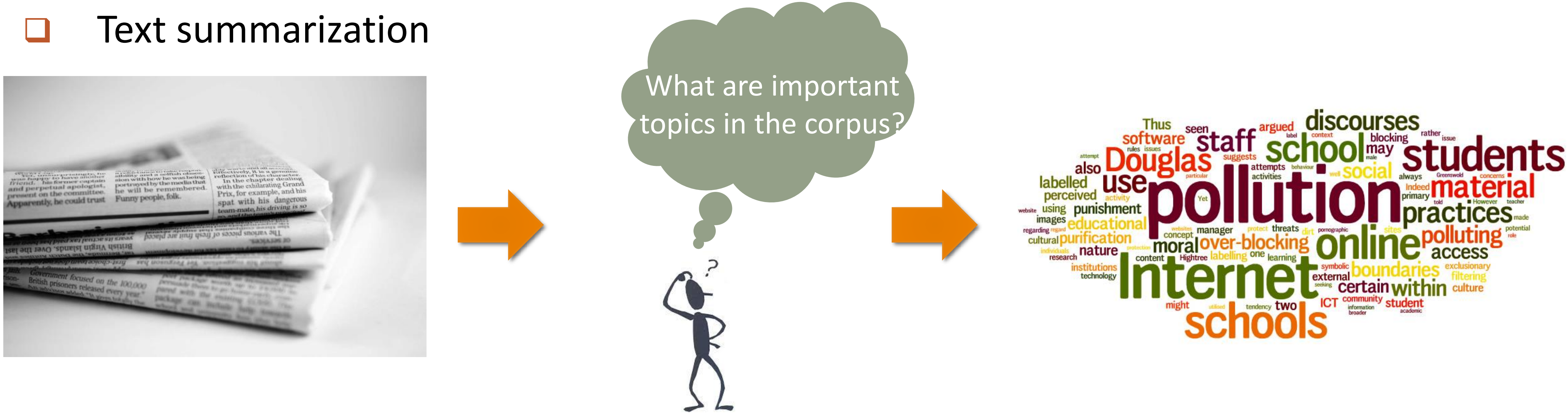
August 14, 2021

Outline

- Unsupervised Topic Modeling
- Supervised & Seed-Guided Topic Modeling
- Clustering-Based Topic Discovery
- Discriminative Topic Mining

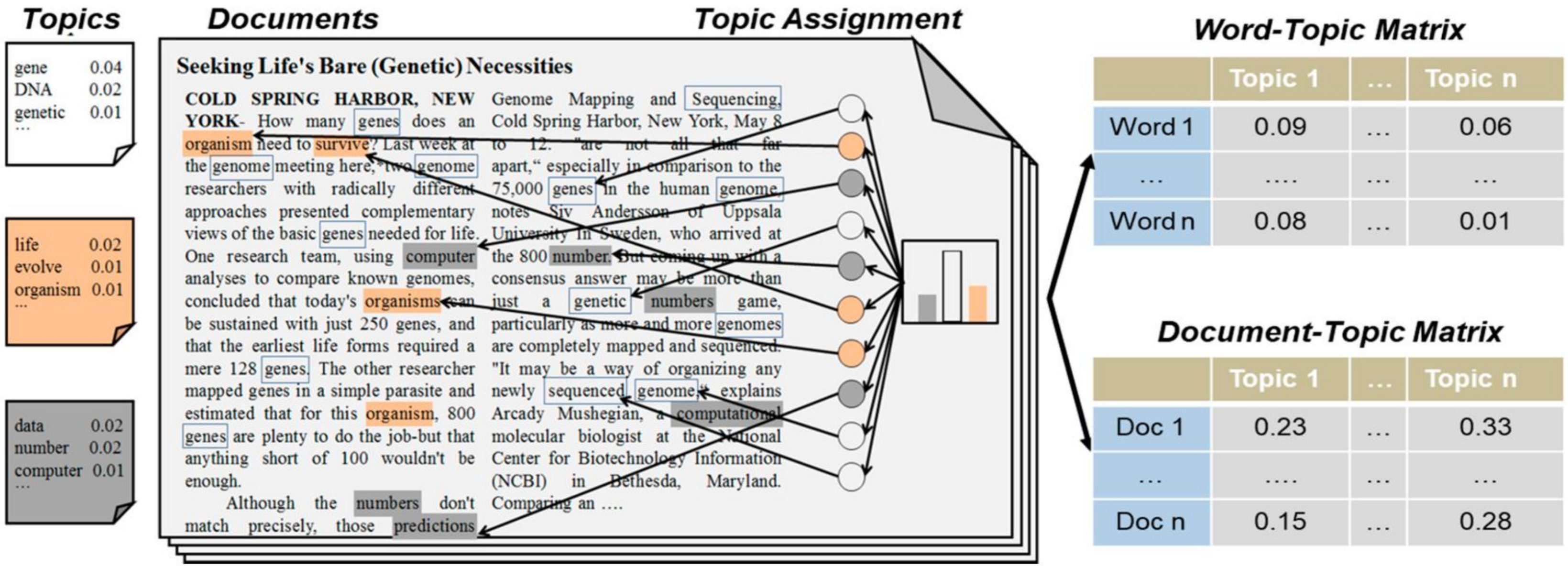
Topic Modeling: Introduction

- How to effectively & efficiently comprehend a large text corpus?
- Knowing what important topics are there is a good starting point!
- Topic discovery facilitates a wide spectrum of applications
  - Document classification/organization
  - Document retrieval/ranking
  - Text summarization



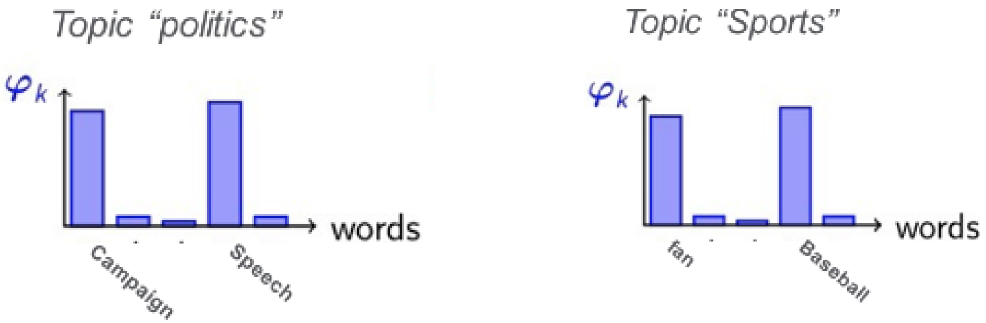
Topic Modeling: Overview

- How to discover topics automatically from the corpus?
- By modeling the corpus statistics!
  - Each document has a latent topic distribution
  - Each topic is described by a different word distribution



Latent Dirichlet Allocation (LDA): Overview

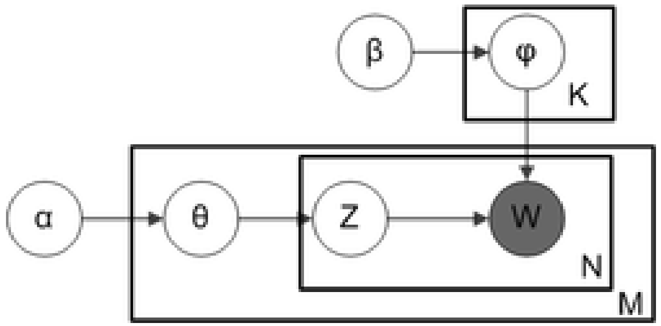
- Each document is represented as a mixture of various topics
  - Ex. A news document may be 40% on politics, 50% on economics, and 10% on sports
- Each topic is represented as a probability distribution over words
  - Ex. The distribution of “politics” vs. “sports” might be like:



- Dirichlet priors are imposed to enforce sparse distributions:
  - Documents cover only a small set of topics (sparse document-topic distribution)
  - Topics use only a small set of words frequently (sparse topic-word distribution)

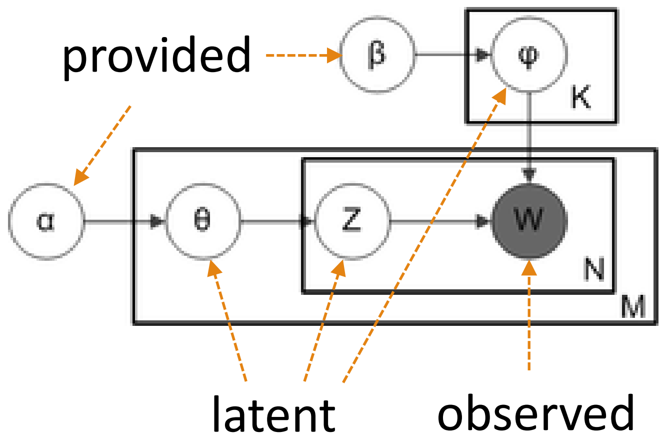
LDA: Generative Model

- Formulating the statistical relationship between words, documents and latent topics as a generative process describing how documents are created:
  - For the ith document, choose $\theta_i \sim \mathrm{Dir}(\alpha)$ document's topic distribution
  - For the kth topic, choose $\varphi_k \sim \mathrm{Dir}(\beta)$ topic's word distribution
  - For the jth word in the ith document,
    - choose topic $z_{i,j} \sim \mathrm{Categorical}(\theta_i)$ word's topic
    - choose a word $w_{i,j} \sim \mathrm{Categorical}(\varphi_{z_{i,j}})$



LDA: Inference

- Learning the LDA model (Inference)
- What need to be learned
  - Document topic distribution θ (for assigning topics to documents)
  - Topic-word distribution φ (for topic interpretation)
  - Words' latent topic z
- How to learn the latent variables? – complicated due to intractable posterior
  - Monte Carlo simulation
  - Gibbs sampling
  - Variational inference
  - …



Outline

- Unsupervised Topic Modeling
- Supervised & Seed-Guided Topic Modeling
- Clustering-Based Topic Discovery
- Discriminative Topic Mining

Issues with LDA

- LDA is completely unsupervised (i.e., users only input number of topics)
- Cannot take user supervision
  - Ex. What if a user is specifically interested in some topics but LDA doesn't discover them?

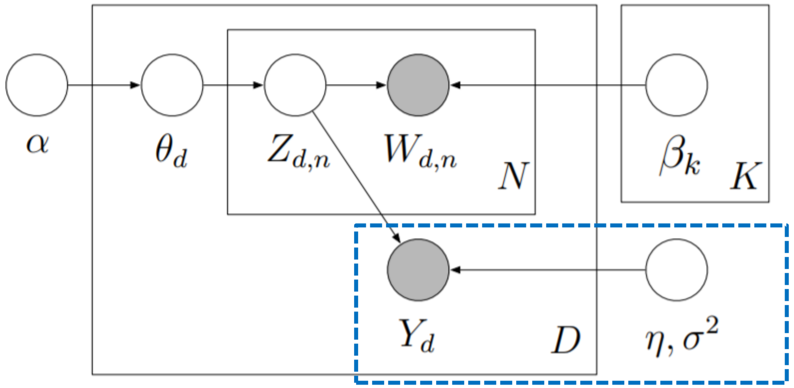
	Topic 1	Weight	Topic 2	Weight	Topic 3	Weight	Topic 4	Weight	Topic 5	Weight
0	life	0.018076	father	0.059603	official	0.017620	case	0.021908	art	0.010555
1	man	0.017714	graduate	0.048363	force	0.015388	law	0.020698	open	0.010413
2	woman	0.016657	son	0.042746	military	0.014587	court	0.019967	room	0.010363
3	book	0.010486	mrs	0.041379	war	0.011381	lawyer	0.016935	house	0.009002
4	family	0.010382	daughter	0.037156	government	0.010564	state	0.014501	building	0.008722
5	young	0.009896	mother	0.034542	troop	0.008949	judge	0.012487	artist	0.008264
6	write	0.009493	receive	0.029211	attack	0.008886	legal	0.011141	design	0.008162
7	child	0.009460	marry	0.029038	leader	0.008082	rule	0.009854	floor	0.008034
8	live	0.008819	yesterday	0.024107	peace	0.006835	decision	0.009261	museum	0.007917
9	love	0.007814	degree	0.022899	soldier	0.006562	file	0.008289	exhibition	0.007222

	Topic 6	Weight	Topic 7	Weight	Topic 8	Weight	Topic 9	Weight	Topic 10	Weight
0	group	0.051052	market	0.024976	serve	0.010918	change	0.007661	city	0.021776
1	member	0.040683	stock	0.024874	add	0.010185	system	0.007233	area	0.014865
2	meeting	0.016390	share	0.020583	minute	0.009301	problem	0.006835	build	0.014361
3	issue	0.014988	price	0.018141	pepper	0.009235	power	0.005400	building	0.014326
4	official	0.013069	sell	0.016564	oil	0.008976	create	0.005056	home	0.013632
5	support	0.011994	buy	0.015415	cook	0.008711	research	0.004712	resident	0.013483
6	leader	0.011799	company	0.015249	food	0.008689	produce	0.004574	community	0.012479
7	organization	0.011135	investor	0.015062	cup	0.008682	far	0.004447	local	0.010686
8	meet	0.010235	yesterday	0.012813	sauce	0.008209	result	0.004280	live	0.010661
9	effort	0.008479	analyst	0.010768	small	0.007864	kind	0.004166	project	0.010459

10 topics generated by LDA on The New York Times dataset

Supervised LDA (sLDA)

- Allow users to provide document annotations/labels
- Incorporate document labels into the generative process
  - For the ith document, choose $\boxed{\theta_i} \sim \mathrm{Dir}(\alpha)$ document's topic distribution
  - For the jth word in the ith document,
    - choose topic $\boxed{z_{i,j}} \sim \mathrm{Categorical}(\theta_i)$ word's topic
    - choose a word $w_{i,j} \sim \mathrm{Categorical}(\beta_{z_{i,j}})$
  - For the ith document, choose $\boxed{y_i \sim N(\eta^\top \bar{z}_i, \sigma^2)}$, $\bar{z}_i = \frac{1}{L}\sum_{j=1}^{L} z_{i,j}$



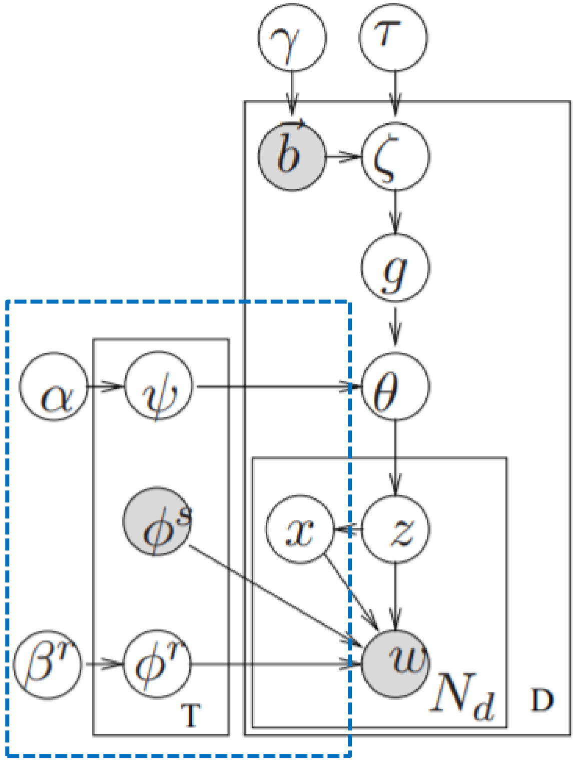
generate document's label

Seeded LDA: Guided Topic-Word Distribution

- Another form of user supervision: several seed words for each topic

1. For each k=1 $\cdots$ T,
   (a) Choose regular topic $\phi_k^r \sim \text{Dir}(\beta_r)$.
   (b) Choose *seed* topic $\phi_k^s \sim \text{Dir}(\beta_s)$.
   (c) Choose $\pi_k \sim \text{Beta}(1,1)$.
2. For each seed set $s = 1 \cdots$ S,
   (a) Choose group-topic distribution $\psi_s \sim \text{Dir}(\alpha)$.
3. For each document d,
   (a) Choose a binary vector $\vec{b}$ of length S.
   (b) Choose a document-group distribution $\zeta^d \sim \text{Dir}(\tau\vec{b})$.
   (c) Choose a group variable $g \sim \text{Mult}(\zeta^d)$.
   (d) Choose $\theta_d \sim \text{Dir}(\psi_g)$. // of length T
   (e) For each token $i = 1 \cdots N_d$:
      i. Select a topic $z_i \sim \text{Mult}(\theta_d)$.
      ii. Select an indicator $x_i \sim \text{Bern}(\pi_{z_i})$.
      iii. if x_i is 0
         - Select a word $w_i \sim \text{Mult}(\phi_{z_i}^r)$.
      iv. if x_i is 1
         - Select a word $w_i \sim \text{Mult}(\phi_{z_i}^s)$.

Seed topics used to improve the topic-word distribution:
Each word comes from either "regular topics" with a distribution over all word like in LDA, or "seed topics" which only generate words from the seed set

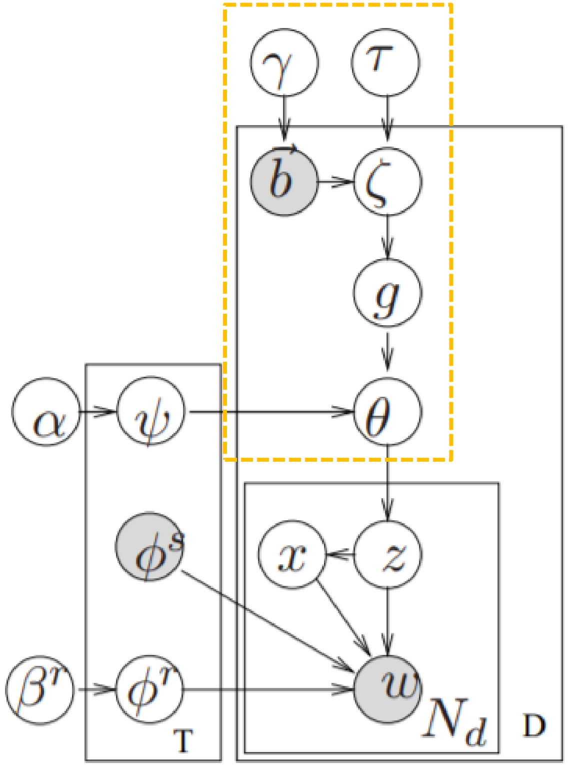


Seeded LDA: Guided Document-Topic Distribution

- Another form of user supervision: several seed words for each topic

1. For each k=1 $\cdots$ T,
   (a) Choose regular topic $\phi_k^r \sim \mathrm{Dir}(\beta_r)$.
   (b) Choose *seed* topic $\phi_k^s \sim \mathrm{Dir}(\beta_s)$.
   (c) Choose $\pi_k \sim \mathrm{Beta}(1,1)$.
2. For each seed set $s = 1 \cdots$ S,
   (a) Choose group-topic distribution $\psi_s \sim \mathrm{Dir}(\alpha)$.
3. For each document d,
   (a) Choose a binary vector $\vec{b}$ of length S.
   (b) Choose a document-group distribution $\zeta^d \sim \mathrm{Dir}(\tau\vec{b})$.
   (c) Choose a group variable $g \sim \mathrm{Mult}(\zeta^d)$.
   (d) Choose $\theta_d \sim \mathrm{Dir}(\psi_g)$. // of length T
   (e) For each token $i = 1 \cdots N_d$:
      i. Select a topic $z_i \sim \mathrm{Mult}(\theta_d)$.
      ii. Select an indicator $x_i \sim \mathrm{Bern}(\pi_{z_i})$.
      iii. if x_i is 0
         - Select a word $w_i \sim \mathrm{Mult}(\phi_{z_i}^r)$.
      iv. if x_i is 1
         - Select a word $w_i \sim \mathrm{Mult}(\phi_{z_i}^s)$.

Seed topics used to improve the document-topic distribution:
Group-topic distribution = seed set distribution over regular topics
Group-topic distribution used as prior to draw document-topic distribution



Outline

- Unsupervised Topic Modeling
- Supervised & Seed-Guided Topic Modeling
- Clustering-Based Topic Discovery
- Discriminative Topic Mining

Clustering-Based Topic Discovery

- Topic modeling frameworks use **bag-of-words** features (i.e., only word counts in documents matter; word ordering is ignored)
- In Part I of the tutorial, we introduced distributed text representations (text embeddings and language models) that better model sequential information in text
- Can we take advantage of those advanced text representations for the topic discovery task, as an alternative to topic modeling?

Word Embedding + Clustering

- Cast “topics” as clusters of word types — similar to taking the top-ranked words from each topic’s distribution in topic modeling
- How to obtain word clusters? Run clustering algorithms on word embeddings
- Since the text embedding space captures word semantic similarity (i.e., high vector similarity implies high semantic similarity), using distance-based clustering algorithms (like K-means) will naturally group semantically similar words into the same cluster

Clustering-Based Topic Discovery: A benchmark study

- Clustering algorithms:
  - k-means (KM)
  - Gaussian Mixture Models (GMM)
- Embeddings:
  - Word2Vec
  - GloVe
  - fastText
  - Spherical text embedding
  - ELMo
  - BERT

Sia, S., Dalmia, A., & Mielke, S. J. (2020). Tired of Topic Models? Clusters of Pretrained Word Embeddings Make for Fast and Good Topics too! EMNLP

Clustering-Based Topic Discovery: Word Frequency

- One thing to consider is that text embeddings do not explicitly encode frequency information, which is important for topic discovery (i.e., more frequent words in the corpus may be more representative)
- Two ways to incorporate frequency information
  - Weighted clustering: Frequent words weigh more when computing cluster centroids
  - Rerank words in clusters: Rerank terms by frequency in each cluster when selecting representative terms

Clustering-Based Topic Discovery: Results

- Using k-means (KM)/Gaussian Mixture Models (GMM) as clustering algorithm and using Spherical text embedding/BERT as representations leads to comparable results with LDA
- Future work
  - More advanced clustering algorithms?
  - Joint modeling of document-topic distribution via clustering?

weighted clustering + reranking

	Reuters								20 Newsgroups							
	$\diamond$		$\diamond^w$		$\diamond_r$		$\diamond_r^w$		$\diamond$		$\diamond^w$		$\diamond_r$		$\diamond_r^w$	
	KM	GMM	KM	GMM	KM	GMM	KM	GMM	KM	GMM	KM	GMM	KM	GMM	KM	GMM
Word2vec	-0.39	-0.47	-0.21	-0.09	0.02	0.01	0.03	0.08	-0.21	-0.10	-0.11	0.13	0.18	0.16	0.19	0.20
ELMo	-0.73	-0.55	-0.43	0.00	-0.10	-0.08	-0.02	0.06	-0.56	-0.13	-0.38	0.18	0.13	0.14	0.16	0.19
GloVe	-0.67	-0.59	-0.04	0.01	-0.27	-0.03	0.01	0.05	-0.18	-0.12	0.06	0.24	0.22	0.23	0.23	0.23
Fasttext	-0.68	-0.70	-0.46	-0.08	0.00	0.00	0.06	0.11	-0.32	-0.20	-0.18	0.21	0.24	0.23	0.25	0.24
Spherical	-0.53	-0.65	-0.07	0.09	0.01	-0.05	0.10	0.12	-0.05	-0.24	0.24	0.23	0.25	0.22	0.26	0.24
BERT	-0.43	-0.19	-0.07	0.12	0.00	-0.01	0.12	0.15	0.04	0.14	0.25	0.25	0.17	0.19	0.25	0.25
average	-0.57	-0.52	-0.21	0.01	-0.06	-0.03	0.05	0.10	-0.21	-0.11	-0.02	0.21	0.20	0.20	0.23	0.23
std. dev.	0.14	0.18	0.19	0.09	0.12	0.03	0.05	0.04	0.21	0.13	0.25	0.05	0.04	0.04	0.04	0.02

Table 1: NPMI Results (higher is better) for pre-trained word embeddings and k-means (KM), and Gaussian Mixture Models (GMM). $\diamond^w$ indicates weighted and $\diamond_r$ indicates reranking of top words. For Reuters (left table), LDA has an NPMI score of 0.12, while GMM_r^w BERT achieves 0.15. For 20NG (right), both LDA and KM_r^w Spherical achieve a score of 0.26. All results are averaged across 5 random seeds.

Outline

- Unsupervised Topic Modeling
- Supervised & Seed-Guided Topic Modeling
- Clustering-Based Topic Discovery
- Discriminative Topic Mining
  - Introduction of the Task
  - CatE: Discriminative Topic Mining via Category-Name Guided Text Embedding [WWW'20]
  - Demo: TopicMine (based on CatE)
  - JoSH: Hierarchical Topic Mining via Joint Spherical Tree and Text Embedding [KDD'20]

Motivations

- What are the limitations of topic models?
- **Failure to incorporate user guidance:** 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 skewed and biased summarization of the corpus
- **Failure to enforce distinctiveness among retrieved topics:** Topic models do not impose discriminative constraints
  - concepts are most effectively interpreted via their uniquely defining features
  - e.g. Egypt is known for pyramids and China is known for the Great Wall

Motivations

- **(Cont'd) Failure to enforce distinctiveness among retrieved topics:** Topic models do not impose discriminative constraints
  - three retrieved topics from the New York Times annotated corpus via LDA:

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

  - 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)

Introduction

- **A New Task: Discriminative Topic Mining**
  - Given a text corpus and a set of **category names**, discriminative topic mining aims to retrieve a set of terms that **exclusively belong to** each category
  - Ex. Given c_1: “The United States”, c_2: “France”, c_3: “Canada”
    - correct to retrieve “Ontario” under c_3: Ontario is a province in Canada and exclusively belongs to Canada
    - incorrect to retrieve “North America” under c_3: North America is a continent and does not belong to any countries (reversed belonging relationship)
    - incorrect to retrieve “English” under c_3: English is also the national language of the United States (not discriminative)

Discriminative Topic Mining

- **A New Task: Discriminative Topic Mining**
  - Difference from topic modeling
    - requires **a set of user provided category names** and only focuses on retrieving terms belonging to the given categories
    - imposes strong discriminative requirements that each retrieved term under the corresponding category must **belong to and only belong to** that category semantically

Outline

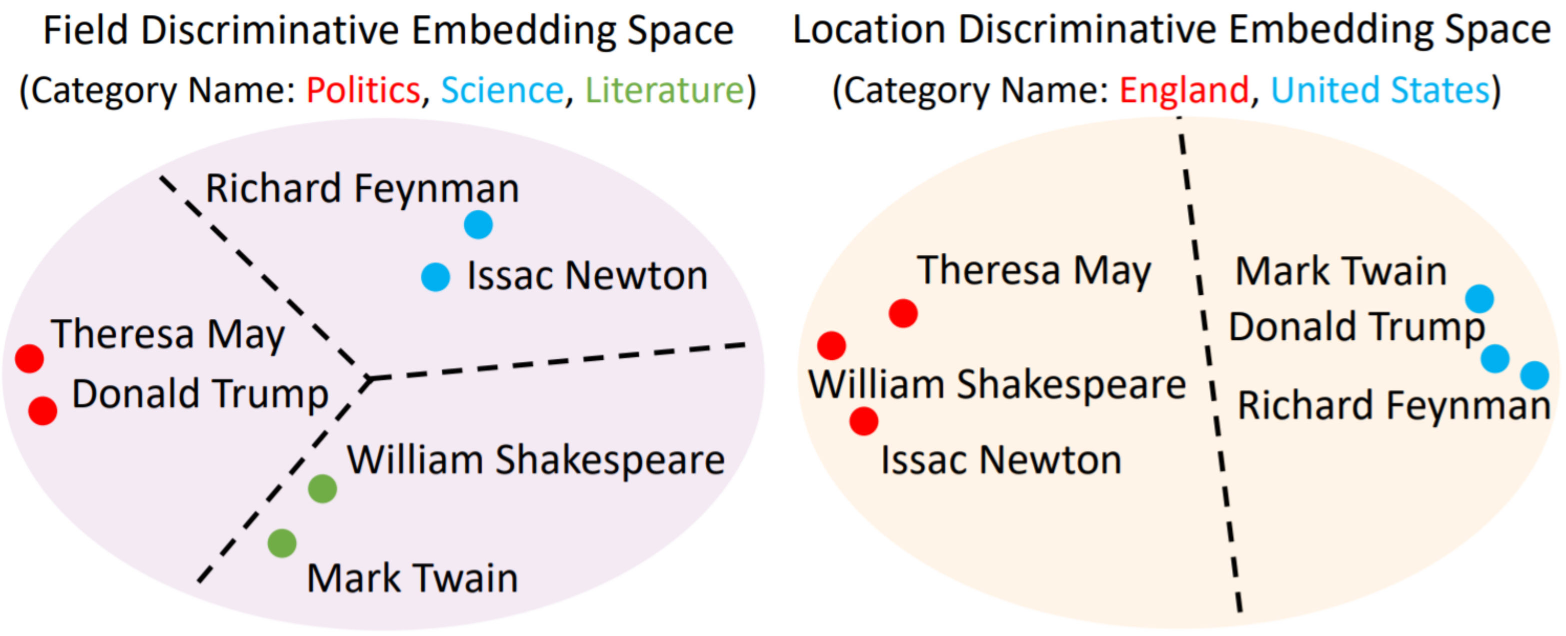
- ❑ Unsupervised Topic Modeling
- ❑ Supervised & Seed-Guided Topic Modeling
- ❑ Clustering-based Topic Discovery
- ❑ Discriminative Topic Mining
  - ❑ Introduction of the Task
  - ❑ CatE: Discriminative Topic Mining via Category-Name Guided Text Embedding [WWW’20]
  - ❑ Demo: TopicMine (based on CatE)
  - ❑ JoSH: Hierarchical Topic Mining via Joint Spherical Tree and Text Embedding [KDD’20]

CatE Embedding: Overview

- Motivation:
  - Topic models use document-topic and topic-word distributions to model the text generation process
    - able to discover hidden topic semantics
    - bag-of-words generation assumption
  - Word embeddings capture word semantic correlations via the distributional hypothesis
    - captures local context similarity
    - not exploit document-level statistics (global context)
    - not model topics
- Take advantage of both frameworks!

CatE Embedding: Discriminative Embedding

- Intuitively, with different categories to be discriminated, the embedding space should have different distribution
- How to achieve this property?



CatE Embedding: Text Generation Modeling

- Modeling text generation under user guidance
- A three-step process:
  1. A document d is generated conditioned on one of the n categories
  2. Each word w_i is generated conditioned on the semantics of the document d
  3. Surrounding words w_{i+j} in the local context window of w_i are generated conditioned on the semantics of the center word w_i
- Likelihood of corpus generation conditioned on user-given categories

1. Topic assignment

2. Global context

3. Local context

CatE Embedding: Objective

- Objective: negative log-likelihood

$$P(\mathcal{D} \mid C) = \underbrace{\prod_{d \in \mathcal{D}} p(d \mid c_d)}_{\text{1. Topic assignment}} \underbrace{\prod_{w_i \in d} p(w_i \mid d)}_{\text{2. Global context}} \prod_{\substack{w_{i+j} \in d \\ -h \le j \le h, j \ne 0}} \underbrace{p(w_{i+j} \mid w_i)}_{\text{3. Local context}}$$

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

Decompose into word-topic distribution

- How do we know which word belongs to which category (word-topic distribution)?

Category Representative Word Retrieval

- As a starting point, we propose to retrieve representative words by jointly considering two separate aspects:
  - Relatedness: measured by embedding cosine similarity
  - Specificity: category representative words should be more specific than the category name
- Ex. "Ontario" can be selected as a category representative word of "Canada" since it is **related** to "Canada" and **more specific** than "Canada".
- How do we know the specificity of words?

Word Semantic Specificity

- Word distributional specificity:

> **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.

- Ex. "seafood" has a higher word distributional specificity than "food", because seafood is a specific type of food

Jointly Learning Word Embedding and Specificity

- Our model:

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

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

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

- κ_w is the distributional specificity of w.

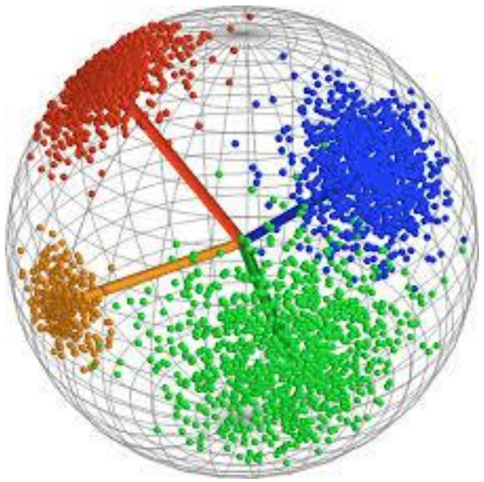
Interpreting The Model

- Preliminary: The vMF distribution – A distribution defined on unit sphere

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

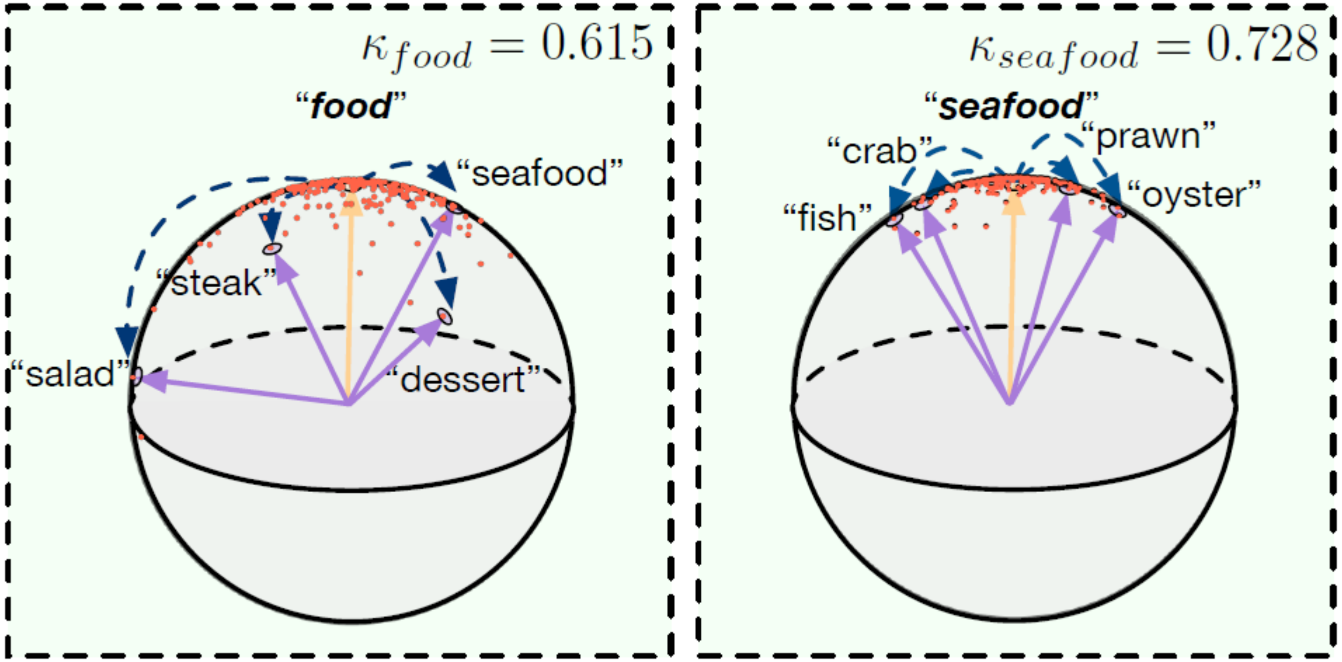
Concentration Parameter

Center Direction



Interpreting The Model

- (Theorem) 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



Category Representative Word Retrieval

- Ranking Measure for Selecting Class Representative Words:
- We find a representative word of category c_i and add it to the set S by

Prefer words having high embedding cosine similarity with the category name

Prefer words with low distributional specificity (more general)

$$w = arg\, min_w \mathrm{rank}_{sim}(w, c_i) \cdot \mathrm{rank}_{spec}(w)$$

$$s.t. \quad w \notin \mathcal{S} \quad \text{and} \quad \kappa_w > \kappa_{c_i}.$$

w hasn't been a representative word

w must be more specific than the category name

Overall Algorithm

Algorithm 1: Discriminative Topic Mining.

Input: A text corpus $\mathcal{D}$; a set of category names $\mathcal{C} = \{c_i\}|_{i=1}^{n}$.

Output: Discriminative topic mining results $\mathcal{S}_i|_{i=1}^{n}$.

```
for i ← 1 to n do
    S_i ← {c_i}        ▹ initialize S_i with category names;
for t ← 1 to max_iter do
    Train W, C on D according to Equation (2);
    for i ← 1 to n do
        w ← Select representative word of c_i by Eq. (12);
        S_i ← S_i ∪ {w};
for i ← 1 to n do
    S_i ← S_i \ {c_i}        ▹ exclude category names;
Return S_i|_{i=1}^n;
```

Experiment Settings

- Datasets
- New York Times annotated corpus (Sandhaus, 2008)
  - topic
  - location
- Recently released Yelp Dataset Challenge
  - food type
  - sentiment

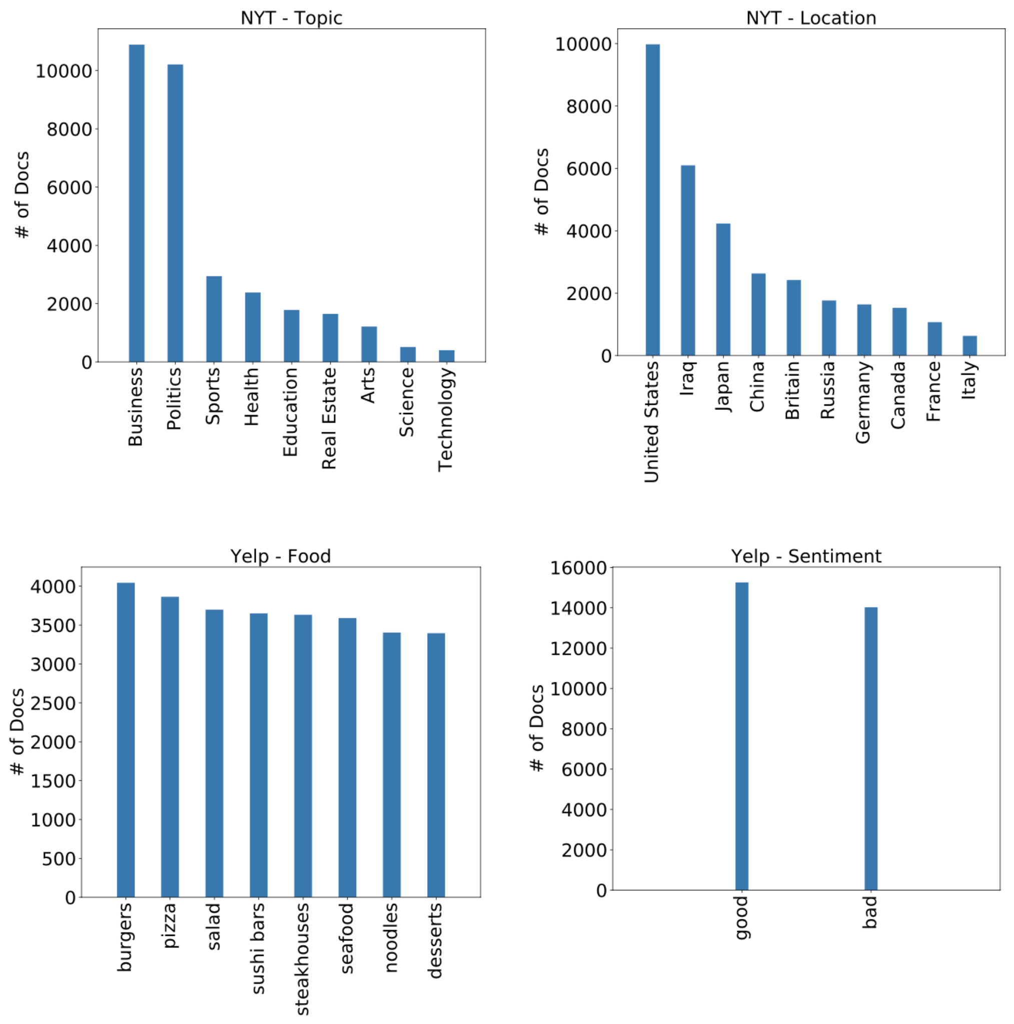


Figure 2: Dataset statistics.

Experiments

- Discriminative Topic Mining:
- Baselines
  - LDA (NIPS 2003) Manual select
  - Seeded LDA (EACL 2012) Seed-guided
  - TWE (AAAI 2015) Embedding-based
  - Anchored CorEx (TACL 2017) Seed-guided
  - Labeled ETM (arXiv 2019) Embedding-based
- Metrics:
  - Averaged topic coherence: how coherent the mined topics are
  - Mean accuracy: how accurately the retrieved terms belong to the category

Qualitative Results

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

Quantitative Results

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**

Experiments: Weakly-Supervised Text Classification:

- Use different embedding features to WeSTClass model
- Baselines:
  - Word2Vec (NIPS 2013)
  - GloVe (EMNLP 2014)
  - fastText (TACL 2017)
  - BERT (NAACL 2019)

Experiments: Weakly-Supervised Text Classification:

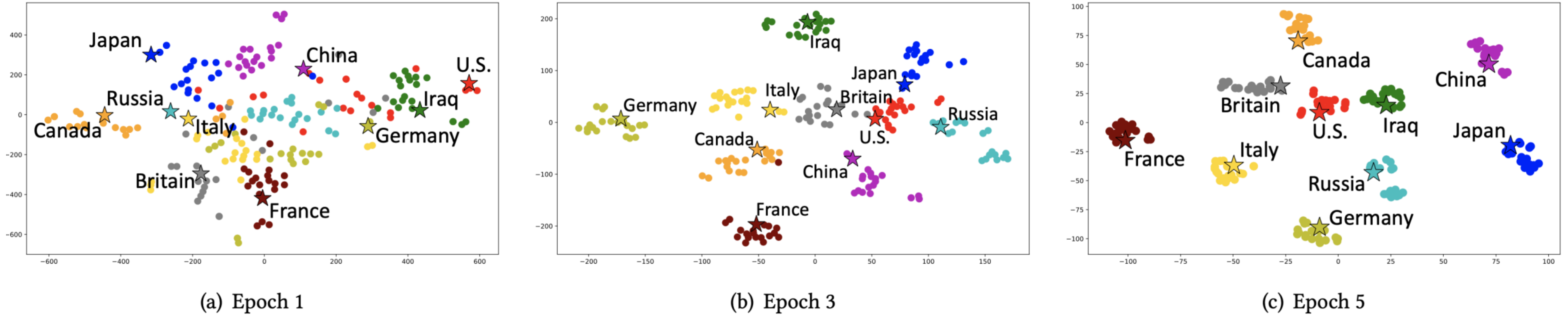
- Text Classification results

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**

Case Study

- Discriminative Embedding Space



Case Study

- Coarse-to-Fine Topic Presentation

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

Outline

- Unsupervised Topic Modeling
- Supervised & Seed-Guided Topic Modeling
- Clustering-based Topic Discovery
- Discriminative Topic Mining
  - Introduction of the Task
  - CatE: Discriminative Topic Mining via Category-Name Guided Text Embedding [WWW'20]
  - Demo: TopicMine (based on CatE)
  - JoSH: Hierarchical Topic Mining via Joint Spherical Tree and Text Embedding [KDD'20]

Project Goal

- Topic discovery in massive text corpora presents a holistic view to users of the contents
- However, traditional unsupervised methods like Latent Dirichlet Allocation (LDA) fail to provide completely meaningful and user-interested topics
- We develop TopicMine, a user-guided topic mining system that takes user-interested category names as input and retrieve category representative phrases to form coherent topics

Inputs → Outputs

Categories:

- Britain → London, England, Scotland, Wales, ...
- China → Beijing, Shanghai, Hong Kong, Fujian, ...
- Canada → Ontario, Toronto, Quebec, Montreal, ...

Category representative phrases

Project Goal

- TopicMine presents a category in a coarse-to-fine manner: The category representative phrases are first selected by category relevance, and then ranked by semantic specificity
- Our framework learns an additional parameter κ for each phrase which reflects how specific the phrase meaning is based on how variant the phrase's local contexts are in the entire corpus
- For example, "California" will be ranked higher than "Log Angeles" as representative phrases for category "The United States"

$$\kappa_{United\ States} < \kappa_{California} < \kappa_{Los\ Angeles} < \kappa_{USC}$$

Input		Output
Category: United States	→	California, Los Angeles, USC, ...

Category Representative Phrases

- User Inputs: (truth discovery, text mining, pattern mining)

TRUTH DISCOVERY	TEXT MINING	PATTERN MINING
misinformation	text_analysis	sequential_pattern_mining
faitcrowd	document_retrieval	frequent_sequence_mining
rumors	text_processing	frequent_itemset_mining
veracity	text_analytics	motif_discovery
missing_values	information_extraction	pattern_discovery
untrustworthy	biomedical_informatics	minimum_spanning_tree
multiple_sources	latent_semantic_analysis	a-priori
multi-source	unstructured_text	pattern_matching

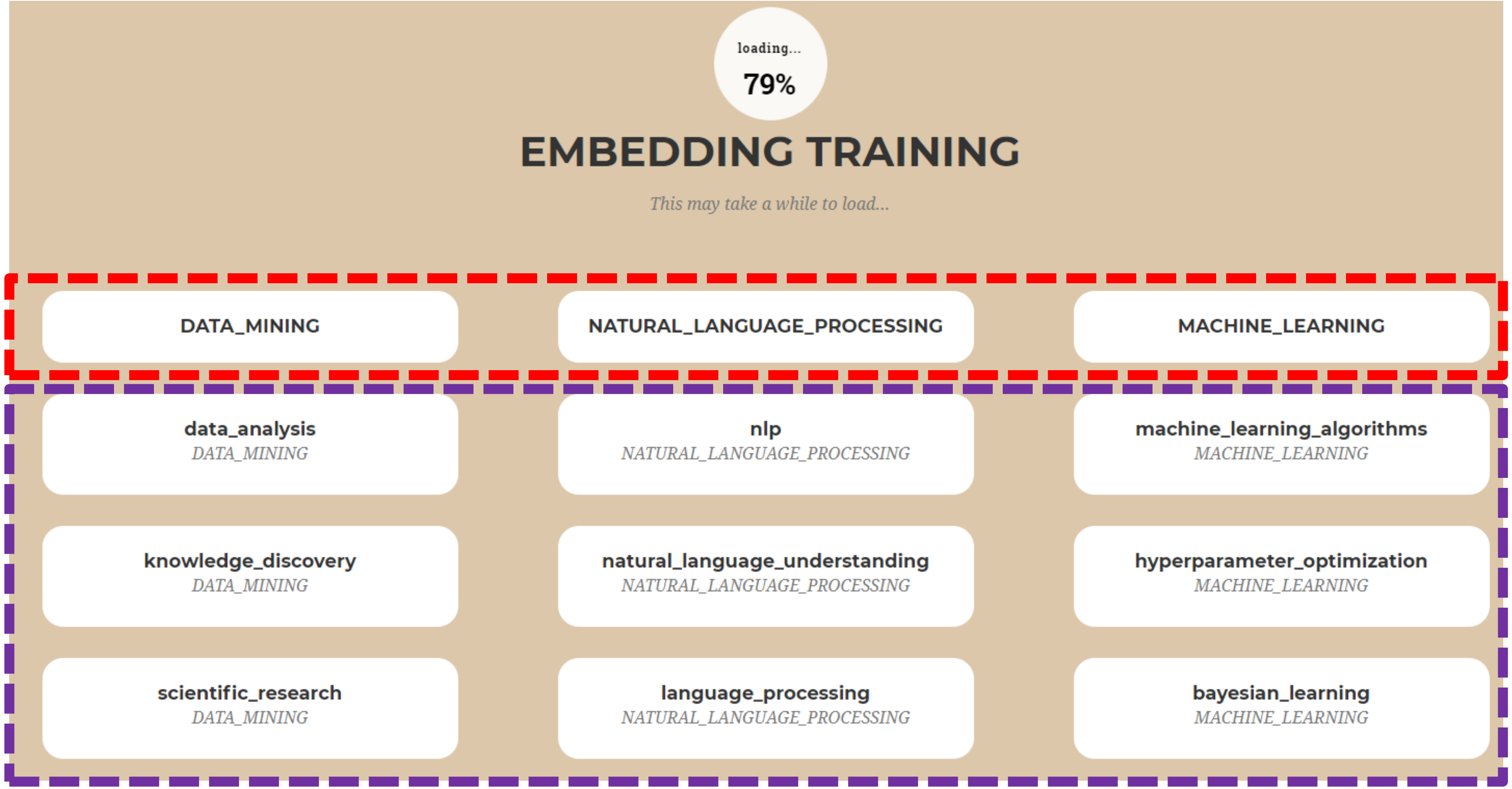
Category Phrases Sort By Specificity Range

- Coarse-to-fine topic presentation

RANGE OF κ	TRUTH DISCOVERY	TEXT MINING	PATTERN MINING
$1 < \frac{\kappa}{\kappa_c} < 1.25$	misinformation common_sense_knowledge rumors	text_analysis text_processing unstructured_text	sequential_pattern_mining frequent_sequence_mining frequent_itemset_mining
$1.25 < \frac{\kappa}{\kappa_c} < 1.5$	multiple_sources decision_problem fact-checking	document_retrieval information_extraction topic_extraction	minimum_spanning_tree pruning_techniques association_rules
$1.5 < \frac{\kappa}{\kappa_c} < 1.75$	faitcrowd hyptrails timing_information	latent_semantic_analysis tf-idf semeval-2015	trajectory-based a-priori community-level

Demo System Showcase

Inputs



Class representative phrases

Demo System Showcase

CATEGORY REPRESENTATIVE PHRASES

DATA_MINING	NATURAL_LANGUAGE_PROCESSING	MACHINE_LEARNING
scientific_data	language_processing	machine_learning_algorithms
pattern_mining	natural_language_understanding	hyperparameter_optimization
data_analysis	linguistic	supervised_learning
text_mining	linguistic_resources	multinomial_naive_bayes
data_warehousing	nlp_tasks	nonlinear_regression
biomedical_informatics	language_acquisition	hyperparameters
data_visualization	text_understanding	regression
information_network	lexical_semantics	variational_bayesian_inference
scientific_applications	computational_linguistics	nonparametric_regression
correlation_analysis	natural_languages	poisson_regression

Demo System Showcase

CATEGORY PHRASES SORT BY SPECIFICITY RANGE

RANGE OF K	DATA_MINING	NATURAL_LANGUAGE_PROCESSING	MACHINE_LEARNING
1.0<k<1.25	data_mining scientific_data data_analysis data_warehousing data_visualization	natural_language_processing computational_linguistics linguistic_resources semantic_representation spoken_dialogue	machine_learning statistical_methods regression hyperparameter kernel_machines
1.25<k<1.5	web_mining graph_mining pattern_mining market_analysis bioinformatics	language_identification natural_language_understanding semantic_relations natural_language_generation knowledge_extraction	machine_learning_algorithms hyperparameter_optimization bayesian_optimization supervised_learning logistic_regression
1.5<k<1.75	social_network_analysis biological_networks sequential_pattern_mining frequent_itemset_mining community_discovery	named_entity_recognition word_sense_disambiguation semantic_role_labeling visual_question_answering sentiment_analysis	online_learning_algorithms kernel_ridge_regression em_algorithm support_vector_machines variational_inference

Outline

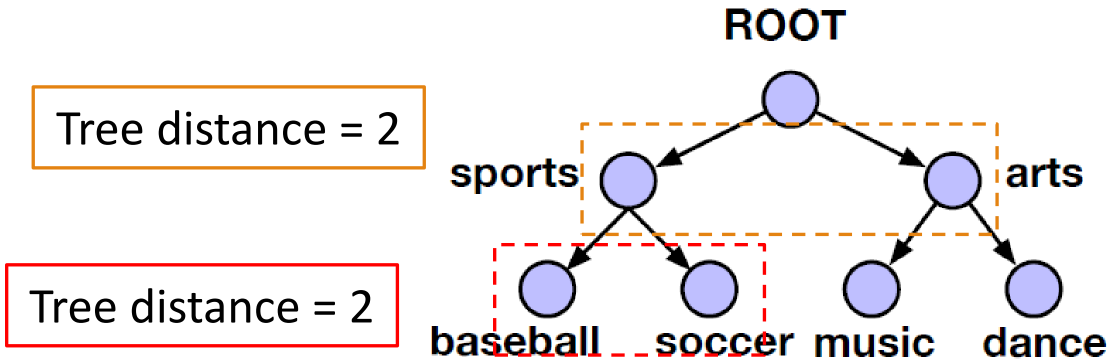
- Unsupervised Topic Modeling
- Supervised & Seed-Guided Topic Modeling
- Clustering-based Topic Discovery
- Discriminative Topic Mining
  - Introduction of the Task
  - CatE: Discriminative Topic Mining via Category-Name Guided Text Embedding [WWW'20]
  - Demo: TopicMine (based on CatE)
  - JoSH: Hierarchical Topic Mining via Joint Spherical Tree and Text Embedding [KDD'20]

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

JoSH Embedding

- 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 "soccer" 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

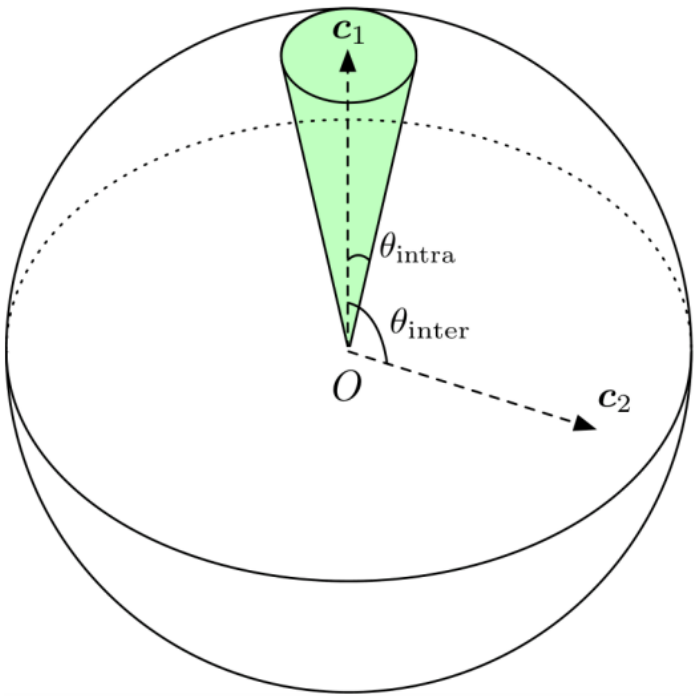
$$\mathcal{L}_{\text{intra}} = \sum_{c_i \in \mathcal{T}} \sum_{w_j \in C_i} \min(0, \boldsymbol{u}_{w_j}^\top \boldsymbol{c}_i - m_{\text{intra}}),$$

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

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

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

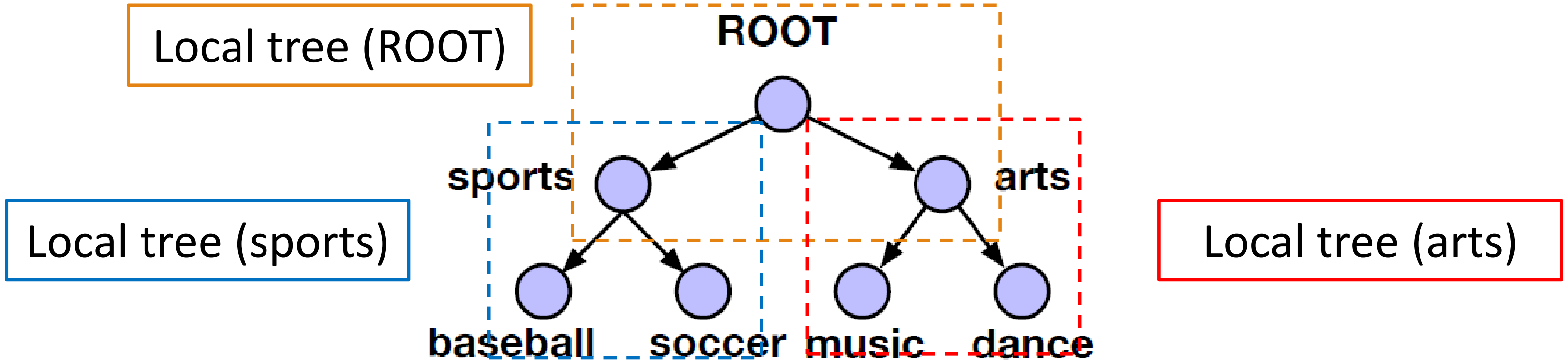
$$\theta_{\text{inter}} \geq \arccos(1 - m_{\text{inter}})$$



(a) Intra- & Inter-Category Configuration.

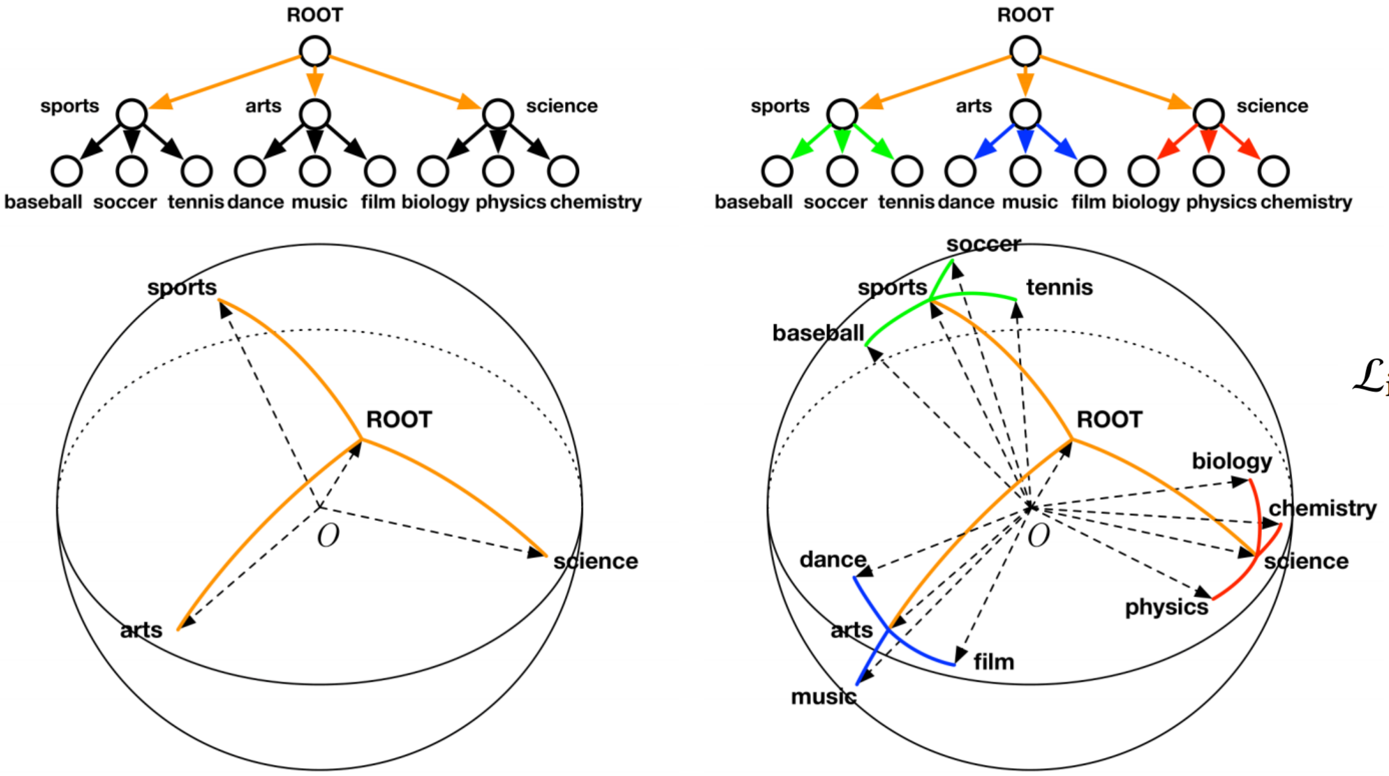
JoSH Tree Embedding

- **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 of 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 (Similar to CatE)
- 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

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

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 name;

while *True* **do**

  $t \leftarrow t + 1$;

  // E-Step (representative term retrieval);

  $C_i^{(t)}|_{i=1}^{n} \leftarrow$ Eq. (11);

  // M-Step (embedding training);

  $\boldsymbol{u}_w, \boldsymbol{v}_w, \boldsymbol{d}, \boldsymbol{c} \leftarrow$ Eqs. (12), (13), (14), (15), (16);

  if $\forall i, C_i^{(t)}$ agrees with $C_i^{(t-1)}$ on top-K terms **then**

    Break;

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

Experiments: 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: Qualitative Results

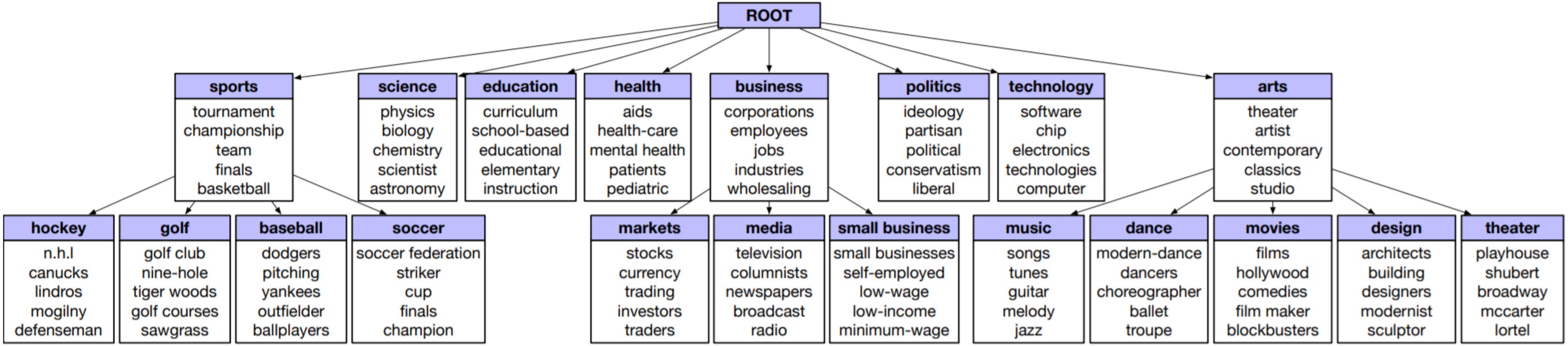
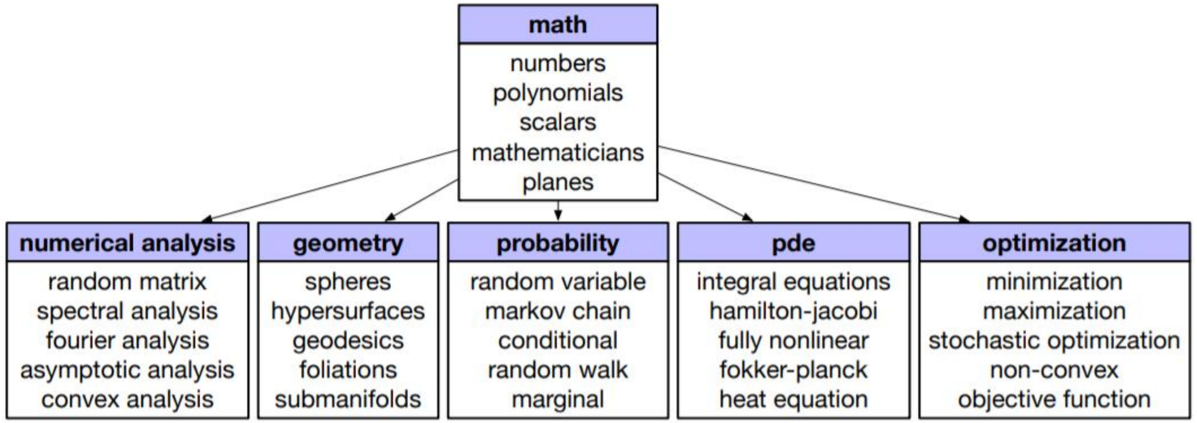
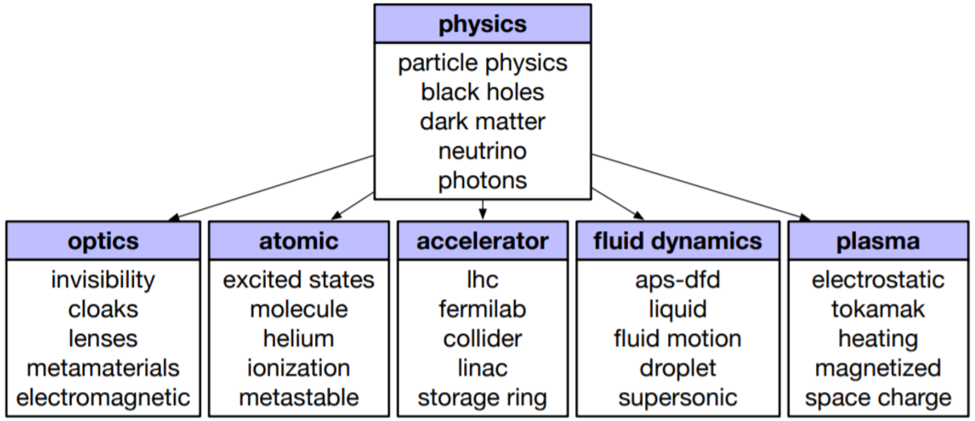


Figure 3: Hierarchical Topic Mining results on NYT.

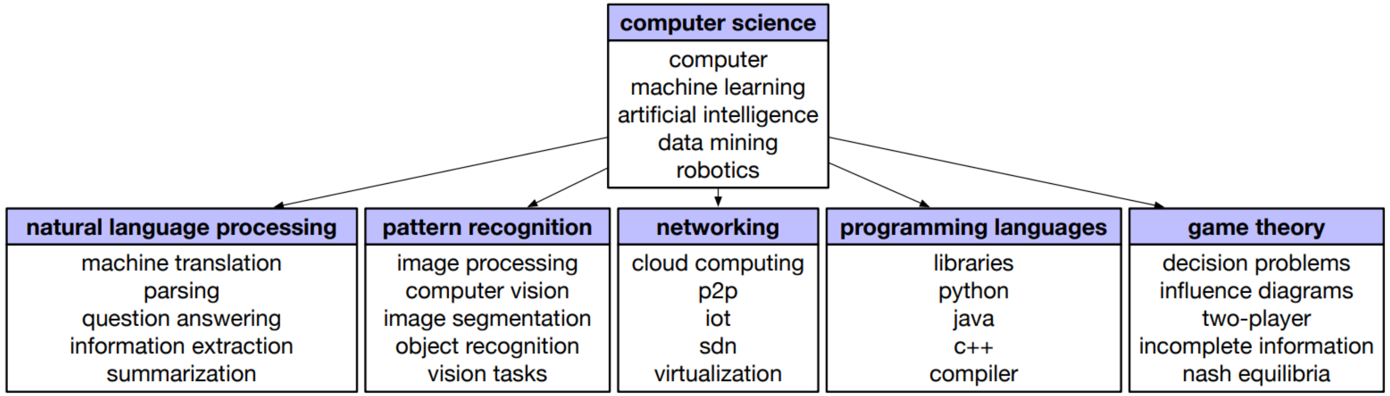
Experiments: Qualitative Results



(a) “Math” subtree.



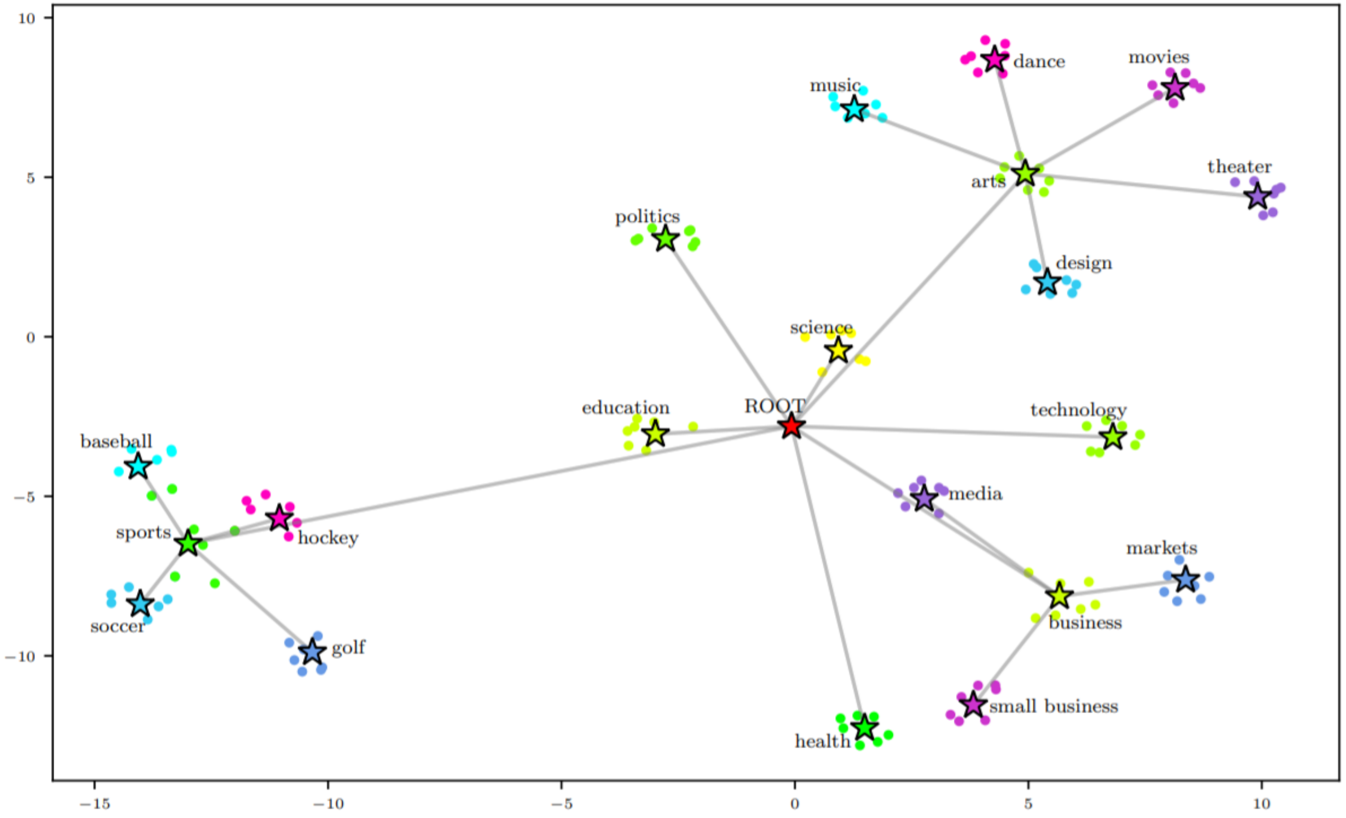
(b) “Physics” subtree.



(c) “Computer Science” subtree.

Experiments: Joint Embedding Space Visualization

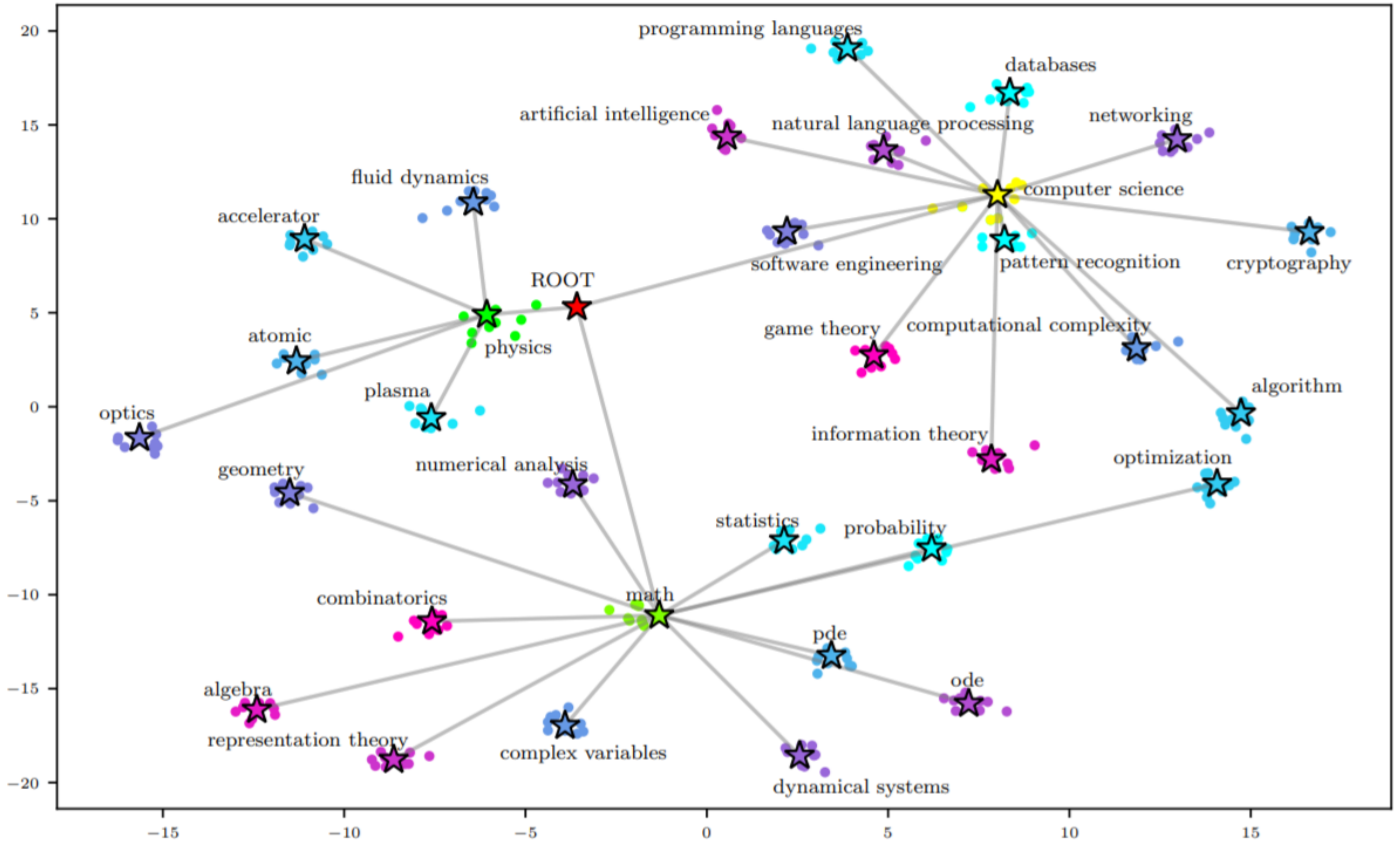
- T-SNE visualization (stars=category embeddings; dots=representative word embeddings)



(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.

References

- Blei, D. M., Griffiths, T. L., Jordan, M. I., & Tenenbaum, J. B. (2003). Hierarchical topic models and the nested Chinese restaurant process. NIPS.
- Blei, D. M., & McAuliffe, J. D. (2007). Supervised topic models. NIPS.
- Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. Journal of machine Learning research.
- Mimno, D., Li, W., & McCallum, A. (2007). Mixtures of hierarchical topics with pachinko allocation. ICML.
- Jagarlamudi, J., Daumé III, H., & Udupa, R. (2012). Incorporating lexical priors into topic models. EACL.
- Meng, Y., Huang, J., Wang, G., Wang, Z., Zhang, C., Zhang, Y., & Han, J. (2020). Discriminative topic mining via category-name guided text embedding. WWW.
- Meng, Y., Zhang, Y., Huang, J., Zhang, Y., Zhang, C., & Han, J. (2020). Hierarchical topic mining via joint spherical tree and text embedding. KDD.
- Sia, S., Dalmia, A., & Mielke, S. J. (2020). Tired of Topic Models? Clusters of Pretrained Word Embeddings Make for Fast and Good Topics too! EMNLP.